

TAUTOLOGY  
INNOVATION  
SCHOOL



LOGISTIC REGRESSION

# DEEP102

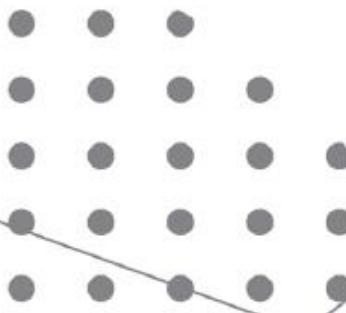
DEEP LEARNING

BY TAUTOLOGY

# KRIN CHINPRASATSAK

DATASCIENTIST  
INSTRUCTOR  
SPEAKER

Experienced data scientist, instructor and speaker,  
AI consultant for the public and private sectors,  
CEO & Co-founder of MADEBYAI and QuantMetric



# DL102 : Logistic Regression



# Lecture

**Formation**

**Logistic Regression**

Logistic Regression เป็นหนึ่งใน algorithm ใช้ใน supervised learning

**Logistic Reg**

Regression เป็นหนึ่งใน learning ที่ใช้สำหรับการคำนวณ การล่วงเวลา (เวลา) ไปในช่วง (0,1) เพื่อ

**Logistic Regression with Multiclass**

**k classes**

**Extension to Neural Network**

**How to create model**

**How to create model**

สร้าง feature ให้โดยการหา unique word ของ dataset ที่มีอยู่ใน dataset ตามที่กำหนด ข้อมูลประมวลผล unique word ที่มีอยู่ใน dataset ที่ต้องการ

**How to create model**

การคำนวณค่าของ logistic regression ที่มีค่าต่ำกว่า 0.5 ให้เป็น 0 และค่าต่ำกว่า 0.5 ให้เป็น 1

**Data Stating**

**k classes**

**How to create model**

การคำนวณค่าของ logistic regression ที่มีค่าต่ำกว่า 0.5 ให้เป็น 0 และค่าต่ำกว่า 0.5 ให้เป็น 1

**Origin of the Equation**

**Logi**

**k classes**

**How to create model**

**How to create model**

# Workshop



# Workshop



# All Topics



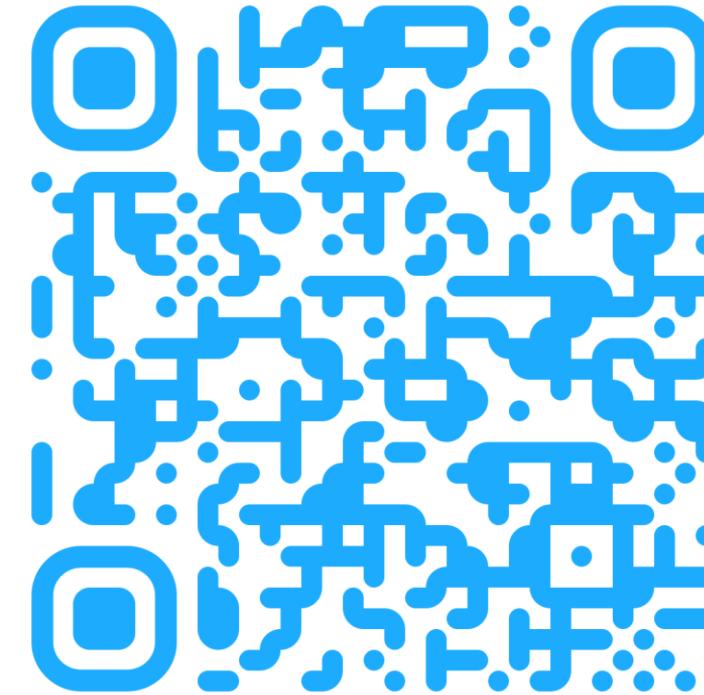
# DL102 : Logistic Regression



# Course Journey



# Facebook Group



TAUTOLOGY Deep Learning

<https://www.facebook.com/groups/2723268457980889>

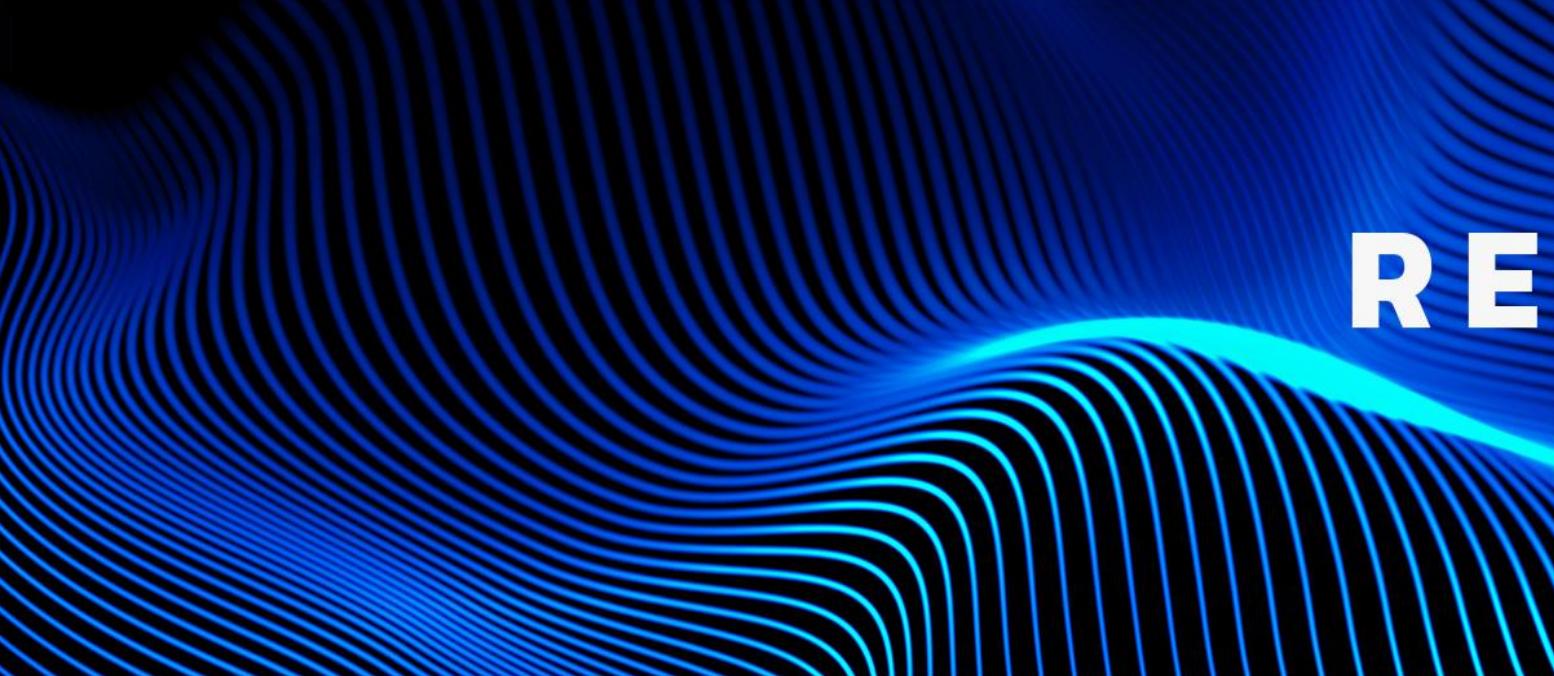
# Document for DL102



[https://github.com/TAUTOLOGY-EDUCATION/Deep-  
Learning-the-Series/tree/main/DL102](https://github.com/TAUTOLOGY-EDUCATION/Deep-Learning-the-Series/tree/main/DL102)

# DL102 : Logistic Regression





# LOGISTIC REGRESSION (2-CLASS)

BY TAU TOLOGY

# Logistic Regression (Binary Classification)



# Introduction

# Introduction

What is Logistic  
Regression?

Extension to  
Neural Network

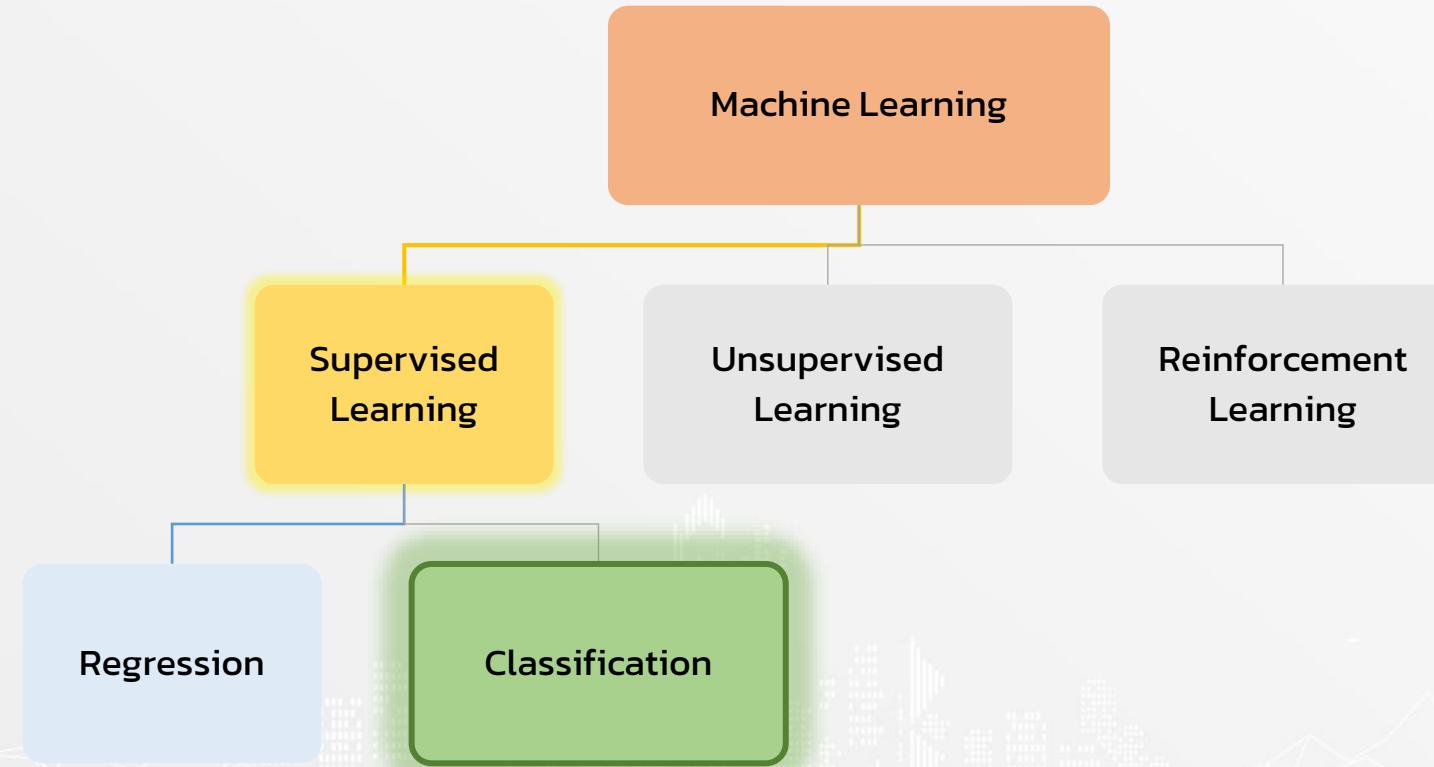
Real World  
Application



# What is Logistic Regression?

**Logistic Regression** เป็นหนึ่งใน algorithm ประเภท supervised learning ที่ใช้สำหรับแก้ปัญหา **classification** โดยมีหลักการทำงานคือ จำแนกประเภทของข้อมูล ด้วยการสร้างสมการเชิงเส้น

# What is Logistic Regression?



# What is Logistic Regression?

**Logistic Regression**

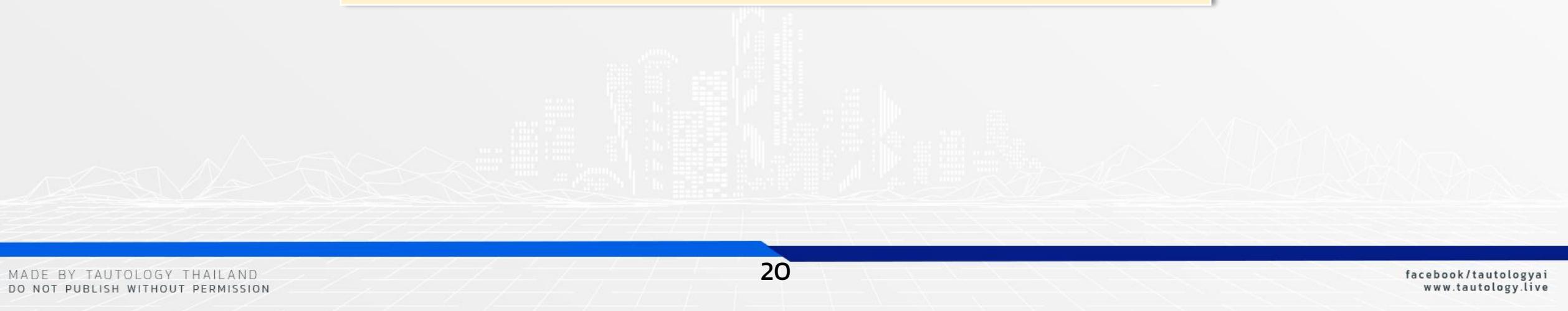
**Binary**

**Multi-Class**

# What is Logistic Regression?

## Binary

**Binary** คือการ map ค่าไปยัง **(0, 1)** เพื่อจำแนก  
ข้อมูล 2 class ด้วย function **sigmoid**



# What is Logistic Regression?

สมการคงตัวสตร์ของ **Logistic Regression**

$$z = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p$$

$$\hat{y} = \sigma(z) \quad (\text{sigmoid})$$

- โดยที่
- ◆  $\hat{y}$  คือ ตัวแปรตาม (predicted target)
  - ◆  $z$  คือ สมการเชิงเส้น (linear regression)
  - ◆  $x_1, x_2, x_3, \dots, x_p$  คือ ตัวแปรตัว (feature)
  - ◆  $w_0, w_1, w_2, \dots, w_p$  คือ สัมประสิทธิ์ (coefficient)

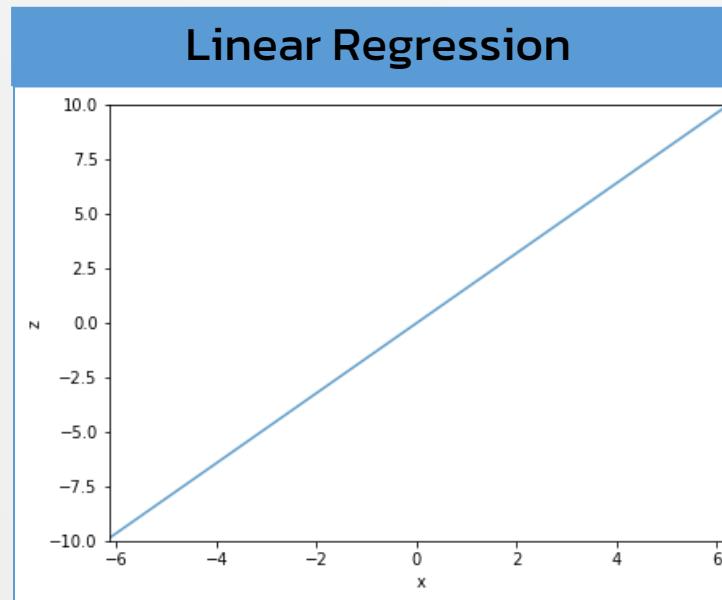
# What is Logistic Regression?

สมการคณิตศาสตร์ของ **Sigmoid Function**  $\sigma : (-\infty, \infty) \rightarrow (0, 1)$

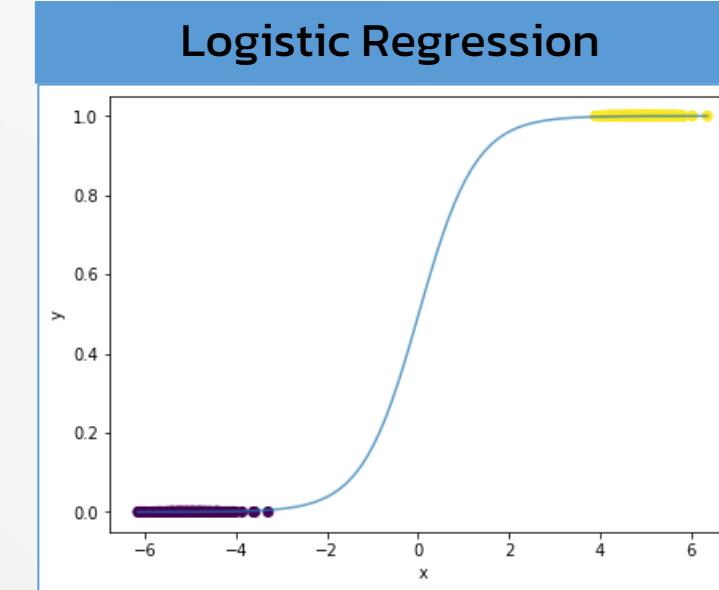
$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{e^z}{e^z + 1}$$

- โดยที่
- ◆  $\hat{y}$  คือ ตัว預測 (predicted target)
  - ◆  $z$  คือ สมการเชิงเส้น (linear regression)
  - ◆  $e$  คือ ค่าคงที่ และมีค่าเท่ากับ 2.718281828...

# What is Logistic Regression?



$$z = w_0 + w_1 x_1$$



$$\hat{y} = \sigma(z)$$

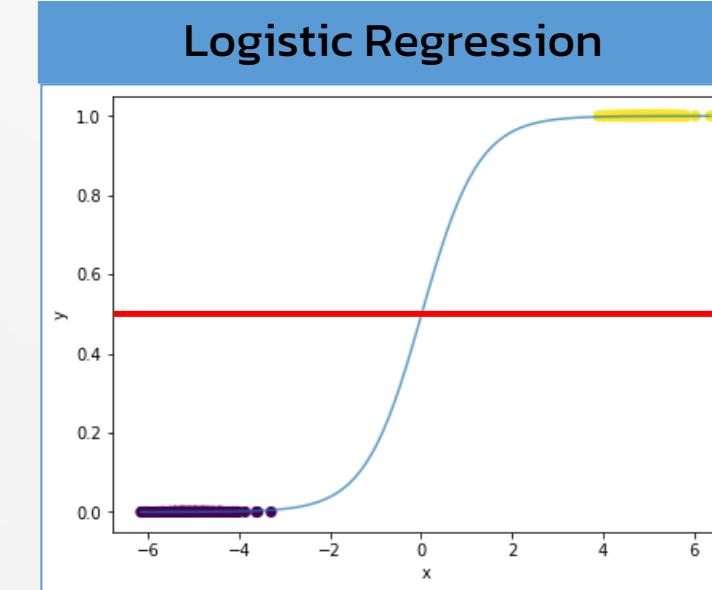
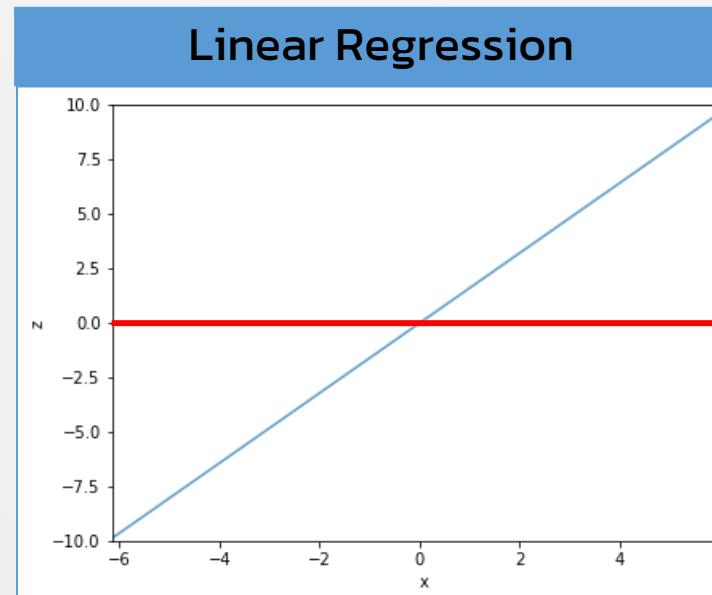
# What is Logistic Regression?

Why 0, 1?

# What is Logistic Regression?

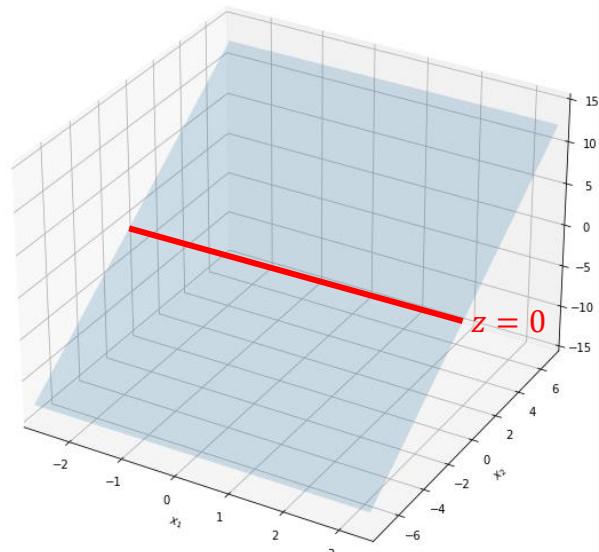
0, 1 เป็นตัวแทนของ **class** ก็ง 2 กี่เราต้องการ classification เช่น เป็นโรค/ไม่เป็นโรค,  
ปลอยกู้/ไม่ปลอยกู้, Fake/True เป็นต้น

# What is Logistic Regression?



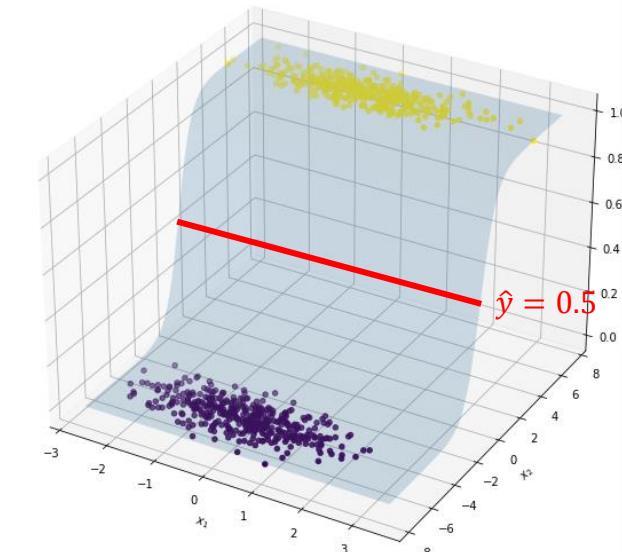
# What is Logistic Regression?

Linear Regression



$$z = w_0 + w_1 x_1 + w_2 x_2$$

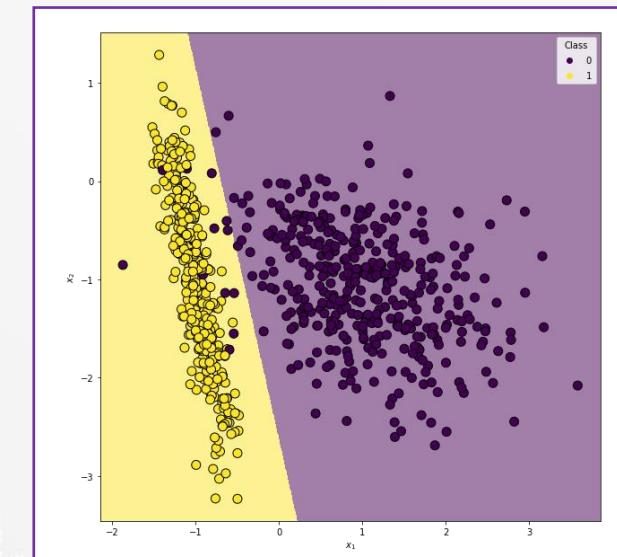
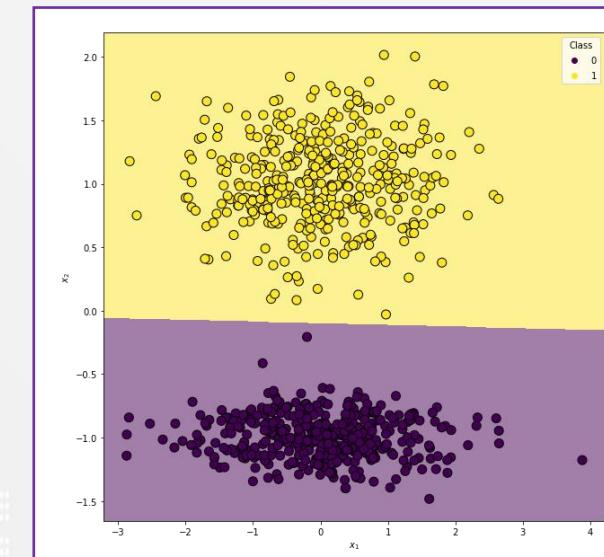
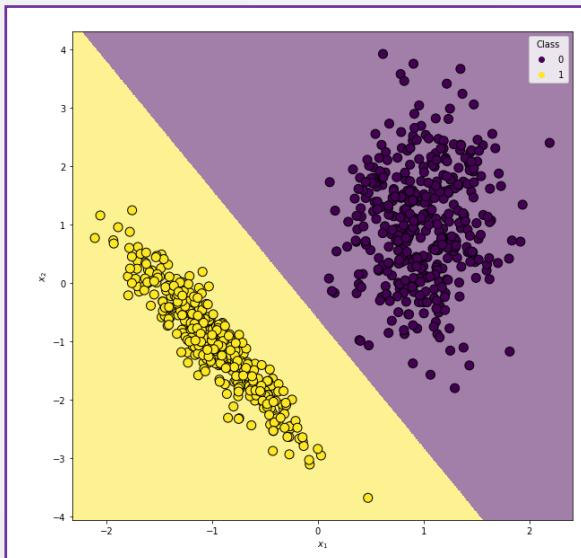
Logistic Regression



$$\hat{y} = \sigma(z)$$

# What is Logistic Regression?

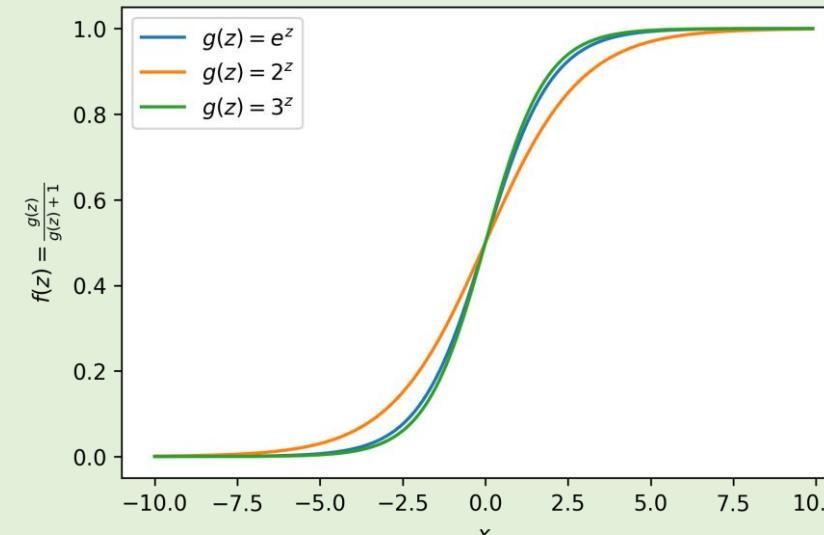
**Decision Boundary** คือ สมการที่ใช้แบ่ง class ( $z = w_0 + w_1x_1 + \cdots + w_px_p = 0$ )



# What is Logistic Regression?

Why  $e$  ?

# What is Logistic Regression?



- ◆  $\frac{\partial}{\partial z} \left( \frac{e^z}{e^z + 1} \right) = \frac{e^z}{(e^z + 1)^2}$
- ◆  $\frac{\partial}{\partial z} \left( \frac{2^z}{2^z + 1} \right) = \frac{2^z \log(2)}{(2^z + 1)^2}$
- ◆  $\frac{\partial}{\partial z} \left( \frac{3^z}{3^z + 1} \right) = \frac{3^z \log(3)}{(3^z + 1)^2}$

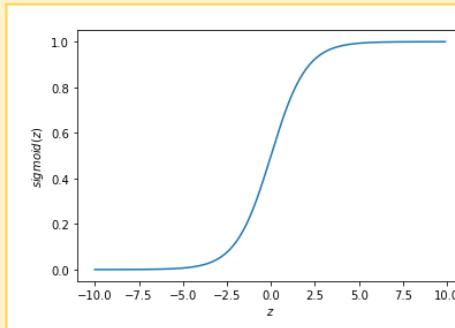
# What is Logistic Regression?

Why not step function ?

# What is Logistic Regression?

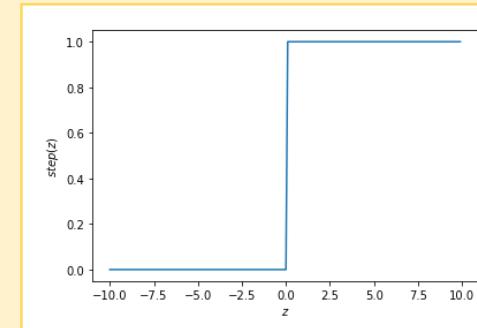
## Sigmoid Function

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



## Step Function

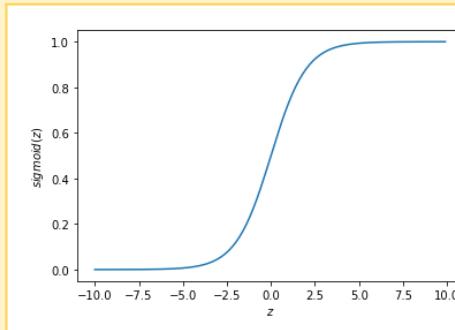
$$f(z) = \begin{cases} 1 & ; \quad z \geq 0 \\ 0 & ; \quad z < 0 \end{cases}$$



# What is Logistic Regression?

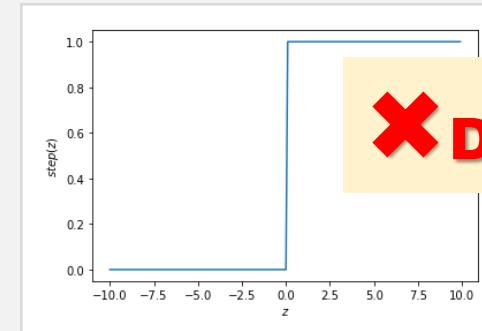
## Sigmoid Function

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



## Step Function

$$f(z) = \begin{cases} 1 & ; z \geq 0 \\ 0 & ; z < 0 \end{cases}$$



✗ Diff ไม่ได้

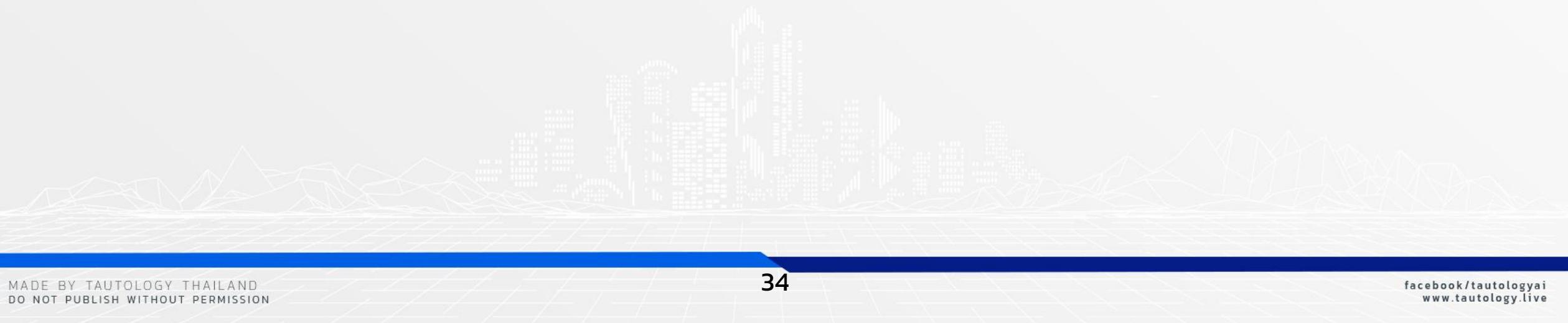
# Introduction

**What is Logistic  
Regression?**

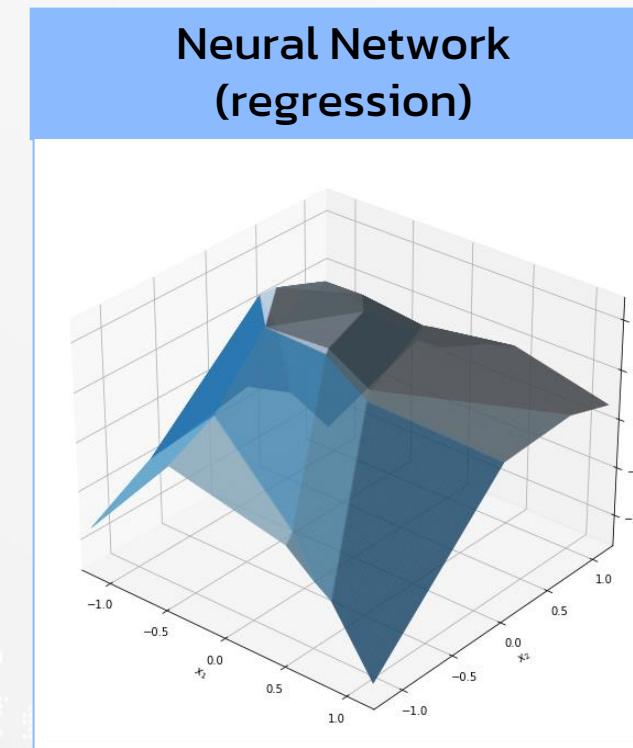
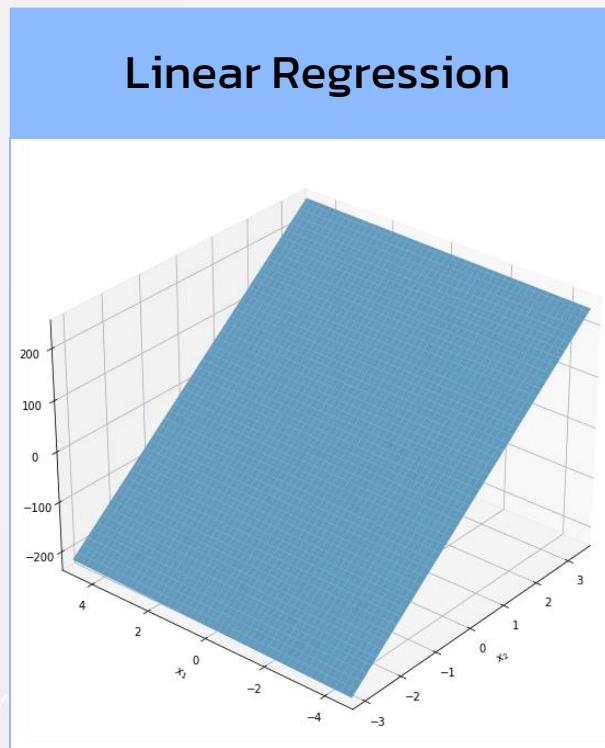


Extension to  
Neural Network

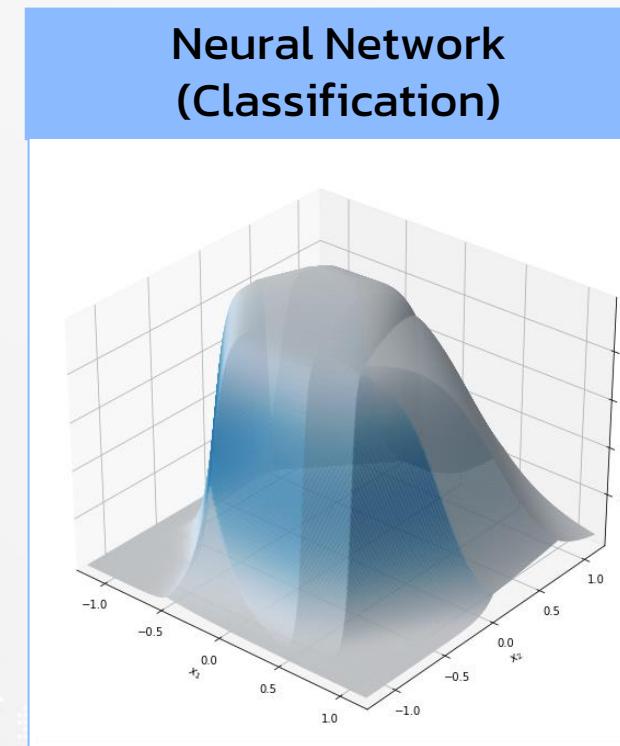
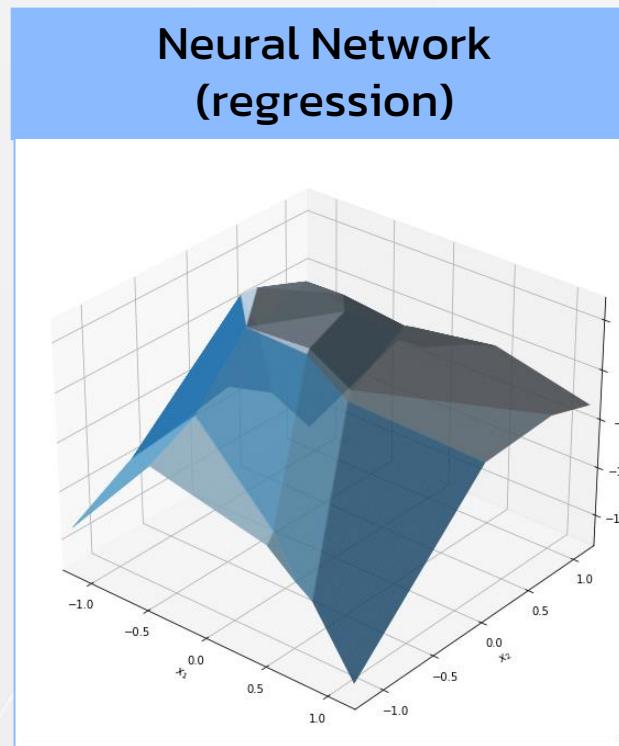
Real World  
Application



# Extension to Neural Network

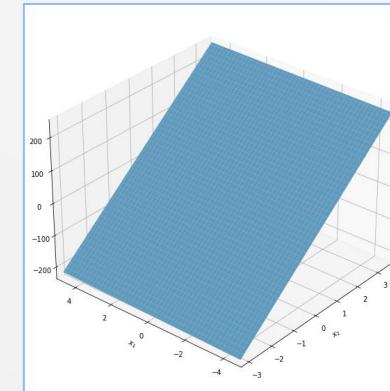


# Extension to Neural Network

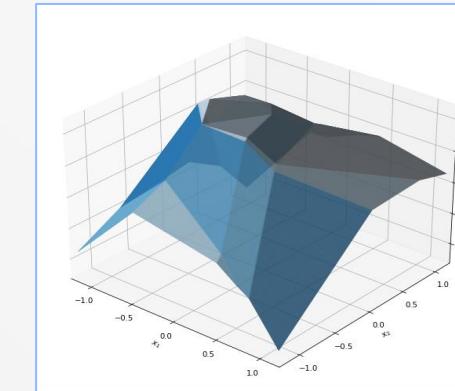


# Extension to Neural Network

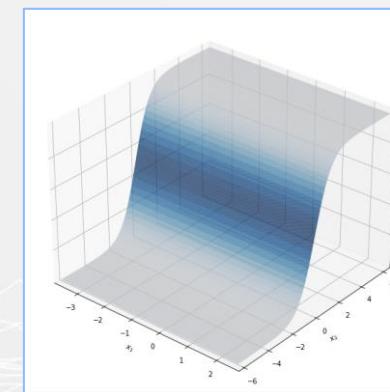
Linear  
Regression



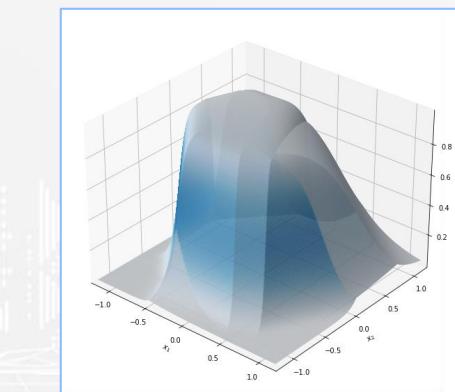
Neural Network  
(regression)



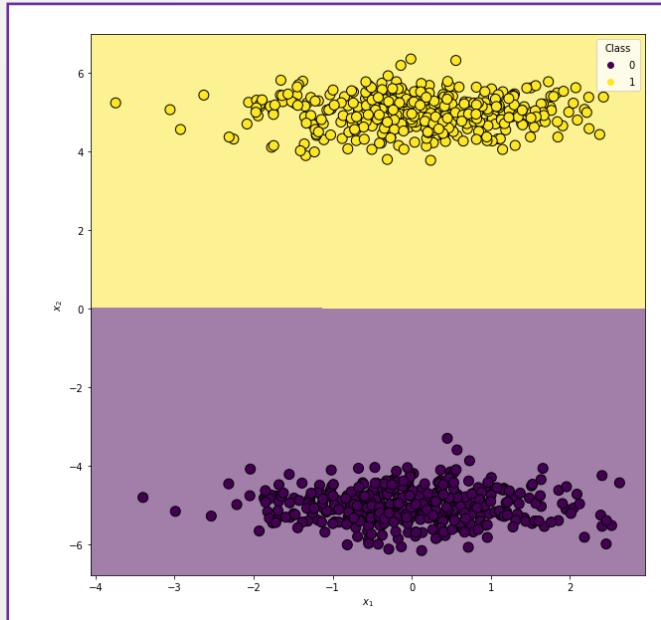
Logistic  
Regression



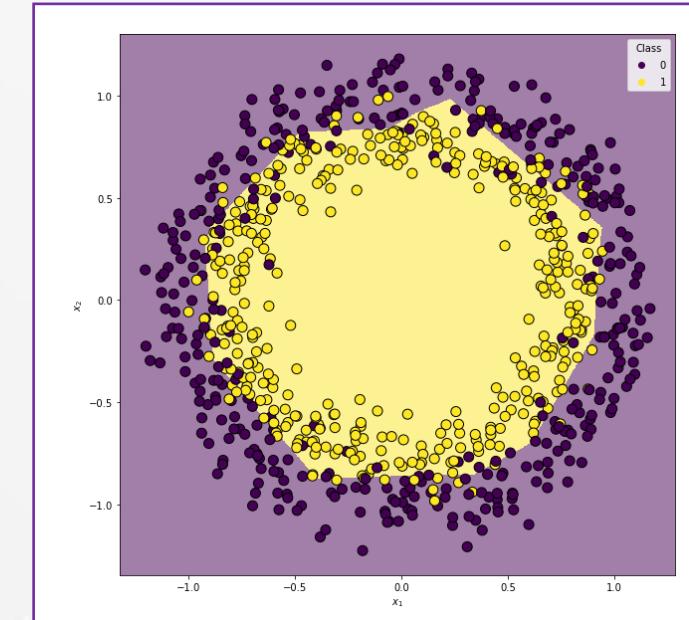
Neural Network  
(Classification)



# Extension to Neural Network



Logistic Regression



Neural Network  
(Classification)

# Introduction

**What is Logistic  
Regression?**



**Extension to  
Neural Network**



**Real World  
Application**



# Real World Application



## การตรวจสอบไขข้อความเชื่อ

โดยพิจารณาจาก จำนวนของคำต่าง ๆ ที่  
ปรากฏในข้อความ, ความยาวของข้อความ,  
คะแนนความอ่านง่ายของโนนุชนา เป็นต้น

<https://aclanthology.org/d19-5017/>

# Real World Application



## การจำแนกรุกิจ SME ที่มีโอกาส ล้มละลาย

โดยพิจารณาจาก ผลประกอบการ, ปริมาณ  
ดอกเบี้ยของหนี้สินบริษัท, ช่วงเวลาที่ธุรกิจค้าง  
ชำระค่าสินค้าหรือค่าบริการ เป็นต้น

<https://www.mdpi.com/2227-9091/8/4/107>

# Real World Application



## การคัดแยกผู้ป่วยที่เป็นโรคอัลไซเมอร์ ขั้นต้น

โดยพิจารณาจาก ภาพที่ได้จากการแสกนสมอง  
ด้วยวิธี MRI

<https://www.sciencedirect.com/science/article/abs/pii/s1746809420304705>

# Introduction

**What is Logistic  
Regression?**



**Extension to  
Neural Network**



**Real World  
Application**

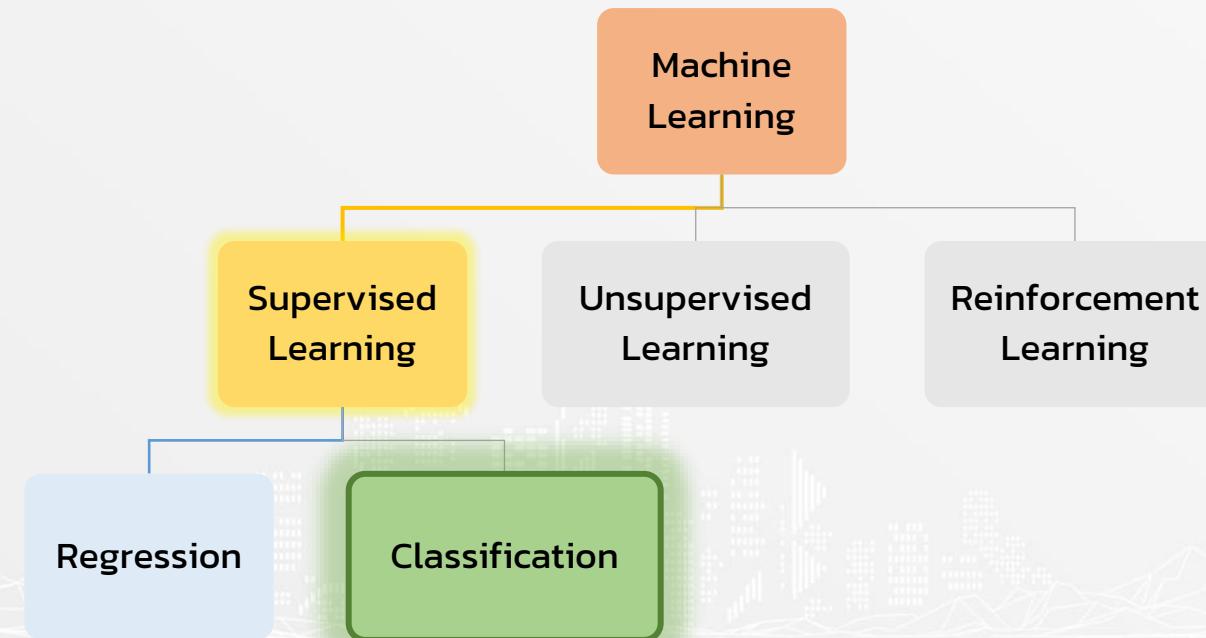


# Logistic Regression (Binary Classification)



# Logistic Regression

**Logistic Regression** ធន្តុកប៉ុងឲ្យ algorithm ប្រែក់នៅ **supervised learning**



# Concept of Supervised Learning

**Data   ⇒  Model   ⇒  Prediction**

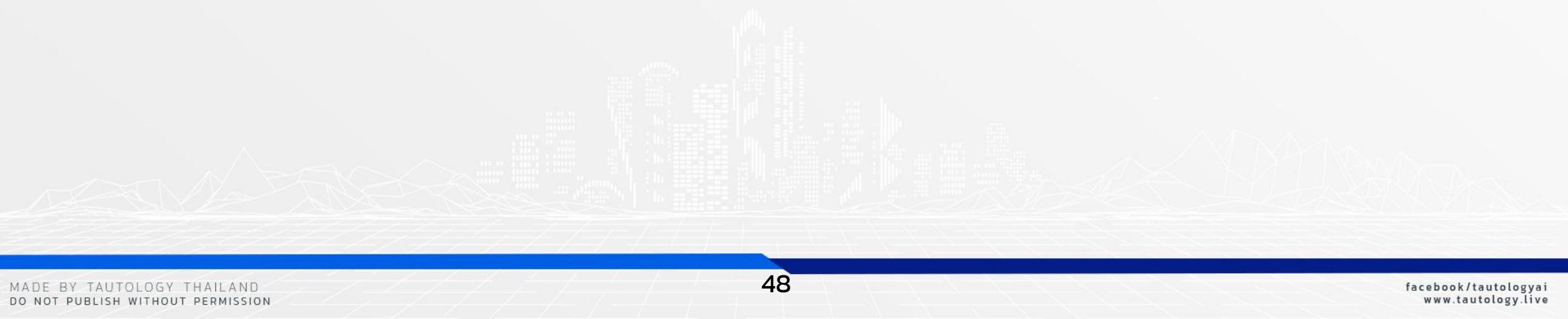
# Data

# Data

Data Stating

Data Requirement

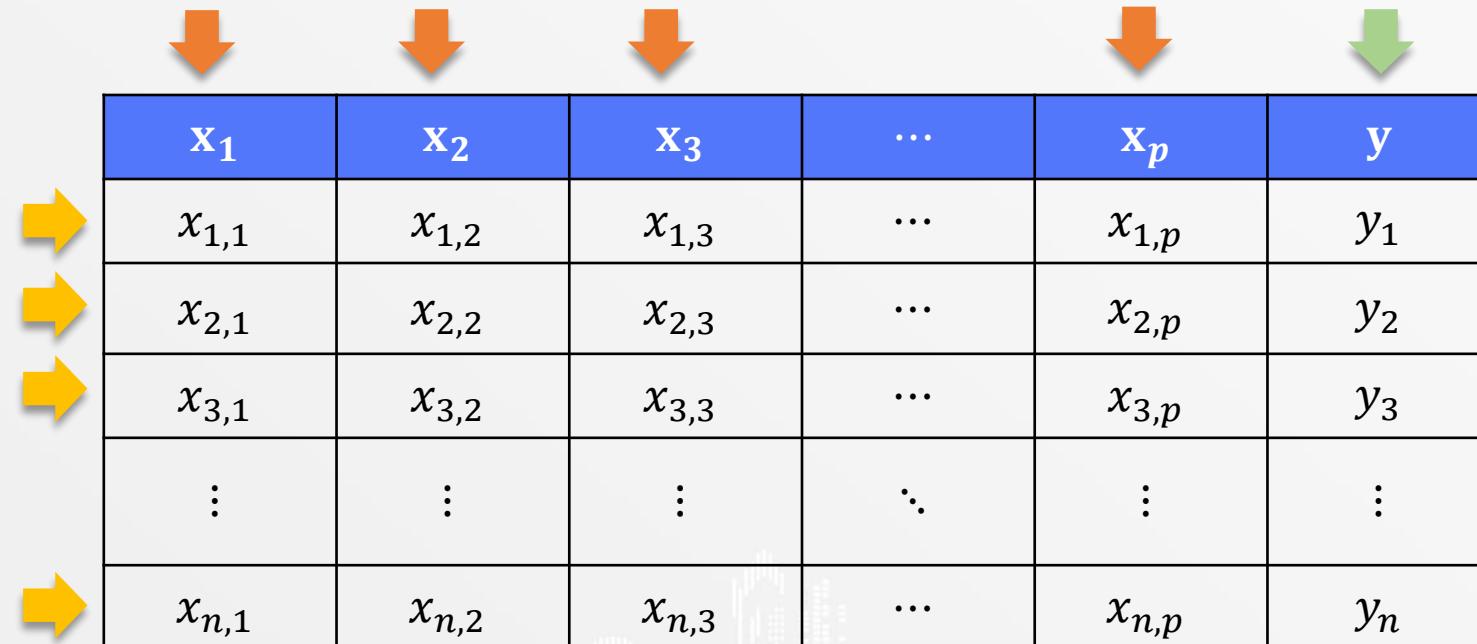
More about  
Target



# Data Stating

$x_1$	$x_2$	$x_3$	...	$x_p$	$y$
$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	...	$x_{1,p}$	$y_1$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	...	$x_{2,p}$	$y_2$
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	...	$x_{3,p}$	$y_3$
⋮	⋮	⋮	⋮	⋮	⋮
$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	...	$x_{n,p}$	$y_n$

# Data Stating



The diagram illustrates a data matrix structure. At the top, five orange arrows point downwards, each aligned with a column header:  $x_1$ ,  $x_2$ ,  $x_3$ ,  $\dots$ , and  $x_p$ . A green arrow points downwards to the  $y$  column header. The matrix itself is a 6x6 grid. The columns are labeled  $x_1$  through  $x_p$  and the rows are labeled  $y_1$  through  $y_n$ . The matrix contains data elements  $x_{i,j}$  where  $i$  is the row index and  $j$  is the column index. The matrix is as follows:

$x_1$	$x_2$	$x_3$	$\dots$	$x_p$	$y$
$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$\dots$	$x_{1,p}$	$y_1$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$\dots$	$x_{2,p}$	$y_2$
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$\dots$	$x_{3,p}$	$y_3$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	$\dots$	$x_{n,p}$	$y_n$

- ♦  $n$  คือ จำนวน sample
- ♦  $p$  คือ จำนวน feature

# Data Stating

$x_1$	$x_2$	$x_3$	...	$x_p$	$y$
$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	...	$x_{1,p}$	$y_1$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	...	$x_{2,p}$	$y_2$
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	...	$x_{3,p}$	$y_3$
:	:	:	..	:	:
$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	...	$x_{n,p}$	$y_n$

- $x_{2,3}$  คือ sample ที่ 2 feature ที่ 3
- $x_{3,p}$  คือ sample ที่ 3 feature ที่  $p$
- $x_{n,p}$  คือ sample ที่  $n$  feature ที่  $p$
- $y_2$  คือ target ของ sample ที่ 2
- $y_3$  คือ target ของ sample ที่ 3
- $y_n$  คือ target ของ sample ที่  $n$

# Data Stating

## Example

- เราต้องการจำแนกผู้ป่วยโรคหัวใจ โดยพิจารณาจาก อายุ เพศ ความดันโลหิต คอลเลสเตอรอล

## Data

อายุ	เพศ	ความดันโลหิต (mmHg)	คอลเลสเตอรอล (mg/dL)	เป็นโรคหัวใจ
42	0	120	209	1
57	1	150	168	1
58	1	128	259	0
59	0	174	249	0

# Data Stating

- ข้อมูลตามแนวแอกว คือ Sample



อายุ	เพศ	ความดันโลหิต (mmHg)	คอลเลสเตอรอล (mg/dL)	เป็นโรคหัวใจ
42	0	120	209	1
57	1	150	168	1
58	1	128	259	0
59	0	174	249	0

# Data Stating

- ข้อมูลตามแนวหลัก คือ Feature and Target
  - Feature (ตัวแปรต้น) คือ ข้อมูลที่ส่งผลให้เกิด target
  - Target (ตัวแปรตาม) คือ ข้อมูลที่เราสนใจจะพยากรณ์

**Feature**

อายุ	เพศ	ความดันโลหิต (mmHg)	คอลเลสเตอรอล (mg/dL)
42	0	120	209
57	1	150	168
58	1	128	259
59	0	174	249

**Target**

เป็นโรคหัวใจ
1
1
0
0

# Data Stating

- Feature and Target
  - เราสามารถแยก และปรับให้เป็น matrix ได้ดังนี้

$$X = \begin{bmatrix} 42 & 0 & 120 & 209 \\ 57 & 1 & 150 & 168 \\ 58 & 1 & 128 & 259 \\ 59 & 0 & 174 & 249 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

# Data

**Data Stating**



**Data Requirement**

**More about  
Target**

# Data Requirement

- ข้อมูลต้องอยู่ในรูปแบบของตาราง
- ข้อมูลต้องเป็น numerical

อายุ	เพศ	ความดันโลหิต (mmHg)	คอลเลสเตอรอล (mg/dL)	เป็นโรคหัวใจ
42	0	120	209	1
57	1	150	168	1
58	1	128	259	0
59	0	174	249	0

# Data Requirement

- ตัวอย่างข้อมูลที่สามารถใช้งานได้เลย และยังไม่สามารถใช้งานได้

อายุ	เพศ	เป็นโรคหัวใจ
42	0	1
57	1	1
58	1	0
59	0	0



อายุ	เพศ	เป็นโรคหัวใจ
42	female	1
57	male	1
58	male	0
59	female	0



# Data Requirement

- เราสามารถแปลงได้โดยสามารถใช้ความรู้ในส่วนของ Data Preparation

อายุ	เพศ	เป็นโรคหัวใจ
42	female	1
57	male	1
58	male	0
59	female	0



อายุ	female	male	เป็นโรคหัวใจ
42	1	0	1
57	0	1	1
58	0	1	0
59	1	0	0

# Data Requirement



For more information



Data Preparation

# Data

**Data Stating**



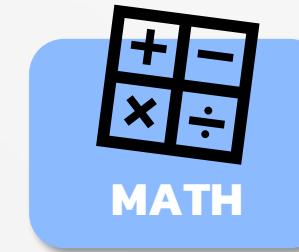
**Data Requirement**



More about  
Target



# More about Target



อายุ	female	male	เป็น โรคหัวใจ
42	1	0	True
57	0	1	True
58	0	1	False
59	1	0	False



อายุ	female	male	เป็น โรคหัวใจ
42	1	0	1
57	0	1	1
58	0	1	0
59	1	0	0



# More about Target



อายุ	female	male	เป็นโรคหัวใจ
42	1	0	True
57	0	1	True
58	0	1	False
59	1	0	False

อายุ	female	male	เป็นโรคหัวใจ
42	1	0	1
57	0	1	1
58	0	1	0
59	1	0	0

# More about Target

อายุ	female	male	เป็นโรคหัวใจ
42	1	0	True
57	0	1	True
58	0	1	False
59	1	0	False

อายุ	female	male	เป็นโรคหัวใจ
42	1	0	1
57	0	1	1
58	0	1	0
59	1	0	0

Prediction

เป็นโรคหัวใจ
True
False
True
False

Prediction

เป็นโรคหัวใจ
1
0
1
0

Model



# Data

**Data Stating**



**Data Requirement**



**More about  
Target**



# Logistic Regression (Binary Classification)



# Model

# Model

Assumption

Real Face of the  
Model

Cost Function and  
Cost Landscape

How to Create  
Model (Math)

How to Create  
Model (Code)

# Assumption

1. Linear Relationship
2. Normality of Residuals
3. Homoscedasticity
4. No Missing Features
5. No Multicollinearity

# Assumption



For more information



Model Improvement  
in DL101

# Model

**Assumption**



Real Face of the  
Model

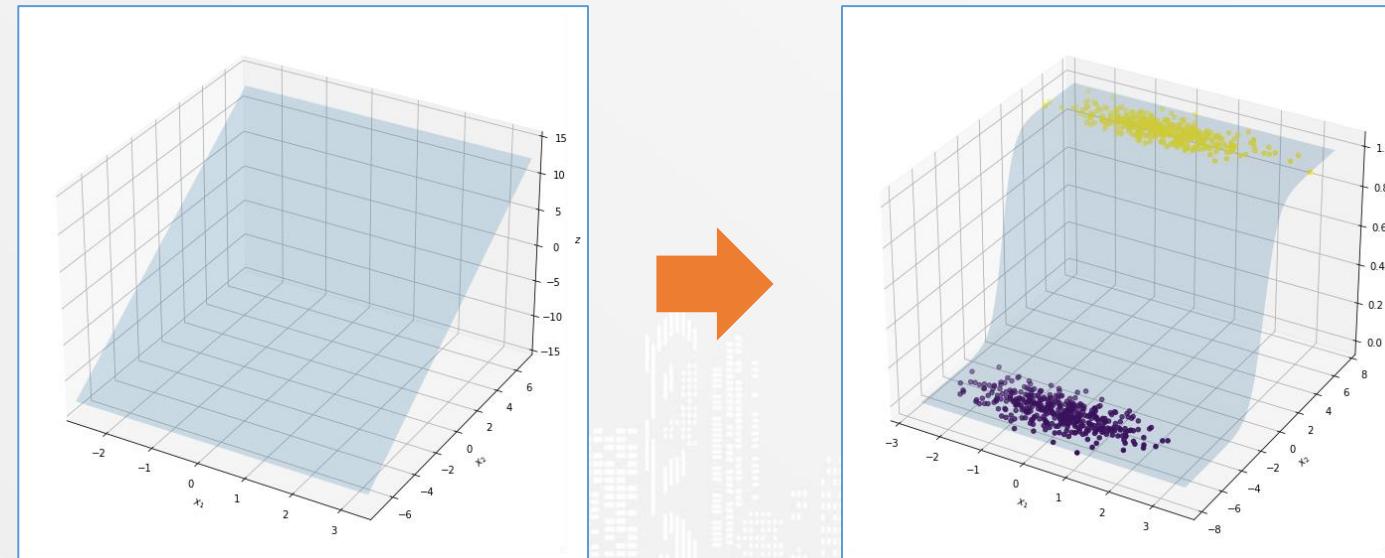
Cost Function and  
Cost Landscape

How to Create  
Model (Math)

How to Create  
Model (Code)

# Real Face of the Model

**Logistic regression** คือ โมเดลคณิตศาสตร์ที่จำแนกประเภทของข้อมูล ด้วยการสร้างสมการเชิงเส้น (linear regression)



# Real Face of the Model

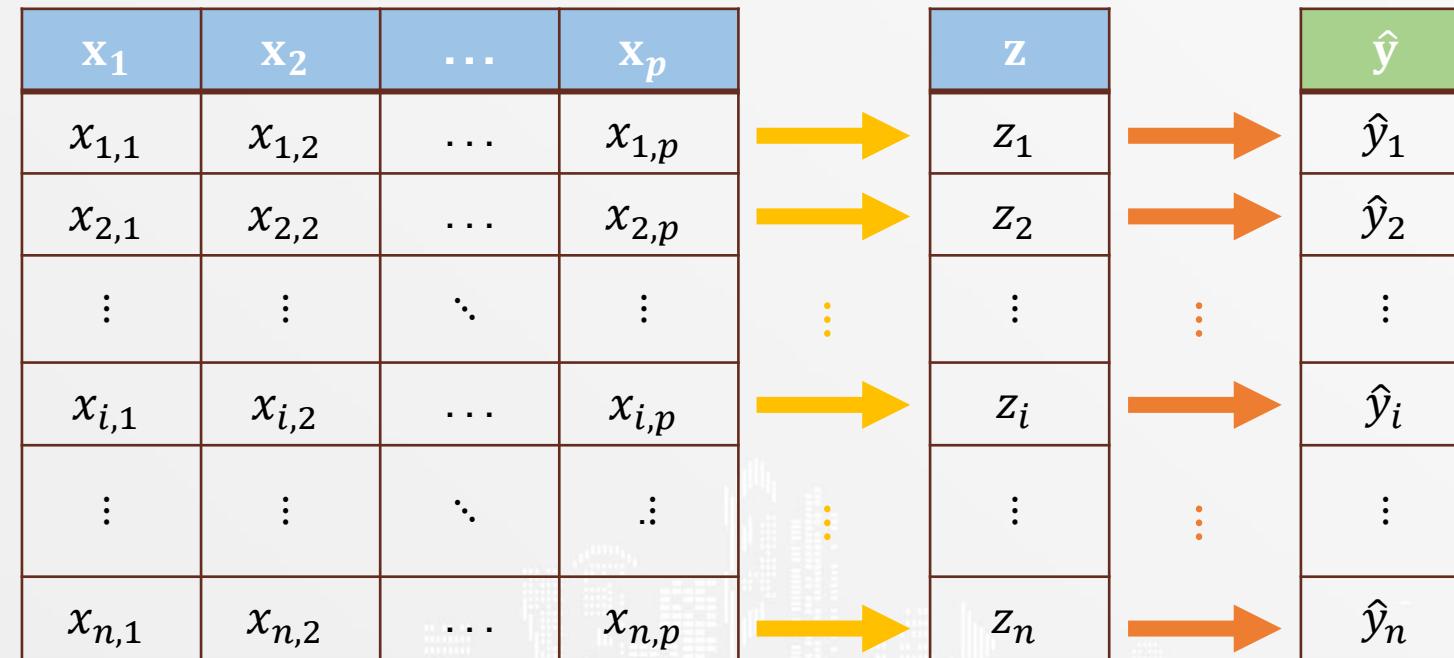
**Logistic regression** คือ โมเดลคณิตศาสตร์ที่จำแนกประเภทของข้อมูล ด้วยการสร้างสมการเชิงเส้น (linear regression)

$$z = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p$$

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

- โดยที่
- ◆  $\hat{y}$  คือ ตัว預測 (predicted target)
  - ◆  $z$  คือ สมการเชิงเส้น (linear regression)
  - ◆  $x_1, x_2, x_3, \dots, x_p$  คือ ตัว特征 (feature)
  - ◆  $w_0, w_1, w_2, \dots, w_p$  คือ 係數 (coefficient)

# Real Face of the Model



# Real Face of the Model

$$z = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p$$



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



$$z_1 = w_0 + w_1 x_{1,1} + w_2 x_{1,2} + \cdots + w_p x_{1,p}$$



$$\hat{y}_1 = \frac{1}{1 + e^{-z_1}}$$

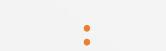
$$z_2 = w_0 + w_1 x_{2,1} + w_2 x_{2,2} + \cdots + w_p x_{2,p}$$



$$\hat{y}_2 = \frac{1}{1 + e^{-z_2}}$$

⋮

$$z_i = w_0 + w_1 x_{i,1} + w_2 x_{i,2} + \cdots + w_p x_{i,p}$$



$$\hat{y}_i = \frac{1}{1 + e^{-z_i}}$$

⋮

$$z_n = w_0 + w_1 x_{n,1} + w_2 x_{n,2} + \cdots + w_p x_{n,p}$$



$$\hat{y}_n = \frac{1}{1 + e^{-z_n}}$$

# Real Face of the Model



“ เราต้องการหา  $w_0, w_1, w_2, \dots, w_p$  ที่ทำให้ cost function ต่ำที่สุด ”

# Model

**Assumption**



**Real Face of the Model**



**Cost Function and Cost Landscape**



**How to Create Model (Math)**



**How to Create Model (Code)**



# Cost Function and Cost Landscape

**Cost function** กี่เราจะใช้ในการสร้าง model คือ

$$-\frac{1}{n} \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$

โดยสูตรข้างต้นมีชื่อว่า **Cross Entropy**

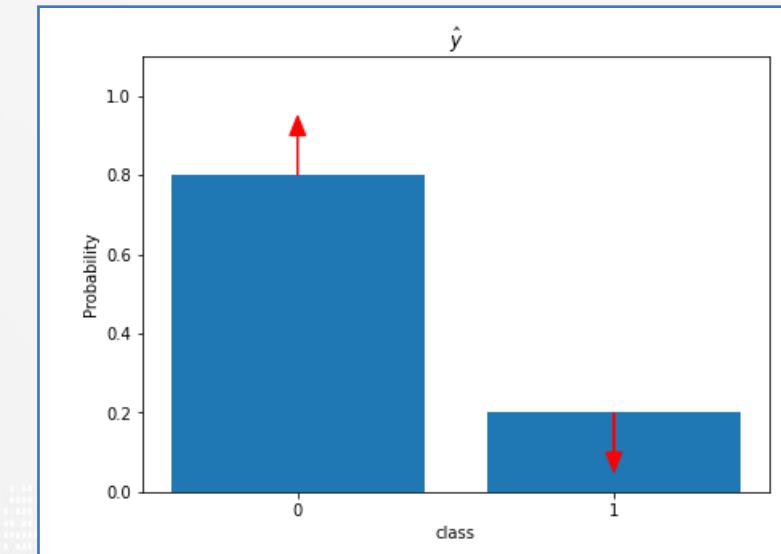
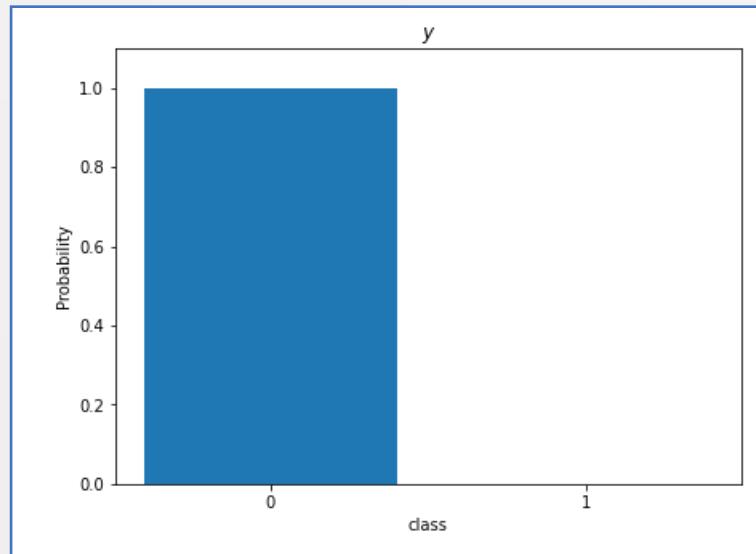
# Cost Function and Cost Landscape

## เหตุผลที่เลือกใช้ Cross Entropy

1. สามารถเปรียบเทียบ distribution ของ  $y$  กับ  $\hat{y}$  ได้
2. Cost function ที่นิยามแบบ cross entropy สามารถ diff ได้ และมีความต่อเนื่องทุกจุด
3. Cost function ที่นิยามแบบ cross entropy เป็น convex function และมีจุดต่ำสุดเพียงจุดเดียว

# Cost Function and Cost Landscape

1. สามารถเปรียบเทียบ distribution ของ  $y$  กับ  $\hat{y}$  ได้



# Cost Function and Cost Landscape



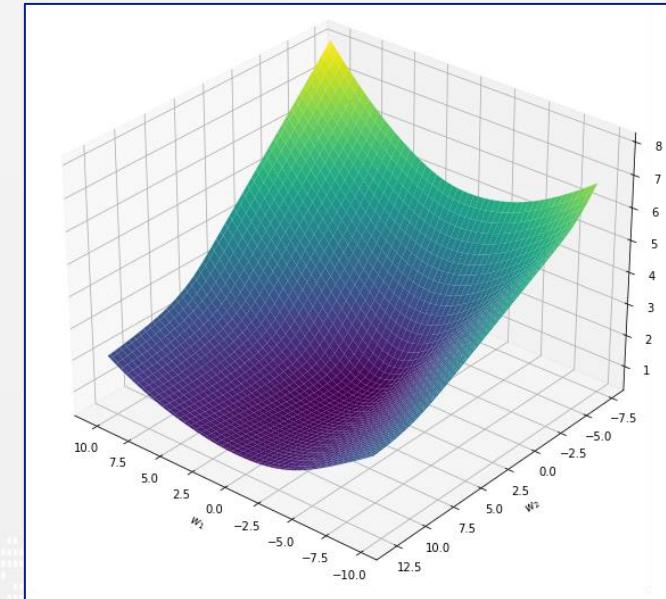
For more information



Cross Entropy

# Cost Function and Cost Landscape

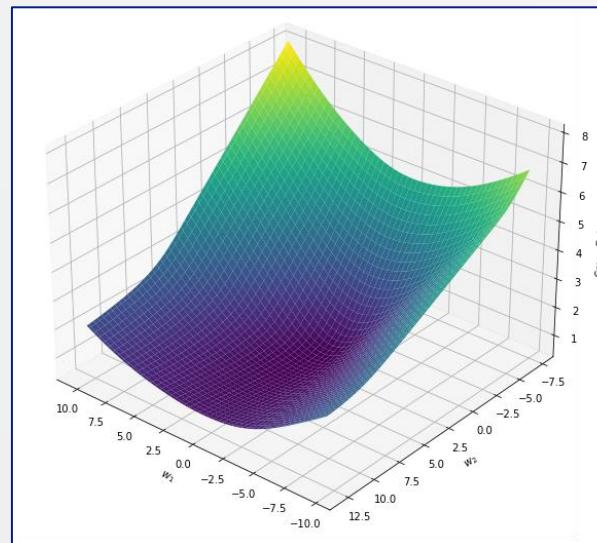
2. Cost function กี่นิยามแบบ cross entropy สามารถ diff ได้ และมีความต่อเนื่องทุกจุด



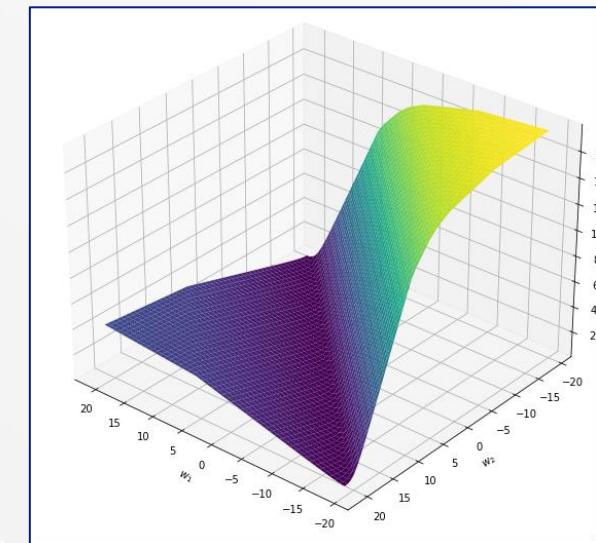
กราฟแสดง cost landscape ของ logistic regression  
โดยที่ cost function เป็น cross entropy

# Cost Function and Cost Landscape

3. Cost function ที่นิยามแบบ cross entropy เป็น convex function และมีจุดต่ำสุดเพียงจุดเดียว



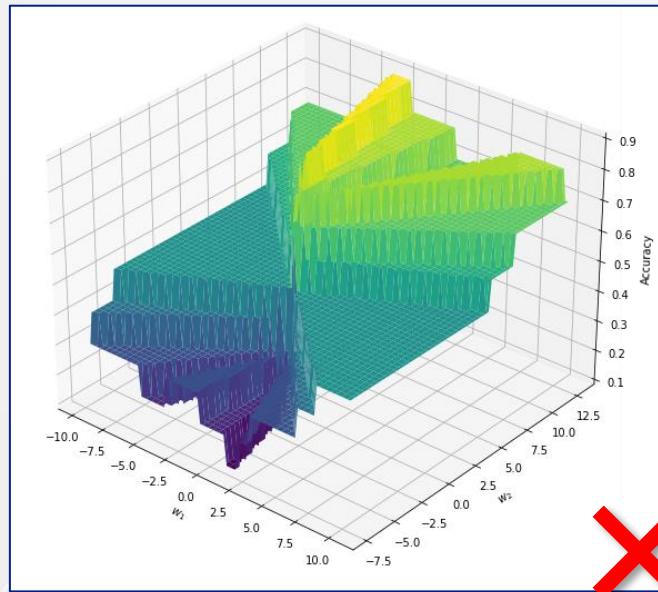
กราฟแสดง cost landscape ของ logistic regression โดยที่ feature เป็น no multicollinearity และ cost function เป็น cross entropy



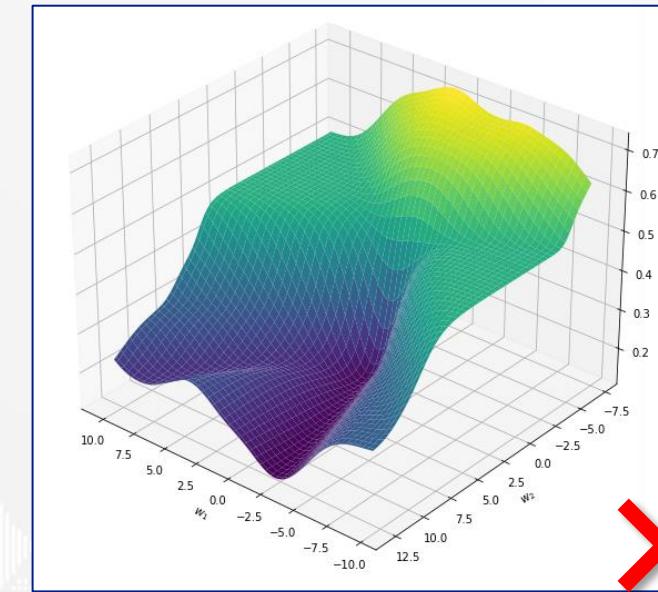
กราฟแสดง cost landscape ของ logistic regression โดยที่ feature เป็น linearly dependent และ cost function เป็น cross entropy

# Cost Function and Cost Landscape

3. Cost function ที่นิยามแบบ cross entropy เป็น convex function และมีจุดต่ำสุดเพียงจุดเดียว



กราฟแสดง cost landscape ของ logistic regression  
โดยที่ cost function เป็น accuracy

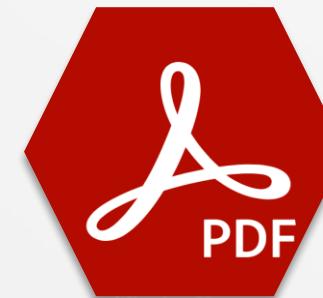


กราฟแสดง cost landscape ของ logistic regression  
โดยที่ cost function เป็น mean squared error

# Cost Function and Cost Landscape



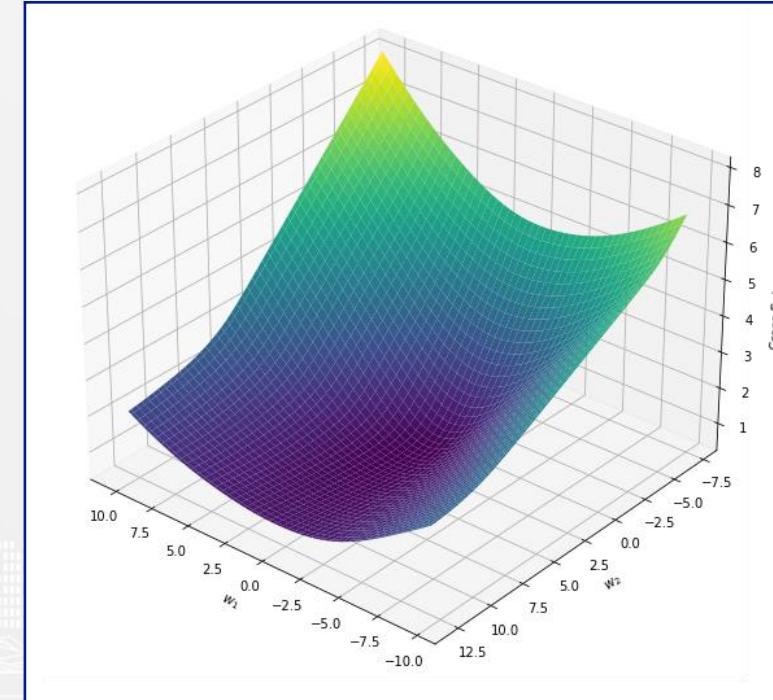
Convexity of Cross Entropy  
for Logistic Regression (Binary)



Open File  
**Convexity\_CE\_LoR\_(Binary).pdf**

# Cost Function and Cost Landscape

$$Cost = -\frac{1}{n} \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$



# Model

**Assumption**



**Real Face of the Model**



**Cost Function and Cost Landscape**



**How to Create Model (Math)**



**How to Create Model (Code)**



# How to Create Model (Math)

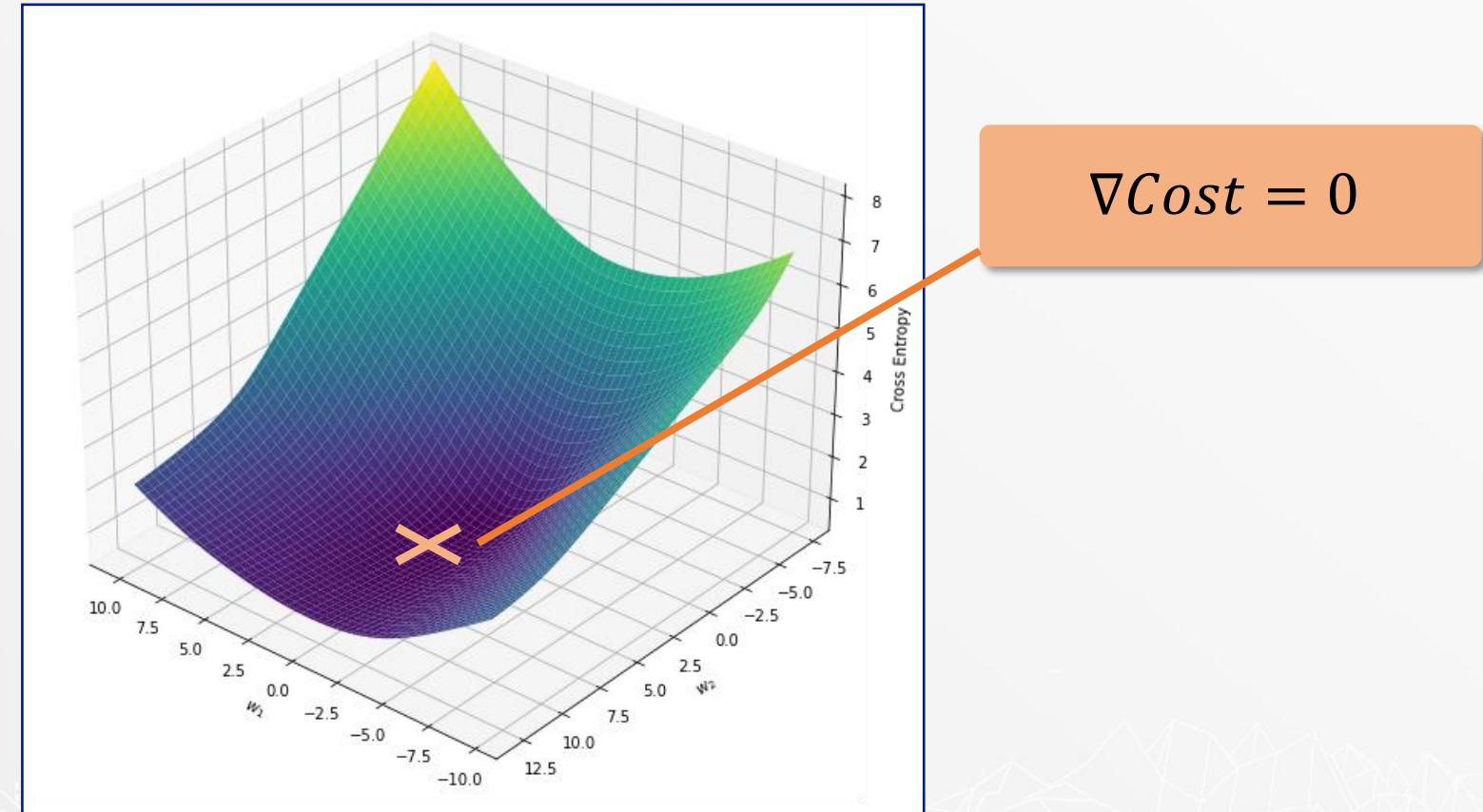
- Problem with Least Squares Method
- Gradient Descent (Logistic Regression)
- Calculation Example

# Problem with Least Squares Method

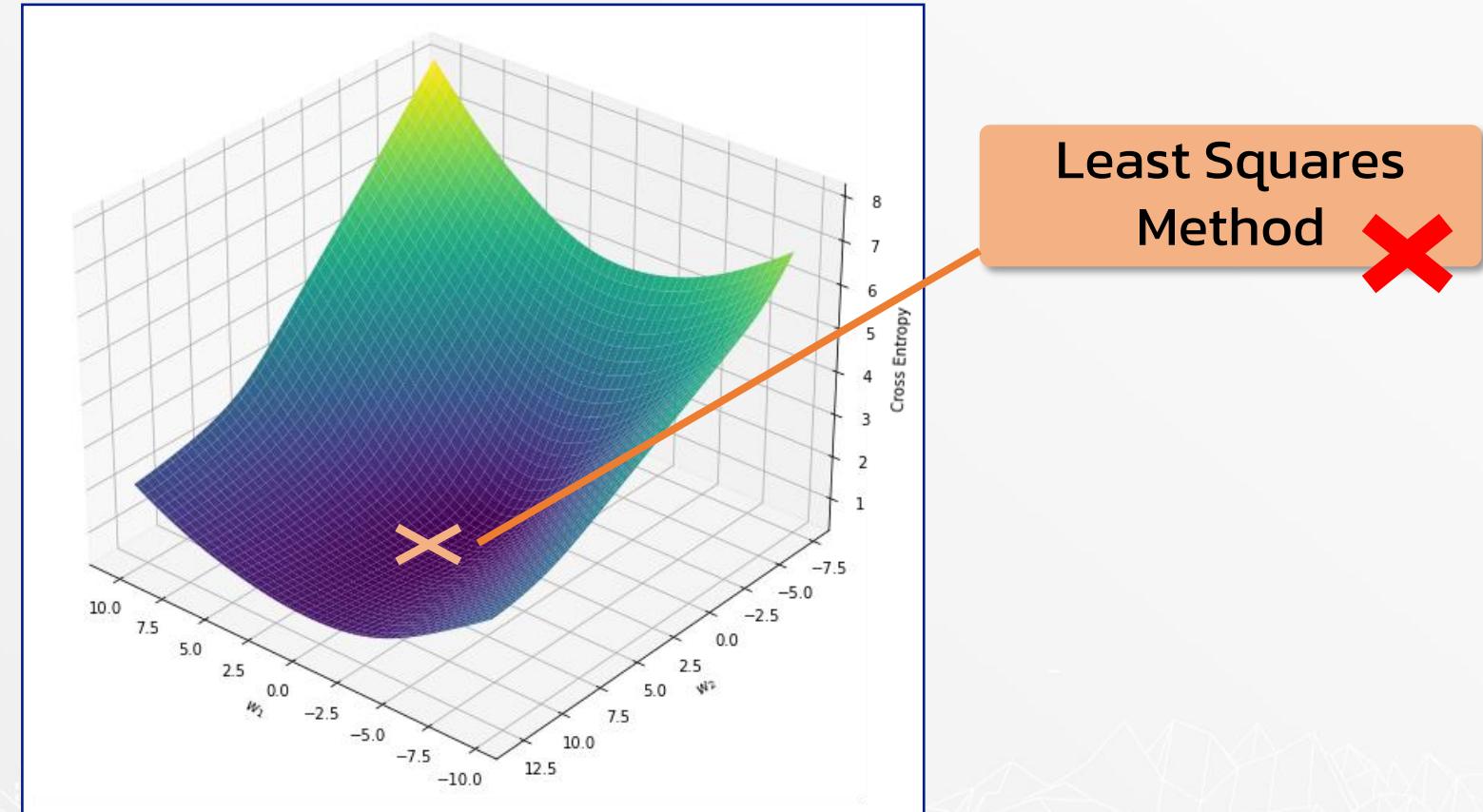


“ เราต้องการหา  $w_0, w_1, w_2, \dots, w_p$  ที่ทำให้ cost function ต่ำที่สุด ”

# Problem with Least Squares Method



# Problem with Least Squares Method



# Problem with Least Squares Method

การที่ logistic regression ไม่มี closed form solution เป็นผลเนื่องมาจากการที่ติดอยู่กับ  $z$  อยู่ในรูปของเลขซึ่งกำลังไม่สามารถถดงลงมาเพื่อแก้สมการได้

$$X_b^T \hat{\mathbf{y}} = X_b^T \mathbf{y}$$

$$\hat{\mathbf{y}} = \begin{bmatrix} \frac{1}{1+e^{-z_1}} \\ \vdots \\ \frac{1}{1+e^{-z_n}} \end{bmatrix} = \begin{bmatrix} \frac{1}{1+e^{-(w_0+w_1x_{1,1}+w_2x_{1,2}+\dots+w_px_{1,p})}} \\ \vdots \\ \frac{1}{1+e^{-(w_0+w_1x_{n,1}+w_2x_{n,2}+\dots+w_px_{n,p})}} \end{bmatrix}$$



# Problem with Least Squares Method

Note : Linear Regression

$$X_b^T \hat{\mathbf{y}} = X_b^T \hat{\mathbf{y}}$$

$$X_b^T X_b \mathbf{w} = X_b^T \hat{\mathbf{y}} \quad (\because \mathbf{z} = X_b \mathbf{w})$$

$$\mathbf{w} = (X_b^T X_b)^{-1} X_b^T \hat{\mathbf{y}}$$

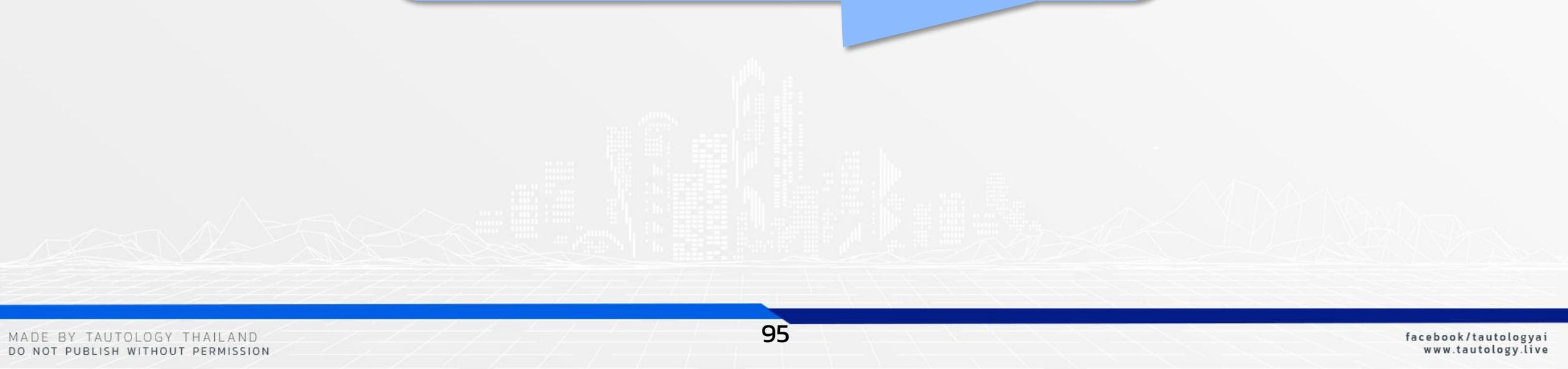


# Problem with Least Squares Method

**Q : แล้วมีวิธีอื่นในการหาค่าตอบใหม่ ?**

# Problem with Least Squares Method

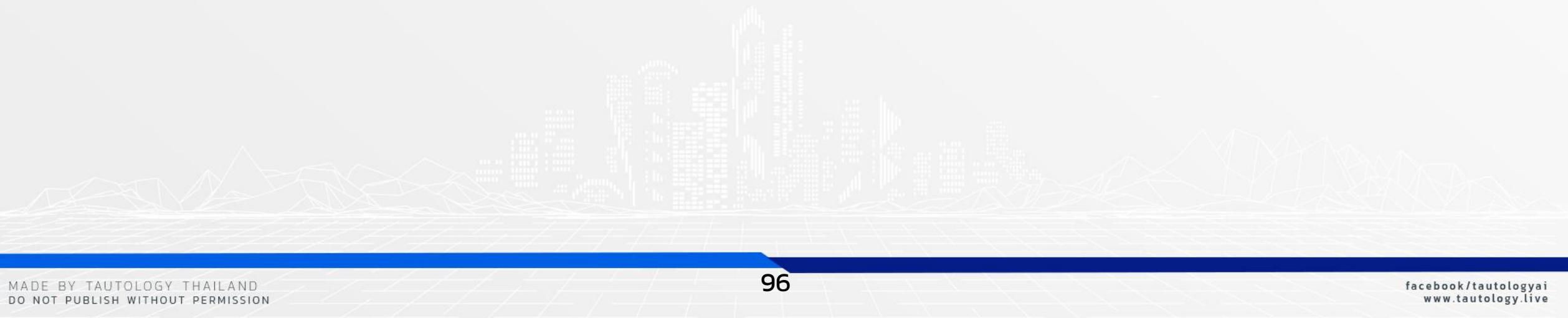
A : มี แล้ววิธีการนั้นมีชื่อว่า  
**Gradient Descent**



# How to Create Model (Math)

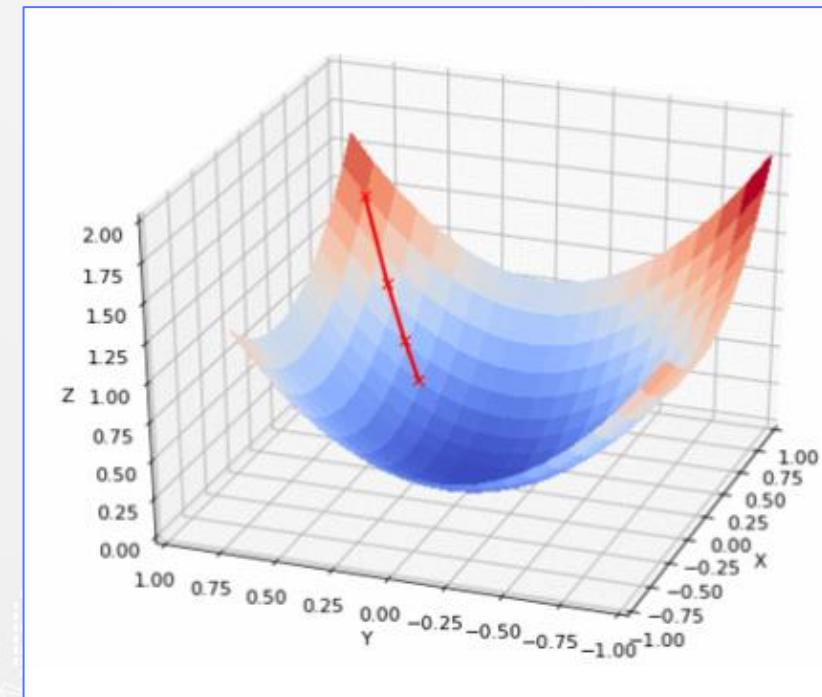
## **Problem with Least Squares Method**

- Gradient Descent (Logistic Regression)
- Calculation Example



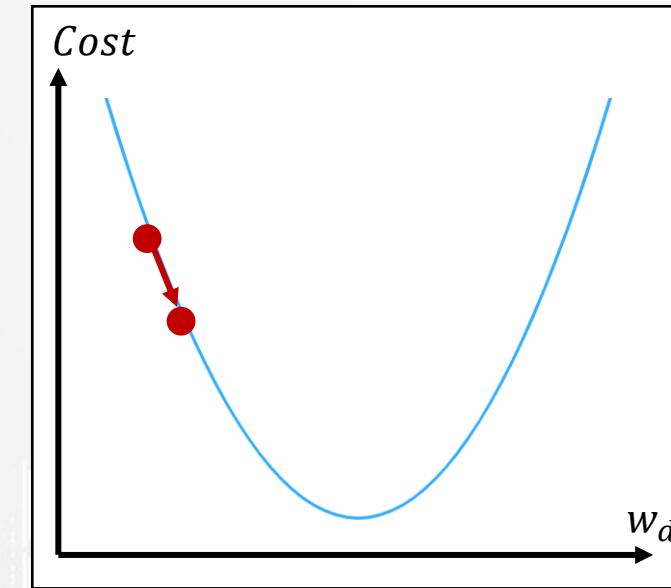
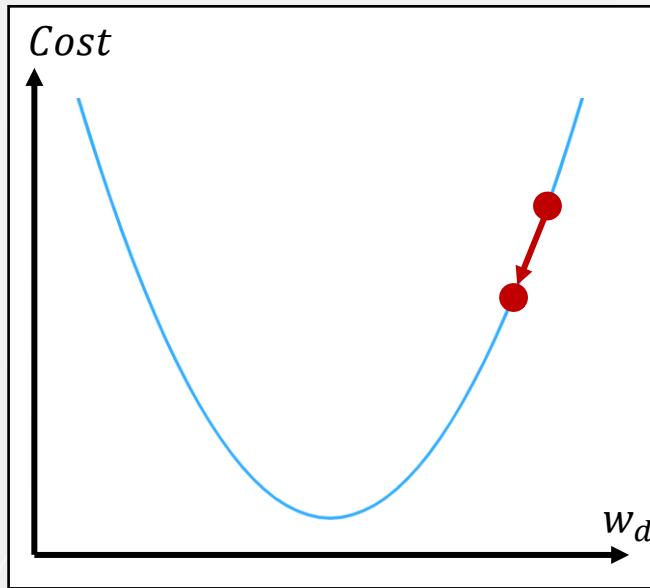
# Gradient Descent (Logistic Regression)

“ Gradient Descent ”



# Gradient Descent (Logistic Regression)

“ Gradient Descent ”



# Gradient Descent (Logistic Regression)

## Logic of Gradient Descent

- ถ้าความซันเป็น  $+$   $\rightarrow$  ลดค่า  $w_d$
- ถ้าความซันเป็น  $-$   $\rightarrow$  เพิ่มค่า  $w_d$

# Gradient Descent (Logistic Regression)

## Equation of Gradient Descent

$$w_d = w_d - \alpha \frac{\partial Cost}{\partial w_d}$$

โดย ◆  $\alpha$  คือ ค่าที่ใช้ควบคุม step size ของ  $w_d$  ( $> 0$ )

# Gradient Descent (Logistic Regression)

เมื่อพิจารณาในทุกมิติพร้อมกัน จะได้ว่า

$$w_0 = w_0 - \alpha \frac{\partial Cost}{\partial w_0}$$

$$w_1 = w_1 - \alpha \frac{\partial Cost}{\partial w_1}$$

⋮

$$w_p = w_p - \alpha \frac{\partial Cost}{\partial w_p}$$

# Gradient Descent (Logistic Regression)

เมื่อพิจารณาในทุกมิติพร้อมกัน จะได้ว่า

$$\begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial Cost}{\partial w_0} \\ \frac{\partial Cost}{\partial w_1} \\ \vdots \\ \frac{\partial Cost}{\partial w_p} \end{bmatrix}$$

# Gradient Descent (Logistic Regression)

## Equation of Gradient Descent

$$\mathbf{w} = \mathbf{w} - \alpha \nabla Cost$$

โดย   $\alpha$  คือ ค่าที่ใช้ควบคุม step size ของ  $\mathbf{w}$

# Gradient Descent (Logistic Regression)

## Equation of Gradient Descent for Logistic Regression

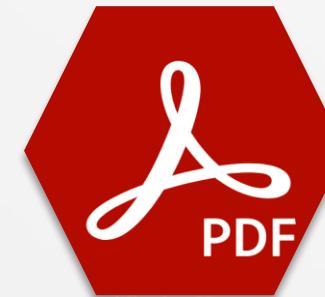
$$\mathbf{w} = \mathbf{w} + \frac{\alpha}{n} X_b^T (\mathbf{y} - \hat{\mathbf{y}})$$

โดย   $\alpha$  คือ ค่าที่ใช้ควบคุม step size ของ  $\mathbf{w}$

# Gradient Descent (Logistic Regression)



Derivation of Gradient Descent  
for Logistic Regression



Open File  
**Derive\_GD\_LoR.pdf**

# Gradient Descent (Logistic Regression)

## Note

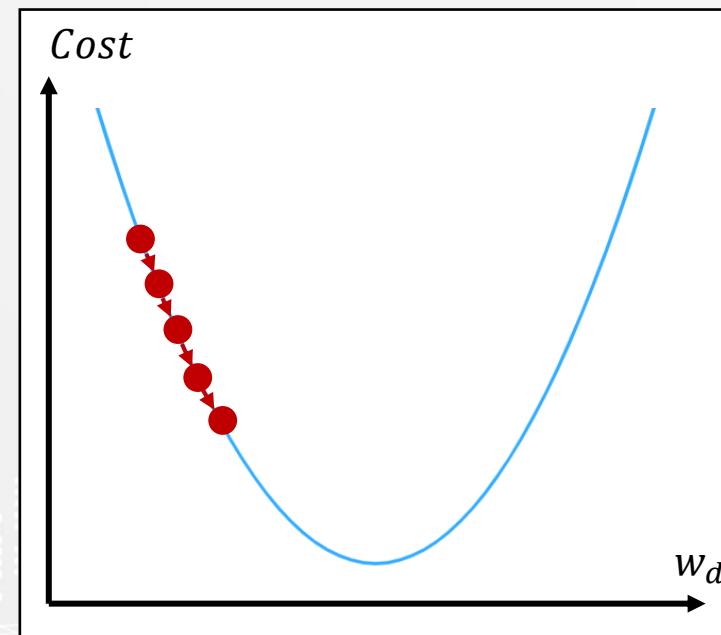
$$\mathbf{w} = \mathbf{w} + \frac{\alpha}{n} \mathbf{X}_b^T (\mathbf{y} - \hat{\mathbf{y}})$$

โดย ◆  $\alpha$  คือ ค่าที่ใช้ควบคุม step size ของ  $\mathbf{w}$

# Gradient Descent (Logistic Regression)

## Note

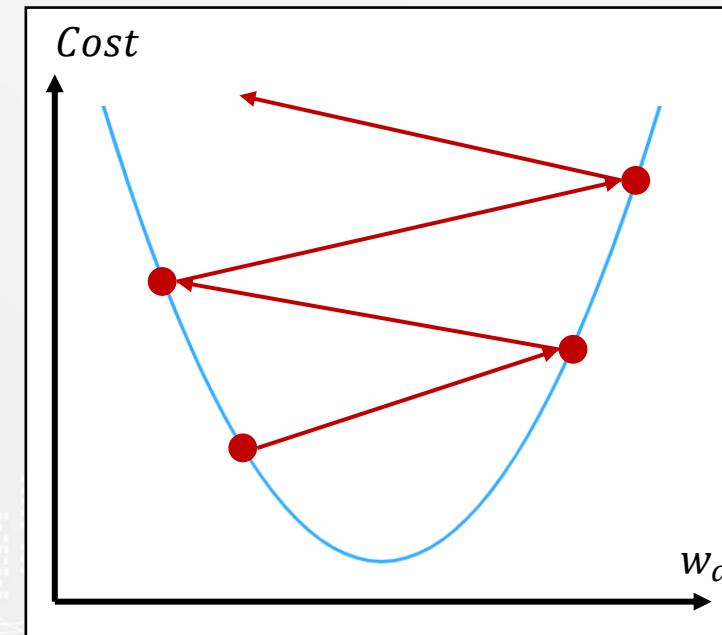
ถ้า  $\alpha$  มีค่าน้อยเกินไป : GD จะลุ่เข้าสู่คำตอบซ้ำ



# Gradient Descent (Logistic Regression)

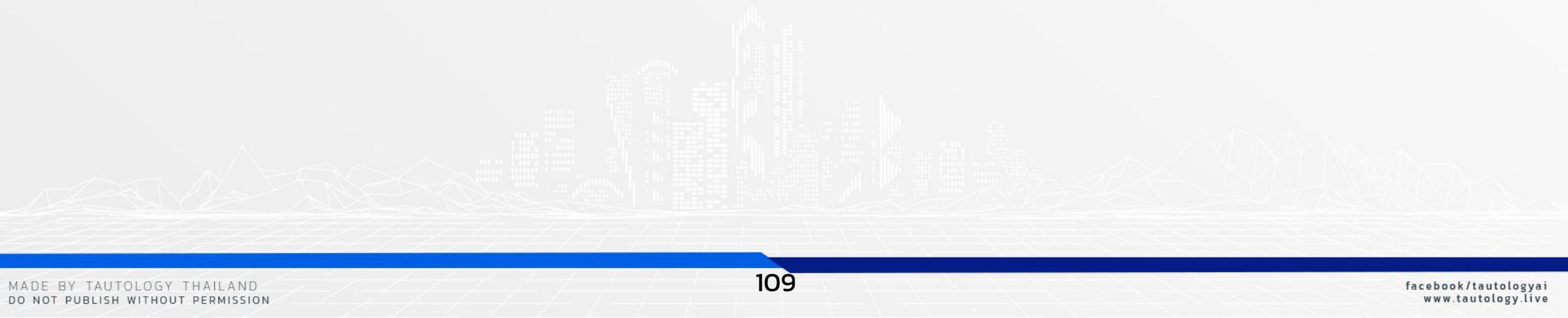
## Note

ถ้า  $\alpha$  มีค่ามากเกินไป : GD จะลุ่วออกจากคำตوب (overflow)



# How to Create Model (Math)

- Problem with Least Squares Method**
- Gradient Descent (Logistic Regression)**
- Calculation Example



# Calculation Example

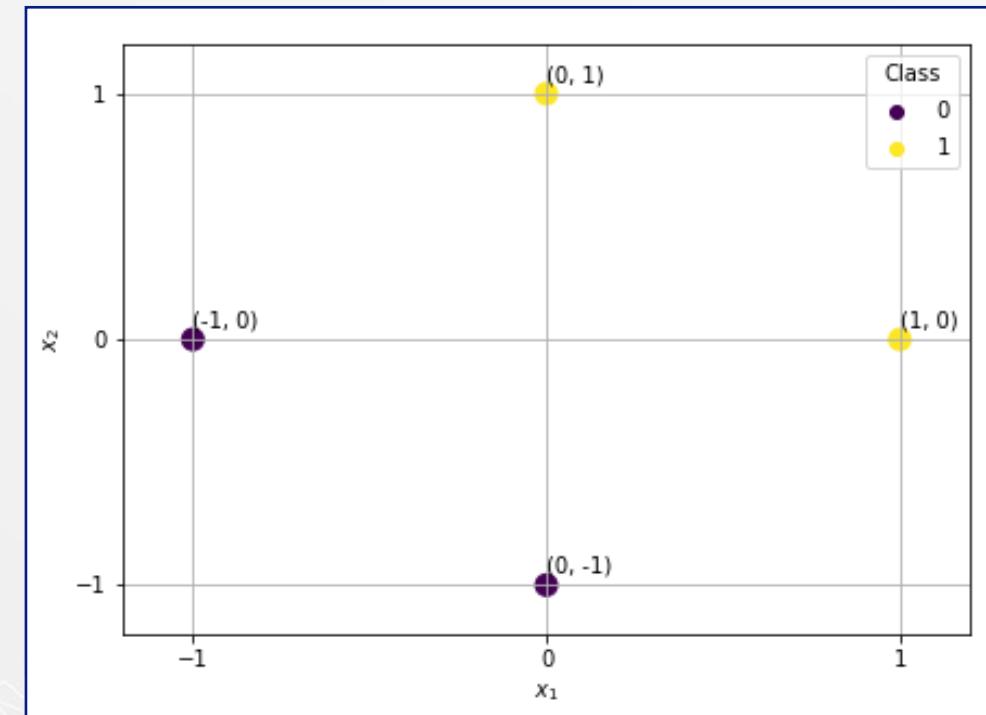
**ตัวอย่างการคำนวณ w สำหรับ logistic regression**  
**ด้วย gradient descent**

$x_1$	$x_2$	y
0	1	1
1	0	1
-1	0	0
0	-1	0

ตารางแสดง dataset

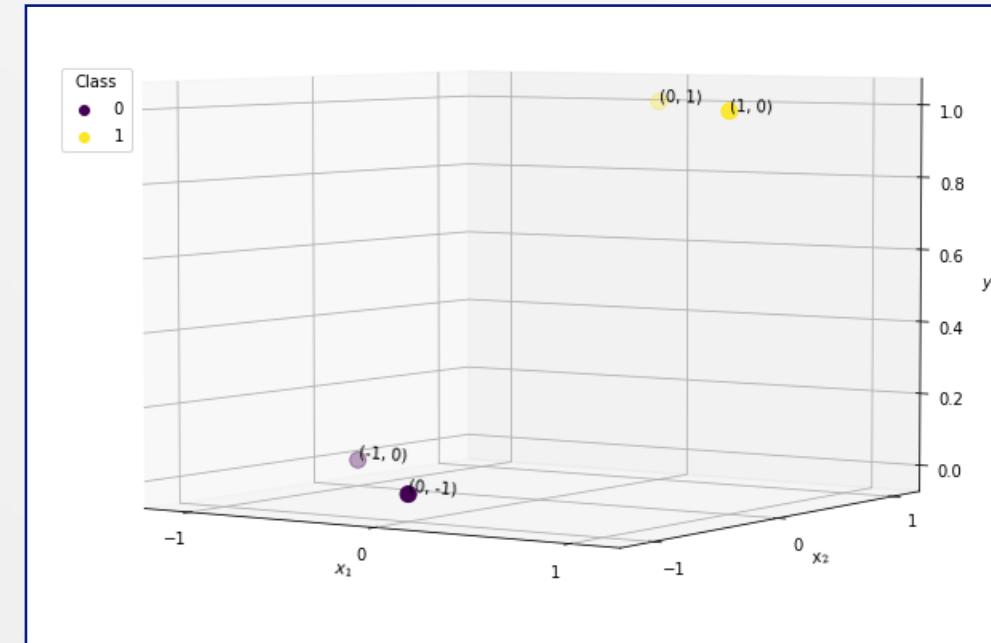
# Calculation Example

- จากข้อมูลใน dataset เราสามารถหาดالةกราฟ 2 มิติ ได้ดังนี้



# Calculation Example

- จากข้อมูลใน dataset เราสามารถหาดالةฟังก์ชัน 3 มิติ ได้ดังนี้



# Calculation Example

## ขั้นตอนในการคำนวณ gradient descent

1. สุ่ม  $w$  เริ่มต้น
2. กำหนดจำนวนรอบที่จะ update  $w$  (epoch)
3. กำหนดค่า  $\alpha$
4. For loop เพื่อ update  $w$ 
  - a) คำนวณ  $z$
  - b) คำนวณ  $\hat{y}$
  - c) คำนวณ  $w$

# Calculation Example

## 1. สรุป $w$ เริ่มต้น

สมมติให้สรุป  $w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

# Calculation Example

2. กำหนดจำนวนรอบที่จะ update w (epoch)

กำหนดให้ epoch = 3

# Calculation Example

## 3. กำหนดค่า $\alpha$

กำหนดให้  $\alpha = 5$

# Calculation Example

## 4. Update $w$

$$w = w + \frac{\alpha}{n} X_b^T (y - \hat{y})$$

โดย  $\alpha$  คือ ค่าที่ใช้ควบคุม step size ของ  $w$

# Calculation Example

จากข้อมูลใน dataset เราสามารถเขียน  $X, y$  และ  $X_b$  ได้ดังต่อไปนี้

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad y = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{ และ } X_b = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

# Calculation Example

## 4. Update w : คำนวณ z

epoch = 1

$$\mathbf{z} = \mathbf{X}_b \mathbf{w} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

# Calculation Example

## 4. Update w : คำนวณ $\hat{y}$

epoch = 1

$$\hat{y} = \begin{bmatrix} \frac{1}{1+e^{-z_1}} \\ \frac{1}{1+e^{-z_2}} \\ \frac{1}{1+e^{-z_3}} \\ \frac{1}{1+e^{-z_4}} \end{bmatrix} = \begin{bmatrix} \frac{1}{1+e^{-1}} \\ \frac{1}{1+e^0} \\ \frac{1}{1+e^0} \\ \frac{1}{1+e^1} \end{bmatrix} = \begin{bmatrix} 0.731 \\ 0.5 \\ 0.5 \\ 0.269 \end{bmatrix}$$

# Calculation Example

## 4. Update $w$ : คำนวณ $w$

epoch = 1

$$\begin{aligned} w &= w + \frac{\alpha}{n} X_b^T (y - \hat{y}) \\ &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{5}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \left( \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.731 \\ 0.5 \\ 0.5 \\ 0.269 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 1.25 \\ 1.672 \end{bmatrix} \end{aligned}$$

# Calculation Example

## 4. Update w : คำนวณ z

epoch = 2

$$\mathbf{z} = \mathbf{X}_b \mathbf{w} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1.25 \\ 1.672 \end{bmatrix} = \begin{bmatrix} 1.672 \\ 1.25 \\ -1.25 \\ -1.672 \end{bmatrix}$$

# Calculation Example

## 4. Update w : คำนวณ $\hat{y}$

epoch = 2

$$\hat{y} = \begin{bmatrix} \frac{1}{1+e^{-z_1}} \\ \frac{1}{1+e^{-z_2}} \\ \frac{1}{1+e^{-z_3}} \\ \frac{1}{1+e^{-z_4}} \end{bmatrix} = \begin{bmatrix} \frac{1}{1+e^{-1.672}} \\ \frac{1}{1+e^{-1.25}} \\ \frac{1}{1+e^{1.25}} \\ \frac{1}{1+e^{1.672}} \end{bmatrix} = \begin{bmatrix} 0.842 \\ 0.777 \\ 0.222 \\ 0.158 \end{bmatrix}$$

# Calculation Example

## 4. Update $w$ : คำนวณ $w$

epoch = 2

$$\begin{aligned} w &= w + \frac{\alpha}{n} X_b^T (y - \hat{y}) \\ &= \begin{bmatrix} 0 \\ 1.25 \\ 1.672 \end{bmatrix} + \frac{5}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \left( \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.842 \\ 0.777 \\ 0.222 \\ 0.158 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 1.807 \\ 2.068 \end{bmatrix} \end{aligned}$$

# Calculation Example

## 4. Update w : คำนวณ z

epoch = 3

$$\mathbf{z} = \mathbf{X}_b \mathbf{w} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1.807 \\ 2.068 \end{bmatrix} = \begin{bmatrix} 2.068 \\ 1.807 \\ -1.807 \\ -2.068 \end{bmatrix}$$

# Calculation Example

## 4. Update w : คำนวณ $\hat{y}$

epoch = 3

$$\hat{y} = \begin{bmatrix} \frac{1}{1+e^{-z_1}} \\ \frac{1}{1+e^{-z_2}} \\ \frac{1}{1+e^{-z_3}} \\ \frac{1}{1+e^{-z_4}} \end{bmatrix} = \begin{bmatrix} \frac{1}{1+e^{-2.068}} \\ \frac{1}{1+e^{-1.807}} \\ \frac{1}{1+e^{1.807}} \\ \frac{1}{1+e^{2.068}} \end{bmatrix} = \begin{bmatrix} 0.888 \\ 0.859 \\ 0.141 \\ 0.112 \end{bmatrix}$$

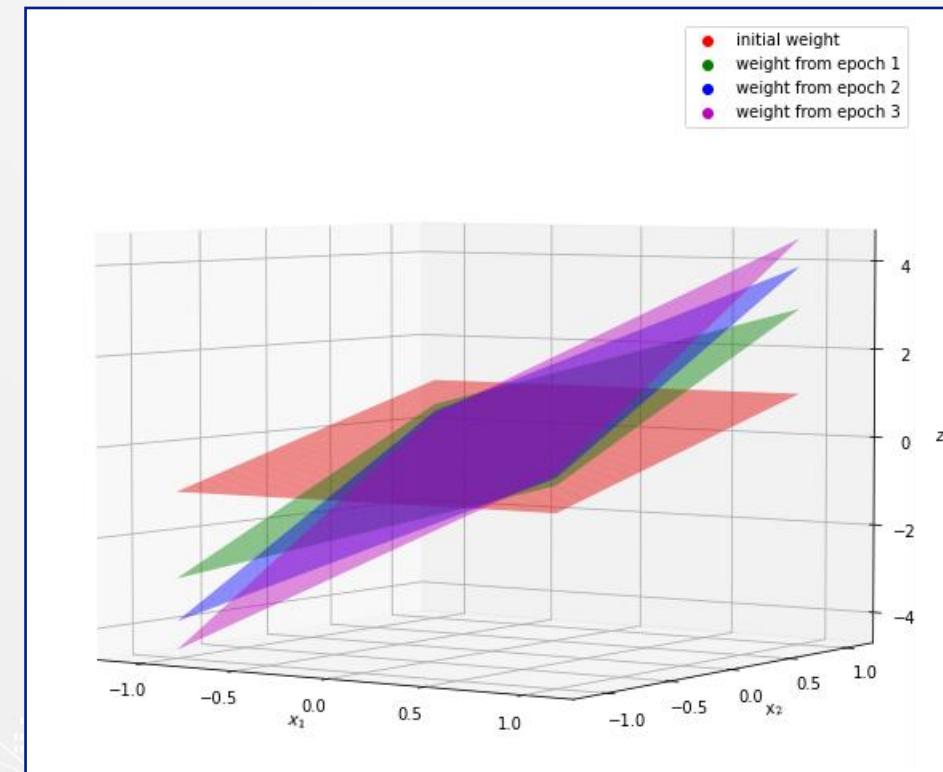
# Calculation Example

## 4. Update $w$ : คำนวณ $w$

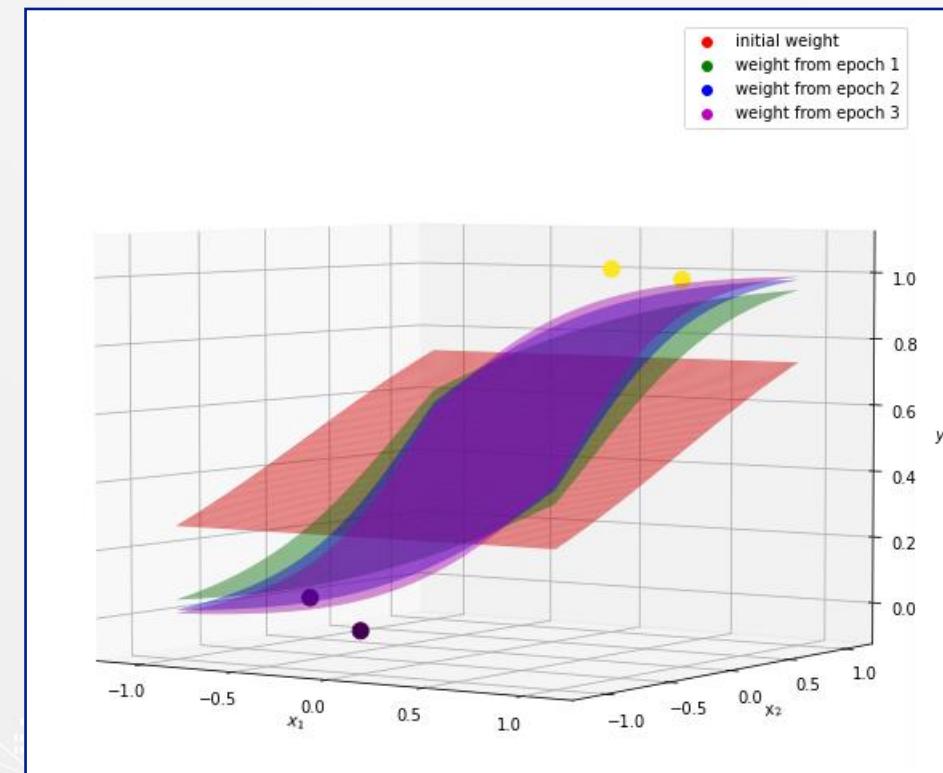
epoch = 3

$$\begin{aligned} w &= w + \frac{\alpha}{n} X_b^T (y - \hat{y}) \\ &= \begin{bmatrix} 0 \\ 1.807 \\ 2.068 \end{bmatrix} + \frac{5}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \left( \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.888 \\ 0.859 \\ 0.141 \\ 0.112 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 2.159 \\ 2.348 \end{bmatrix} \end{aligned}$$

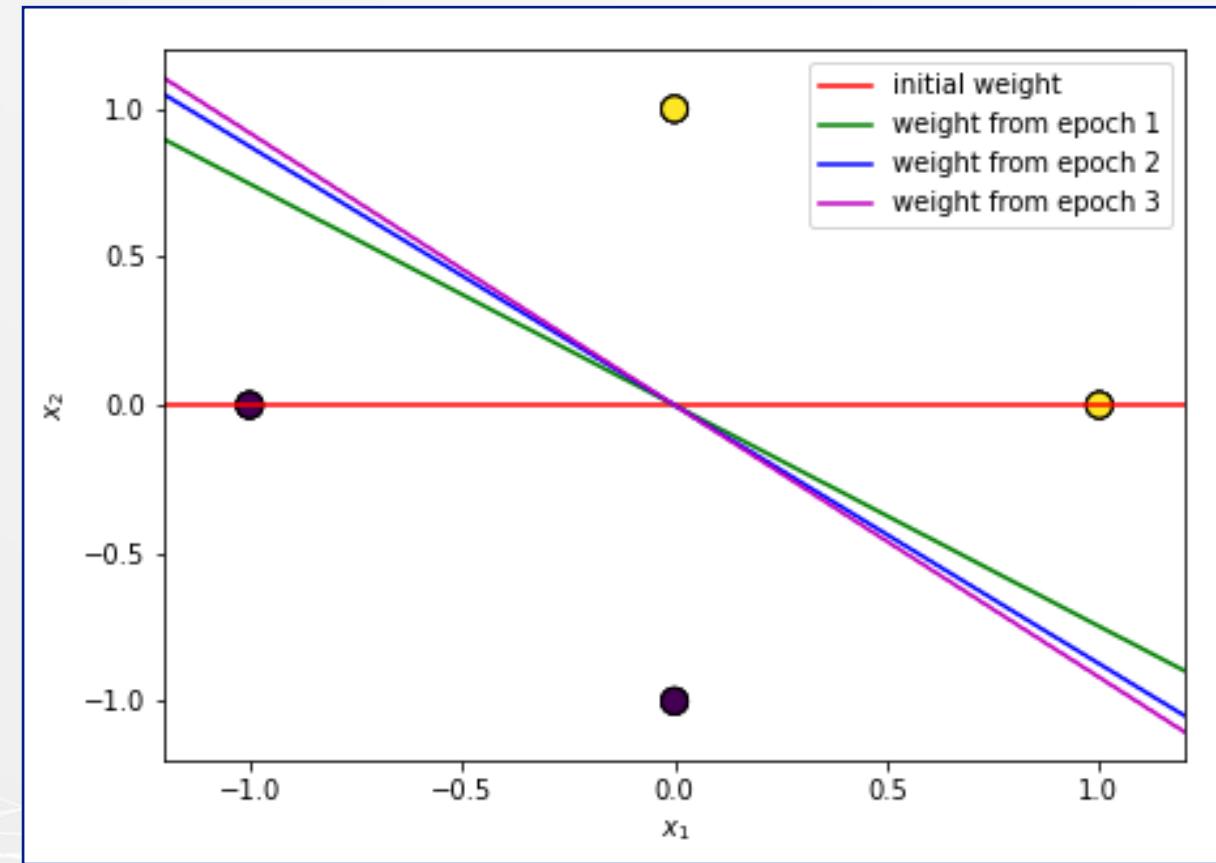
# Calculation Example



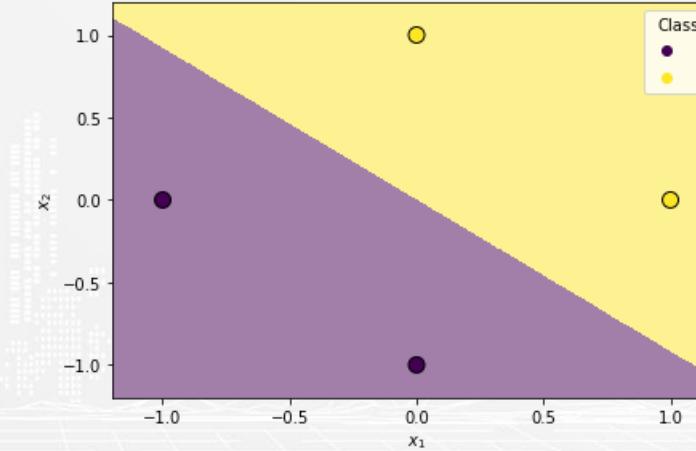
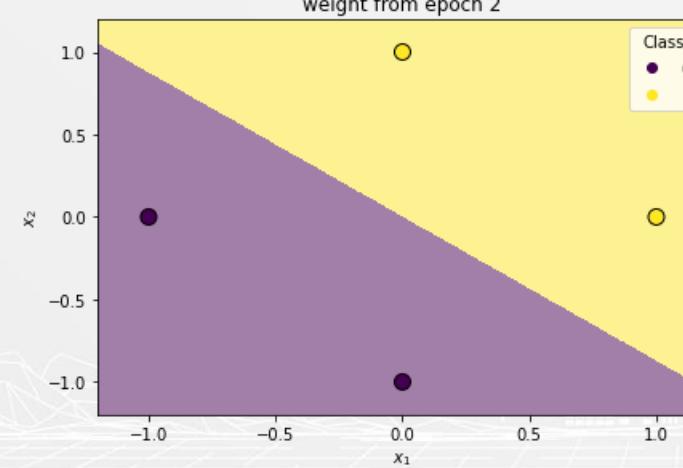
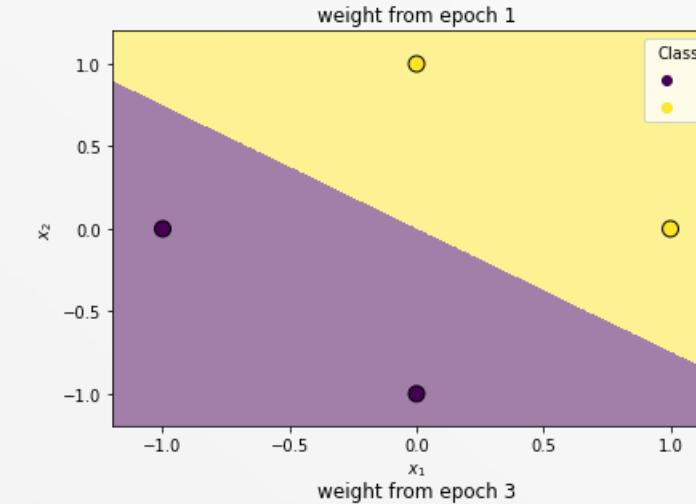
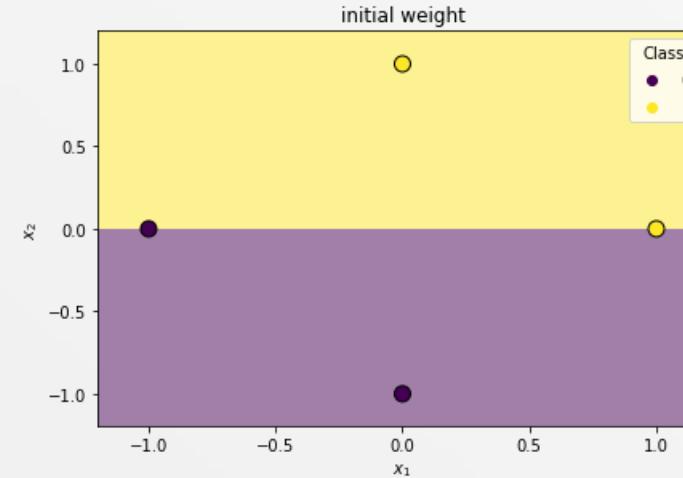
# Calculation Example



# Calculation Example



# Calculation Example



# Calculation Example



Exercise of Logistic Regression (Binary)



Open File  
**Exercise\_LoR\_(Binary).pdf**

# How to Create Model (Math)

- Problem with Least Squares Method**
- Gradient Descent (Logistic Regression)**
- Calculation Example**

# Model

**Assumption**



**Real Face of the Model**



**Cost Function and Cost Landscape**



**How to Create Model (Math)**



**How to Create Model (Code)**



# How to Create Model (Code)

ตัวอย่าง code สำหรับคำนวณ w

$x_1$	$x_2$	y
0	1	1
1	0	1
-1	0	0
0	-1	0

ตารางแสดง dataset

# How to Create Model (Code)

- Code สำหรับสร้าง model จากข้อมูลของเราโดยที่

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \text{ และ } y = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

```
1 clf = LogisticRegression(penalty='none')
2 clf.fit(X, y)
```

LogisticRegression(penalty='none')

# How to Create Model (Code)

- ค่า  $w_0$  ถูกเก็บไว้ใน attribute ชื่อ `intercept_`

```
1 clf.intercept_
array([0.])
```

# How to Create Model (Code)

- ค่า  $w_1, \dots, w_p$  จะเก็บไว้ใน attribute ชื่อ `coef_`

```
1 clf.coef_
```

```
array([[10.52183855, 10.52183855]])
```

# How to Create Model (Code)

ดังนั้น เราจะได้  $w_0 = 0, w_1 = 10.5218, w_2 = 10.5218$  ซึ่งสามารถเขียนเป็น model ของ logistic regression สำหรับข้อมูลชุดนี้ได้ดังนี้

$$z = 0 + 10.5218x_1 + 10.5218x_2$$

$$\hat{y} = \sigma(z)$$

# How to Create Model (Code)

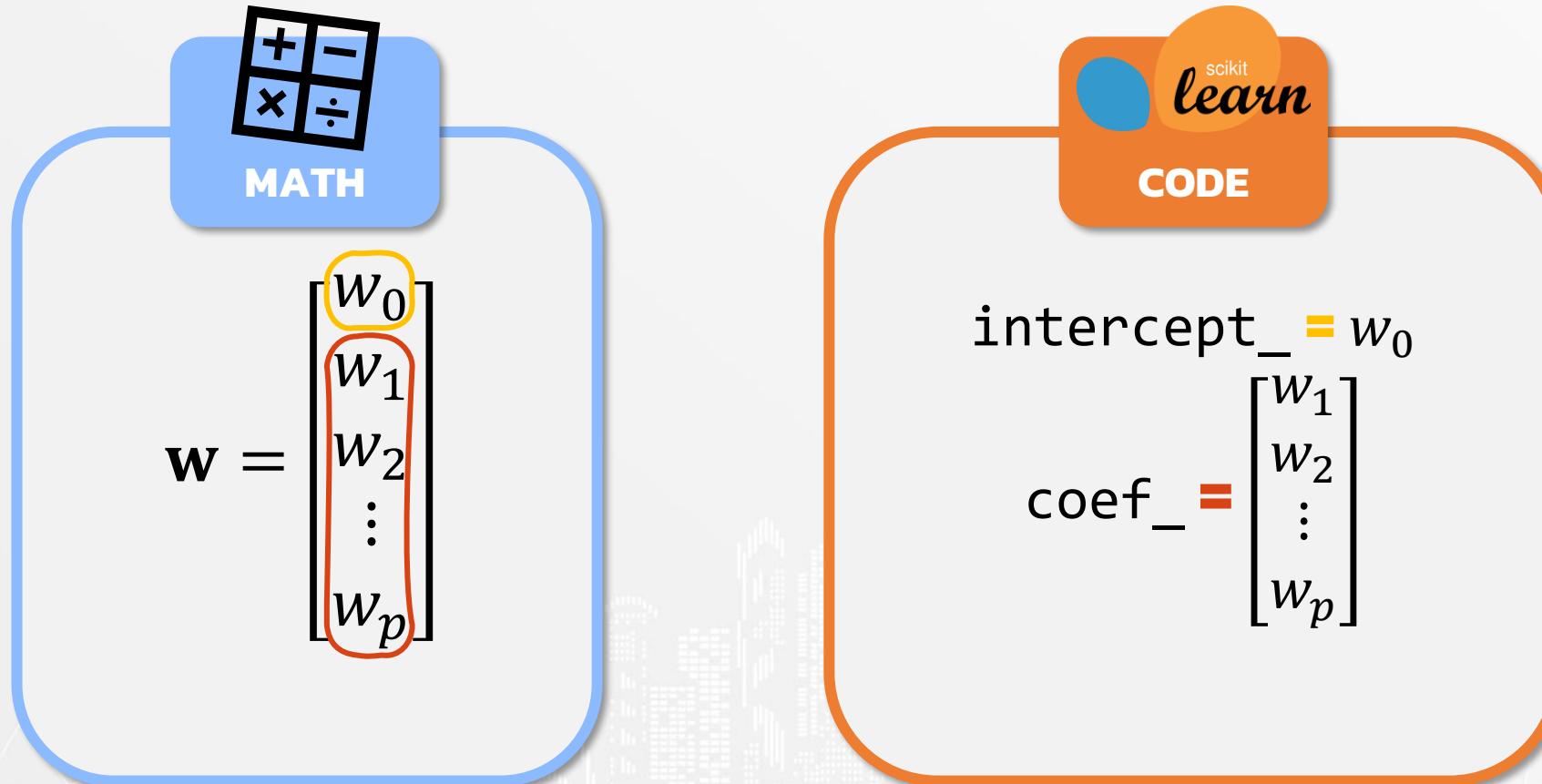


Code for this section

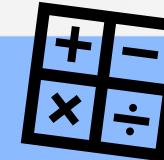


Open File  
**Model Creation (Binary).ipynb**

# How to Create Model (Code)



# How to Create Model (Code)



MATH

1. สูม  $w$  เริ่มต้น
2. กำหนดจำนวนรอบที่จะ  
update  $w$  (epoch)
3. กำหนดค่า  $\alpha$
4. for  $i$  in range(epoch):  
update  $w$

scikit  
learn

CODE

```
1 clf = LogisticRegression(penalty='none')
2 clf.fit(X, y)
```

LogisticRegression(penalty='none')

```
1 clf.intercept_
```

array([0.])

```
1 clf.coef_
```

array([[10.52183855, 10.52183855]])

# Model

**Assumption**



**Real Face of the Model**



**Cost Function and Cost Landscape**



**How to Create Model (Math)**



**How to Create Model (Code)**



# Logistic Regression (Binary Classification)



# Prediction

# Prediction

**Logistic regression** คือ โมเดลคณิตศาสตร์กี่จำแนกประเภทของข้อมูล ด้วยการสร้างสมการเชิงเส้น (linear regression)

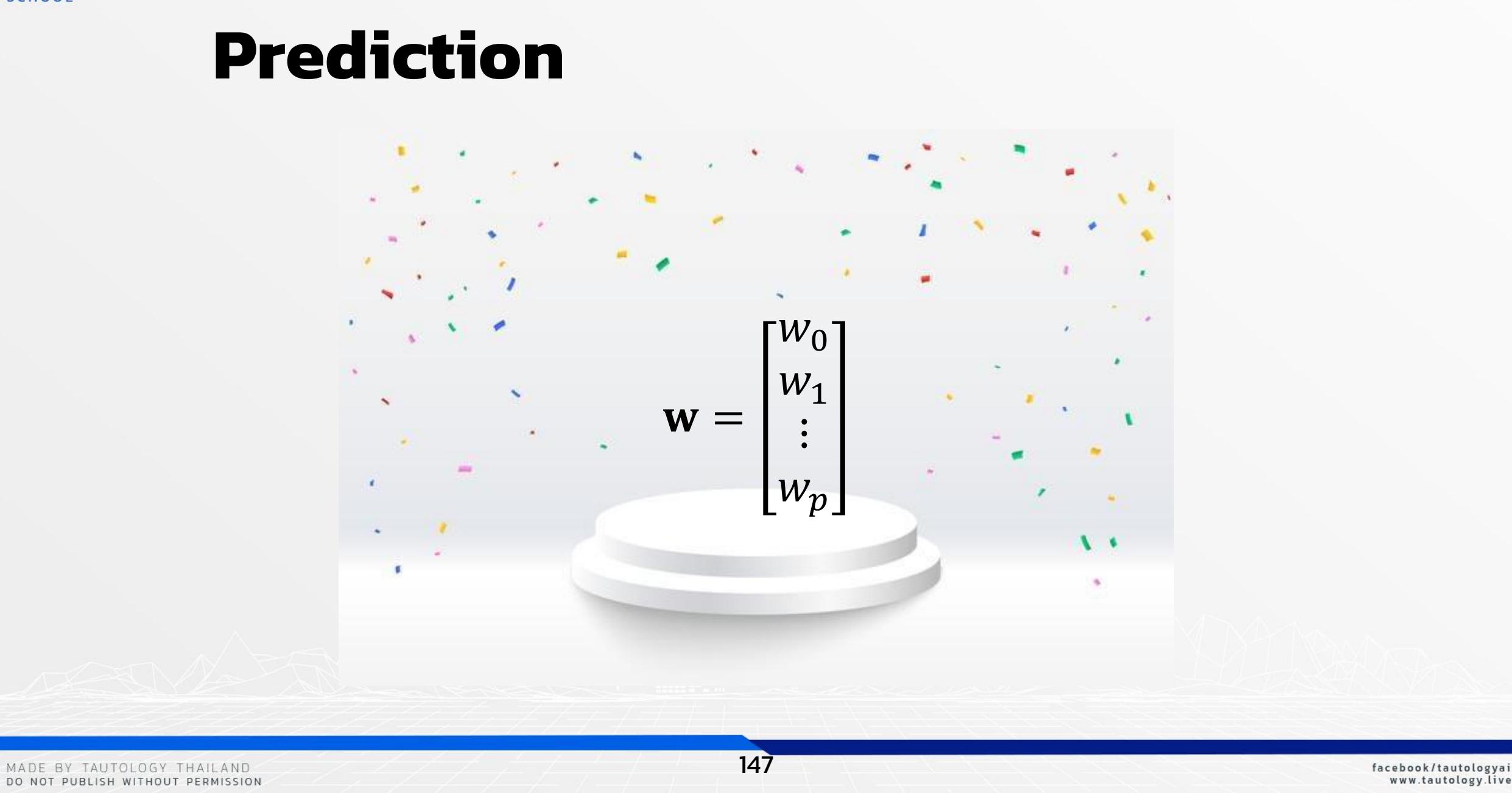
$$z = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p$$

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

- โดยที่
- ◆  $\hat{y}$  คือ ตัว預測 (predicted target)
  - ◆  $z$  คือ สมการเชิงเส้น (linear regression)
  - ◆  $x_1, x_2, x_3, \dots, x_p$  คือ ตัว特征 (feature)
  - ◆  $w_0, w_1, w_2, \dots, w_p$  คือ สัมประสิทธิ์ (coefficient)

# Prediction

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix}$$

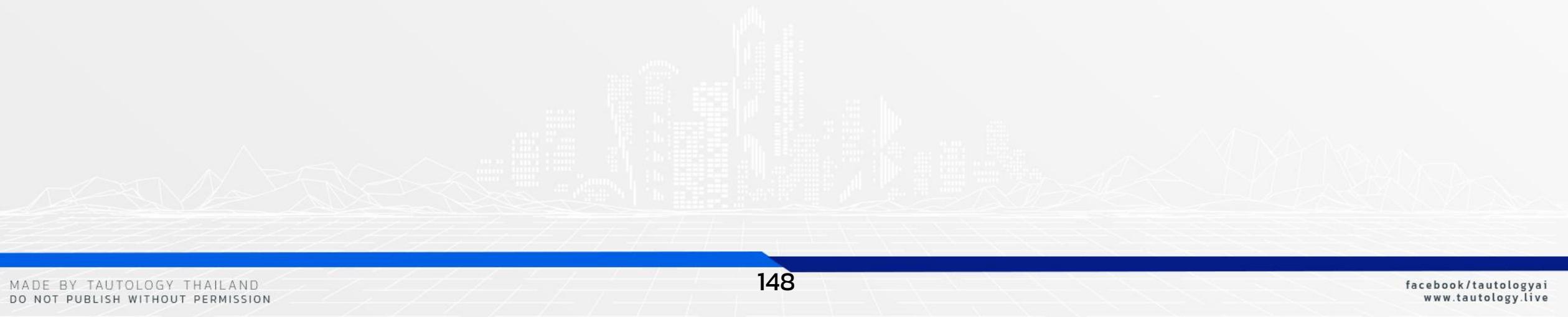


# Prediction

1-Sample

Multi-Sample

Code



# 1-Sample

ตัวอย่างการคำนวณ  $\hat{y}$



# 1-Sample

สมมติว่า  $\mathbf{w}$  ของปัญหานี้ที่เราหามาได้คือ

$$\mathbf{w} = \begin{bmatrix} 0 \\ 10.5218 \\ 10.5218 \end{bmatrix}$$

# 1-Sample

ซึ่งทำให้สามารถเขียนสมการ  $\hat{y}$  ได้ดังต่อไปนี้

$$\begin{aligned}\star \quad z &= 0 + 10.5218x_1 + 10.5218x_2 \\ &= 0 + 10.5218(0) + 10.5218(1.5) \\ &= 15.7827\end{aligned}$$

$$\star \quad \hat{y} = \frac{1}{1+e^z} = \frac{1}{1+e^{15.7827}} = 0.99999986015 \approx 1$$

# 1-Sample

ดังนั้น เราจะได้  $\hat{y}$  ของข้อมูลชุดนี้คือ



# Prediction

1-Sample



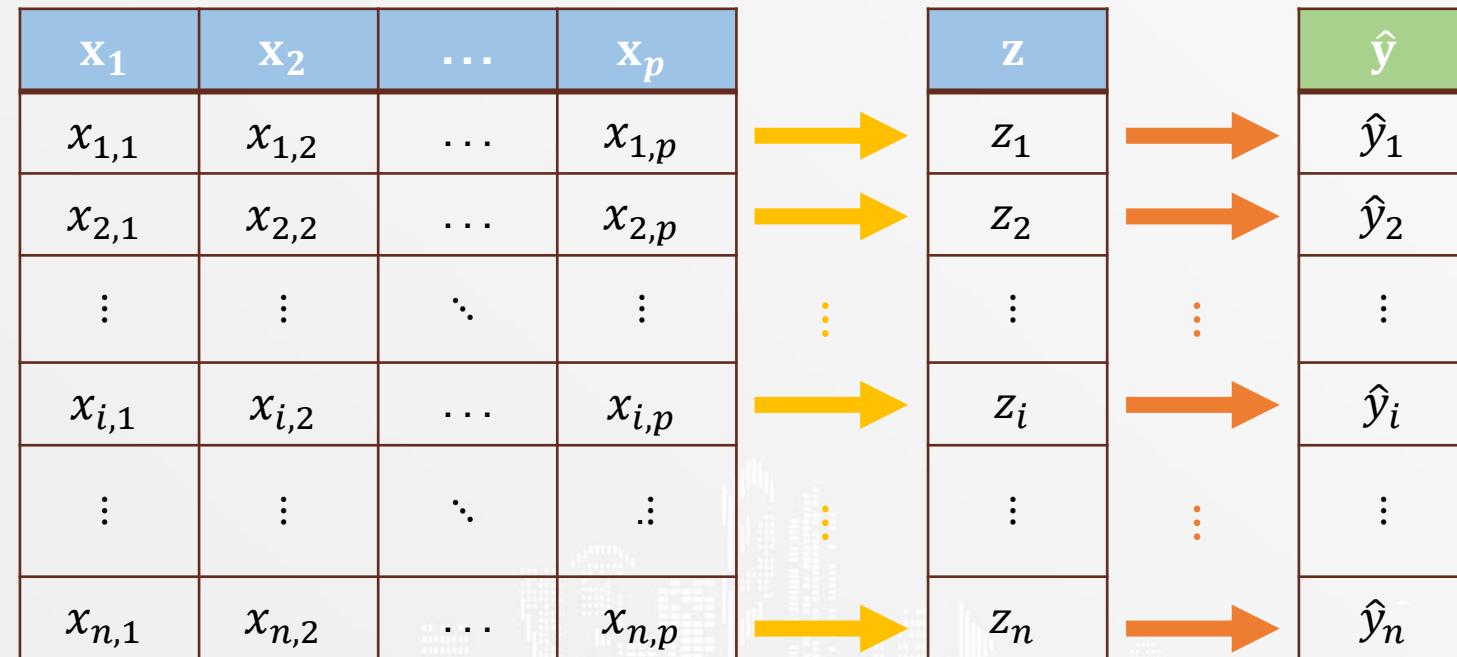
Multi-Sample



Code

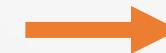


# Multi-Sample



# Multi-Sample

$$z = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p$$



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



$$z_1 = w_0 + w_1 x_{1,1} + w_2 x_{1,2} + \cdots + w_p x_{1,p}$$



$$\hat{y}_1 = \frac{1}{1 + e^{-z_1}}$$

$$z_2 = w_0 + w_1 x_{2,1} + w_2 x_{2,2} + \cdots + w_p x_{2,p}$$



$$\hat{y}_2 = \frac{1}{1 + e^{-z_2}}$$

⋮

$$z_i = w_0 + w_1 x_{i,1} + w_2 x_{i,2} + \cdots + w_p x_{i,p}$$



$$\hat{y}_i = \frac{1}{1 + e^{-z_i}}$$

⋮

$$z_n = w_0 + w_1 x_{n,1} + w_2 x_{n,2} + \cdots + w_p x_{n,p}$$



$$\hat{y}_n = \frac{1}{1 + e^{-z_n}}$$

# Multi-Sample

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix} \rightarrow \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} \frac{1}{1+e^{-z_1}} \\ \frac{1}{1+e^{-z_2}} \\ \vdots \\ \frac{1}{1+e^{-z_n}} \end{bmatrix}$$

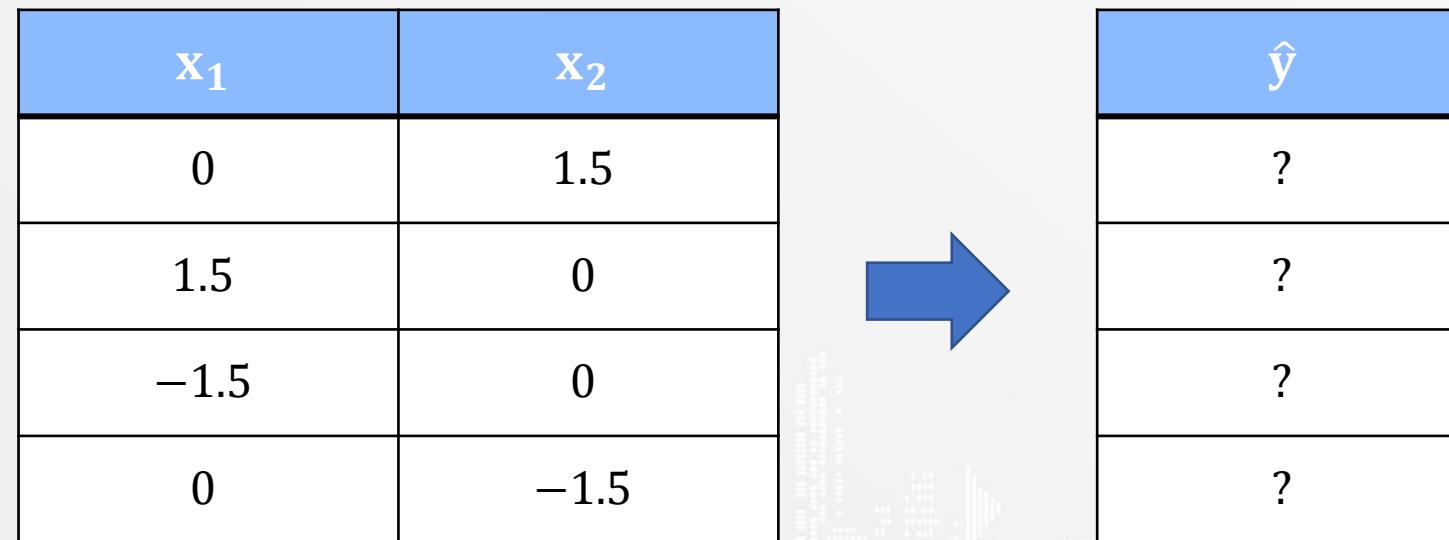
# Multi-Sample

$$\mathbf{z} = X_b \mathbf{w} \rightarrow \hat{\mathbf{y}} = \begin{bmatrix} \frac{1}{1+e^{-z_1}} \\ \frac{1}{1+e^{-z_2}} \\ \vdots \\ \frac{1}{1+e^{-z_n}} \end{bmatrix}$$

โดยที่  $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}$ ,  $X_b = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}$ ,  $\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix}$ ,  $\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}$

# Multi-Sample

ตัวอย่างการคำนวณ  $\hat{y}$



$x_1$	$x_2$	$\hat{y}$
0	1.5	?
1.5	0	?
-1.5	0	?
0	-1.5	?

# Multi-Sample

- สมมติว่า  $w$  ของปัญหานี้ที่เราหมายได้คือ

$$w = \begin{bmatrix} 0 \\ 10.5218 \\ 10.5218 \end{bmatrix}$$

- และจากข้อมูลใน dataset เราสามารถเขียน  $X_b$  ได้ดังต่อไปนี้

$$X_b = \begin{bmatrix} 1 & 0 & 1.5 \\ 1 & 1.5 & 0 \\ 1 & -1.5 & 0 \\ 1 & 0 & -1.5 \end{bmatrix}$$

# Multi-Sample

เราคำนวณค่า  $z$  ได้จาก  $z = X_b w$

$$\begin{aligned} z &= \begin{bmatrix} 1 & 0 & 1.5 \\ 1 & 1.5 & 0 \\ 1 & -1.5 & 0 \\ 1 & 0 & -1.5 \end{bmatrix} \begin{bmatrix} 0 \\ 10.5218 \\ 10.5218 \end{bmatrix} \\ &= \begin{bmatrix} 15.7872 \\ 15.7872 \\ -15.7872 \\ -15.7872 \end{bmatrix} \end{aligned}$$

# Multi-Sample

จาก  $\hat{y} = \frac{1}{1+e^{-z}}$  จะได้ว่า

$$\hat{y} = \begin{bmatrix} \frac{1}{1+e^{-15.7872}} \\ \frac{1}{1+e^{-15.7872}} \\ \frac{1}{1+e^{-15.7872}} \\ \frac{1}{1+e^{15.7872}} \\ \frac{1}{1+e^{15.7872}} \end{bmatrix} = \begin{bmatrix} 0.99 \\ 0.99 \\ 0 \\ 0 \end{bmatrix}$$

# Multi-Sample

ดังนั้น เราจะได้  $\hat{y}$  สำหรับข้อมูลชุดนี้คือ

$x_1$	$x_2$
0	1.5
1.5	0
-1.5	0
0	-1.5



$\hat{y}$
0.99
0.99
0
0

# Prediction

1-Sample



Multi-Sample



Code



# Code

ตัวอย่าง code สำหรับหา  $\hat{y}$

$x_1$	$x_2$
0	1.5
1.5	0
-1.5	0
0	-1.5



$\hat{y}$
?
?
?
?

# Code

- Code สำหรับหา ŷ จากข้อมูลของเราโดยที่  $X = \begin{bmatrix} 0 & 1.5 \\ 1.5 & 0 \\ -1.5 & 0 \\ 0 & -1.5 \end{bmatrix}$

```
1 clf.predict(X)
```

```
array([1, 1, 0, 0])
```

# Code

- Code สำหรับหาค่า probability ของ sample ที่จะถูกพยากรณ์ในแต่ละ class

```
1 clf.predict_proba(X)
```

```
array([[1.39841254e-07, 9.99999860e-01],  
       [1.39841254e-07, 9.99999860e-01],  
       [9.99999860e-01, 1.39841254e-07],  
       [9.99999860e-01, 1.39841254e-07]])
```

# Code

ดังนั้น เราจะได้  $\hat{y}$  สำหรับข้อมูลชุดนี้คือ

$x_1$	$x_2$
0	1.5
1.5	0
-1.5	0
0	-1.5



$\hat{y}$
1
1
0
0

# Code

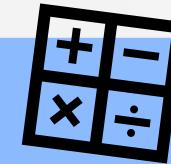


Code for this section



Open File  
**Model Creation (Binary).ipynb**

# Code



MATH

$$\mathbf{z} = \mathbf{X}_b \mathbf{w}$$

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



CODE

```
1 clf.predict(X)
```

```
array([1, 1, 0, 0])
```

# Prediction

1-Sample



Multi-Sample



Code



# Logistic Regression (Binary Classification)



# Improvement

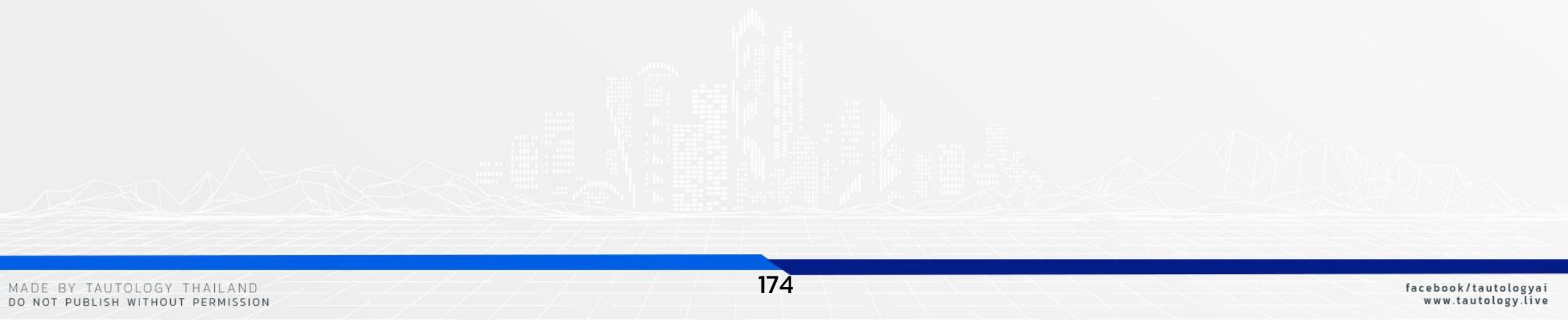
# Improvement

Imbalanced class

Regularization

# Imbalanced class

- Problem with Imbalanced Class
- Solution

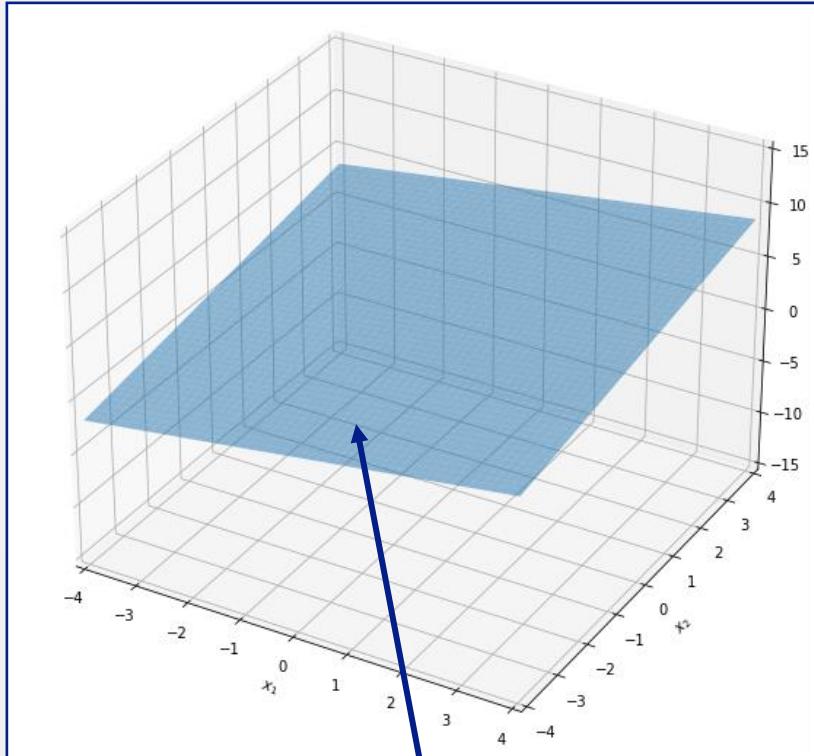


# Problem with Imbalanced Class

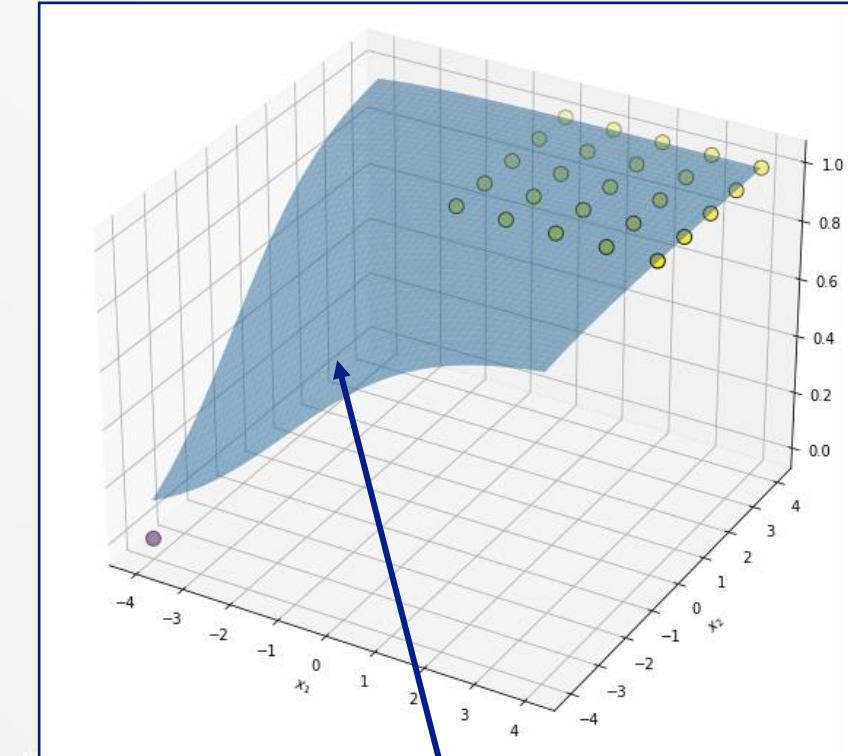
$x_1$	$x_2$	$y$
0	0	1
0	1	1
0	2	1
0	3	1
0	4	1
1	0	1
1	1	1
1	2	1
1	3	1
1	4	1
2	0	1
2	1	1
2	2	1

$x_1$	$x_2$	$y$
2	3	1
2	4	1
3	0	1
3	1	1
3	2	1
3	3	1
3	4	1
4	0	1
4	1	1
4	2	1
4	3	1
4	4	1
-4	-4	0

# Problem with Imbalanced Class

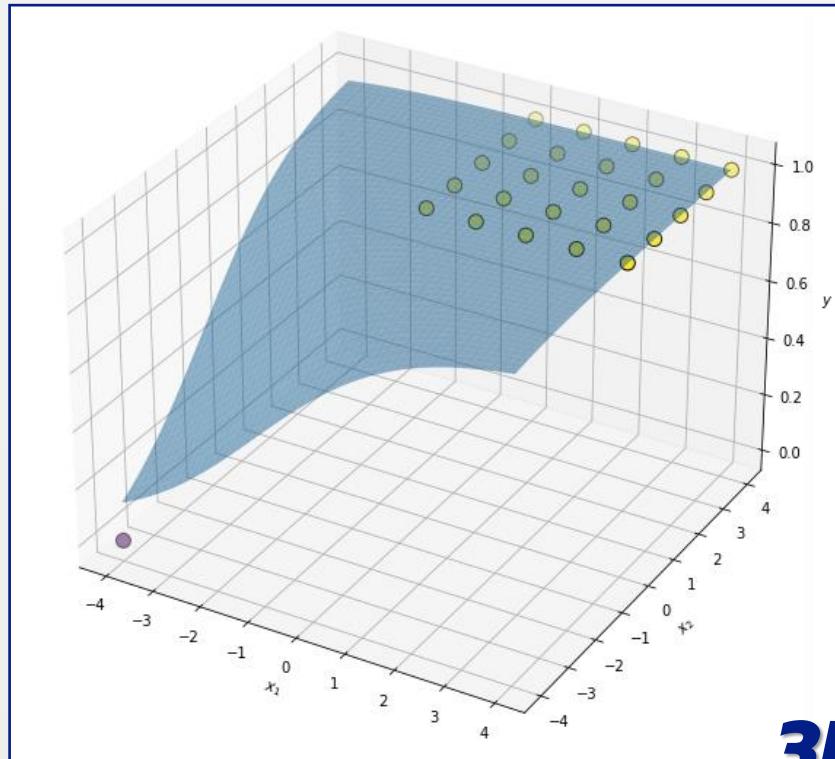


$$z = w_0 + w_1 x_1 + w_2 x_2$$

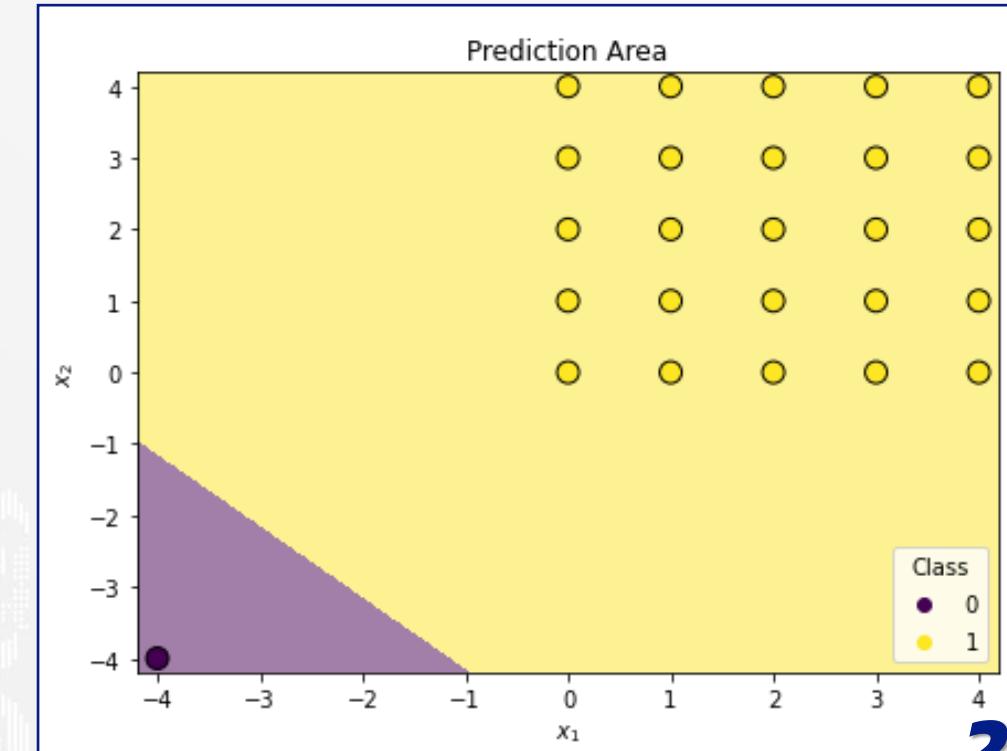


$$\hat{y} = \sigma(z)$$

# Problem with Imbalanced Class

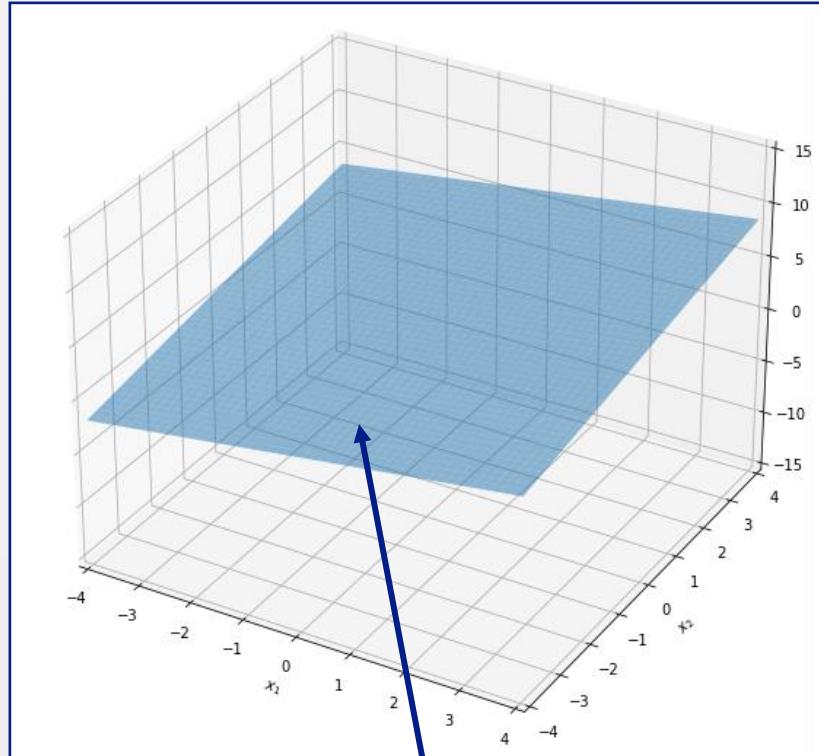


3D

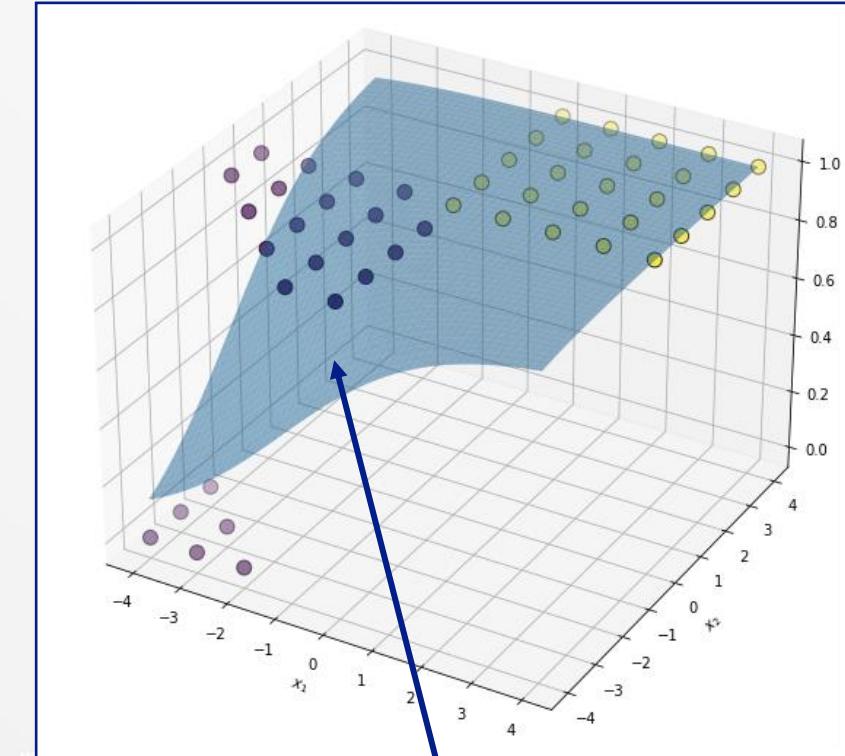


2D

# Problem with Imbalanced Class

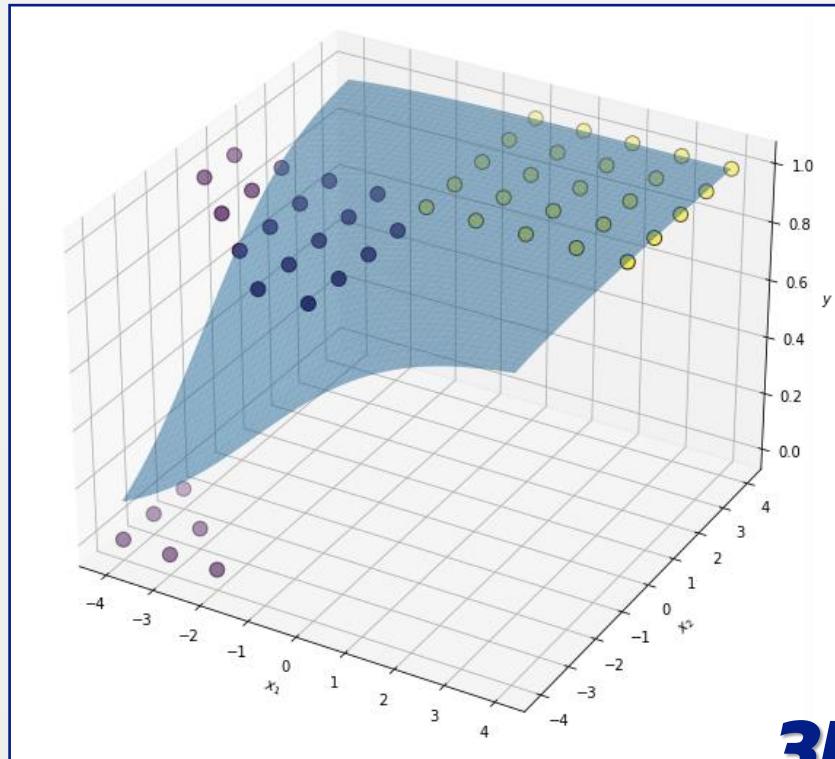


$$z = w_0 + w_1 x_1 + w_2 x_2$$

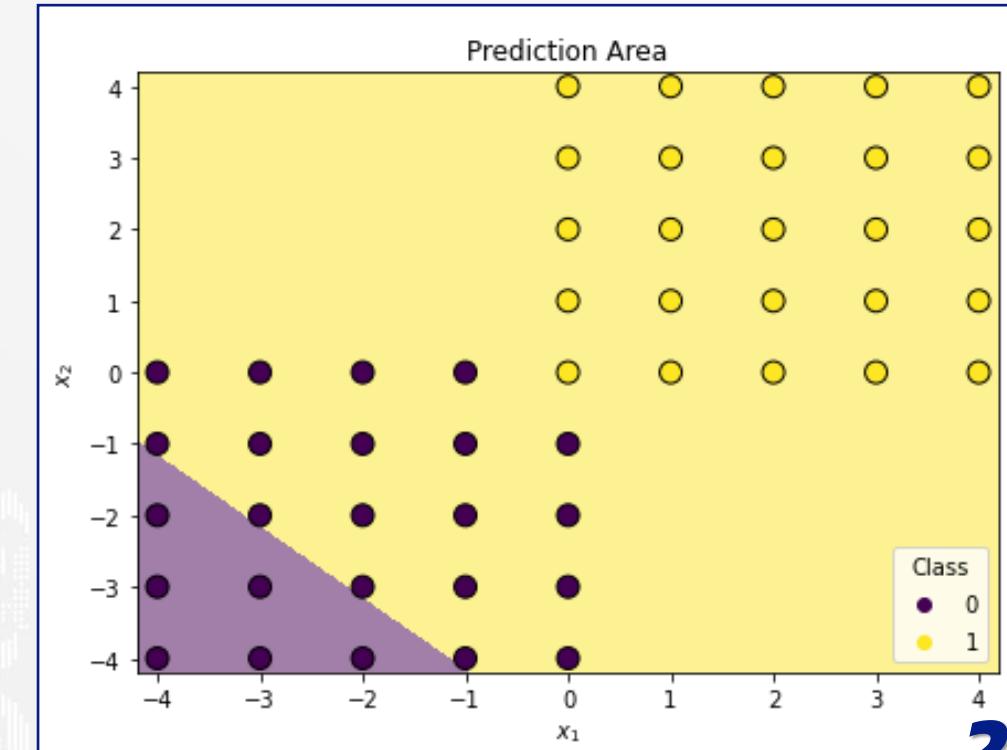


$$\hat{y} = \sigma(z)$$

# Problem with Imbalanced Class

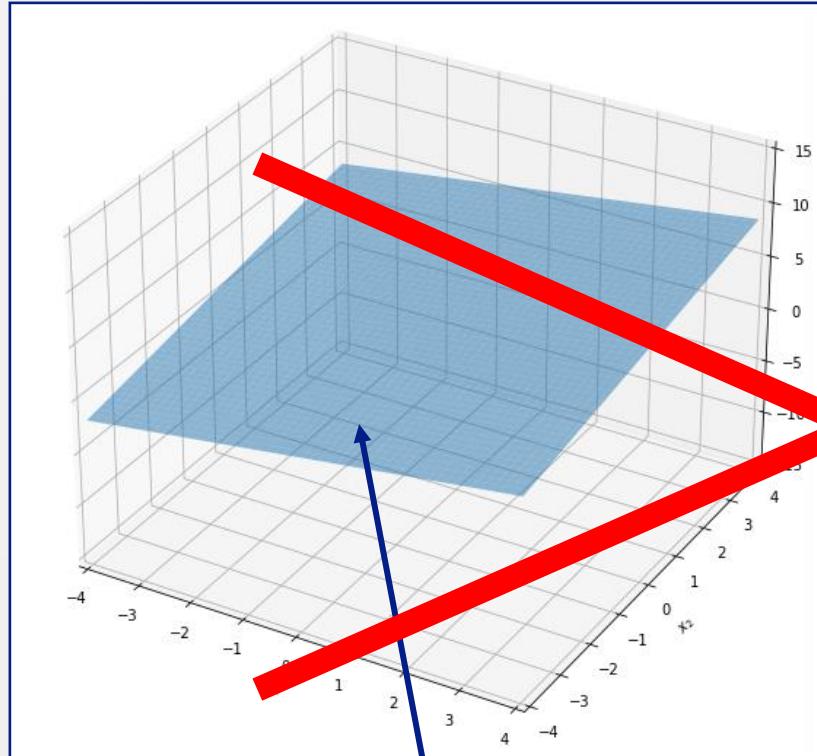


3D

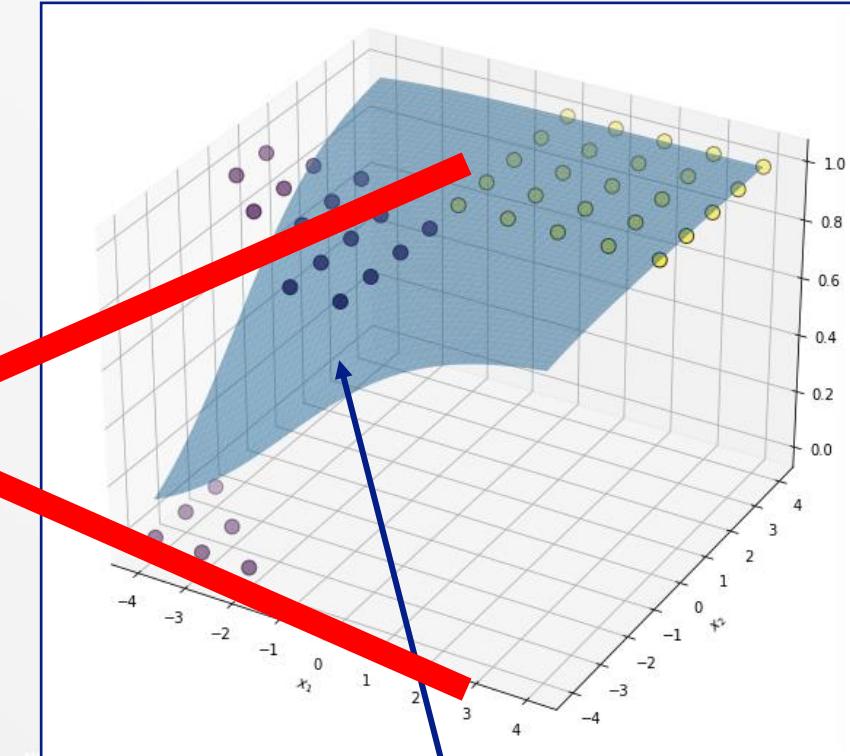


2D

# Problem with Imbalanced Class

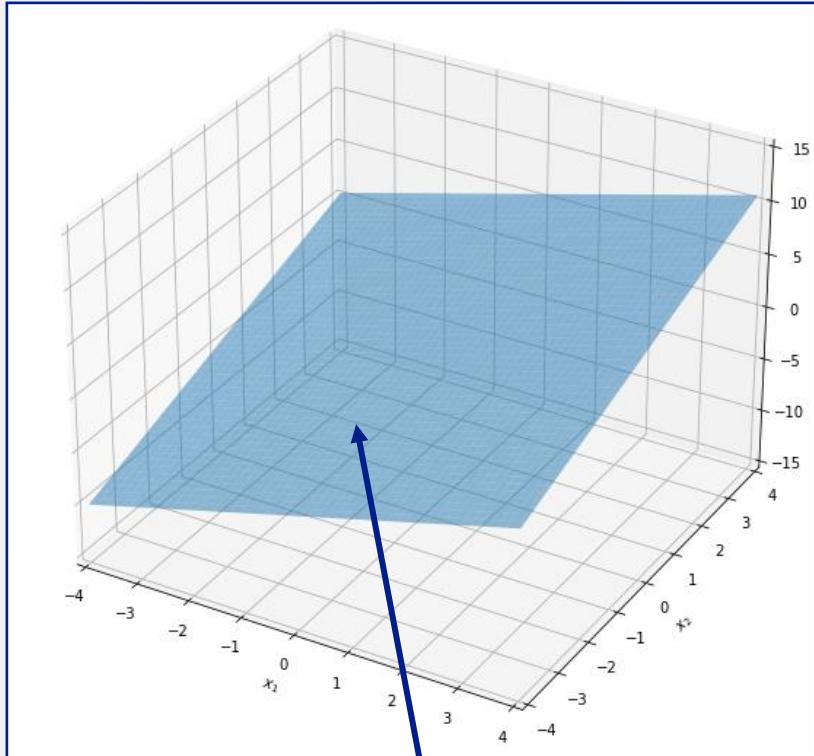


$$z = w_0 + w_1x_1 + w_2x_2$$

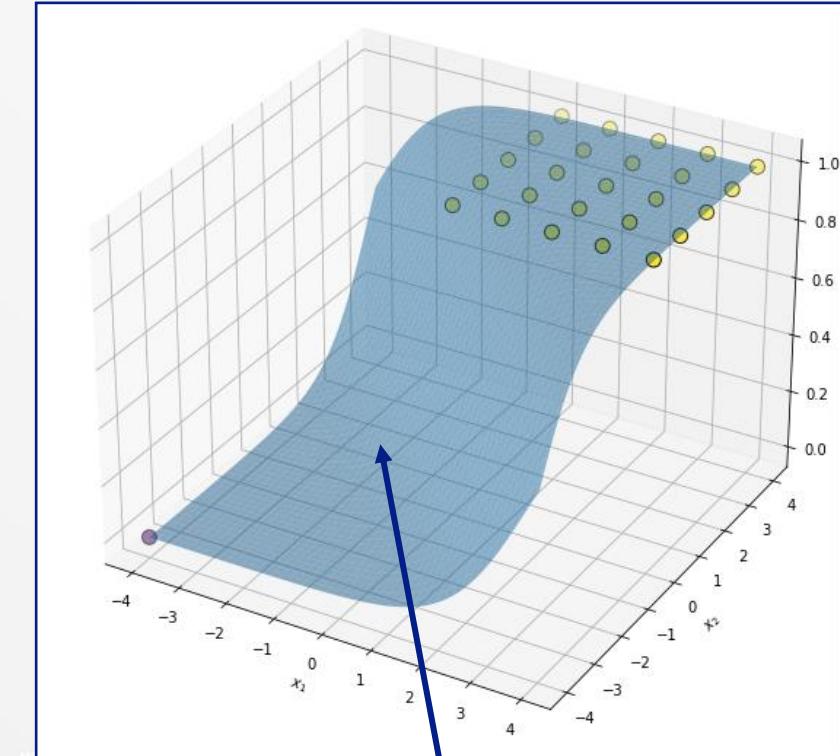


$$\hat{y} = \sigma(z)$$

# Problem with Imbalanced Class

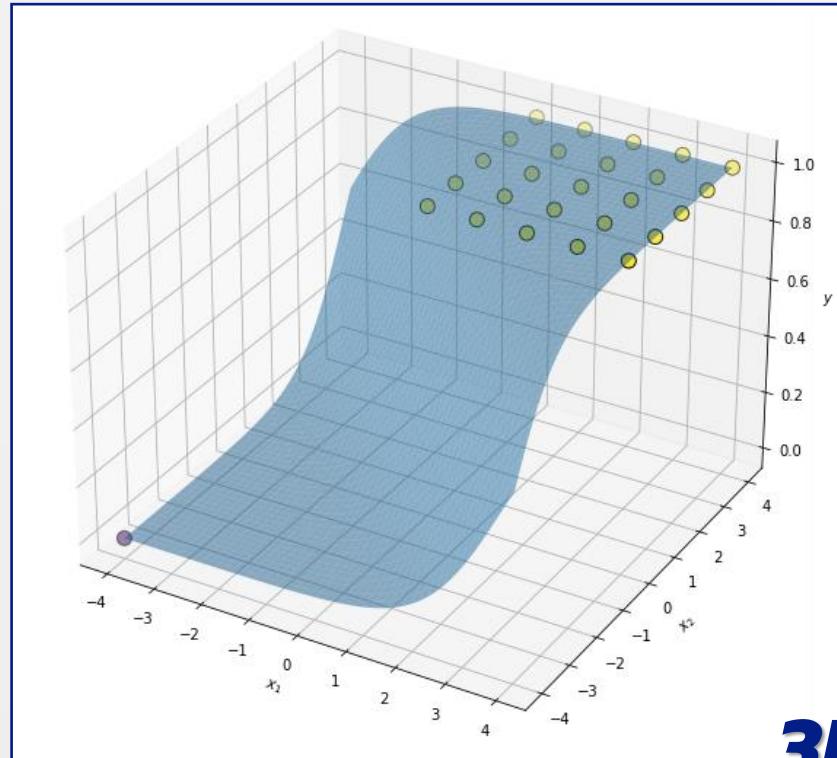


$$z = w_0 + w_1 x_1 + w_2 x_2$$

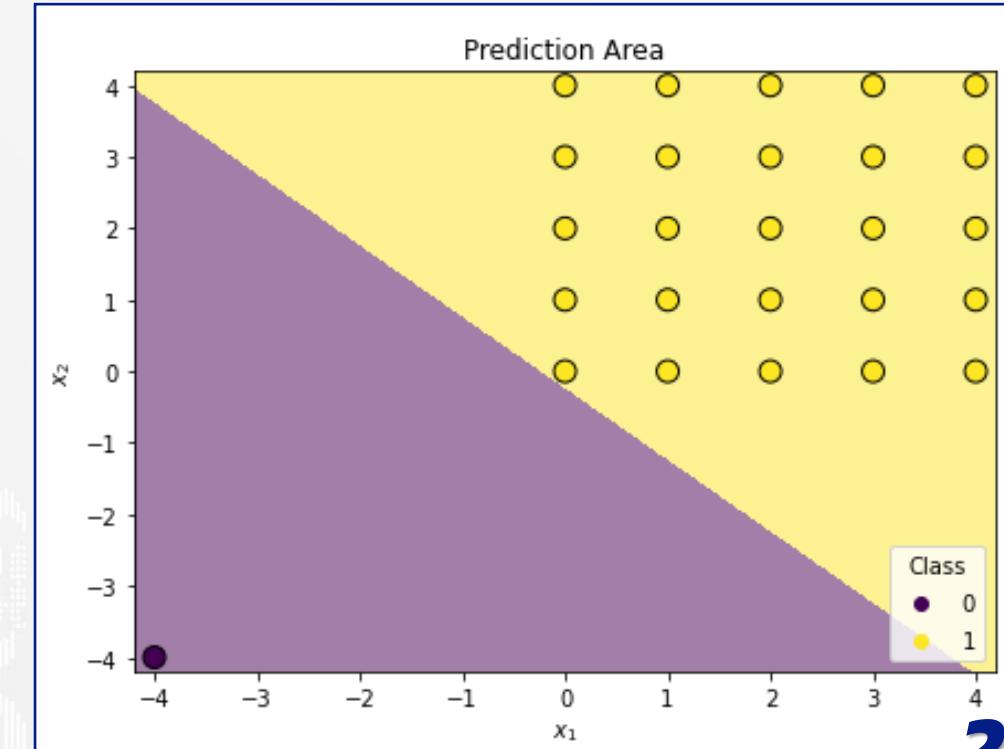


$$\hat{y} = \sigma(z)$$

# Problem with Imbalanced Class

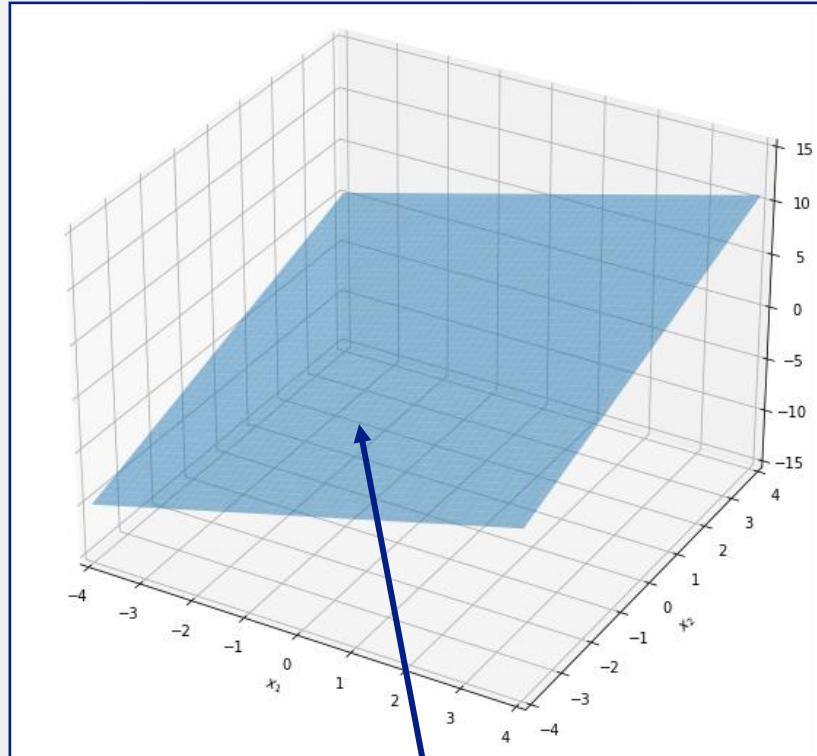


3D

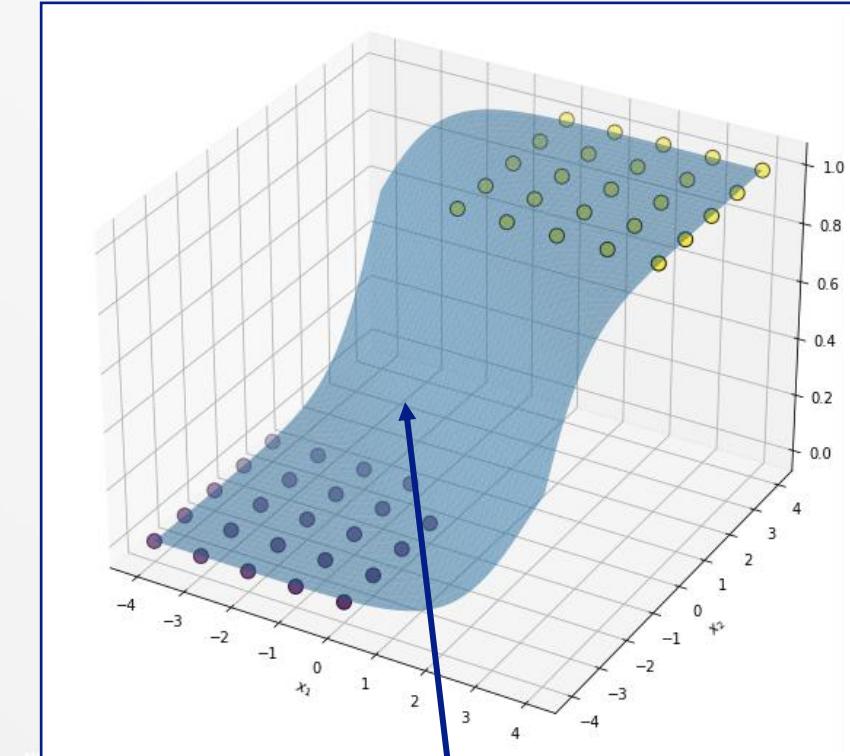


2D

# Problem with Imbalanced Class

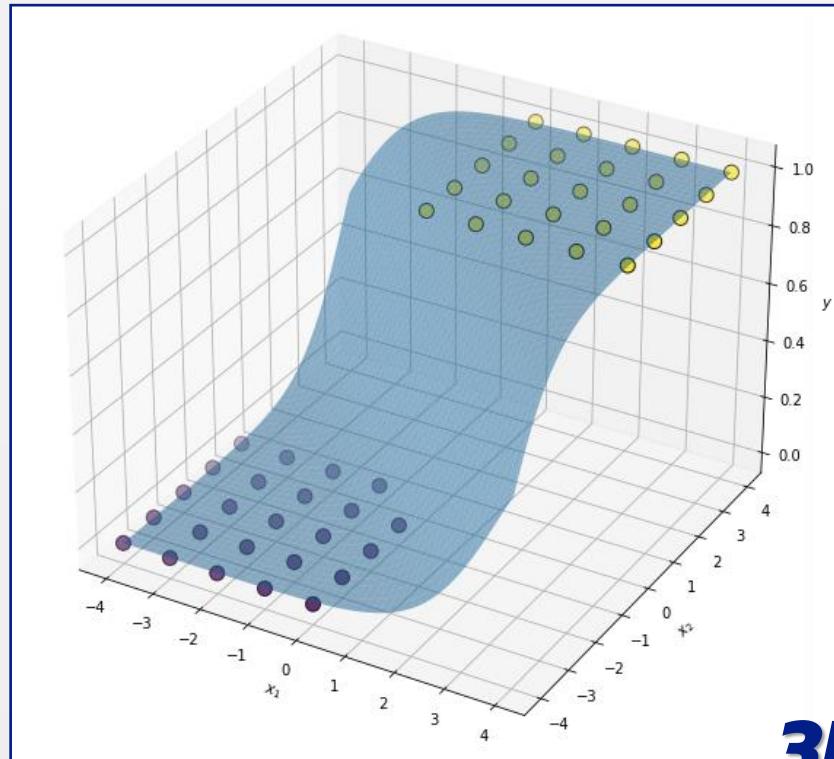


$$z = w_0 + w_1 x_1 + w_2 x_2$$

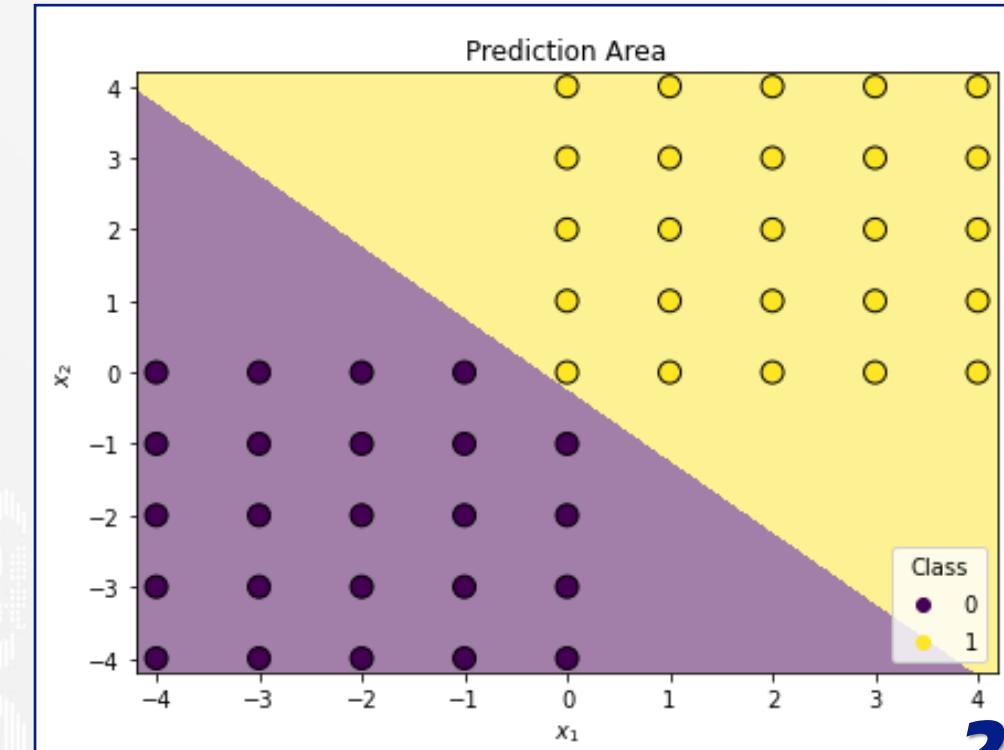


$$\hat{y} = \sigma(z)$$

# Problem with Imbalanced Class



3D

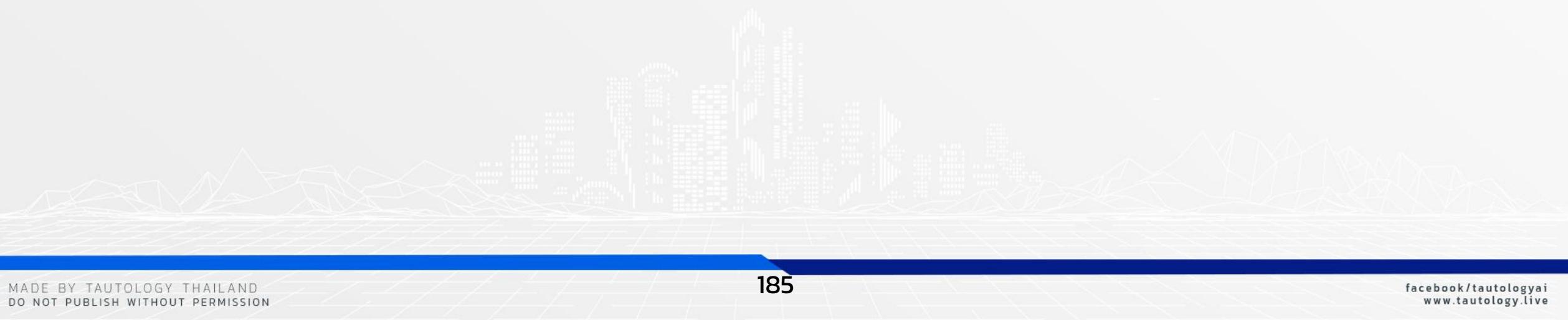


2D

# Imbalanced class

**Problem with Imbalanced Class**

**Solution**



# Solution

Bootstrapping

SMOTE

ADASYN

Balanced  
Class Weight

# Solution

Bootstrapping

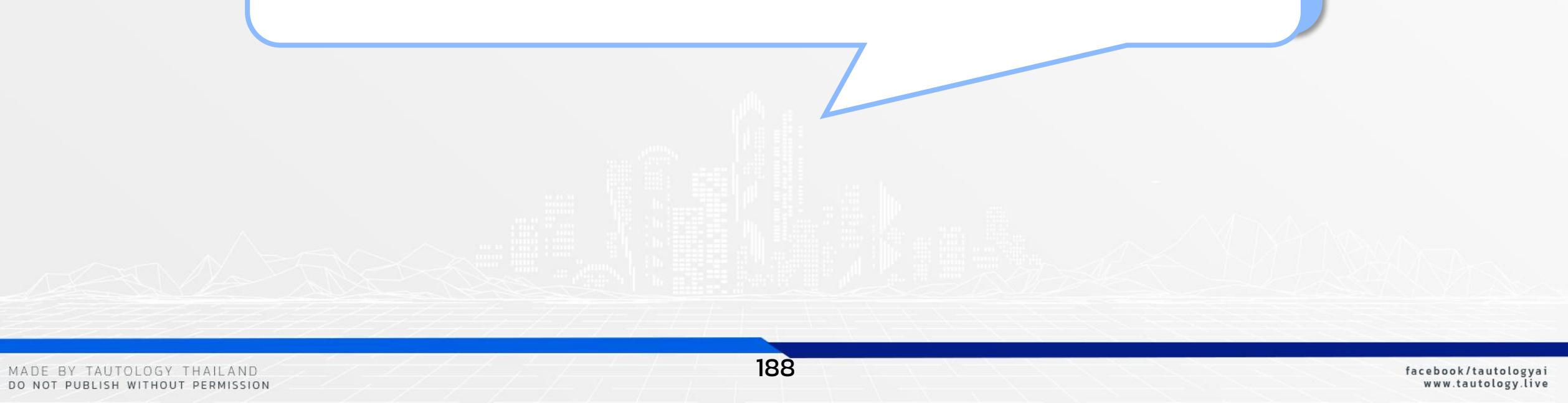
SMOTE

ADASYN

Balanced  
Class Weight

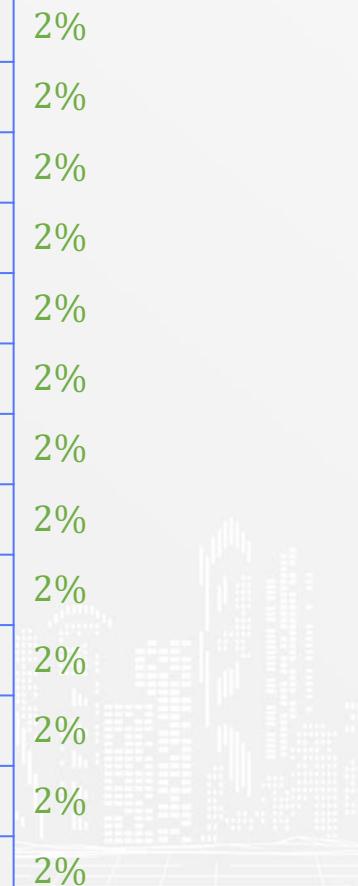
# Solution

หลักการทำงานของวิธี balanced class weight ก็คือ  
เราจะให้ความสำคัญกับแต่ละ class อย่างเท่าเทียมกัน



# Solution

$x_1$	$x_2$	$y$
0	0	1
0	1	1
0	2	1
0	3	1
0	4	1
1	0	1
1	1	1
1	2	1
1	3	1
1	4	1
2	0	1
2	1	1
2	2	1



$x_1$	$x_2$	y
2	3	1
2	4	1
3	0	1
3	1	1
3	2	1
3	3	1
3	4	1
4	0	1
4	1	1
4	2	1
4	3	1
4	4	1
-4	-4	0

# Solution

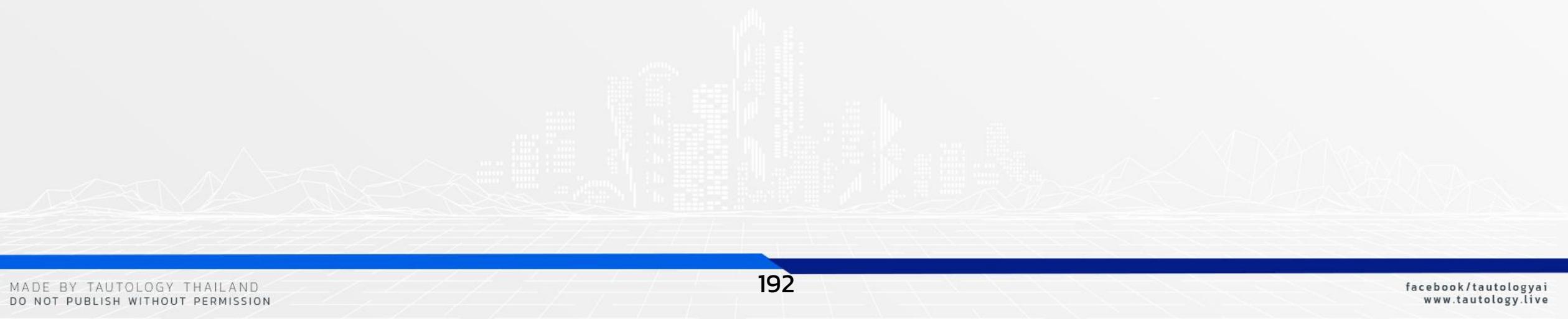
$$\mathbf{w} = \mathbf{w} - \frac{\alpha}{n} X_b^T \left( (\mathbf{y} - \hat{\mathbf{y}}) \cdot \begin{bmatrix} 0.02 \\ 0.02 \\ \vdots \\ 0.02 \\ 0.5 \end{bmatrix} \right)$$

# Solution

```
1 clf = LogisticRegression(penalty='none', class_weight='balanced')
2 clf.fit(X, y)
```

# Imbalanced class

- Problem with Imbalanced Class**
- Solution**



# Improvement

**Imbalanced class**



**Regularization**



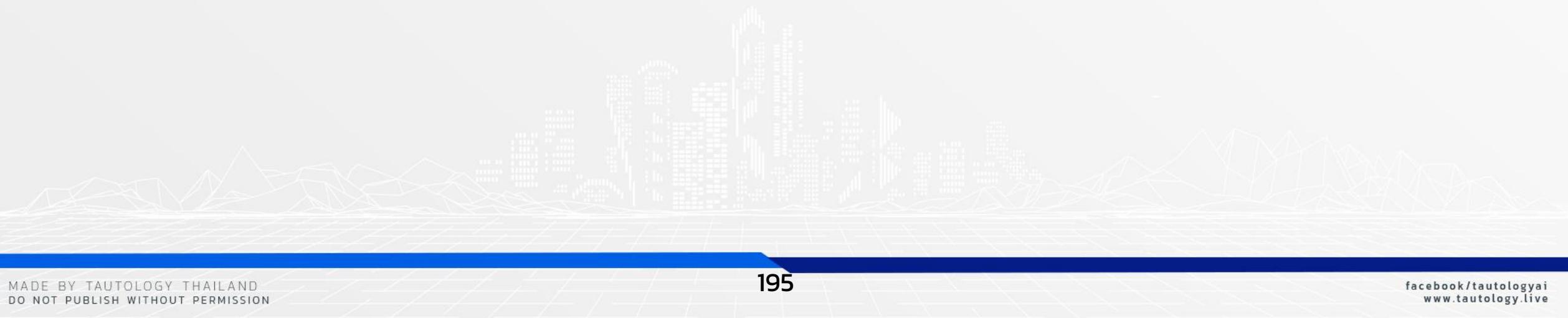
# Regularization

อ้างอิงจากงานวิจัยงานวิจัยของ Linear regression ในปี 1970 นักคณิตศาสตร์ค้นพบว่า “ การใส่  $\lambda > 0$  จะทำให้ error ของ model ลดลงกว่าการไม่ใส่  $\lambda$  ( $\lambda = 0$ ) ”

\*\*ยกเว้น  $y = \hat{y}$  (model ไม่มี error)

# Regularization

และเนื่องจาก logistic regression ถูกสร้างขึ้นจาก  
linear regression



# Regularization

สำหรับ logistic regression “ การใส่  $\lambda > 0$  จะทำให้ error ของ model ลดลงกว่าการไม่ใส่  $\lambda$  ( $\lambda = 0$ ) ”

# Regularization

**L2**  
**Regularization**

**L1**  
**Regularization**

**Elastic Net**

# Regularization

## L2 Regularization

```
1 alphas = np.logspace(-3, 3, num=100)
```

```
1 clf = LogisticRegressionCV(penalty='l2', Cs=alphas, cv=5)
2 clf.fit(X, y)
```

# Regularization

## L1 Regularization

```
1 alphas = np.logspace(-3, 3, num=100)
```

```
1 clf = LogisticRegressionCV(penalty='l1', solver='saga', Cs=alphas, cv=5)
2 clf.fit(X, y)
```

# Regularization

## Elastic Net

```
1 alphas = np.logspace(-3, 3, num=100)
2 l1_ratios = [0.01, 0.05, 0.1, 0.5, 0.7, 0.9, 0.95, 0.99, 1]

1 clf = LogisticRegressionCV(penalty='elasticnet', solver='saga',
2                             l1_ratio=l1_ratio, Cs=alphas, cv=5)
3 clf.fit(X, y)
```

# Regularization



Code for this section



Open File  
**Regularization (Binary).ipynb**

# Improvement

**Imbalanced class**



**Regularization**



# Logistic Regression (Binary Classification)



# DL102 : Logistic Regression



TAUTOLOGY  
INNOVATION  
SCHOOL

TAUTOLOGY



# LOGISTIC REGRESSION WORKSHOP

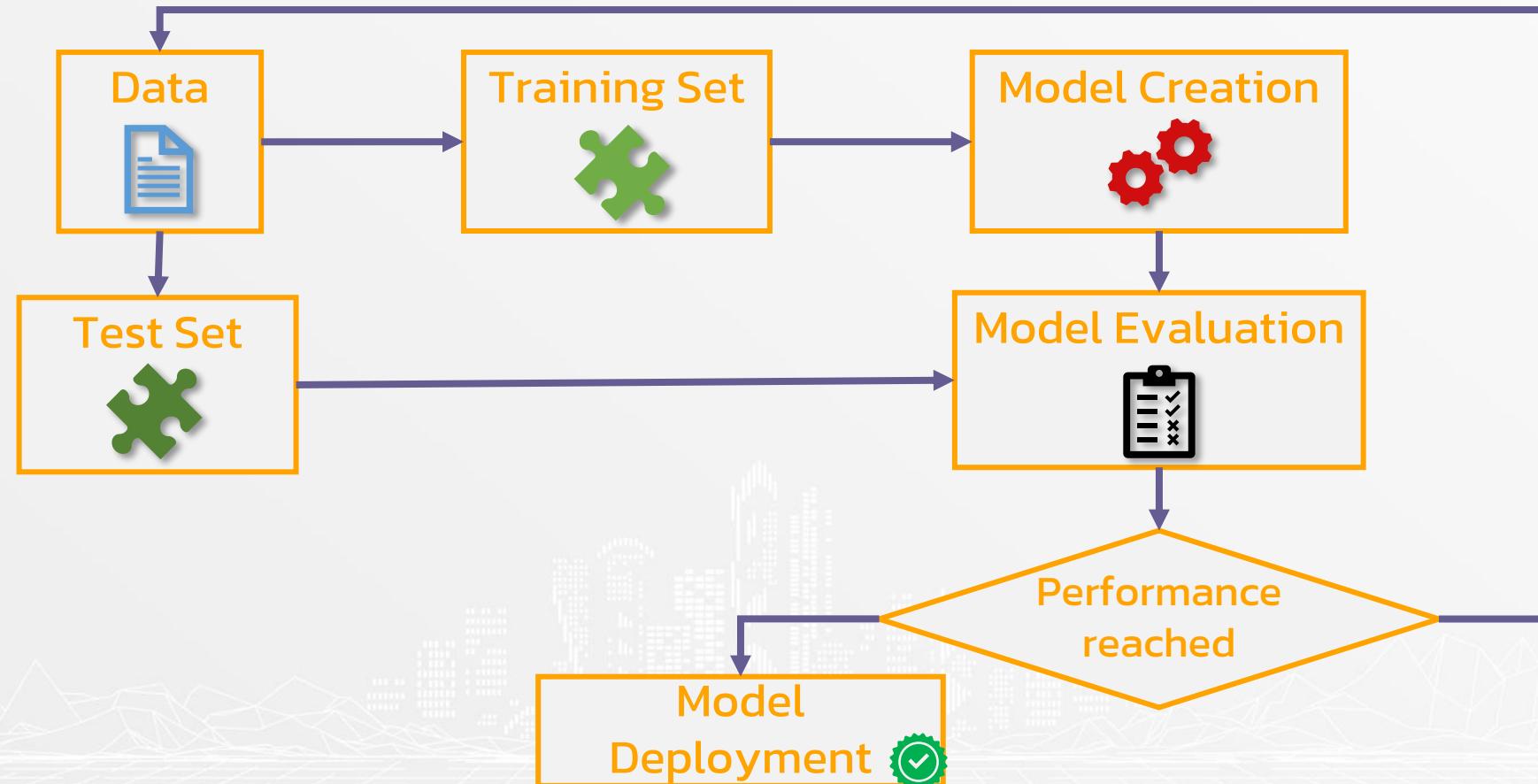
BY TAU TOLOGY

LOGISTIC REGRESSION - WORKSHOP

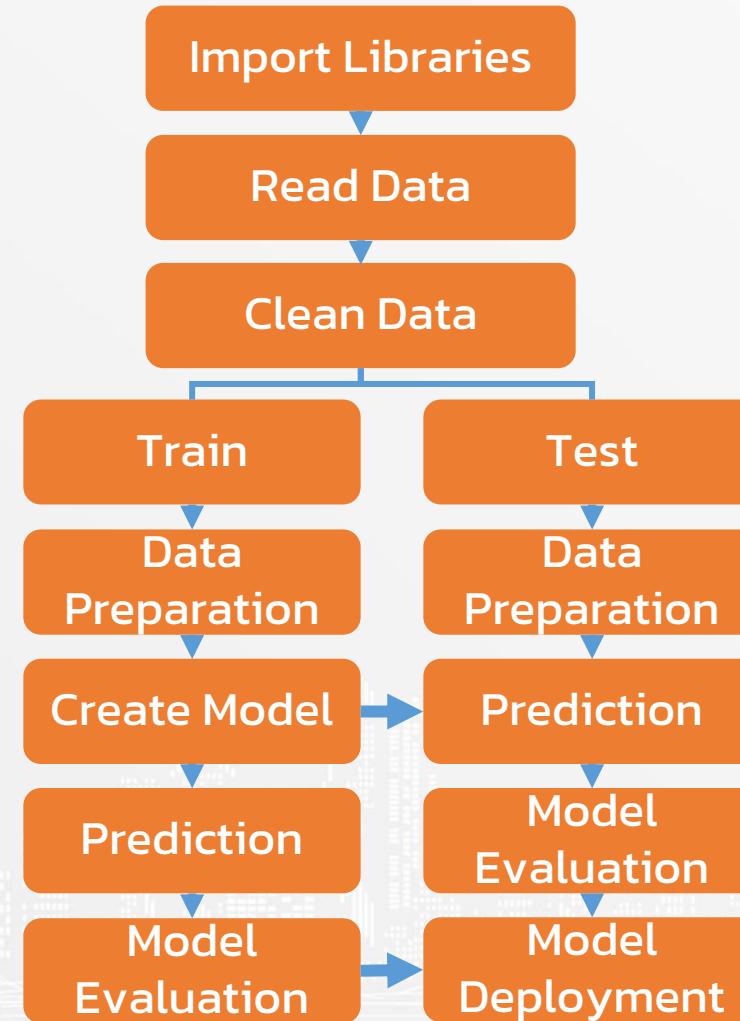
MADE BY TAU TOLOGY THAILAND  
DO NOT PUBLISH WITHOUT PERMISSION

facebook/tautologyai  
www.tautology.live

# Supervised Learning Workflow



# Code Pipeline



Import Libraries

Read Data

Clean Data

Train

Test

Data Preparation

Data Preparation

Create Model

Prediction

Prediction

Model Evaluation

Model Evaluation

Model Deployment

# Import Libraries

- 1  NumPy
- 2  pandas
- 3  scikit-learn
- 4  matplotlib

# Code

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 from sklearn.model_selection import train_test_split
6 from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder, StandardScaler, MinMaxScaler
7 from sklearn.linear_model import LogisticRegression, LogisticRegressionCV
8 from sklearn.metrics import plot_confusion_matrix, classification_report
9
10 import warnings
11 warnings.filterwarnings('ignore')
12
13 np.random.seed(12345)
```

Import Libraries

Read Data

Clean Data

Train

Test

Data Preparation

Data Preparation

Create Model

Prediction

Prediction

Model Evaluation

Model Evaluation

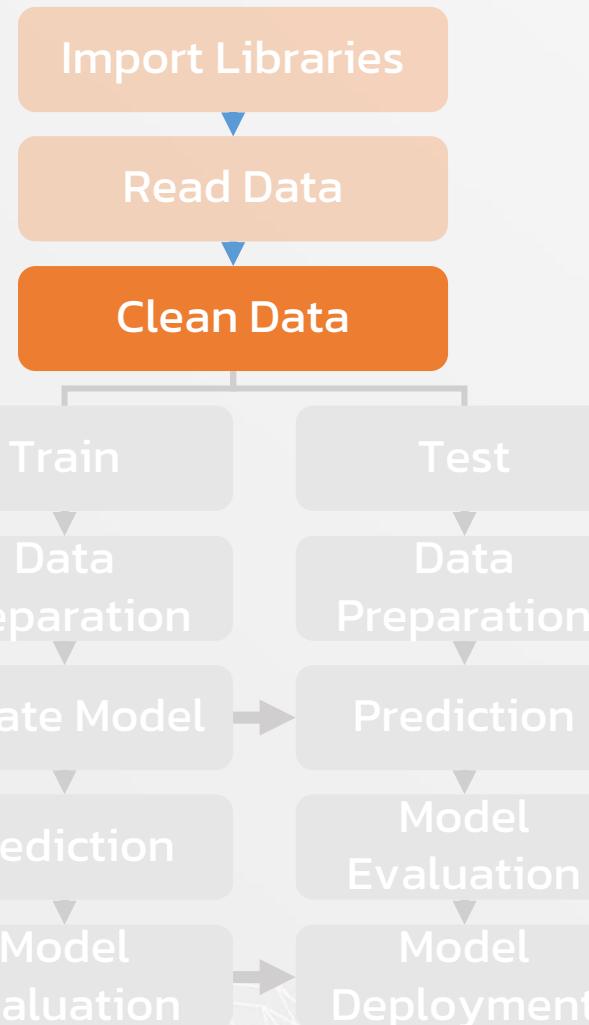
Model Deployment

# Read Data

	age	experience	gpa	degree	position	expected_salary	result
0	29.0	7	2.71	bachelor	secretary	19500	accept
1	29.0	4	NaN	bachelor	secretary	20500	accept
2	27.0	2	3.40	doctorate	accountant	43000	reject
3	33.0	11	3.25	bachelor	NaN	39000	reject
4	34.0	9	3.22	master	engineer	46500	reject
5	26.0	0	3.99	bachelor	accountant	29500	reject
6	23.0	1	3.60	bachelor	accountant	22000	accept
7	27.0	3	2.64	doctorate	accountant	33000	accept
8	23.0	0	2.69	bachelor	secretary	21500	reject
9	25.0	0	2.88	bachelor	engineer	29500	reject
	:	:	:	:	:	:	:

# Code

```
1 data = pd.read_csv('job_acceptance_dataset.csv')
```



# Clean Data

1. Handle Missing Values
2. Handle Outliers





# Clean Data

## 1. Handle Missing Values

## 2. Handle Outliers

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90 entries, 0 to 89
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   age              89 non-null      float64
 1   experience      90 non-null      int64  
 2   gpa              89 non-null      float64
 3   degree           90 non-null      object  
 4   position          89 non-null      object  
 5   expected_salary  90 non-null      int64  
 6   result            89 non-null      object  
dtypes: float64(2), int64(2), object(3)
memory usage: 5.0+ KB
```

# Code

- Check Missing Values

```
1 data.info()
```

- Remove Missing Values

```
1 data.dropna(axis=0, inplace=True)
```



# Clean Data

## 1. Handle Missing Values

## 2. Handle Outliers

	age	experience	gpa	expected_salary
count	86.000000	86.000000	86.000000	86.000000
mean	27.872093	3.186047	3.280581	32209.302326
std	4.375860	3.197468	0.511957	9194.456487
min	21.000000	0.000000	2.520000	15000.000000
25%	24.000000	0.250000	2.890000	25500.000000
50%	28.000000	2.000000	3.280000	32000.000000
75%	31.750000	5.000000	3.567500	38500.000000
max	35.000000	13.000000	6.000000	54500.000000

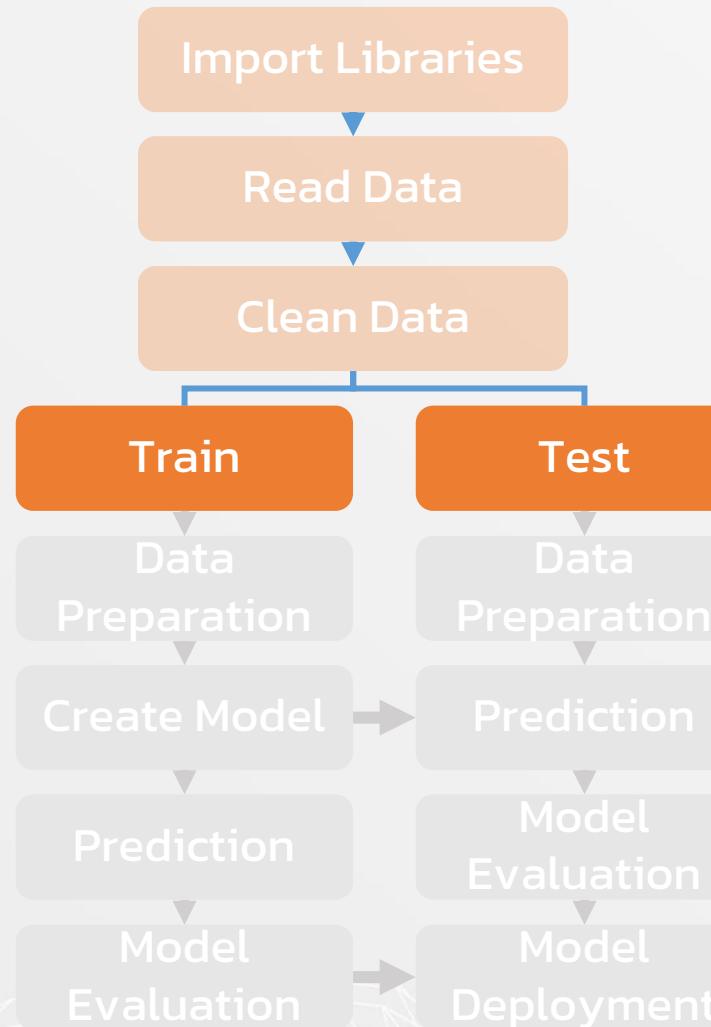
# Code

- Check Outliers

```
1 data.describe()
```

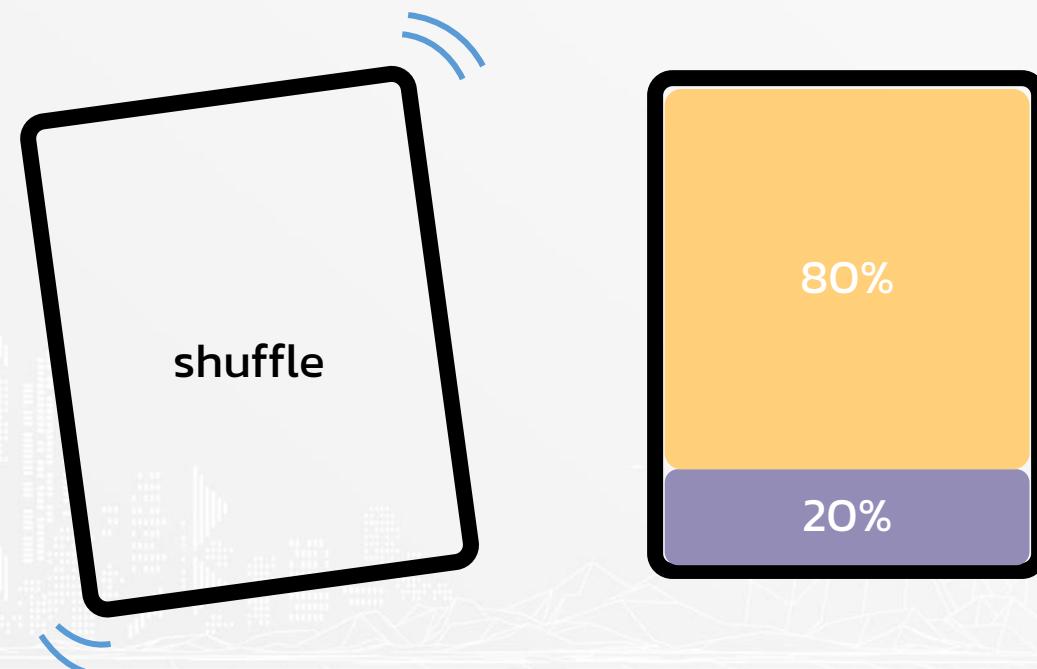
- Remove Outliers

```
1 _filter = data['gpa'] <= 4.00
2 data = data[_filter]
```



## Train/Test

แบ่งข้อมูลออกเป็น 2 ชุด คือ training set และ test set ด้วยอัตราส่วน 80:20 ตามลำดับ

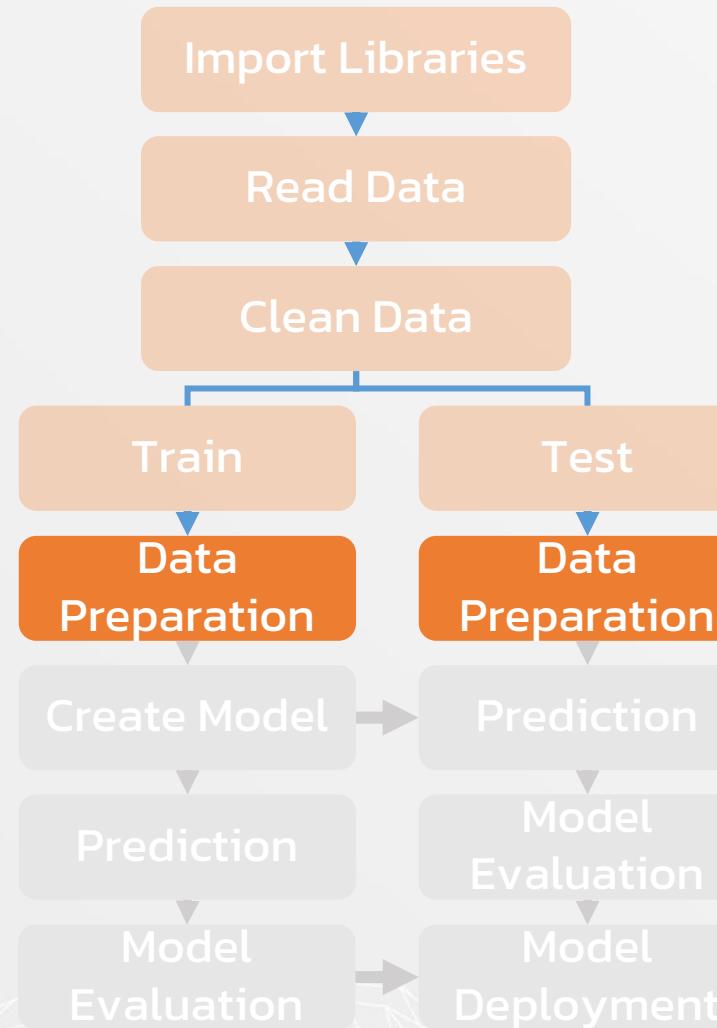


# Code

```
1 target_name = 'result'  
2 feature_name = list(data.columns.drop(target_name))
```

```
1 X = data[feature_name]  
2 y = data[target_name]
```

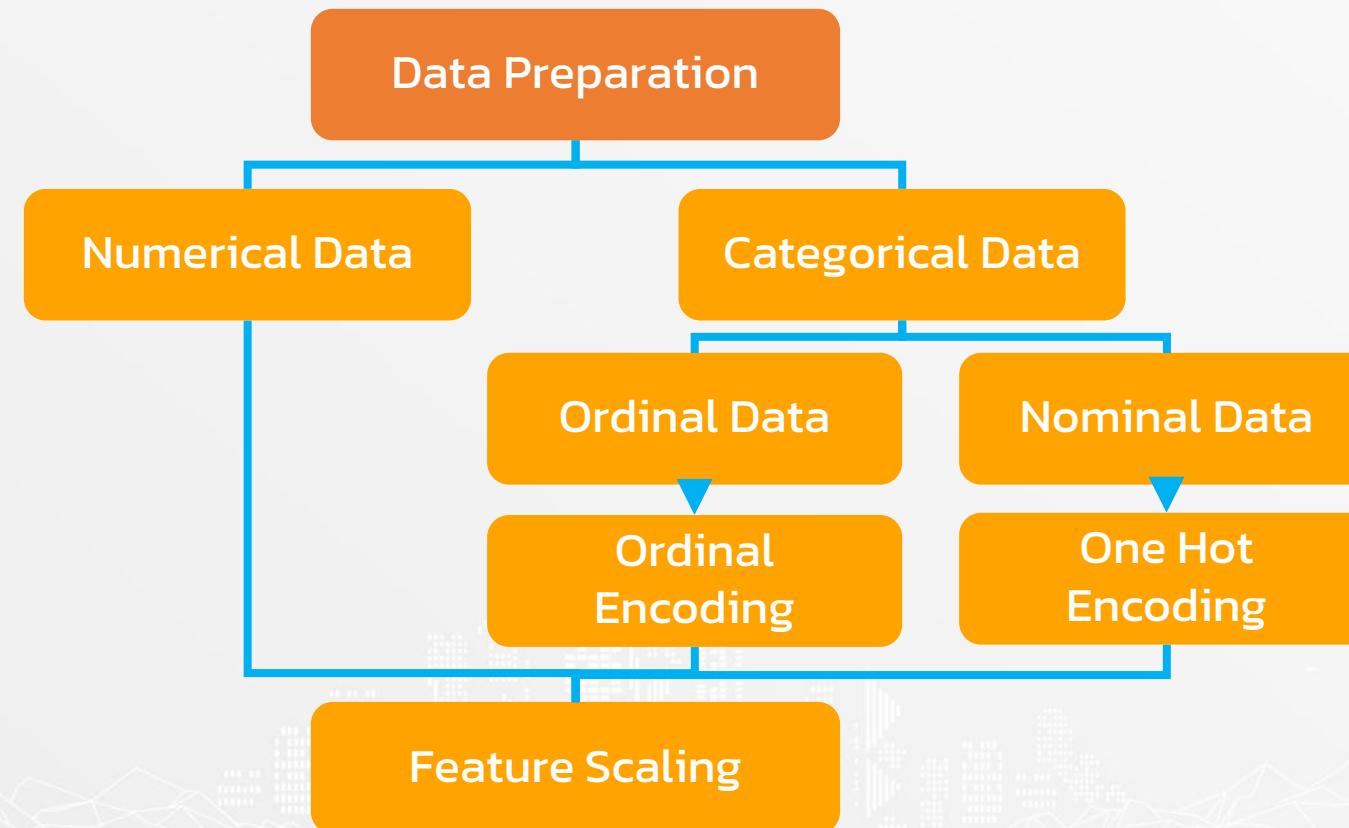
```
1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, shuffle=True)
```



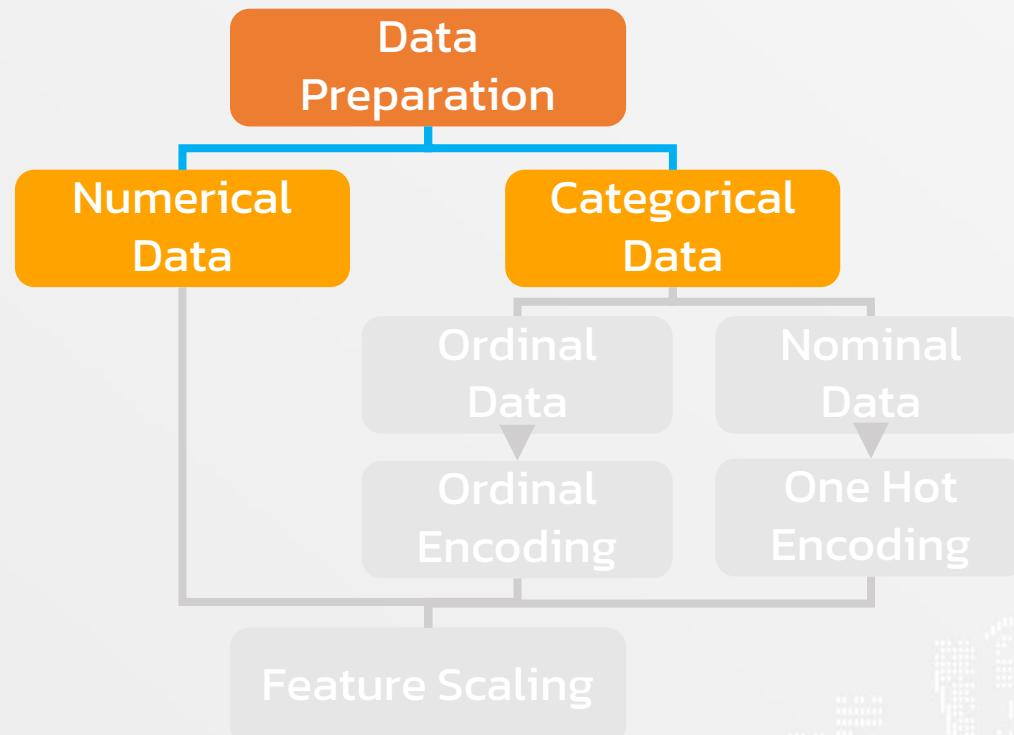
# Data Preparation



# Data Preparation



# Data Preparation



## Type of Features

พิจารณาและจำแนก feature ที่มีลักษณะข้อมูลแบบ numerical data และ categorical data ออกจากกัน

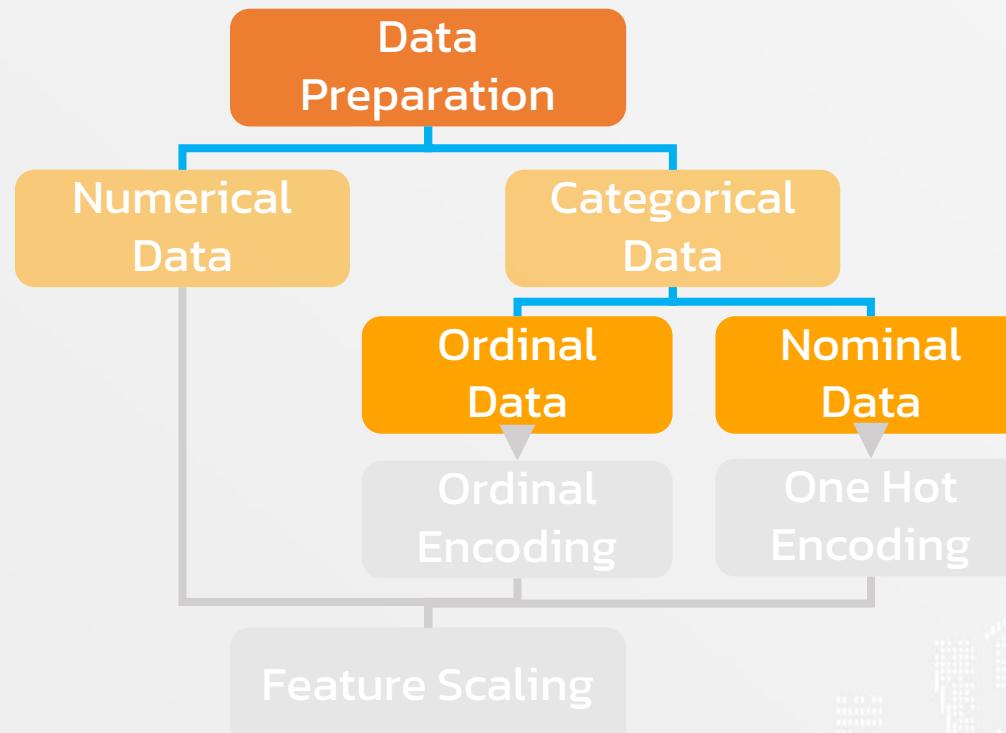
Numerical  
Data

Categorical  
Data

# Code

```
1 numerical_feature = ['age', 'experience', 'gpa', 'expected_salary']  
2 categorical_feature = ['degree', 'position']
```

# Data Preparation



## Type of Categorical Features

พิจารณาและจำแนก feature ที่มีลักษณะข้อมูลแบบ ordinal data และ nominal data ออกจากกัน

```
degree : ['bachelor' 'doctorate' 'master']  
position : ['accountant' 'engineer' 'secretary']
```

# Code

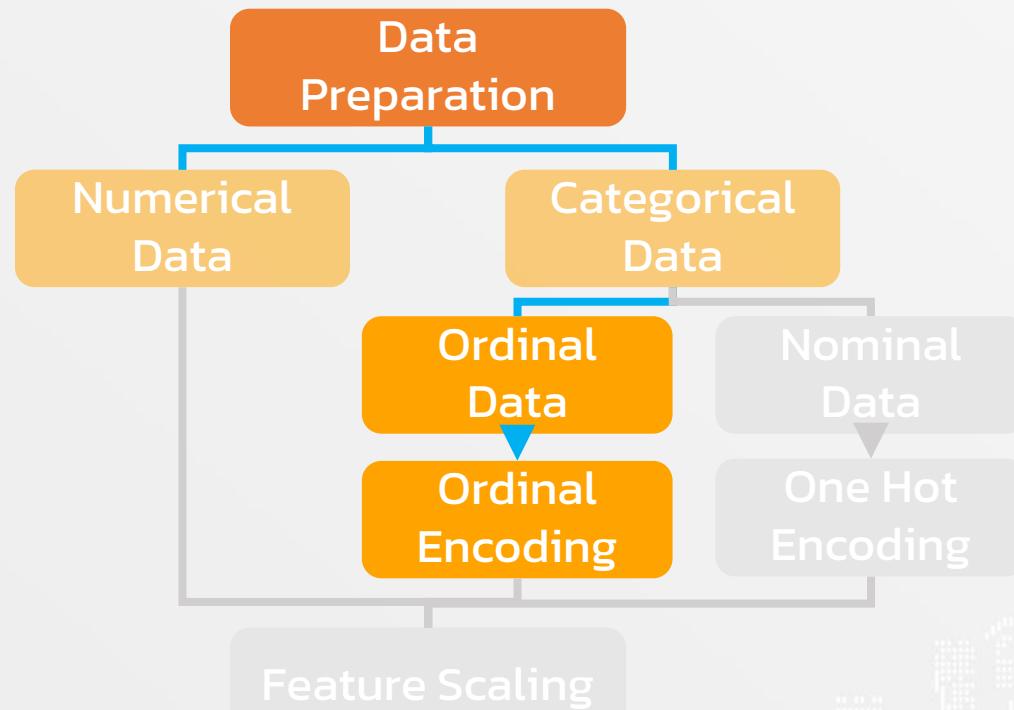
- Consider each Feature of Categorical Features

```
1 for feature in categorical_feature:  
2     print(feature, ':', np.unique(X_train[feature]))
```

- Classify into Ordinal Feature and Nominal Feature

```
1 ordinal_feature = ['degree']  
2 nominal_feature = ['position']
```

# Data Preparation



## Ordinal Encoding

การทำ ordinal encoding จะต้องทำแบบเดียวกันกับใน training set และ test set

degree
0 bachelor
1 master
2 bachelor
3 bachelor
4 bachelor

degree	
0	0.0
1	1.0
2	0.0
3	0.0
4	0.0

# Code

- **Ordinal Data**

```
1 ordinal_feature = ['degree']
```

- **Ordinal Encoding**

```
1 categories = [
2     np.array(['bachelor', 'master', 'doctorate'], dtype=object)
3 ]
```

# Code

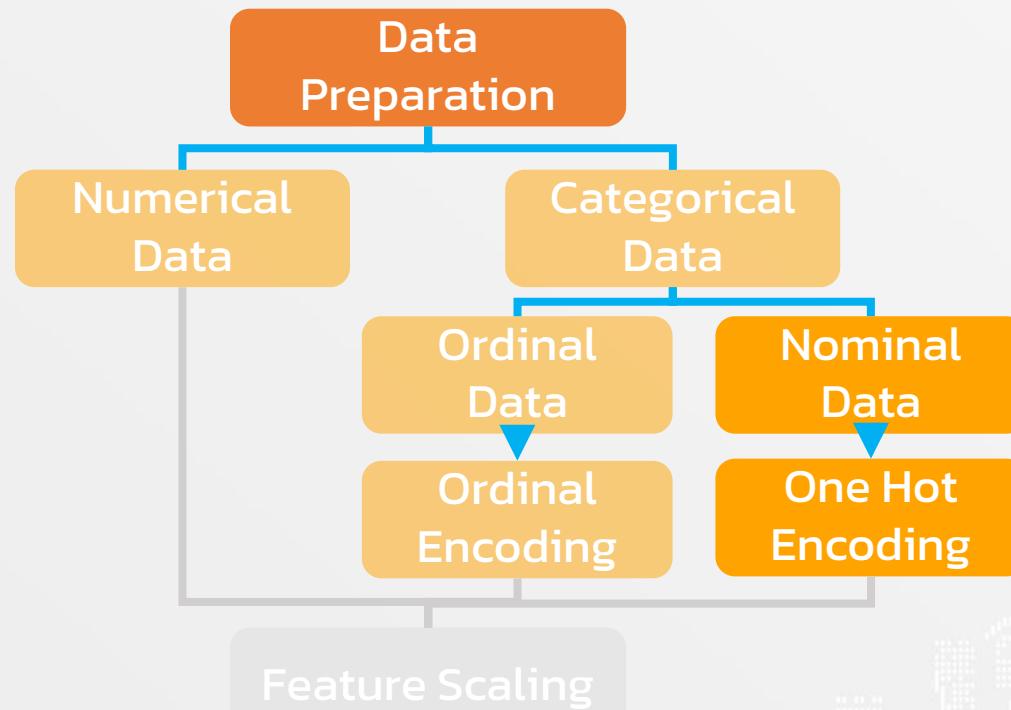
- Ordinal Encoding for **training set**

```
1 ordinal_encoder = OrdinalEncoder(categories=categories)
2 X_train[ordinal_feature] = ordinal_encoder.fit_transform(X_train[ordinal_feature])
```

- Ordinal Encoding for **test set**

```
1 X_test[ordinal_feature] = ordinal_encoder.transform(X_test[ordinal_feature])
```

# Data Preparation



## One Hot Encoding

การทำ one hot encoding จะต้องทำแบบเดียวกันกับใน training set และ test set

position
0 secretary
1 secretary
2 engineer
3 engineer
4 secretary



position	position_accountant	position_engineer	position_secretary	
0 secretary	0	0.0	0.0	1.0
1 secretary	1	0.0	0.0	1.0
2 engineer	2	0.0	1.0	0.0
3 engineer	3	0.0	1.0	0.0
4 secretary	4	0.0	0.0	1.0

# Code

- Nominal Data

```
1 nominal_feature = ['position']
```

- One Hot Encoding

```
1 one_hot_encoder = OneHotEncoder(sparse=False, handle_unknown='ignore')
2 one_hot_encoder.fit(X_train[nominal_feature])
```

```
1 one_hot_feature = []
2 for i, feature in enumerate(nominal_feature):
3     for cate in one_hot_encoder.categories_[i]:
4         one_hot_feature_name = str(feature) + '_' + str(cate)
5         one_hot_feature.append(one_hot_feature_name)
```

# Code

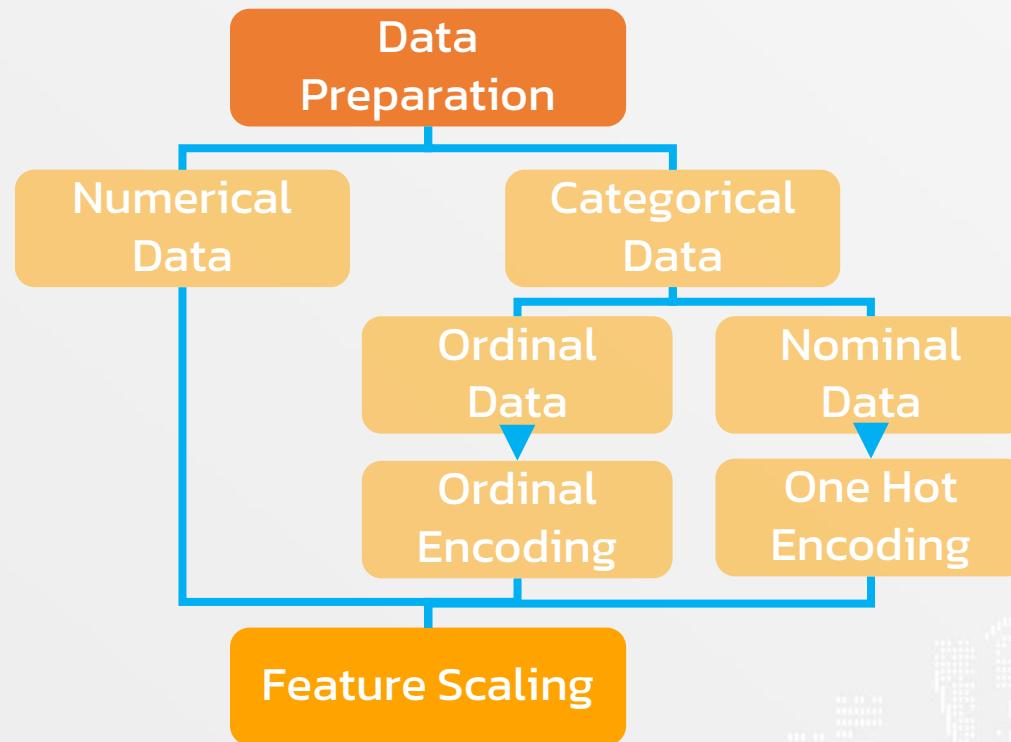
- One Hot Encoding for **training set**

```
1 X_train[one_hot_feature] = one_hot_encoder.transform(X_train[nominal_feature])
2 X_train.drop(nominal_feature, axis=1, inplace=True)
```

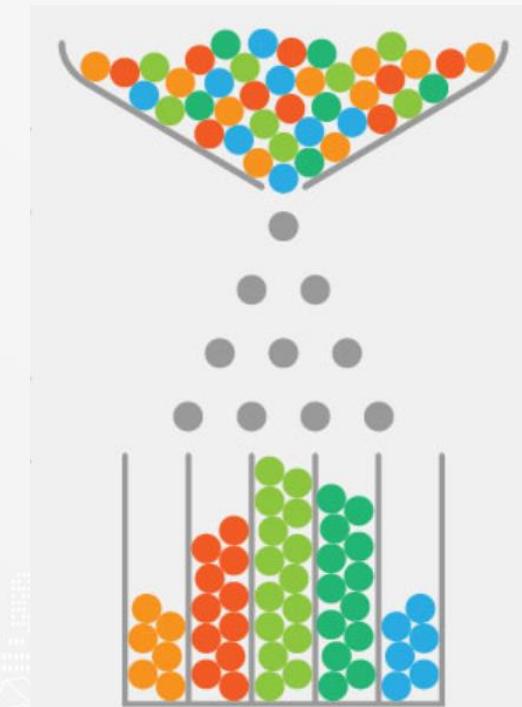
- One Hot Encoding for **test set**

```
1 X_test[one_hot_feature] = one_hot_encoder.transform(X_test[nominal_feature])
2 X_test.drop(nominal_feature, axis=1, inplace=True)
```

# Data Preparation



## Feature Scaling



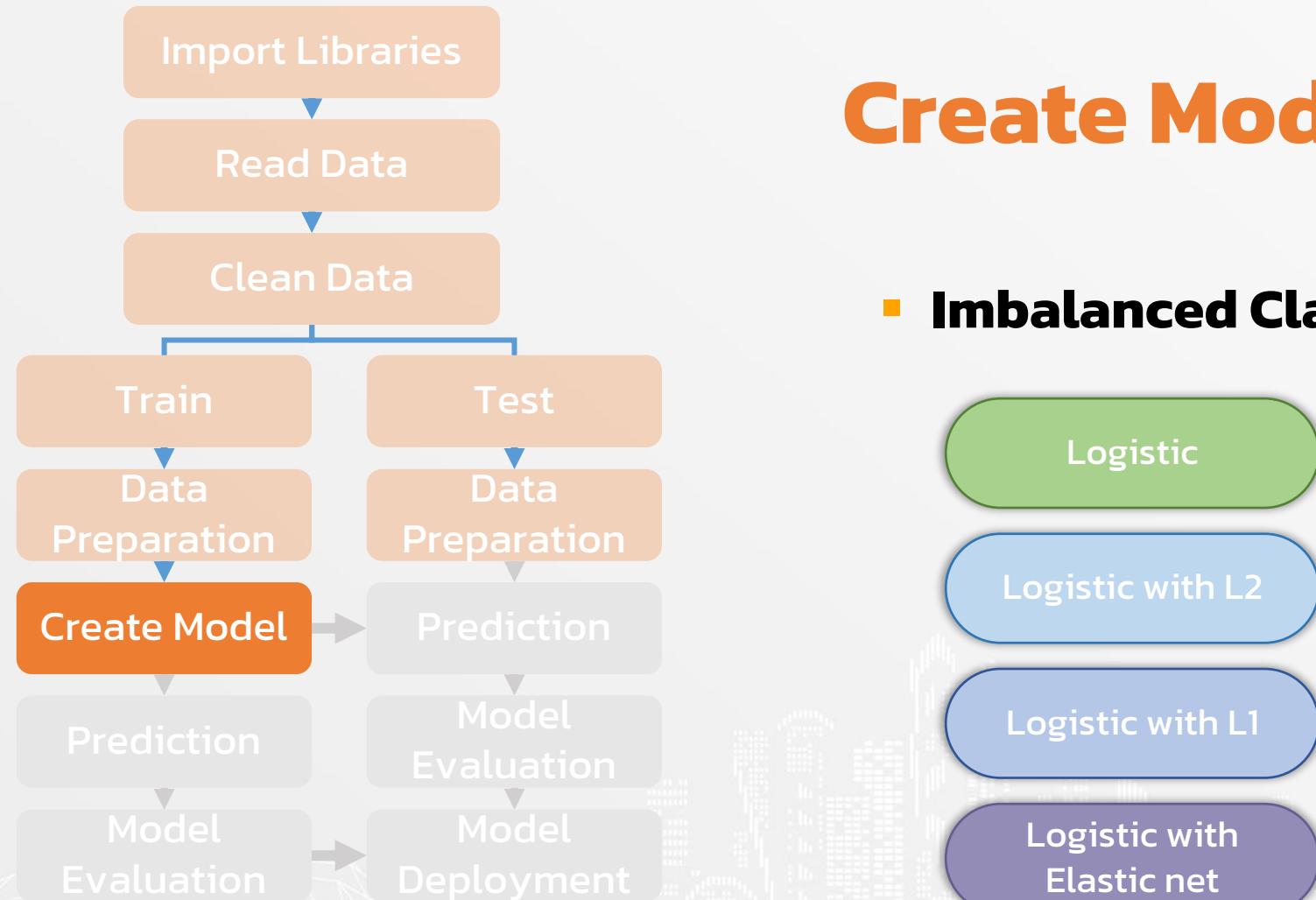
# Code

- Feature Scaling for **training set**

```
1 scaler = StandardScaler()  
2 X_train_scaled = scaler.fit_transform(X_train)
```

- Feature Scaling for **test set**

```
1 scaler = StandardScaler()  
2 X_test_scaled = scaler.transform(X_test)
```

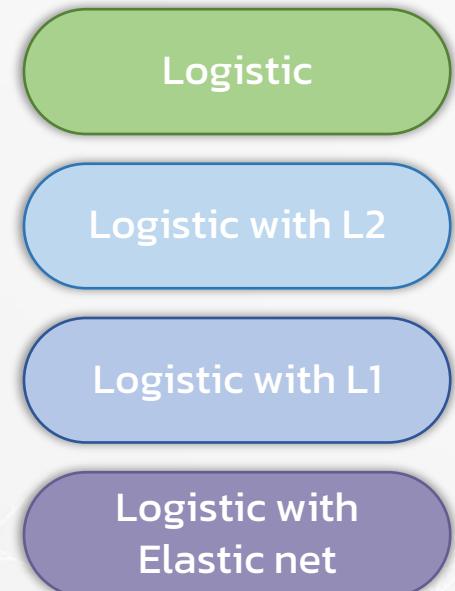


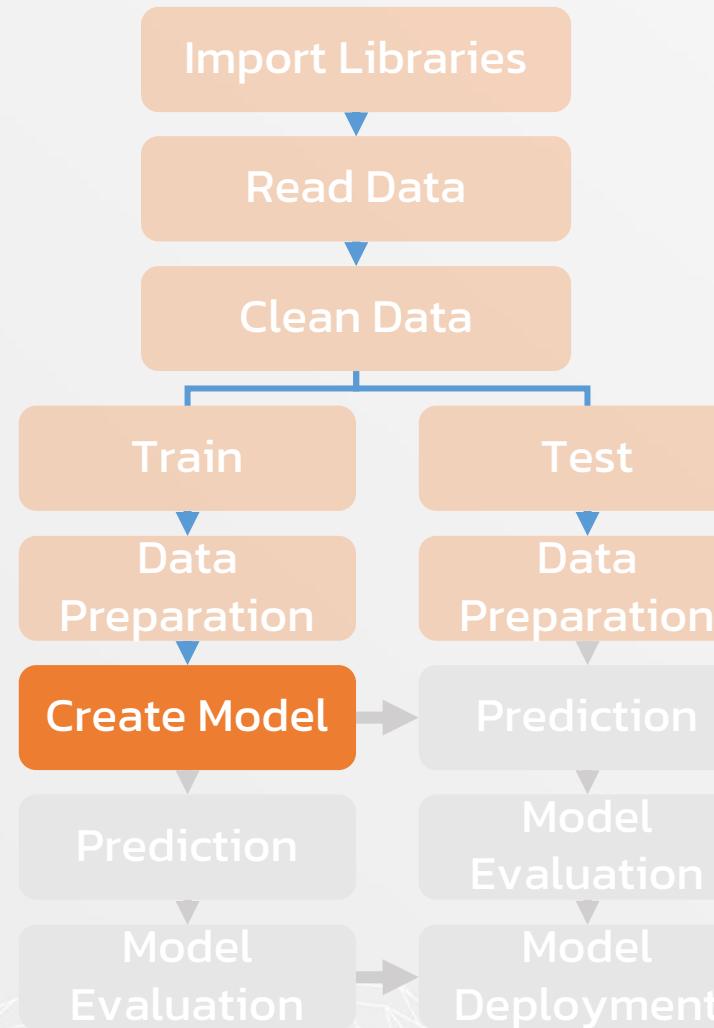
# Create Model

## ■ Imbalanced Class



## ■ Balanced Class





# Create Model

- Setting Parameter
- Train Model
- Model's Weight & Bias

# Code – Setting Parameter

## ■ Imbalanced Class – **Logistic**

```
1 clf = LogisticRegression(penalty='none')
```

## ■ Imbalanced Class – **Logistic with L2**

```
1 alphas = np.logspace(-3, 3, num=100)
2 clf = LogisticRegressionCV(penalty='l2', Cs=alphas, cv=5)
```

# Code – Setting Parameter

## ■ Imbalanced Class – **Logistic with L1**

```
1 alphas = np.logspace(-3, 3, num=100)
2 clf = LogisticRegressionCV(penalty='l1', solver='saga', Cs=alphas, cv=5)
```

## ■ Imbalanced Class – **Logistic with Elastic net**

```
1 alphas = np.logspace(-3, 3, num=100)
2 l1_ratios = [0.01, 0.05, 0.1, 0.5, 0.7, 0.9, 0.95, 0.99, 1]
3 clf = LogisticRegressionCV(penalty='elasticnet', solver='saga',
4                             l1_ratios=l1_ratios, Cs=alphas, cv=5)
```

# Code – Setting Parameter

## ■ Balanced Class – **Logistic**

```
1 clf = LogisticRegression(penalty='none', class_weight='balanced')
```

## ■ Balanced Class – **Logistic with L2**

```
1 alphas = np.logspace(-3, 3, num=100)
2 clf = LogisticRegressionCV(penalty='l2', Cs=alphas, cv=5, class_weight='balanced')
```

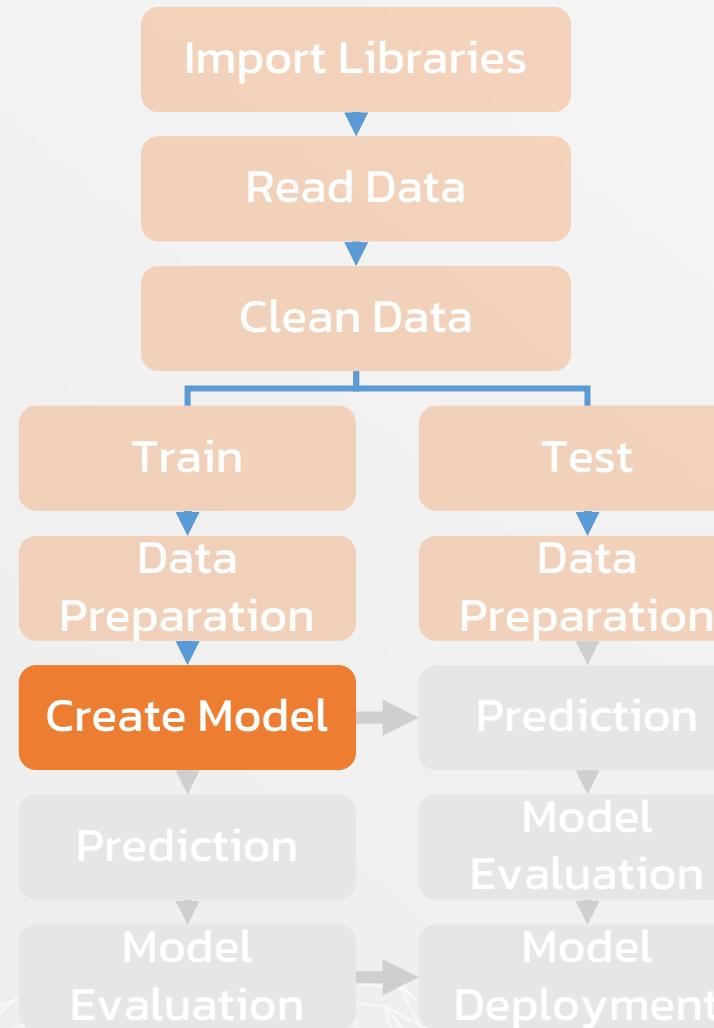
# Code – Setting Parameter

## ■ Balanced Class – **Logistic with L1**

```
1 alphas = np.logspace(-3, 3, num=100)
2 clf = LogisticRegressionCV(penalty='l1', solver='saga', Cs=alphas,
3                             cv=5, class_weight='balanced')
```

## ■ Balanced Class – **Logistic with Elastic net**

```
1 alphas = np.logspace(-3, 3, num=100)
2 l1_ratios = [0.01, 0.05, 0.1, 0.5, 0.7, 0.9, 0.95, 0.99, 1]
3 clf = LogisticRegressionCV(penalty='elasticnet', solver='saga', l1_ratios=l1_ratios,
4                             Cs=alphas, cv=5, class_weight='balanced')
```

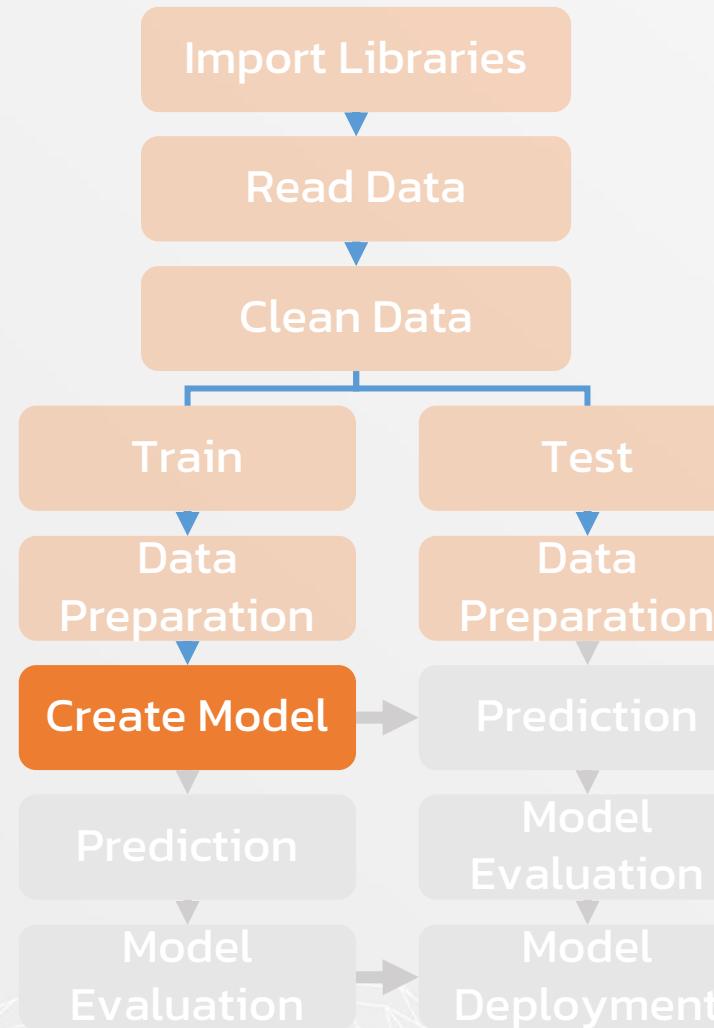


# Create Model

-  **Setting Parameter**
-  **Train Model**
-  **Model's Weight & Bias**

# Code – Train Model

```
1 clf.fit(X_train_scaled, y_train)
```



# Create Model

- Setting Parameter
- Train Model
- Model's Weight & Bias

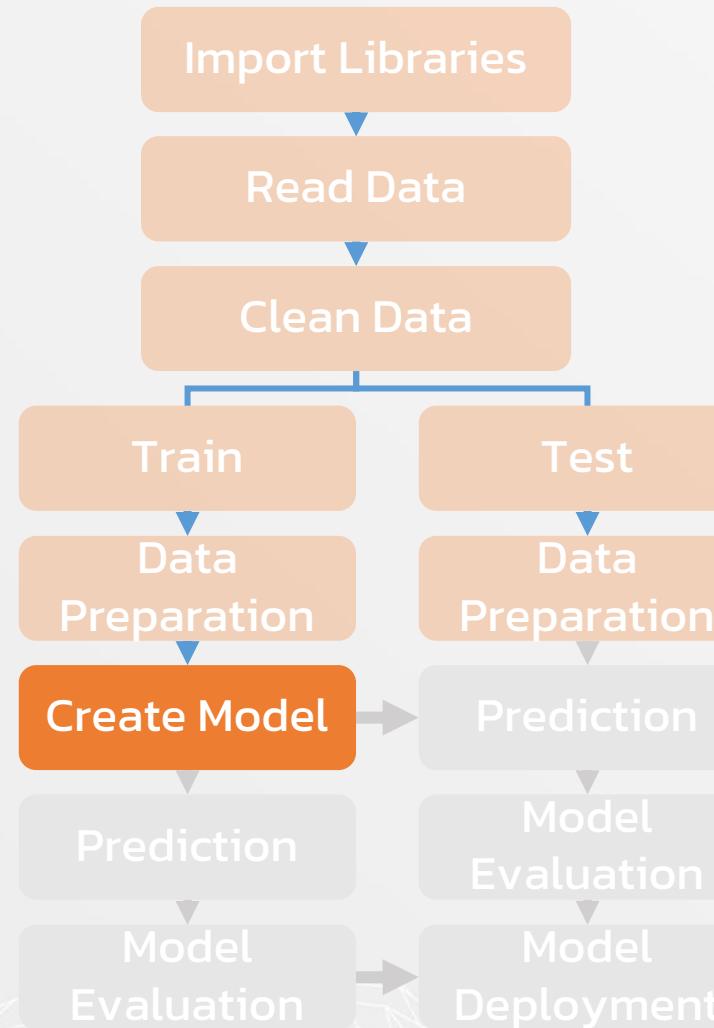
# Code – Model's Weight & Bias

- **Bias ( $w_0$ )**

```
1 clf.intercept_
```

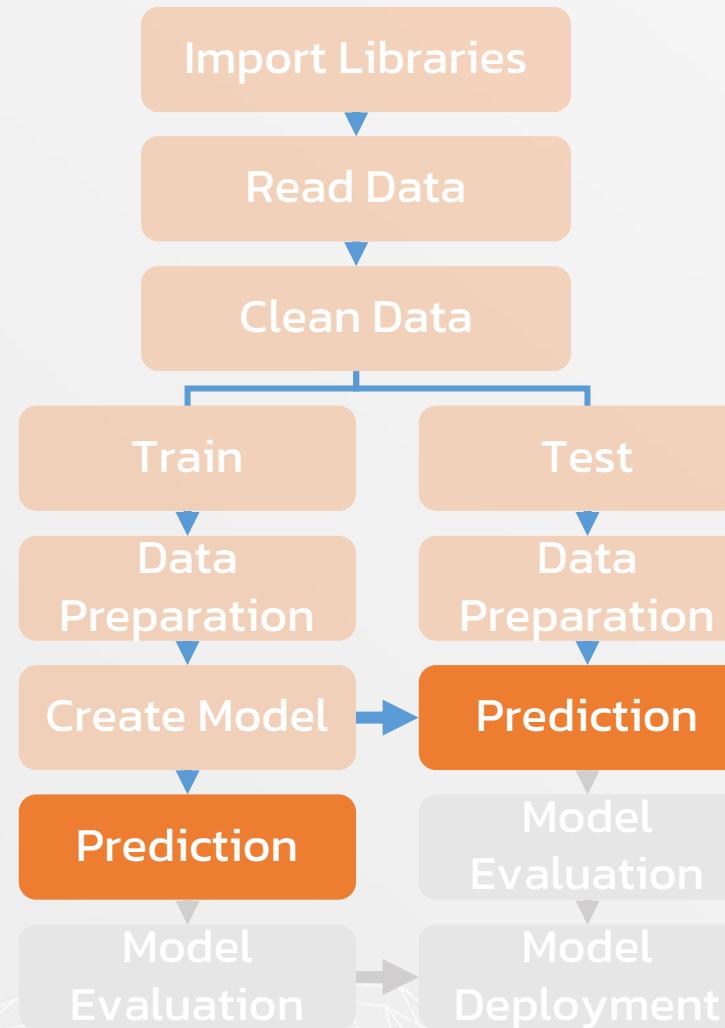
- **Weight ( $w_1, \dots, w_p$ )**

```
1 clf.coef_
```



# Create Model

- Setting Parameter
- Train Model
- Model's Weight & Bias



# Prediction



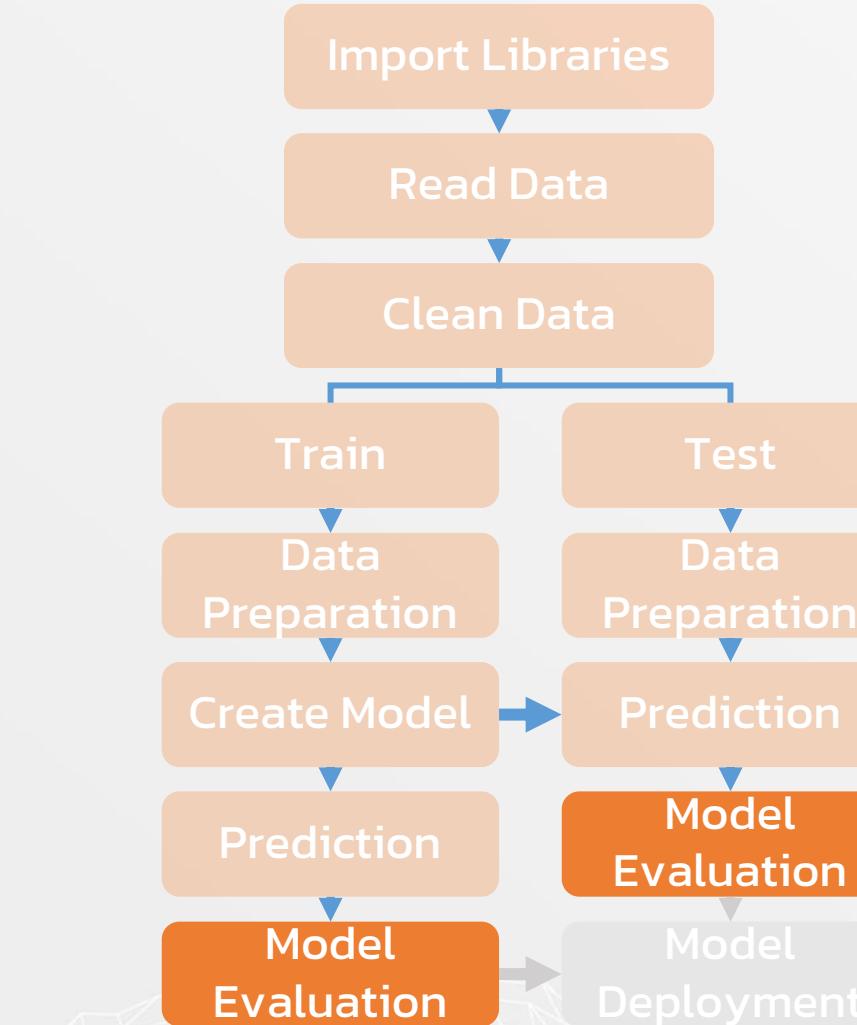
# Code

- Prediction for **training set**

```
1 y_pred_train = clf.predict(X_train_scaled)
```

- Prediction for **test set**

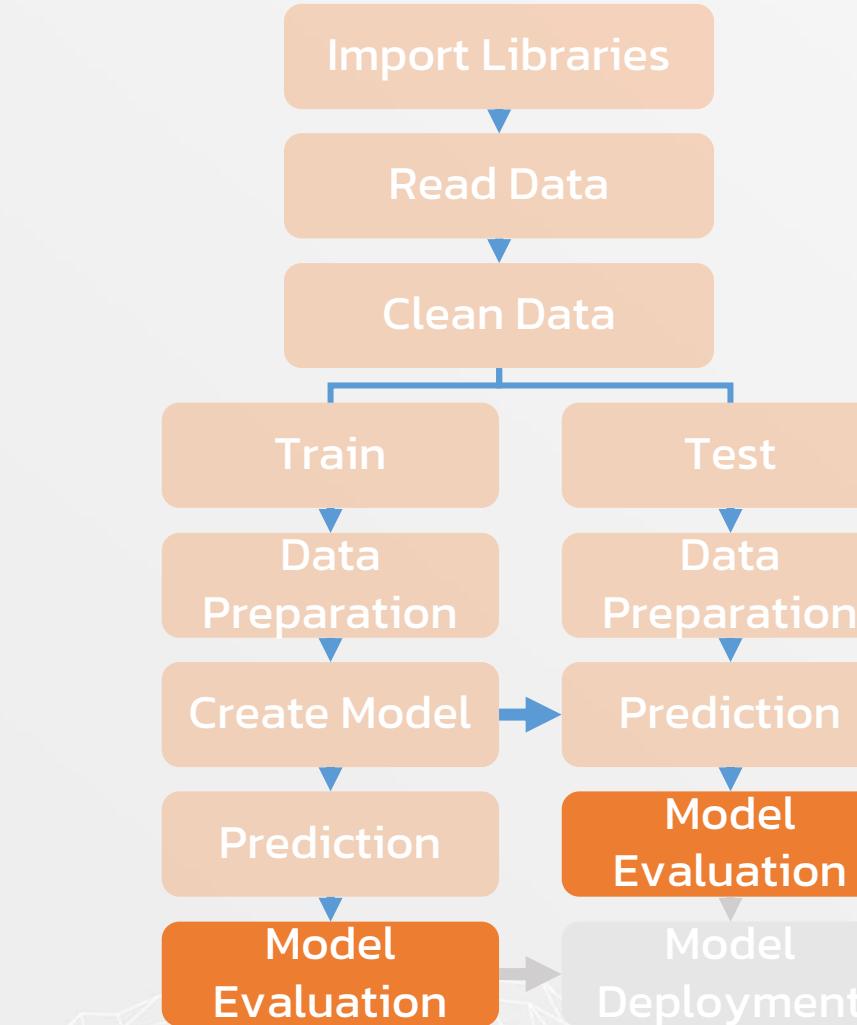
```
1 y_pred_test = clf.predict(X_test_scaled)
```



# Model Evaluation

1. Confusion Matrix
2. Scoring (*accuracy, precision, recall, F1*)





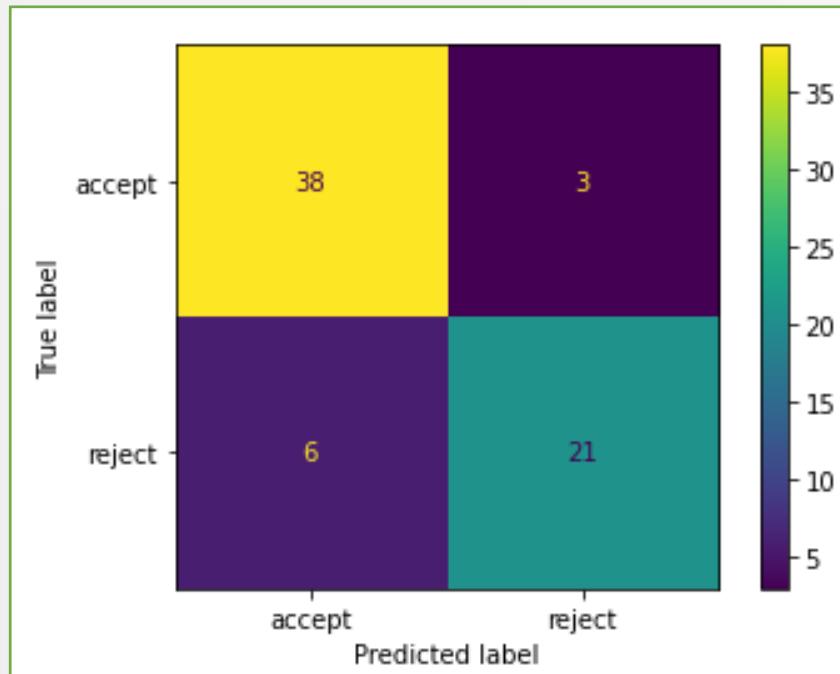
# Model Evaluation

## 1. Confusion Matrix

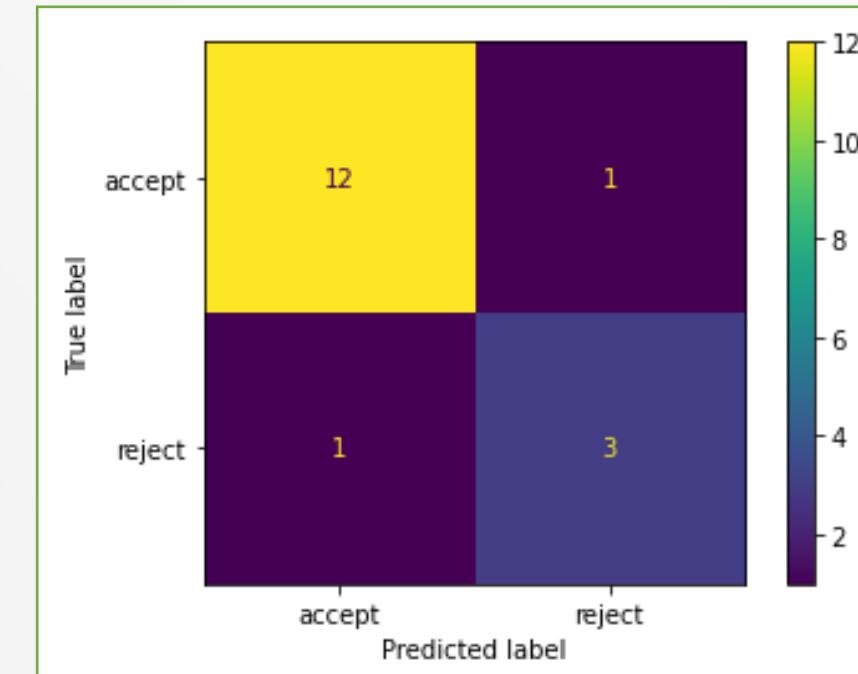
## 2. Scoring (*accuracy, precision, recall, F1*)



- Confusion Matrix for **training set**



- Confusion Matrix for **test set**



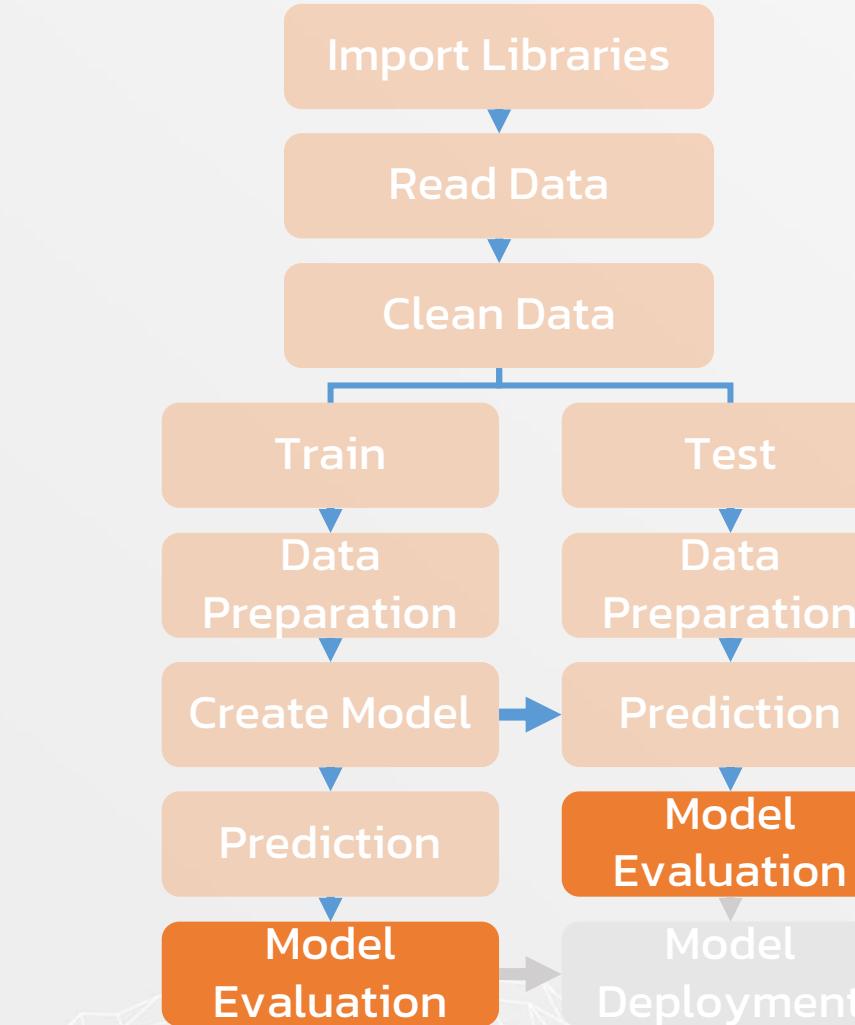
# Code

- Confusion Matrix for **training set**

```
1 plot_confusion_matrix(clf, X_train_scaled, y_train)
```

- Confusion Matrix for **test set**

```
1 plot_confusion_matrix(clf, X_train_scaled, y_train)
```



# Model Evaluation

1. Confusion Matrix
2. **Scoring (accuracy, precision, recall, F1)**



- Scoring for **training set**

- Accuracy

```
accuracy = 0.8676470588235294
```

- Precision, Recall and F1 for class '**accept**'

```
{'precision': 0.8636363636363636, 'recall': 0.926829268292683,  
'f1-score': 0.8941176470588236, 'support': 41}
```

- Precision, Recall and F1 for class '**reject**'

```
{'precision': 0.875, 'recall': 0.7777777777777778,  
'f1-score': 0.823529411764706, 'support': 27}
```

# Code

- Scoring for **training set**

```
1 report = classification_report(y_train, y_pred_train, output_dict=True)
```

```
1 print('accuracy =', report['accuracy'])
```

```
1 report['accept']
```

```
1 report['reject']
```

- Scoring for **test set**

- Accuracy

```
accuracy = 0.8823529411764706
```

- Precision, Recall and F1 for class '**accept**'

```
{'precision': 0.9230769230769231, 'recall': 0.9230769230769231,  
'f1-score': 0.9230769230769231, 'support': 13}
```

- Precision, Recall and F1 for class '**reject**'

```
{'precision': 0.75, 'recall': 0.75, 'f1-score': 0.75, 'support': 4}
```

# Code

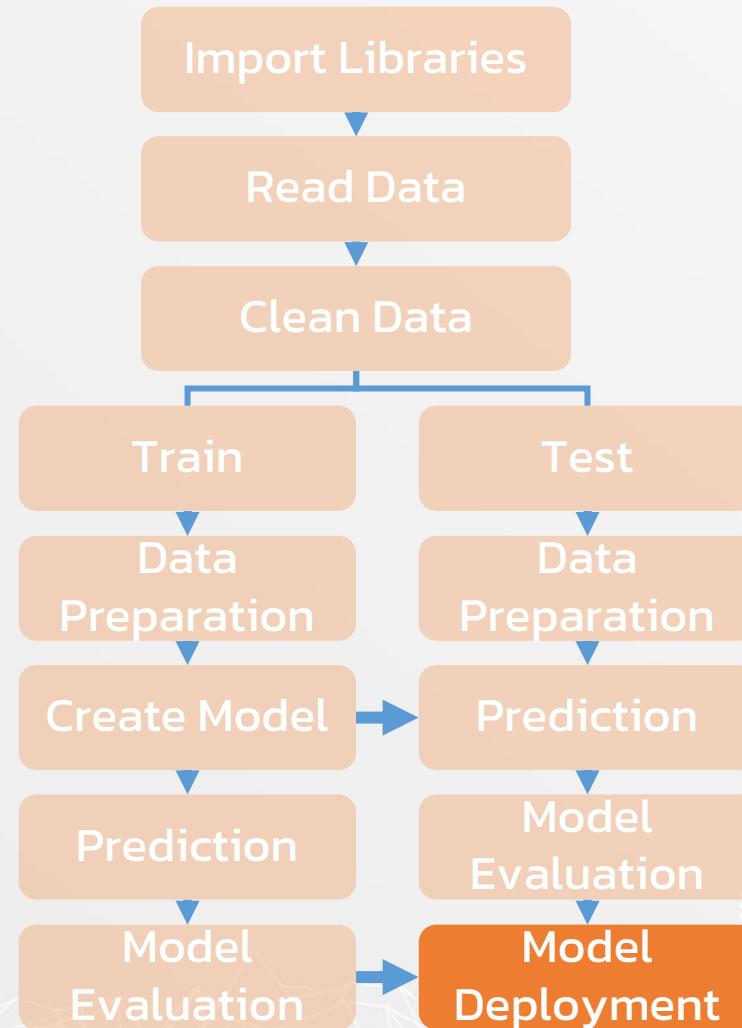
- Scoring for **test set**

```
1 report = classification_report(y_test, y_pred_test, output_dict=True)
```

```
1 print('accuracy =', report['accuracy'])
```

```
1 report['accept']
```

```
1 report['reject']
```



# Model Deployment



# Code

```
1 import pickle
2
3 pickle.dump((clf,
4                 ordinal_encoder,
5                 one_hot_encoder,
6                 scaler,
7                 feature_name,
8                 numerical_feature,
9                 ordinal_feature,
10                nominal_feature),
11                open('job_acceptance_model.pickle', 'wb'))
```





# AI in Healthcare

# Abstract

สร้าง model เพื่อวินิจฉัยผู้ป่วยโรคหัวใจ โดย feature ที่นำมาใช้ คือ ข้อมูลสภาวะร่างกาย เช่น

- ลักษณะการเจ็บหน้าอก
- ค่าความเข้มข้นน้ำตาลในเลือด
- ระดับคอเลสเตอรอลในเลือด



# Why this project important?



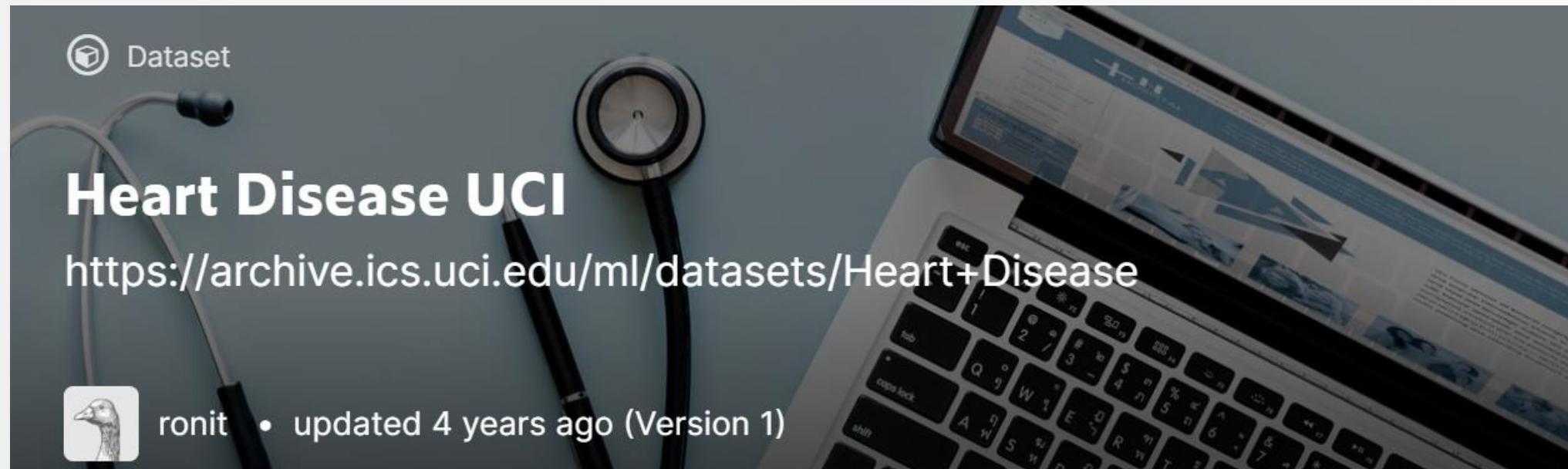
- ◆ สามารถสร้างระบบสำหรับตรวจโรคหัวใจที่ทำงานได้ตลอด 24 ชั่วโมง
- ◆ สามารถนำไปต่อยอดกับการวินิจฉัยโรคอื่น ๆ
- ◆ สามารถใช้เป็นพื้นฐานสำหรับการแพทย์ทางไกล

# Who this project is for?

- ◆ ผู้บริหารโรงพยาบาล
- ◆ บุคลากรทางการแพทย์
- ◆ นักวิเคราะห์ข้อมูล



# Heart Disease Dataset



<https://www.kaggle.com/ronitf/heart-disease-uci>

# Heart Disease Dataset

## Feature

- age : อายุ
- sex : เพศ (1 = ชาย, 0 = หญิง)
- cp : ลักษณะการเจ็บหน้าอก (0, 1, 2, 3)
- tressbph : ความดันโลหิตขณะพักร
- chol : ระดับคอเลสเตอโรลในเลือด
- fbs : ค่าความเข้มข้นน้ำตาลในเลือด  $> 120 \text{ mg/dl}$  (1 = จริง, 0 = เท็จ)
- restecg : ผลคลื่นไฟฟ้าหัวใจขณะพักร

# Heart Disease Dataset

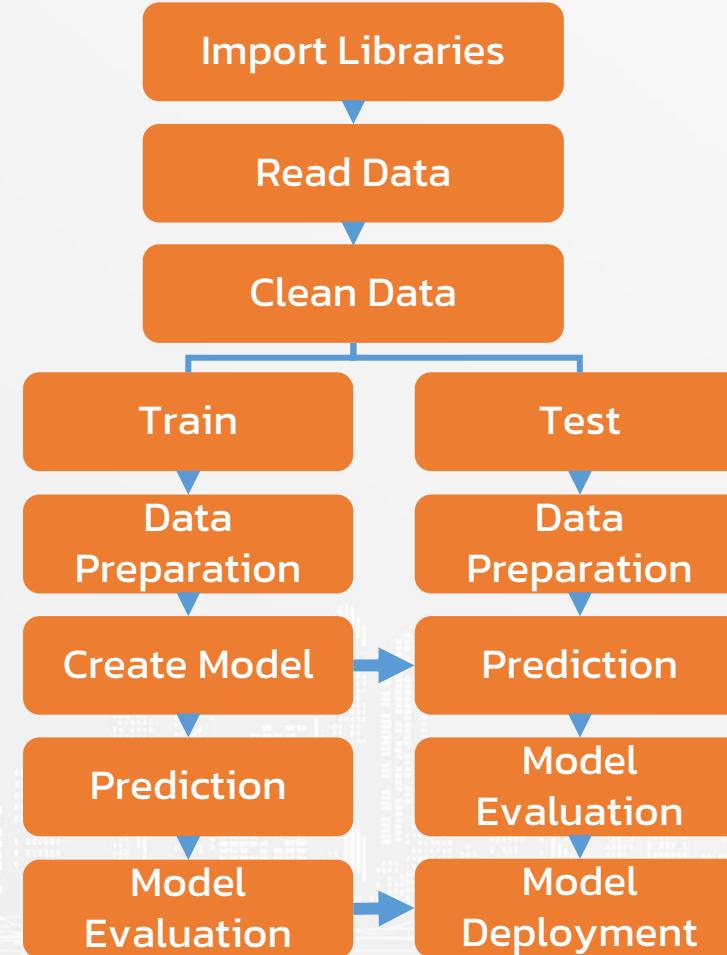
## Feature

- *thalach* : อัตราการเต้นหัวใจสูงสุด
- *exang* : อาการเจ็บหน้าอกระหว่างออกกำลังกาย (1 = เจ็บ, 0 = ไม่เจ็บ)
- *oldpeak* : การเกิดกราฟ ST depression ในผลการตรวจคลื่นไฟฟ้าของหัวใจ
- *slop* : ลักษณะความชันของกราฟ ST segment (0 = ชันขึ้น, 1 = ราบ, 2 = ชันลง)
- *ca* : จำนวนเส้นเลือดตีบ
- *thal* : ลักษณะความเครียดของหัวใจ (0, 1, 2, 3)

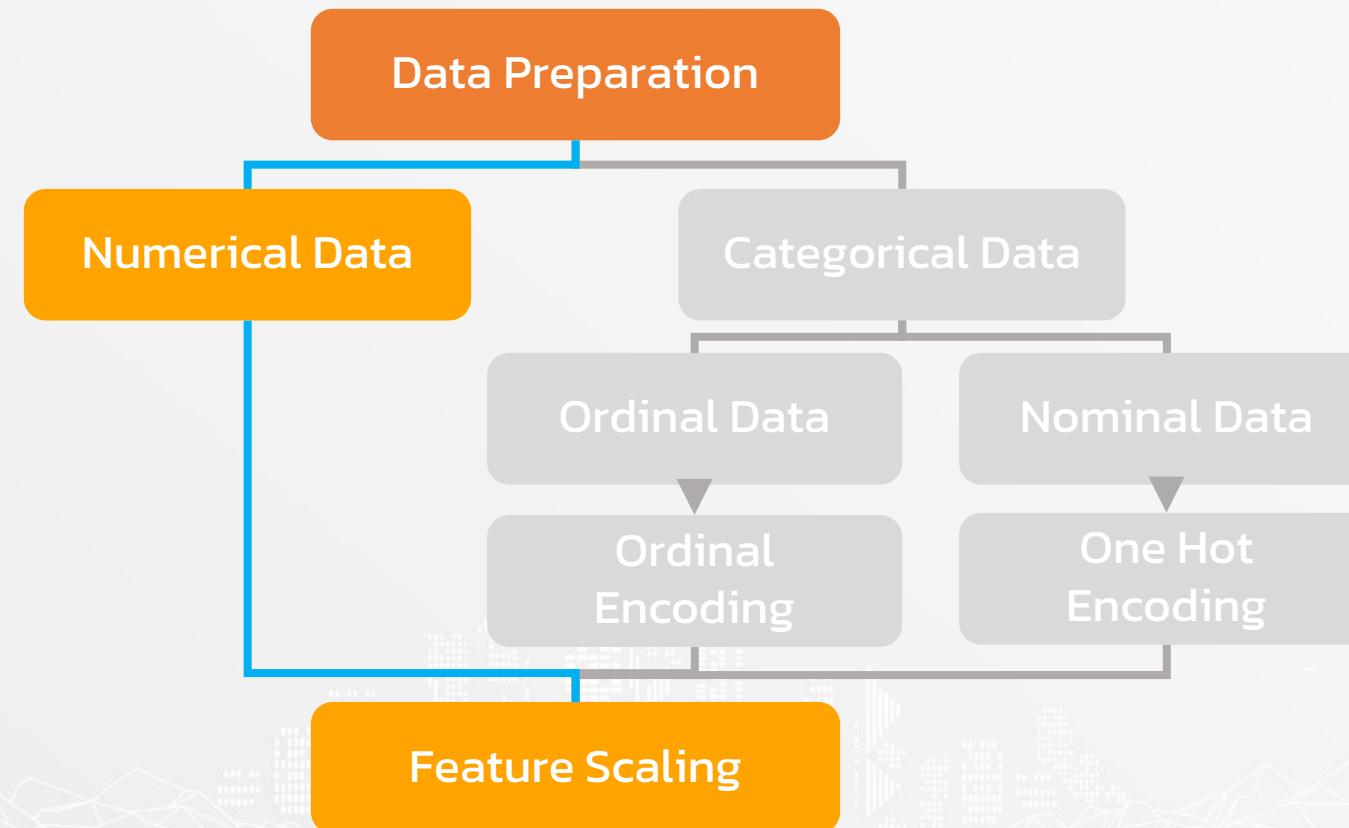
## Target

- *target* : การเป็นโรคหัวใจ (1 = เป็น, 0 = ไม่เป็น)

# What we learn from this project?



# Data Preparation





## 01. HEART DISEASE





# AI in Agriculture

# Abstract

สร้าง model เพื่อจำแนกว่าพืชเป็นโรคหรือไม่ โดยพิจารณาจากใบพืช



# Why this project important?



- ◆ สามารถสร้างระบบตรวจโรคพืชที่ทำงานได้อย่างกันกันกัน
- ◆ สามารถนำความรู้ไปต่อยอดเพื่อสร้าง smart farm
- ◆ สามารถนำไปต่อยอดตรวจโรคในพืชชนิดอื่น ๆ

# Who this project is for?

- ◆ เกษตรกรที่สนใจ AI กับการเกษตร
- ◆ ผู้ควบคุม/วางแผนการผลิต
- ◆ นักวิเคราะห์ข้อมูล



# Plant Disease Dataset



<http://archive.ics.uci.edu/ml/datasets/Leaf?ref=datanews.io#>

# Plant Disease Dataset

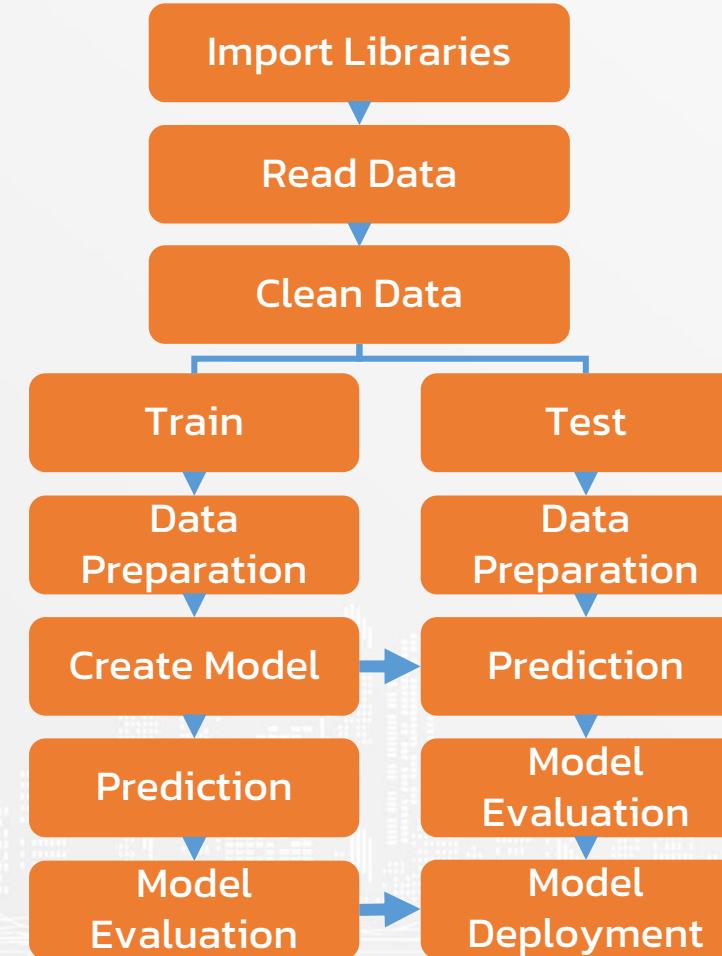
## Feature



## Target

- target : การเป็นโรคของพืช (healthy, unhealthy)

# What we learn from this project?



# Import Libraries

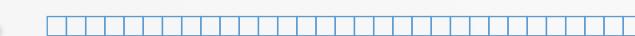
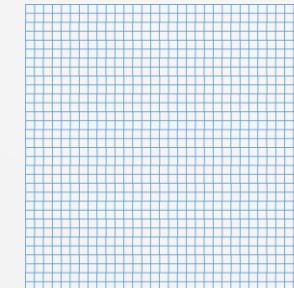
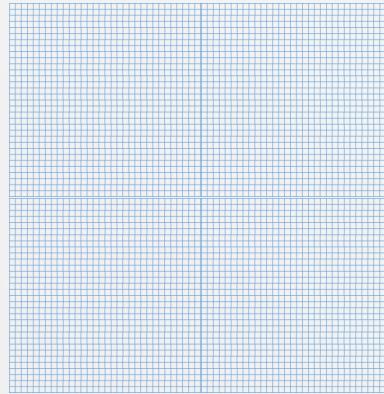
```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 from glob import glob
6 from PIL import Image
7 import cv2
8 from tqdm.auto import tqdm
9 from sklearn.model_selection import train_test_split
10 from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder, StandardScaler, MinMaxScaler
11 from sklearn.linear_model import LogisticRegression
12 from sklearn.metrics import (
13     plot_confusion_matrix,
14     classification_report
15 )
16
17 import warnings
18 warnings.filterwarnings('ignore')
19
20 np.random.seed(12345)
```

# Read Data

```
1 classes = ['angular_leaf_spot', 'bean_rust', 'healthy']
```

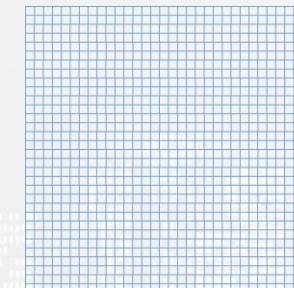
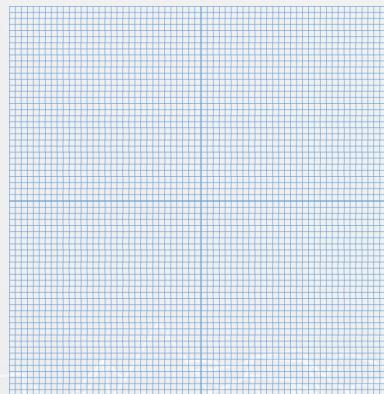
```
1 X = np.empty([0, 32*32*3])
2 y = np.empty([0, 1])
3
4 for _class in tqdm(classes):
5     img_path = glob('dataset/' + _class + '/*')
6     for path in tqdm(img_path):
7         img = Image.open(path)
8         img = img.resize([32, 32])
9         img = np.array(img)
10        if img.shape[2] == 4:
11            img = cv2.cvtColor(img, cv2.COLOR_BGRA2BGR)
12            img = img.reshape(1, -1)
13        X = np.vstack([X, img])
14        if _class == 'healthy':
15            y = np.vstack([y, 'healthy'])
16        else:
17            y = np.vstack([y, 'unhealthy'])
```

# Read Data



healthy

32x32



unhealthy

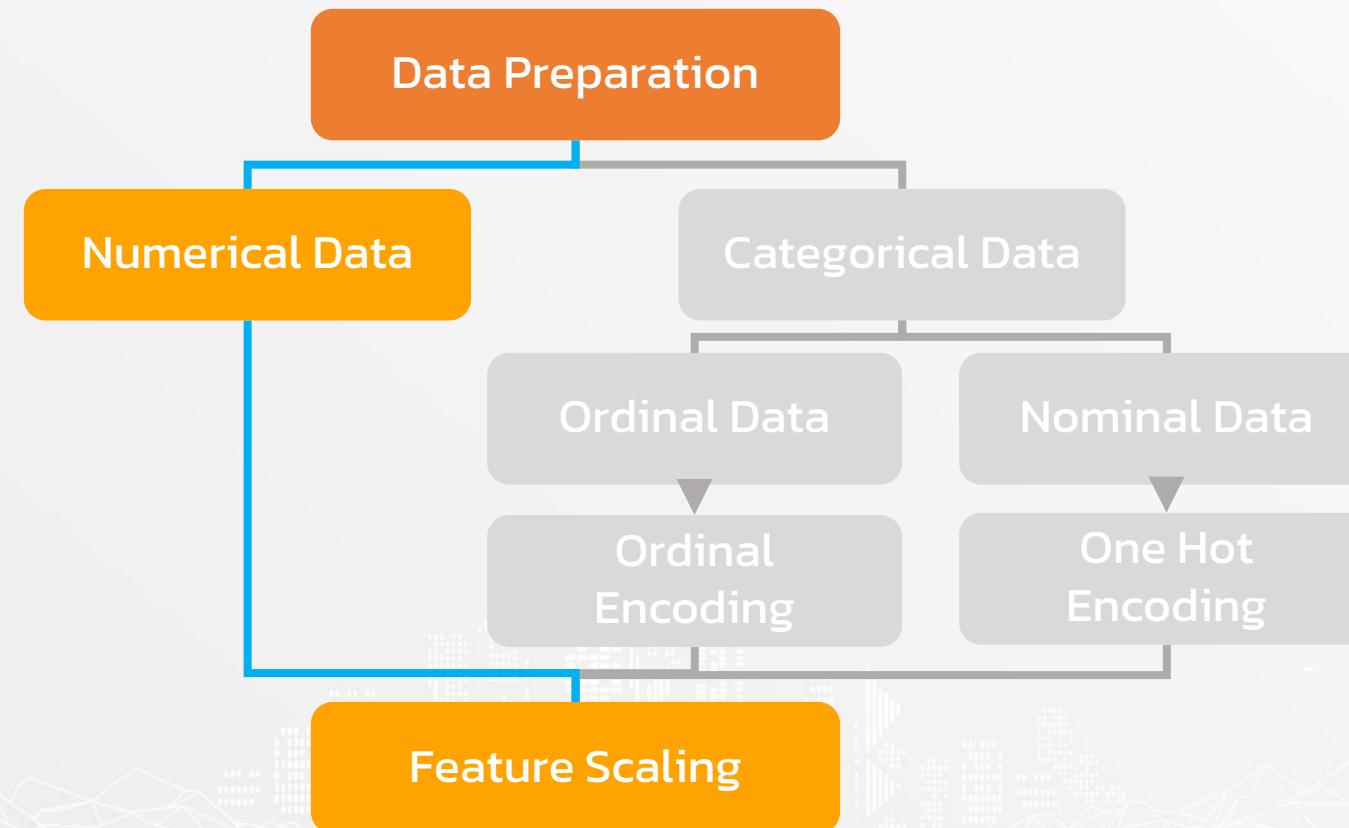
32x32

# Read Data

$x_1$	$x_2$	$x_3$	...	$x_{3072}$	$y$
91.0	127.0	20.0	...	54.0	unhealthy
117.0	97.0	67.0	...	35.0	unhealthy
203.0	206.0	212.0	...	21.0	unhealthy
:	:	:	:	:	:
60.0	122.0	5.0	...	53.0	healthy

 $X$  $y$

# Data Preparation





## 02. PLANT DISEASE





# Fake News

# Abstract

สร้าง model เพื่อตรวจสอบข่าวปลอมโดยพิจารณาจากหัวข้อข่าว, เนื้อหาข่าว และหมวดหมู่ข่าว



# Why this project important?



- ◆ สามารถสร้างระบบตรวจสอบข่าวปลอมที่ทำงานได้ตลอด 24 ชั่วโมง
- ◆ สามารถนำไปต่อยอดเพื่อจัดอันดับความน่าเชื่อถือของสื่อมวลชน
- ◆ สามารถนำไปประยุกต์ใช้กับงานที่มีลักษณะใกล้เคียงได้ เช่น *sentimental analysis*

# Who this project is for?

- ◆ บุคลากรด้านสื่อเมืองชน
- ◆ นักลงทุน
- ◆ นักวิเคราะห์ข้อมูล



# Fake News Dataset



<https://www.kaggle.com/c/fake-news/data>

# Fake News Dataset

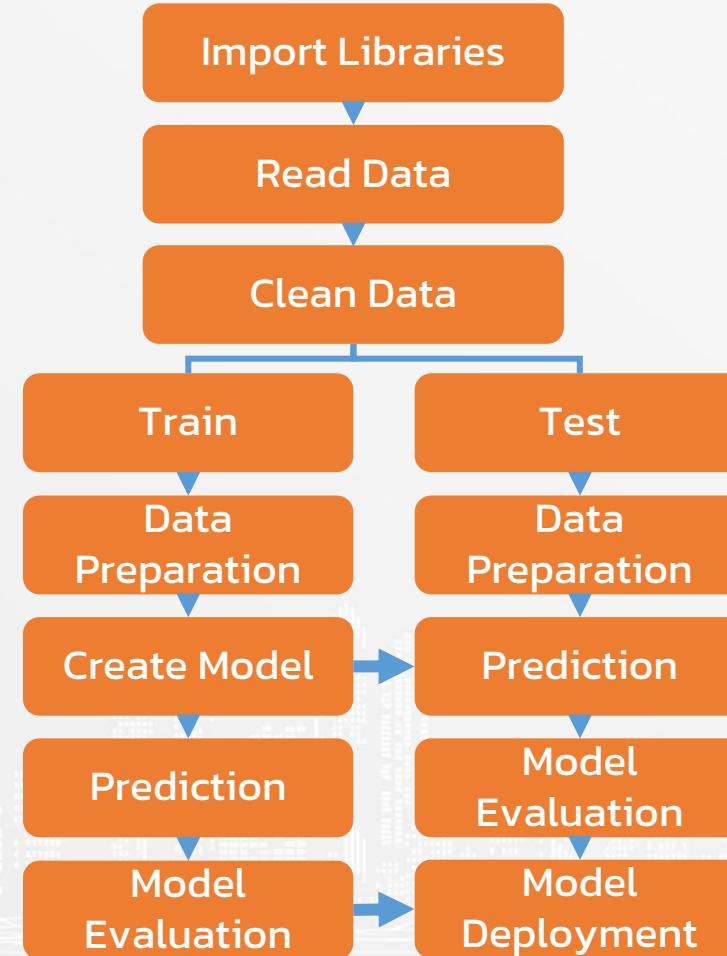
## Feature

- title : หัวข้อข่าว
- text : เนื้อหาข่าว
- subject : หมวดหมู่ข่าว

## Target

- class : ค่าความจริงของข่าว (fake, true)

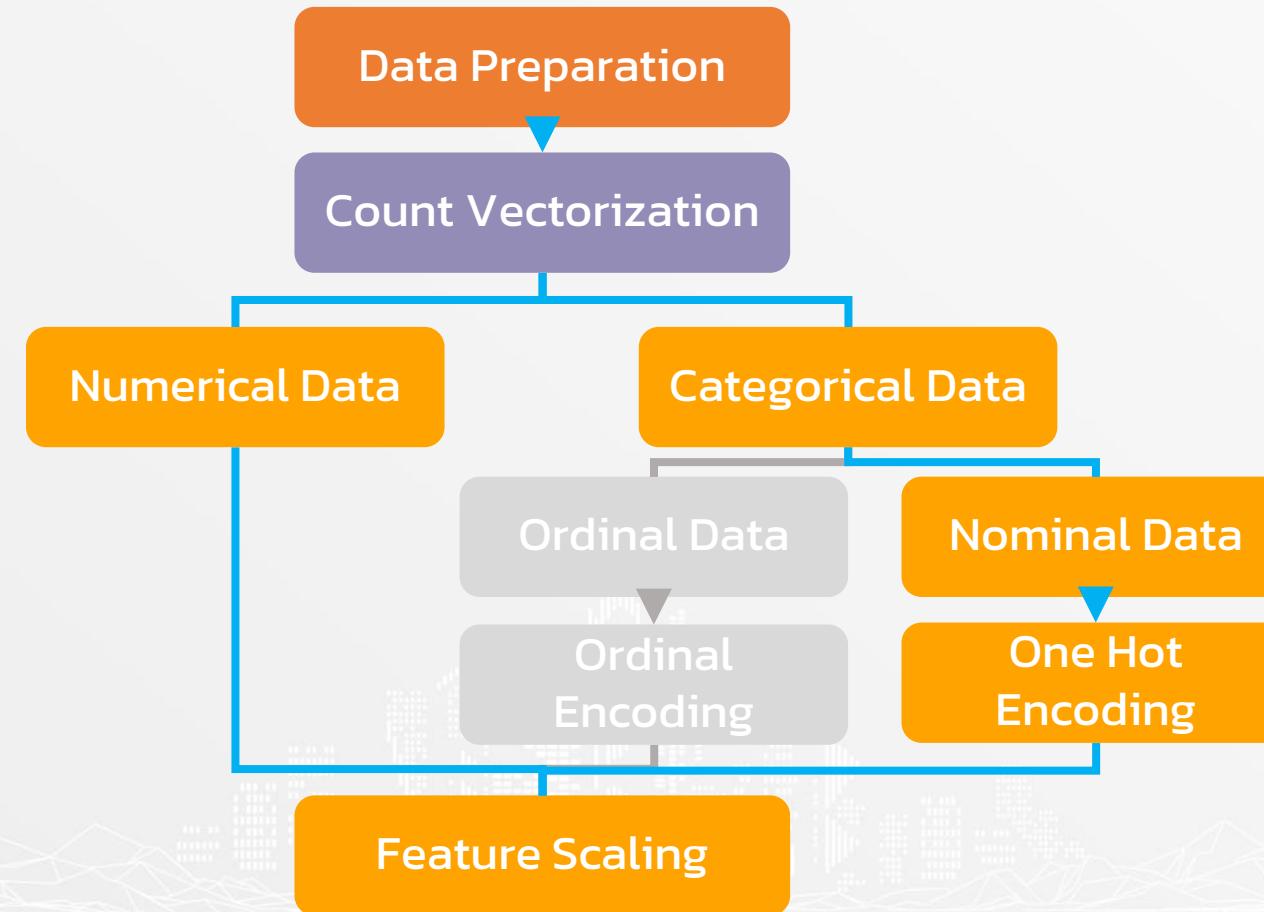
# What we learn from this project?



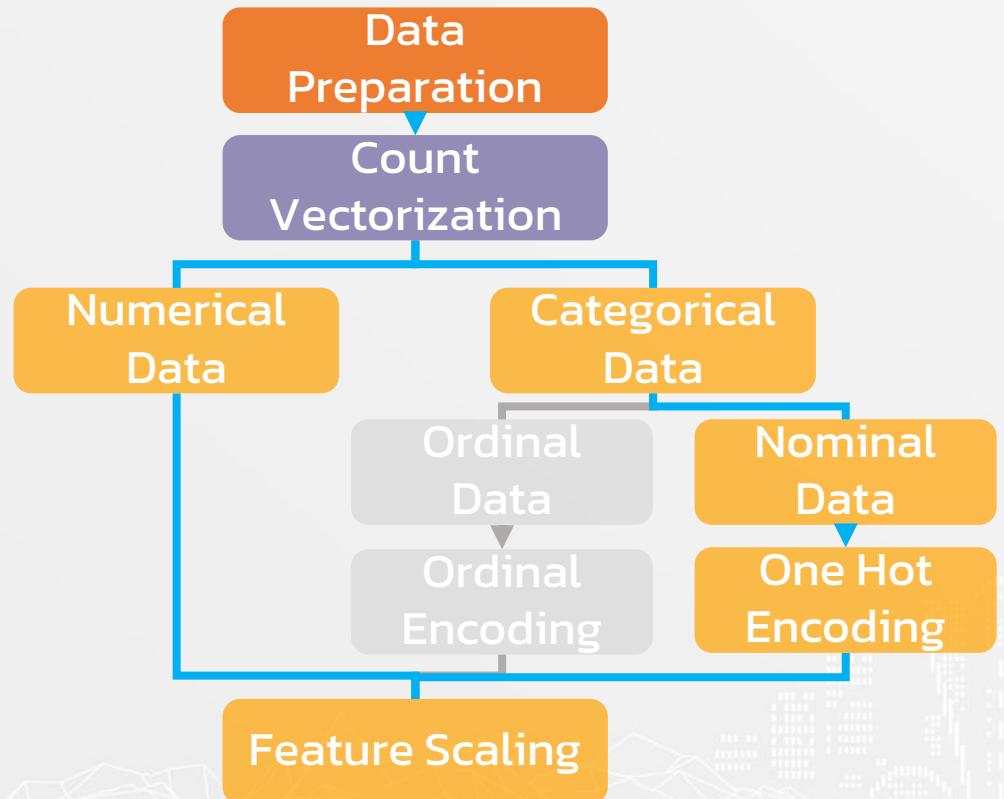
# Import Libraries

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 from sklearn.model_selection import train_test_split
6 from sklearn.feature_extraction.text import CountVectorizer
7 from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder, StandardScaler, MinMaxScaler
8 from sklearn.linear_model import LogisticRegression, LogisticRegressionCV
9 from sklearn.metrics import plot_confusion_matrix, classification_report
10
11 import warnings
12 warnings.filterwarnings('ignore')
13
14 np.random.seed(12345)
```

# Data Preparation



# Data Preparation



## Count vectorization

สร้าง feature ใหม่ โดยการหา unique word จากข้อความทั้งหมดใน dataset จากนั้นให้พิจารณาว่าแต่ละข้อความประกอบด้วย unique word อะไรบ้าง และจำนวนกี่ครั้ง

	'apple'	'green'	'is'	'kiwi'	'orange'	'red'
'Apple is red'	1	0	1	0	0	1
'Kiwi is green'	0	1	1	1	0	0
'Orange is orange'	0	0	1	0	2	0

# Count Vectorization

	cnt_title _000	cnt_title _10	cnt_title _100	...	cnt_title _year	cnt_title _years	...
BAGHDAD (Reuters) – A Russian Islamic State fi...	0	0	0	...	0	0	...
WASHINGTON (Reuters) – President Donald Trump ...	0	0	0	...	0	0	...
WASHINGTON (Reuters) – Russian President Vlad... ...	0	0	0	...	0	0	...
...	...	...	...	...	...	...	...

# Code

- Count vectorization for **training set**

```
1 corpus_train = X_train["title"].tolist()
2 title_vectorizer = CountVectorizer(max_features=1000)
3 title_vectorizer.fit(corpus_train)
4 title_cnt_vec_train = title_vectorizer.transform(corpus_train).toarray()
```

```
1 title_cnt_vec_feature_name = [
2     "cnt_title_" + feature for feature in title_vectorizer.get_feature_names()
3 ]
```

```
1 X_train[title_cnt_vec_feature_name] = title_cnt_vec_train
2 X_train.drop("title", axis=1, inplace=True)
```

# Code

- Count vectorization for **test set**

```
1 corpus_test = X_test['title'].tolist()  
2 title_cnt_vec_test = title_vectorizer.transform(corpus_test).toarray()
```

```
1 X_test[title_cnt_vec_feature_name] = title_cnt_vec_test  
2 X_test.drop('title', axis=1, inplace=True)
```

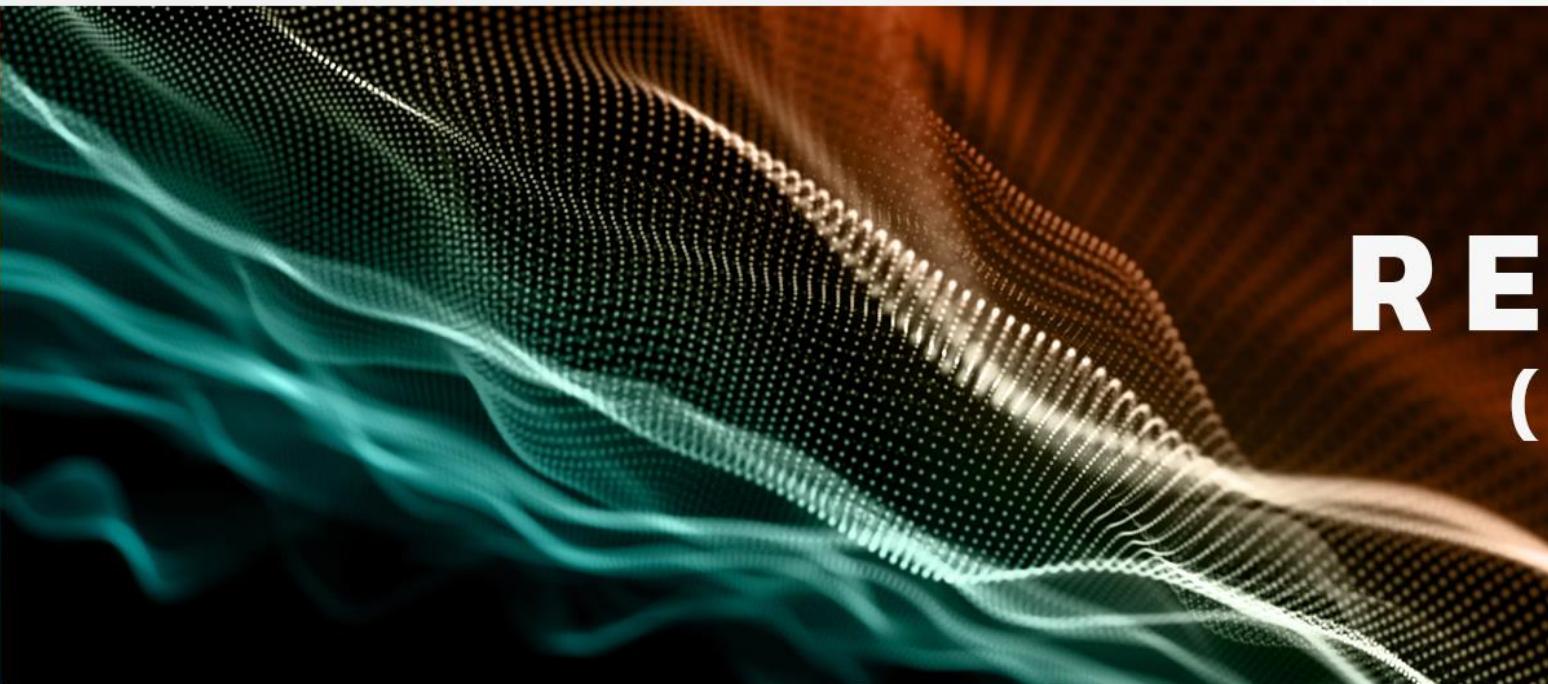


## 03. FAKE NEWS



# DL102 : Logistic Regression





# LOGISTIC REGRESSION (MULTI-CLASS)

BY TAU TOLOGY

# Logistic Regression (Multi-Class Classification)



# Introduction

# Introduction

Logistic Regression  
with Multiclass

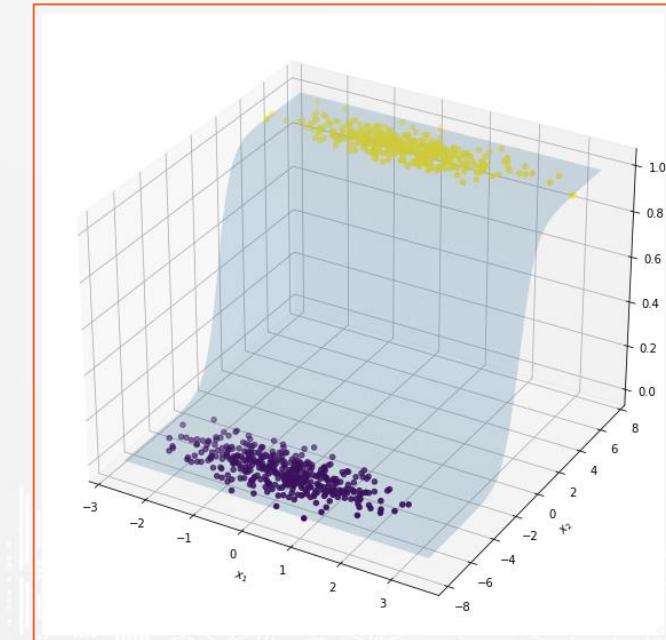
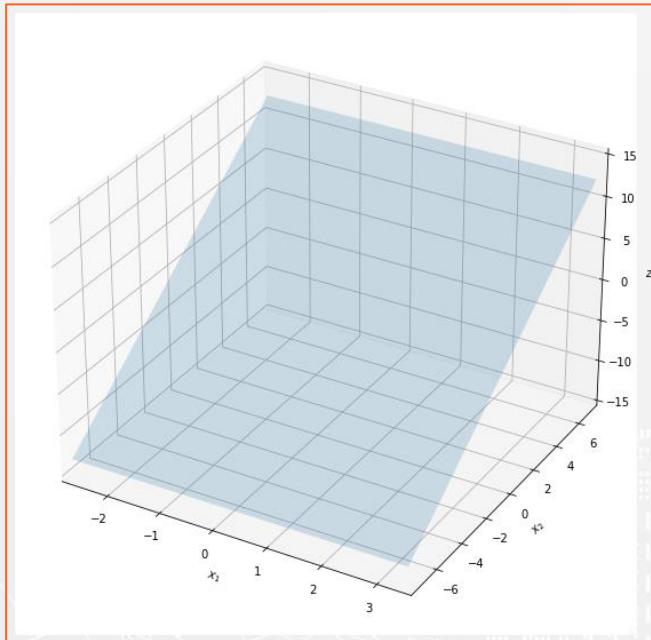
Why Softmax ?

Extension to Neural  
Network

Real World  
Application

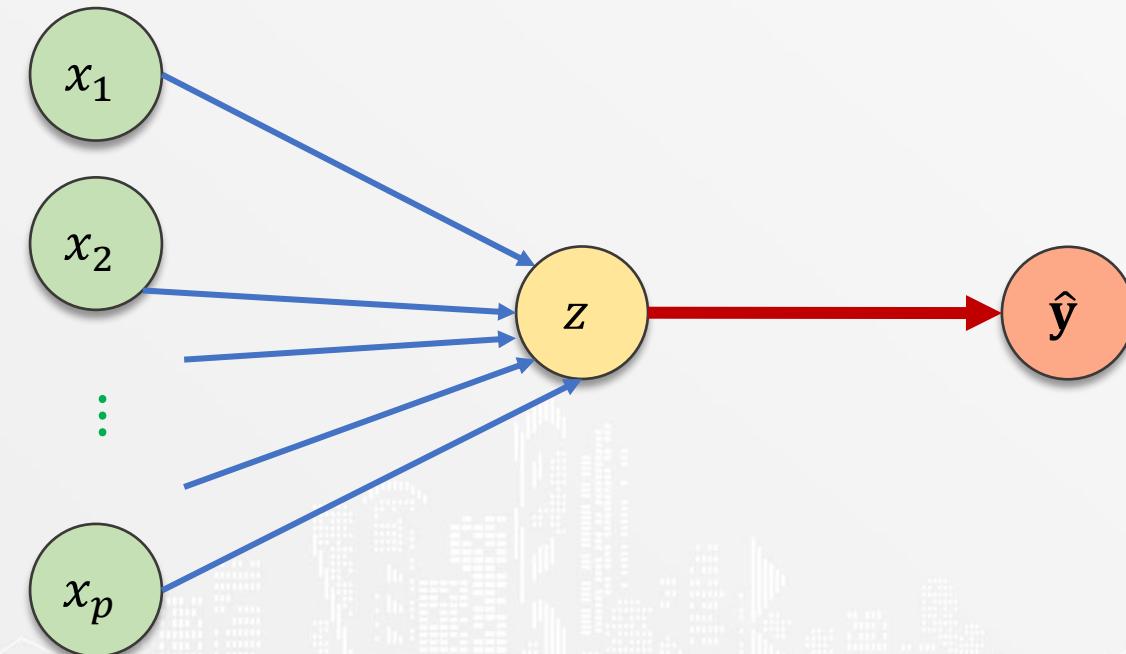
# Logistic Regression with Multiclass

Binary Classification



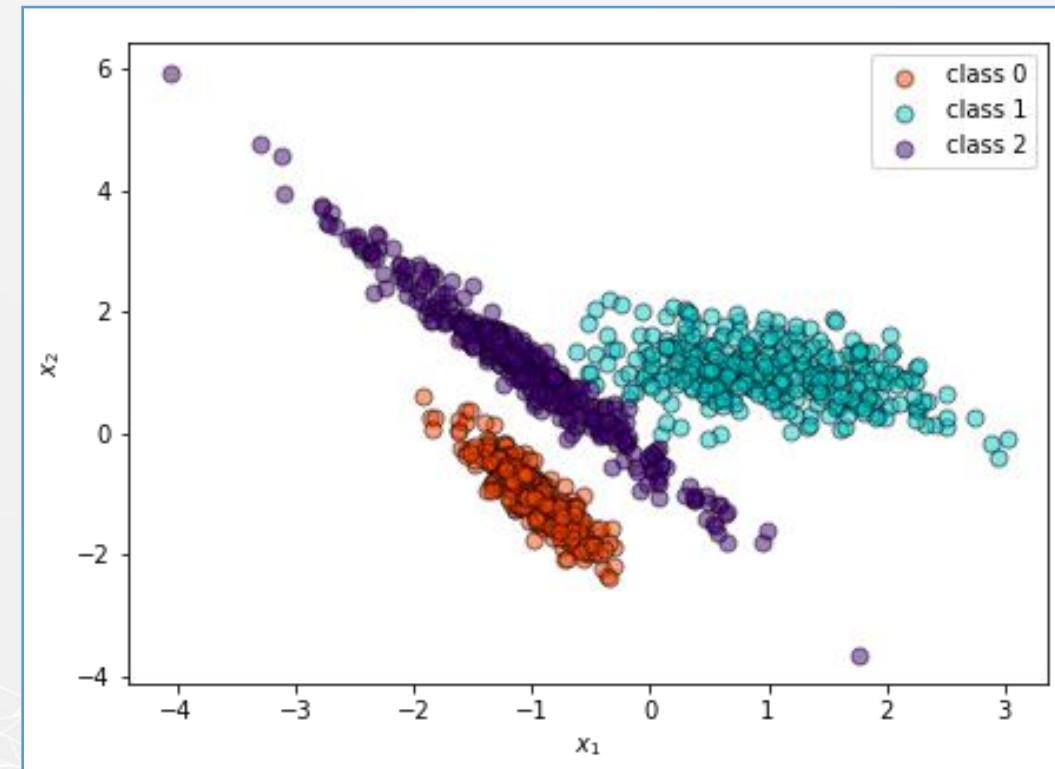
# Logistic Regression with Multiclass

## Binary Classification



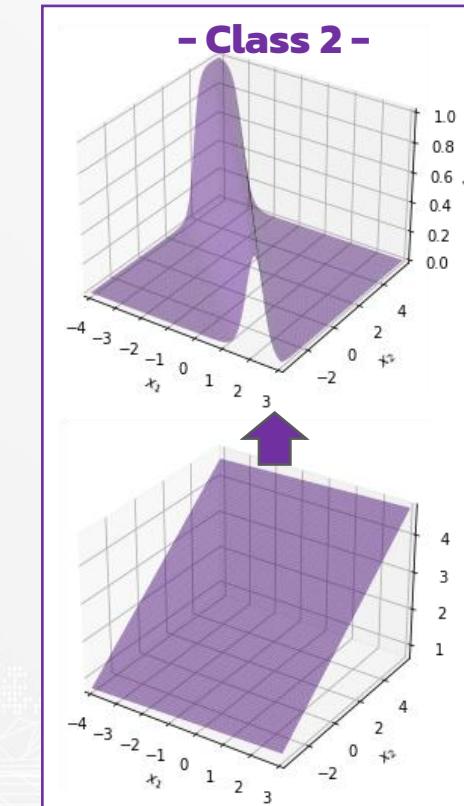
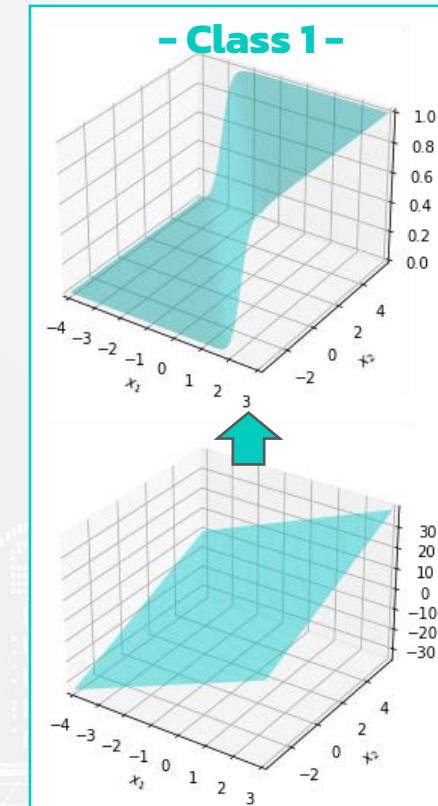
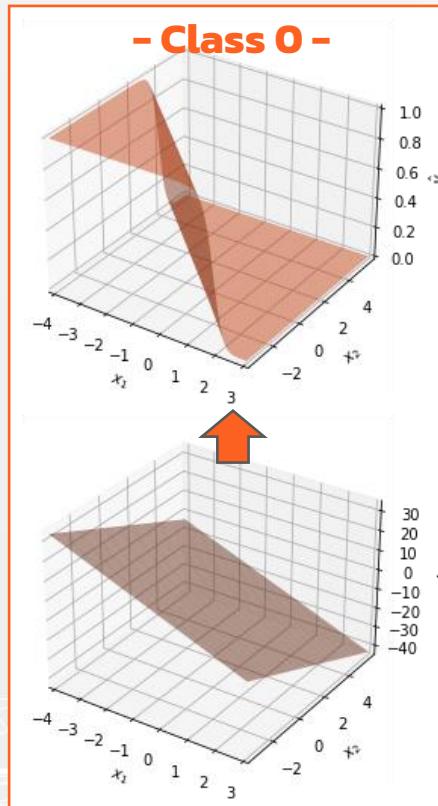
# Logistic Regression with Multiclass

## Muti-Class Classification



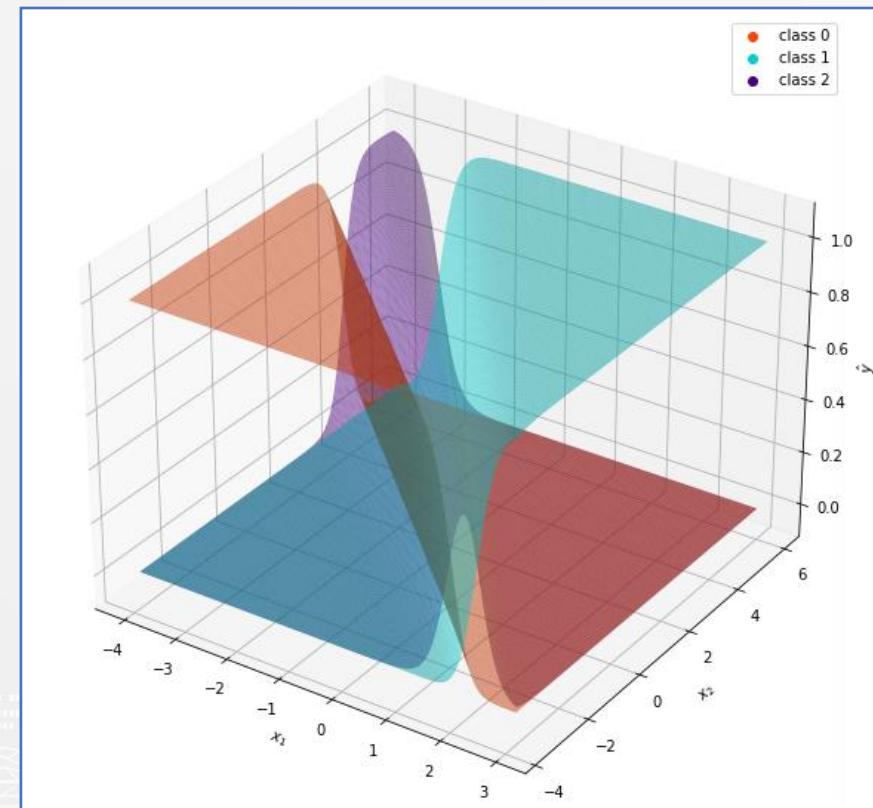
# Logistic Regression with Multiclass

## Muti-Class Classification



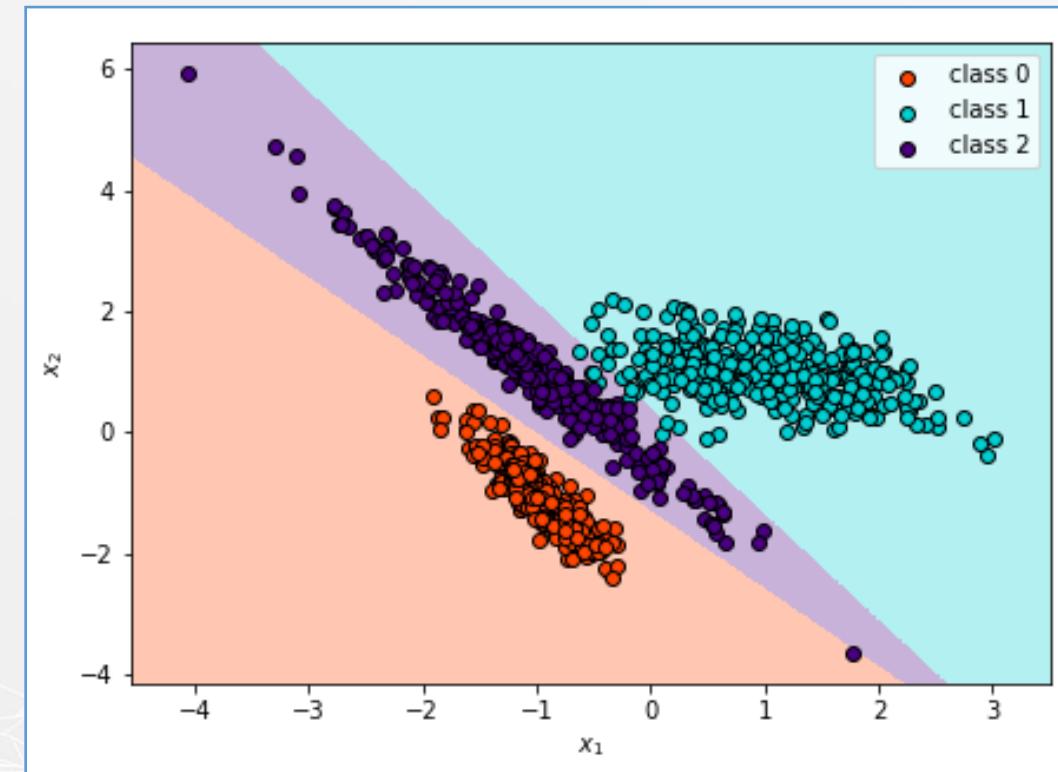
# Logistic Regression with Multiclass

## Muti-Class Classification



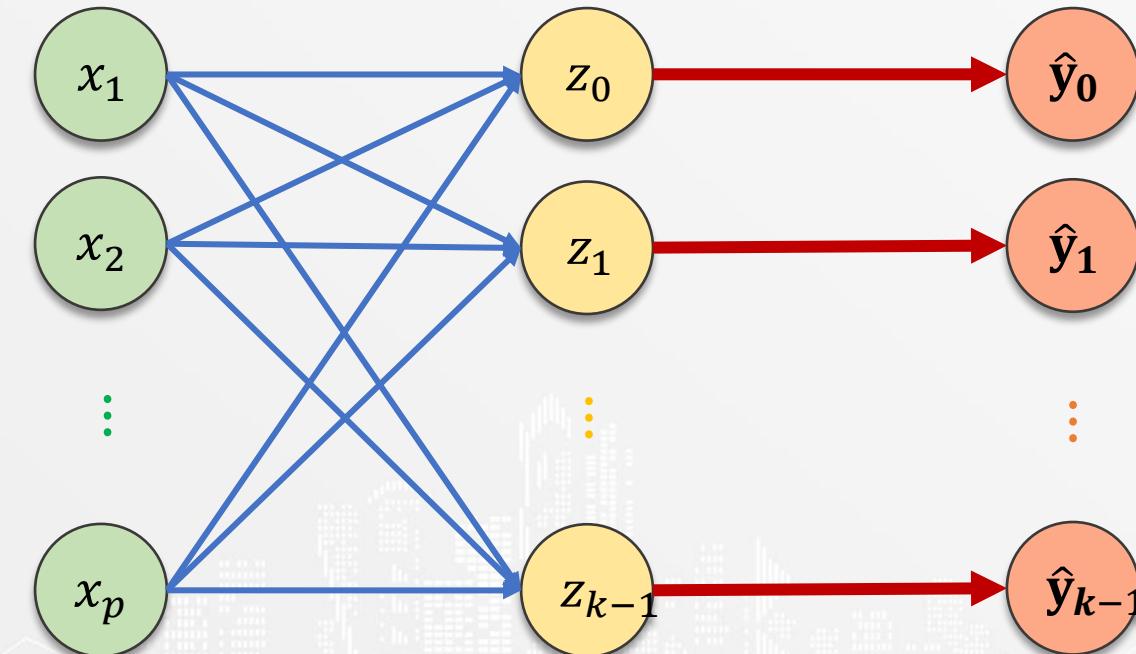
# Logistic Regression with Multiclass

## Muti-Class Classification



# Logistic Regression with Multiclass

## Muti-Class Classification



# Logistic Regression with Multiclass

สมการคณิตศาสตร์ของ Logistic Regression แบบ **Multiclass**

$$z_m = w_{0,m} + w_{1,m}x_1 + w_{2,m}x_2 + \cdots + w_{p,m}x_p$$

$$\hat{y}_m = \frac{e^{z_m}}{\sum_{c=0}^{k-1} e^{z_c}}$$

โดยที่

- ◆  $\hat{y}_m$  คือ ความน่าจะเป็นที่จะเป็น class ที่  $m$  (predicted target)
- ◆  $z_m$  คือ สมการเชิงเส้น (linear regression) ของ class ที่  $m$
- ◆  $x_1, x_2, x_3, \dots, x_p$  คือ ตัวแปรตัว (feature)
- ◆  $w_{0,m}, w_{1,m}, \dots, w_{p,m}$  คือ สัมประสิทธิ์ที่ส่งจาก feature ไป class ที่  $m$  (coefficient)
- ◆  $m \in 0, 1, 2, \dots, k - 1$

# Introduction

**Logistic Regression  
with Multiclass**



**Why Softmax ?**

**Extension to Neural  
Network**

**Real World  
Application**

# Why Softmax ?

**Why**  $\hat{y}_m = \frac{e^{z_m}}{\sum_{c=0}^{k-1} e^{z_c}}$  ?

# Why Softmax ?

**Ans :** เราต้องการให้  $\hat{y}_m$  อยู่ในรูปของความน่าจะเป็น ดังนั้น สมการของ  $\hat{y}_m$  ต้องอยู่ในรูปของ  $\frac{f(z_m)}{\sum_{c=0}^{k-1} f(z_c)}$

# Why Softmax ?

Why  $e^z$  ?

# Why Softmax ?

Logic ของ  $f(z)$  ที่กำให้  $\hat{y}_m$  มีสมบัติของความน่าจะเป็น

1.  $f(z)$  มีค่ามากกว่าหรือเท่ากับ 0 และเป็น monotone function
2. เมื่อ  $z$  ต่างกัน เราอยากรู้  $f(z)$  ต่างกันอย่างชัดเจน
3. เราซับ  $f(z)$  ที่ diff ได้

# Why Softmax ?

- ◆  $f(z) = z$
- ◆  $f(z) = |z|$
- ◆  $f(z) = z^2$
- ◆  $f(z) = z^n$
- ◆  $f(z) = \log z$
- ◆  $f(z) = a^z$
- ◆  $f(z) = \sigma(z)$
- ◆  $f(z) = \text{relu}(z)$
- ◆  $f(z) = \sin(z)$
- ◆  $f(z) = \cos(z)$

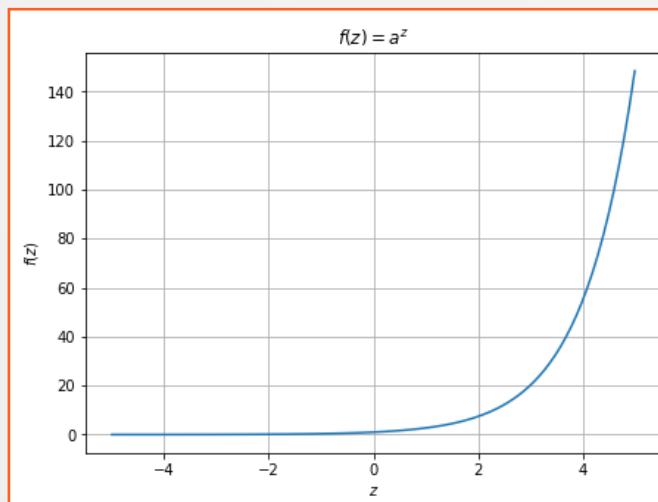
# Why Softmax ?

1.  $f(z)$  มีค่ามากกว่าหรือเท่ากับ 0 และเป็น monotone function

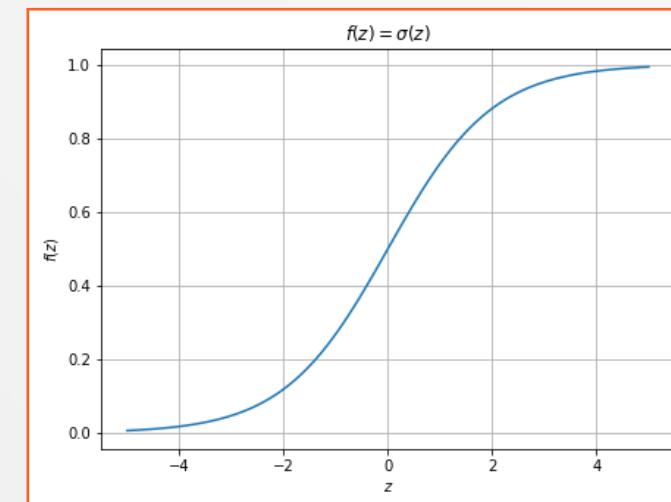
- ◆  ~~$f(z) = z$~~
- ◆  ~~$f(z) = |z|$~~
- ◆  ~~$f(z) = z^2$~~
- ◆  ~~$f(z) = z^n$~~
- ◆  ~~$f(z) = \log z$~~
- ◆  $f(z) = a^z$
- ◆  $f(z) = \sigma(z)$
- ◆  $f(z) = \text{relu}(z)$
- ◆  ~~$f(z) = \sin(z)$~~
- ◆  ~~$f(z) = \cos(z)$~~

# Why Softmax ?

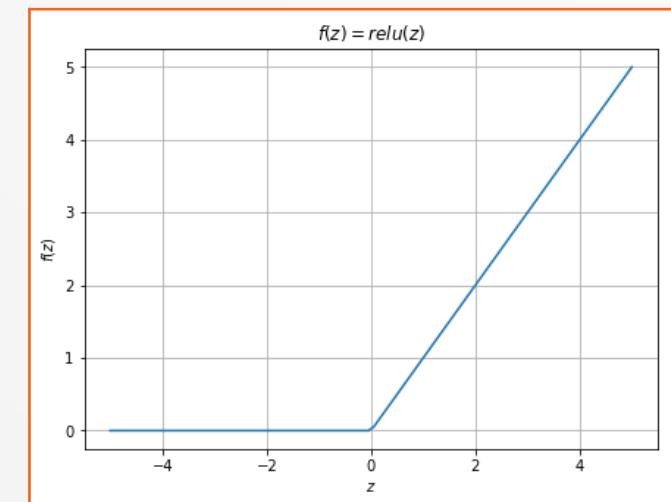
1.  $f(z)$  มีค่ามากกว่าหรือเท่ากับ 0 และเป็น monotone function



$$f(z) = a^z$$



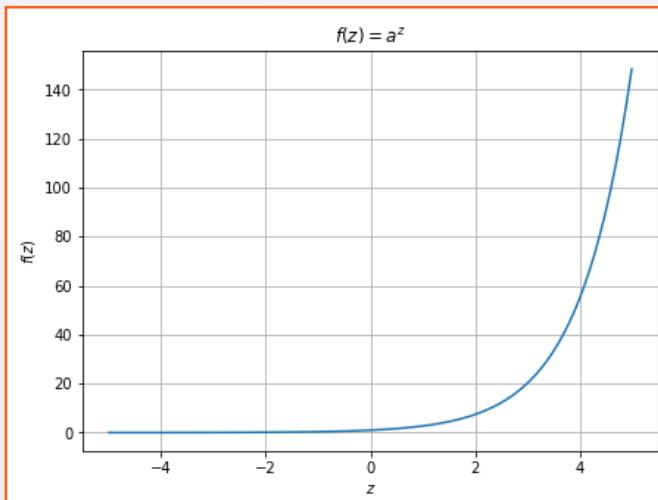
$$f(z) = \sigma(z)$$



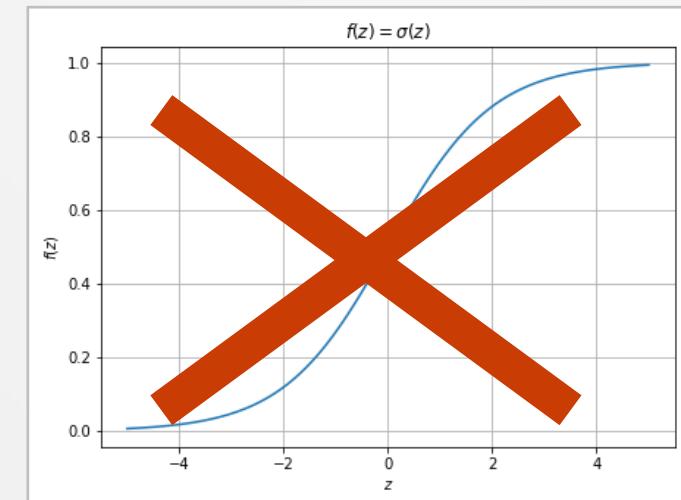
$$f(z) = \text{relu}(z)$$

# Why Softmax ?

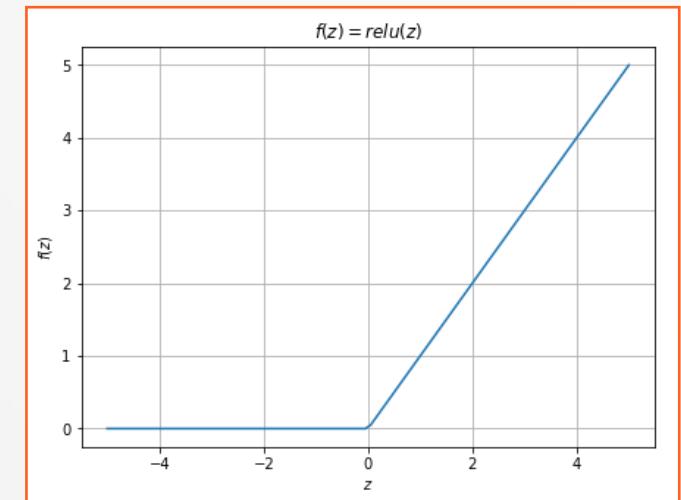
2. เมื่อ  $z$  ต่างกัน เราอยากให้  $f(z)$  ต่างกันอย่างชัดเจน



$$f(z) = a^z$$



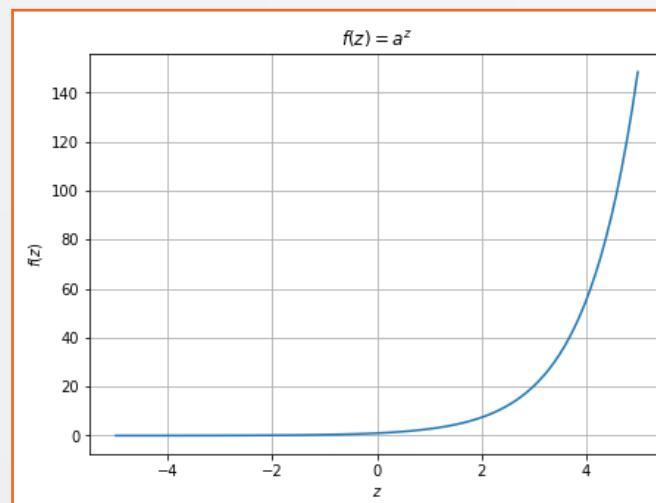
$$f(z) = \sigma(z)$$



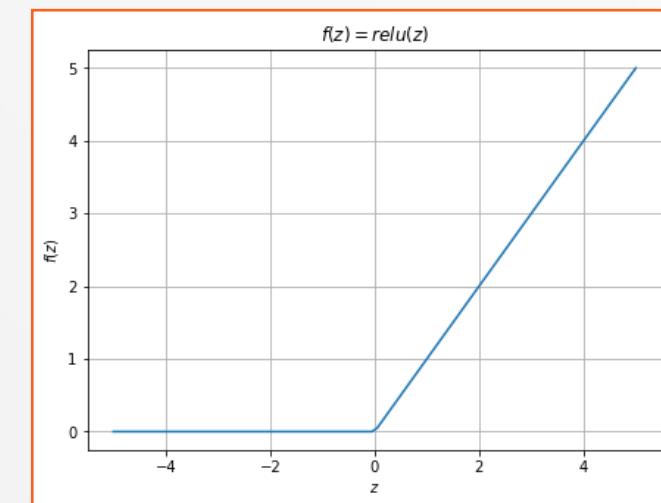
$$f(z) = \text{relu}(z)$$

# Why Softmax ?

2. เมื่อ  $z$  ต่างกัน เราอยากให้  $f(z)$  ต่างกันอย่างชัดเจน



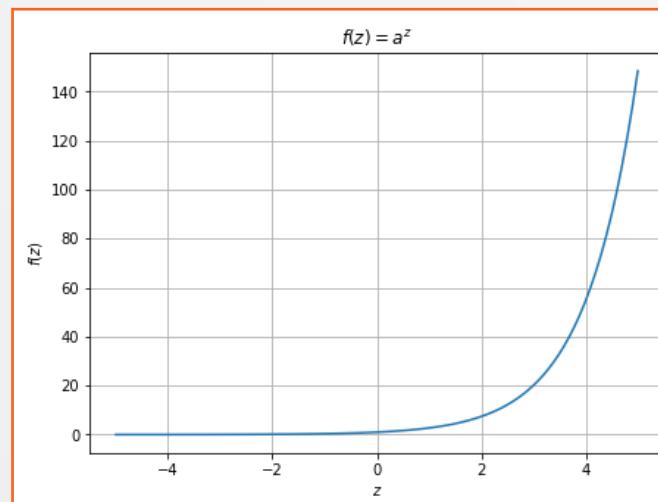
$$f(z) = a^z$$



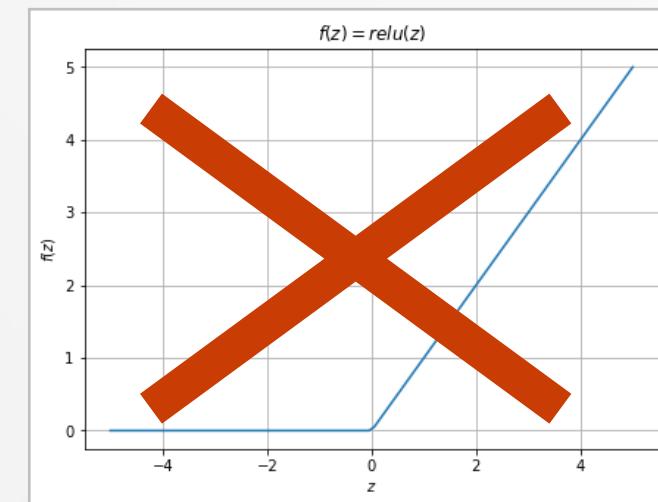
$$f(z) = \text{relu}(z)$$

# Why Softmax ?

3. เราชอบ  $f(z)$  ที่ diff ได้



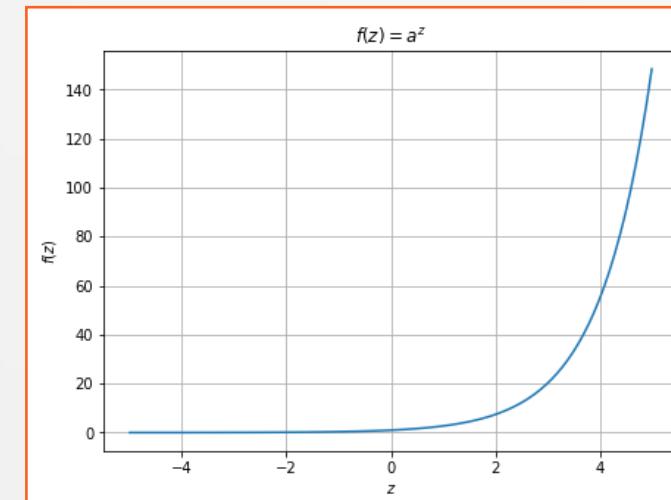
$$f(z) = a^z$$



$$f(z) = \text{relu}(z)$$

# Why Softmax ?

3. เราชอบ  $f(z)$  ที่ diff ได้



$$f(z) = a^z$$

# Why Softmax ?

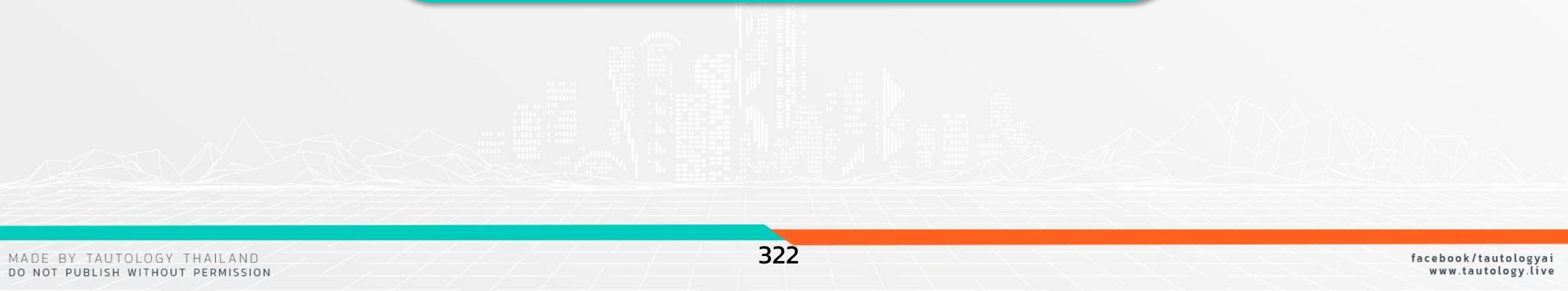
Why  $a = e$  ?

$** e = 2.718281828459045$

# Why Softmax ?

$$\text{Ans: } \frac{\partial a^z}{\partial z} = a^z \log a$$

$$\frac{\partial e^z}{\partial z} = e^z$$



# Why Softmax ?

เราต้องการให้  $\hat{y}_m$  อยู่ในรูปของความน่าจะเป็น  
ดังนั้น สมการของ  $\hat{y}_m$  ต้องอยู่ในรูปของ  $\frac{f(z_m)}{\sum_{c=0}^{k-1} f(z_c)}$

# Why Softmax ?

$$\hat{y}_m = \frac{e^{z_m}}{\sum_{c=0}^{k-1} e^{z_c}}$$

# Introduction

**Logistic Regression  
with Multiclass**



**Why Softmax ?**



**Extension to Neural  
Network**

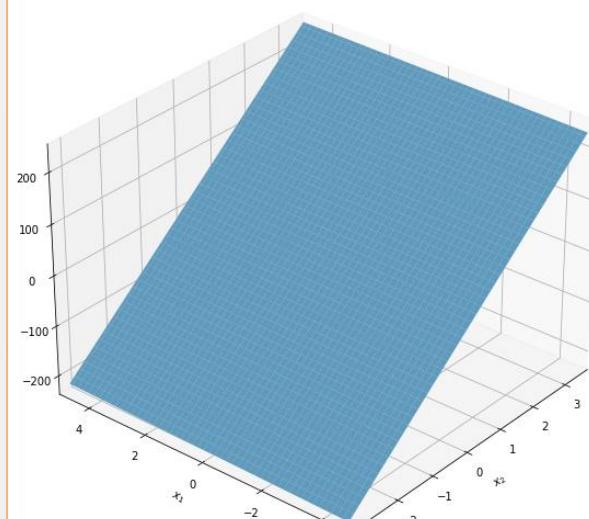


**Real World  
Application**

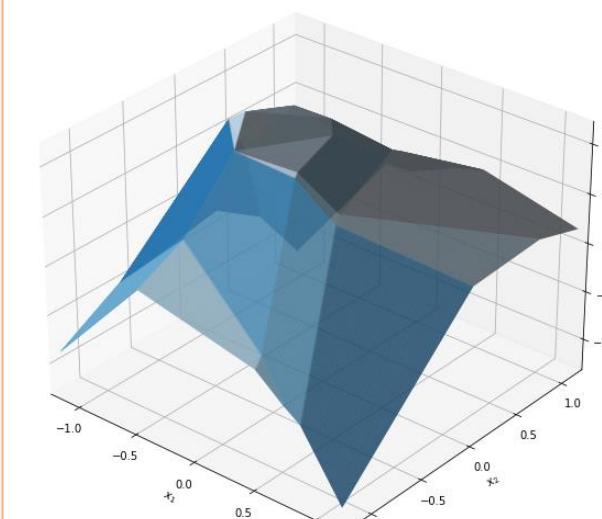


# Extension to Neural Network

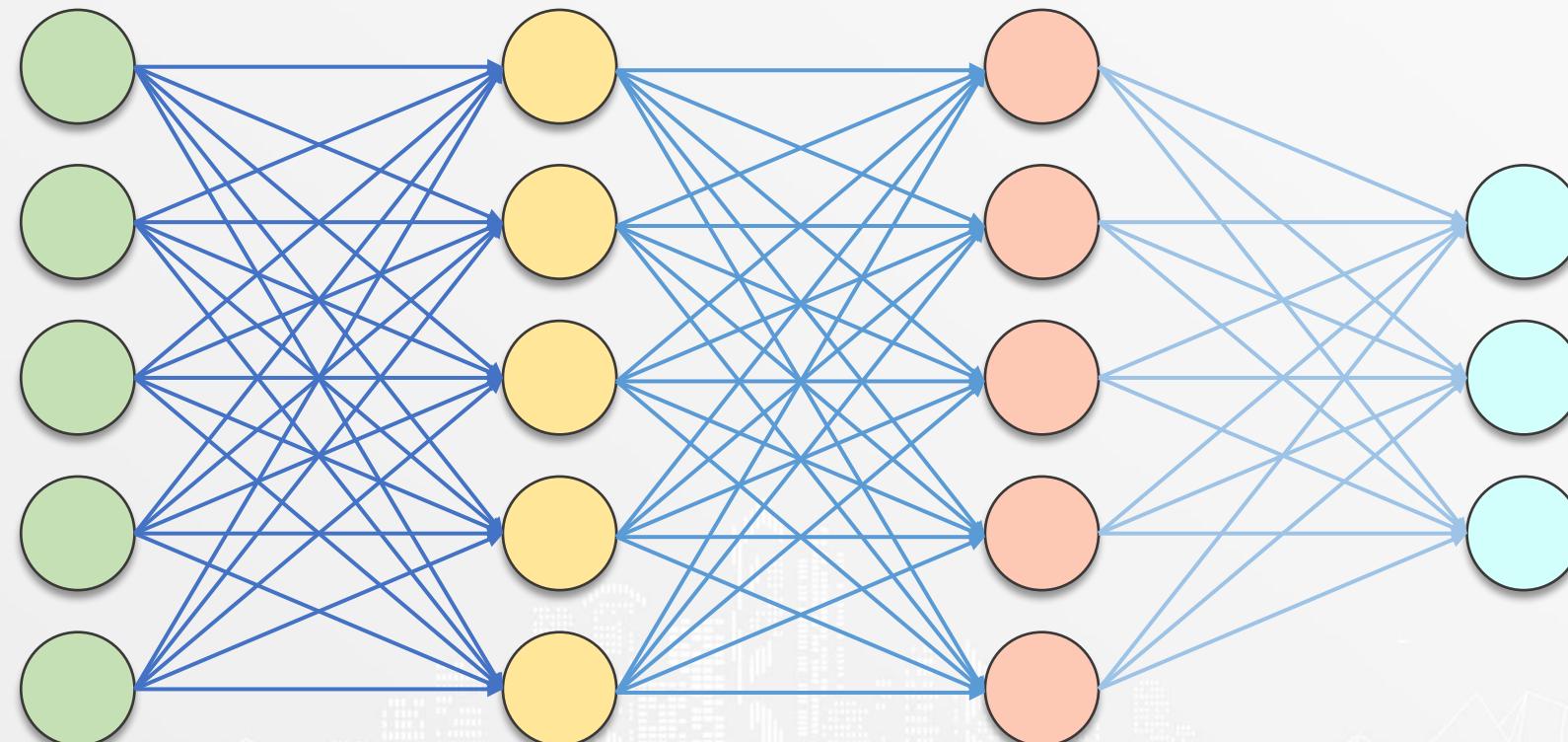
Linear Regression



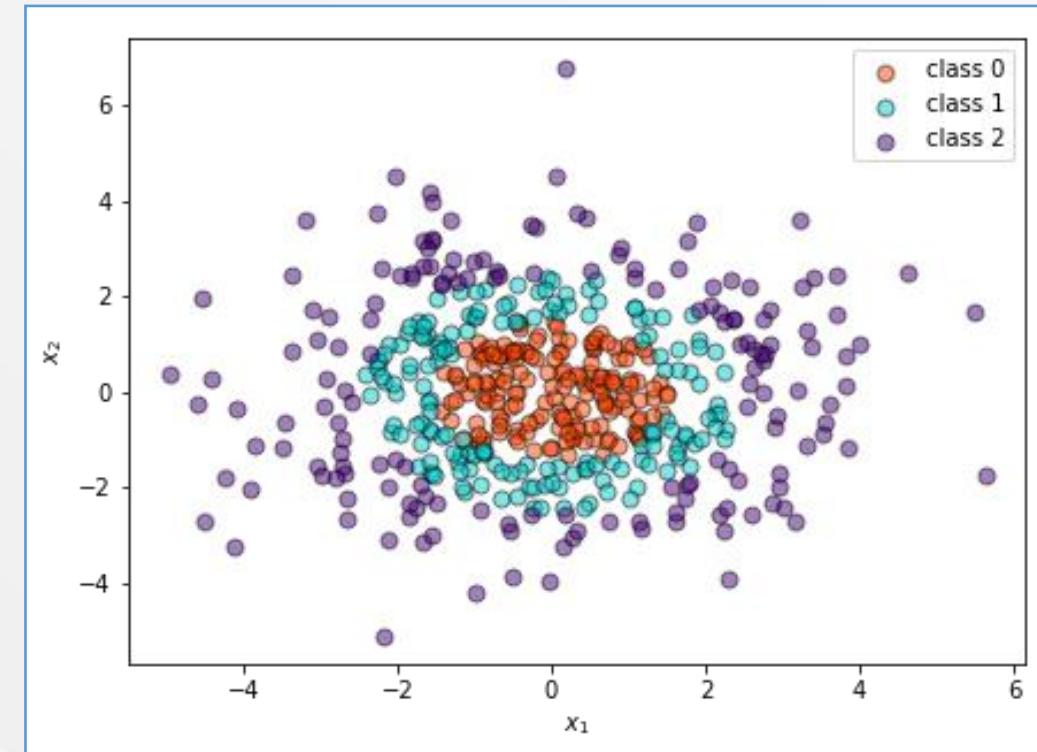
Neural Network  
(regression)



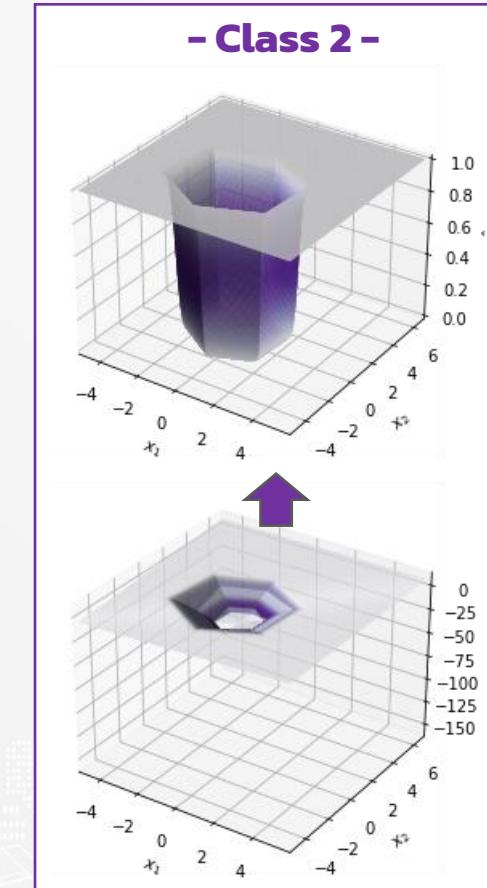
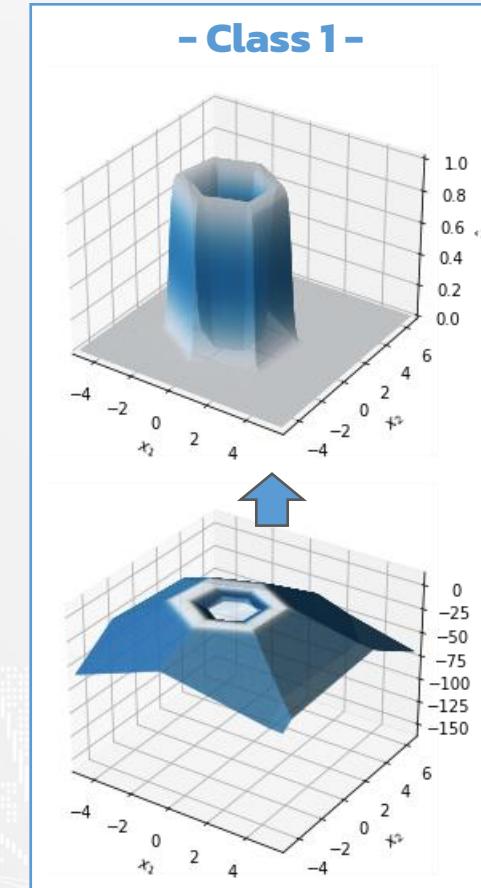
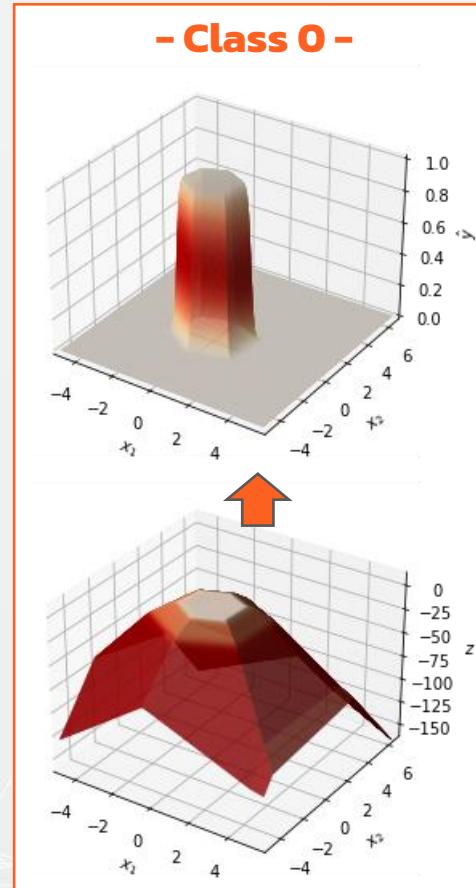
# Extension to Neural Network



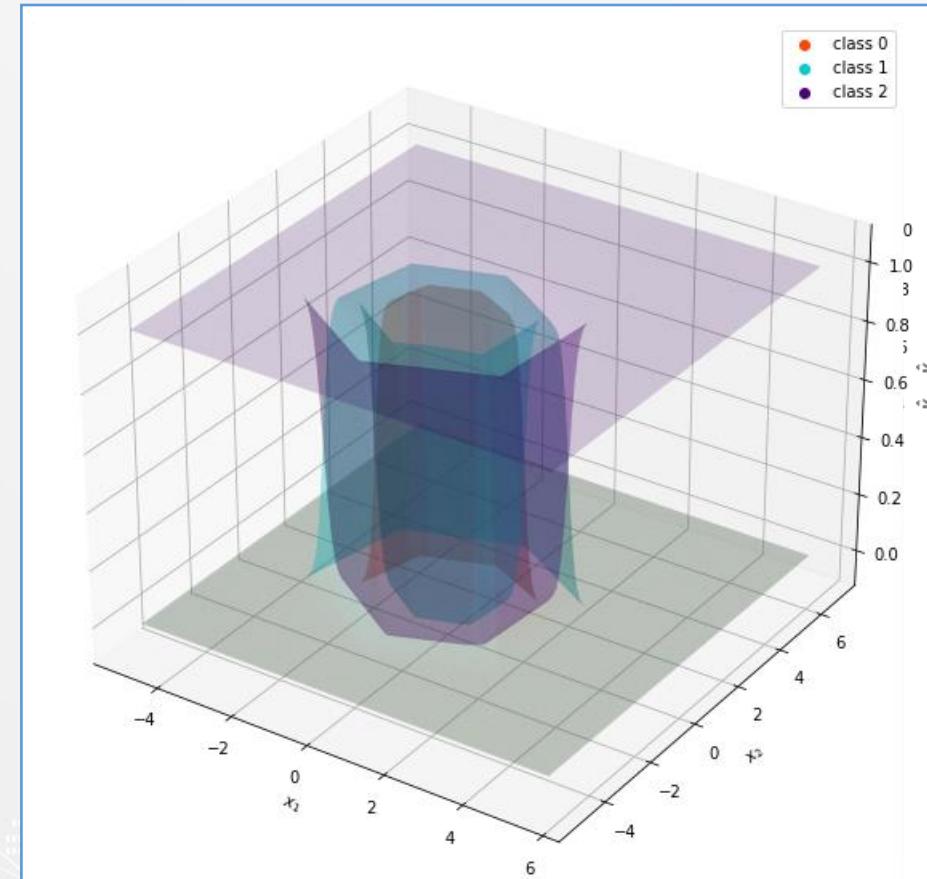
# Extension to Neural Network



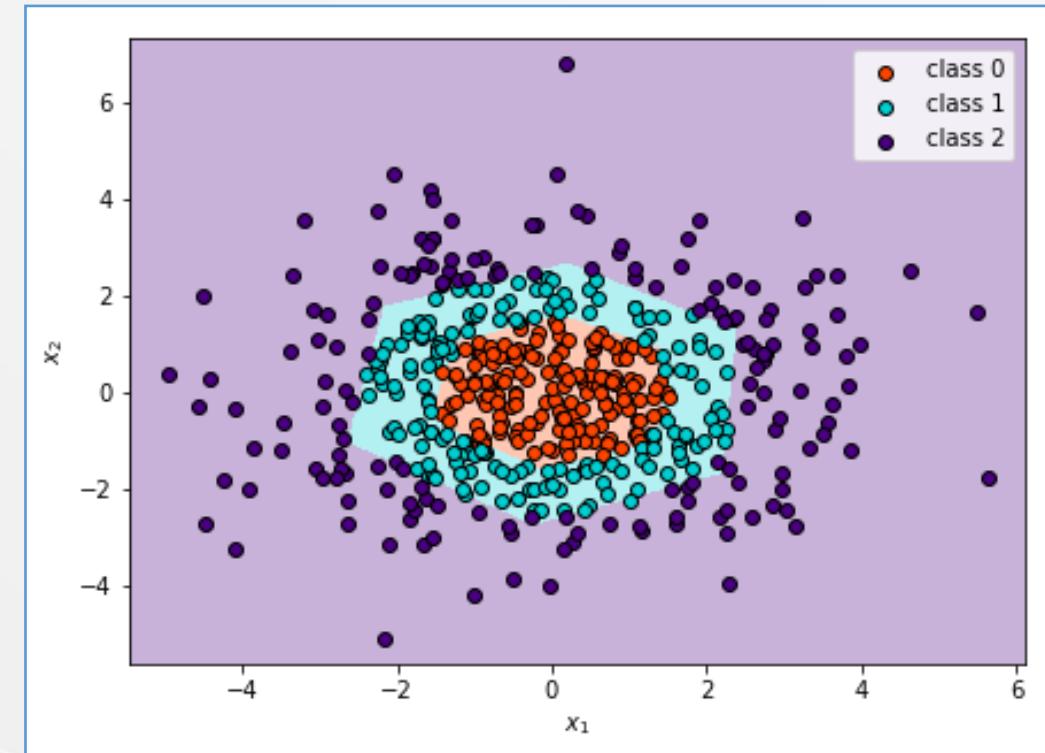
# Extension to Neural Network



# Extension to Neural Network



# Extension to Neural Network



# Introduction

**Logistic Regression  
with Multiclass**



**Why Softmax ?**



**Extension to Neural  
Network**



**Real World  
Application**



# Real World Application



**การจำแนกผู้ป่วยที่เป็นโรคเบาหวาน  
(ไม่เป็นเบาหวาน, มีโอกาสเสี่ยงสูง, เป็น  
เบาหวาน)**  
โดยจะพิจารณาจาก เพศ, อายุ, ระดับการ  
เคลื่อนไหวในชีวิตประจำวัน เป็นต้น

<https://pubmed.ncbi.nlm.nih.gov/26714019/>

# Real World Application



## การวิเคราะห์ความรุนแรงของอุบัติเหตุ บนท้องถนน

(เสียหายเฉพาะทรัพย์สิน, บาดเจ็บเล็กน้อย,  
พิการ, เสียชีวิต)

โดยพิจารณาจาก อายุของคนขับ, ประเภทของ  
ยานพาหนะ, แสงไฟบนท้องถนน, เป็นถนนที่สร้าง  
เสร็จแล้วหรือกำลังก่อสร้าง เป็นต้น

<https://www.scirp.org/journal/paperinformation.aspx?paperid=77395>

# Real World Application



## การวิเคราะห์ performance ของหุ้นใน อนาคต

(ดี, อยู่ในค่าเฉลี่ยของตลาด, ไม่ดี)

โดยพิจารณาจาก กำไรสุทธิ, กำไรต่อหุ้น, มูลค่า  
การบัญชี, สัดส่วนกำไรเกียบปีก่อนหน้า เป็นต้น

<https://www.proquest.com/openview/51d5560e3bea3125896d211c7932d7a9/1?pq-origsite=gscholar&cbl=1056382>

# Introduction

**Logistic Regression  
with Multiclass**



**Why Softmax ?**



**Extension to Neural  
Network**



**Real World  
Application**

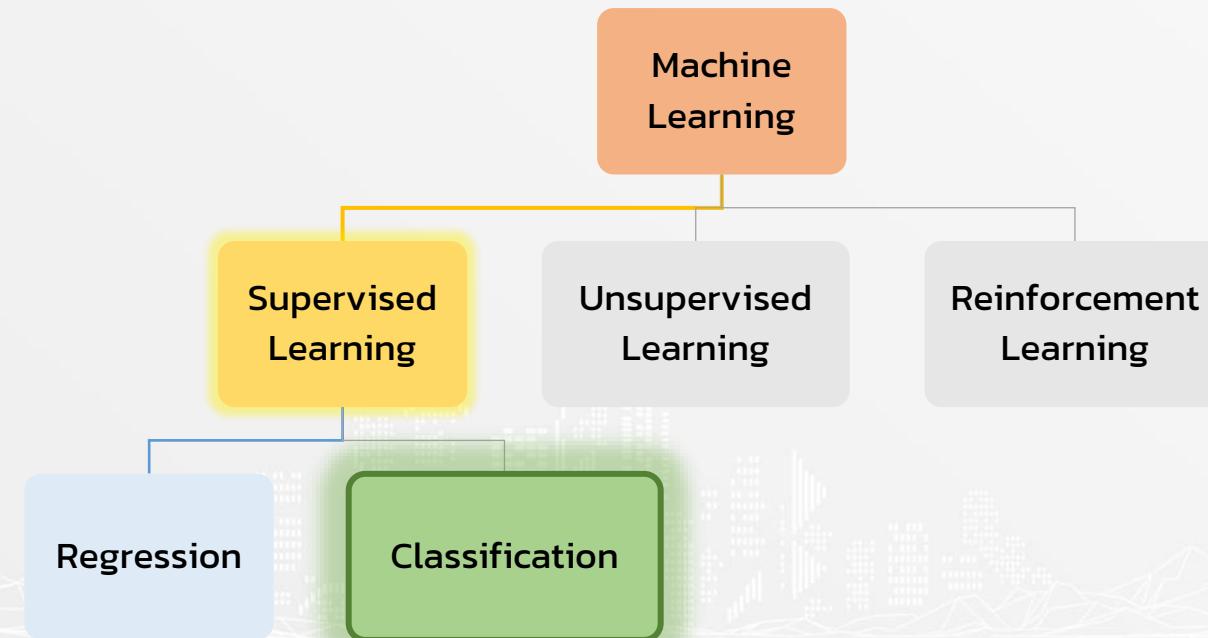


# Logistic Regression (Multi-Class Classification)



# Logistic Regression

**Logistic Regression** ជូនអង់គ្លេស algorithm ប្រភេទ **supervised learning**



# Concept of Supervised Learning

**Data   ⇒  Model   ⇒  Prediction**

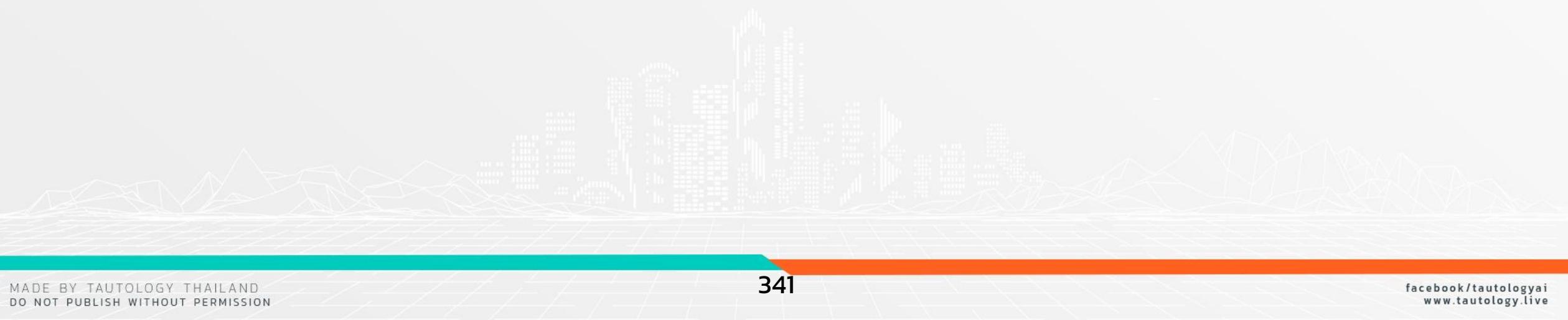
# Data

# Data

Data Stating

Data Requirement

More about  
Target



# Data Stating

The diagram illustrates a data matrix structure. At the top, five orange arrows point downwards, corresponding to the columns labeled  $x_1, x_2, x_3, \dots, x_p$ . On the left side, four yellow arrows point to the left, corresponding to the rows labeled  $x_{1,1}, x_{2,1}, x_{3,1}, \dots, x_{n,1}$ . A green arrow points downwards from the column  $x_p$  to the row  $y$ , indicating the target variable. The matrix itself is a 6x6 grid with the following structure:

$x_1$	$x_2$	$x_3$	$\dots$	$x_p$	$y$
$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$\dots$	$x_{1,p}$	$y_1$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$\dots$	$x_{2,p}$	$y_2$
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$\dots$	$x_{3,p}$	$y_3$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	$\dots$	$x_{n,p}$	$y_n$

- ◆  $n$  คือ จำนวน sample
- ◆  $p$  คือ จำนวน feature

# Data Stating

$x_1$	$x_2$	$x_3$	...	$x_p$	$y$
$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	...	$x_{1,p}$	$y_1$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	...	$x_{2,p}$	$y_2$
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	...	$x_{3,p}$	$y_3$
:	:	:	..	:	:
$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	...	$x_{n,p}$	$y_n$

- $x_{2,3}$  คือ sample ที่ 2 feature ที่ 3
- $x_{3,p}$  คือ sample ที่ 3 feature ที่  $p$
- $x_{n,p}$  คือ sample ที่  $n$  feature ที่  $p$
- $y_2$  คือ target ของ sample ที่ 2
- $y_3$  คือ target ของ sample ที่ 3
- $y_n$  คือ target ของ sample ที่  $n$

# Data Stating

## Example

- เราต้องการจำแนกประเภทของโรคเบ้าหวาน, โควิด และหัวใจ โดยพิจารณาจาก เพศ, อุณหภูมิร่างกาย, น้ำตาลในเลือด และ ความดันโลหิต

## Data

เพศ	อุณหภูมิร่างกาย (°C)	น้ำตาลในเลือด (mg/dl)	ความดันโลหิต (mmHg)	โรค
0	36.6	126	122	0
1	39.5	70	120	1
1	40.1	84	95	1
0	36.5	79	151	2

\*\*0=เบ้าหวาน, 1=โควิด, 2=หัวใจ

# Data Stating

- Feature and Target
  - เราสามารถแยก และปรับให้เป็น matrix ได้ดังนี้

$$X = \begin{bmatrix} 0 & 36.5 & 126 & 122 \\ 1 & 39.5 & 70 & 120 \\ 1 & 40.1 & 84 & 259 \\ 0 & 36.5 & 79 & 249 \end{bmatrix}$$

$$y = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 2 \end{bmatrix}$$

# Data

**Data Stating**



**Data Requirement**



**More about  
Target**



# Data Requirement

- ข้อมูลต้องอยู่ในรูปแบบของตาราง
- ข้อมูลต้องเป็น numerical

เพศ	อุณหภูมิร่างกาย (°C)	น้ำตาลในเลือด (mg/dl)	ความดันโลหิต (mmHg)	โรค
0	36.6	126	122	0
1	39.5	70	120	1
1	40.1	84	95	1
0	36.5	79	151	2

\*\*0=เบาหวาน, 1=โควิด, 2=หัวใจ

# Data Requirement

- ตัวอย่างข้อมูลที่สามารถใช้งานได้เลย และยังไม่สามารถใช้งานได้

อายุ	เพศ	โรค
42	0	0
57	1	1
58	1	1
59	0	2



อายุ	เพศ	โรค
42	female	0
57	male	1
58	male	1
59	female	2



\*\*0=เบาหวาน, 1=โควิด, 2=หัวใจ

# Data Requirement

- เราสามารถแปลงได้โดยสามารถใช้ความรู้ในส่วนของ Data Preparation

อายุ	เพศ	โรค
42	female	0
57	male	1
58	male	1
59	female	2



อายุ	female	male	โรค
42	1	0	0
57	0	1	1
58	0	1	1
59	1	0	2

\*\*0=เบาหวาน, 1=โควิด, 2=หัวใจ

# Data

**Data Stating**



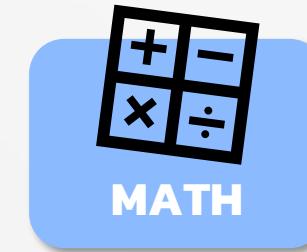
**Data Requirement**



More about  
Target



# More about Target



อายุ	female	male	โรค
42	1	0	เบาหวาน
57	0	1	โควิด
58	0	1	โควิด
59	1	0	หัวใจ



อายุ	female	male	เบาหวาน	โควิด	หัวใจ
42	1	0	1	0	0
57	0	1	0	1	0
58	0	1	0	1	0
59	1	0	0	0	1



\*\*0=เบาหวาน, 1=โควิด, 2=หัวใจ

# More about Target



อายุ	female	male	โรค
42	1	0	เบาหวาน
57	0	1	โควิด
58	0	1	โควิด
59	1	0	หัวใจ

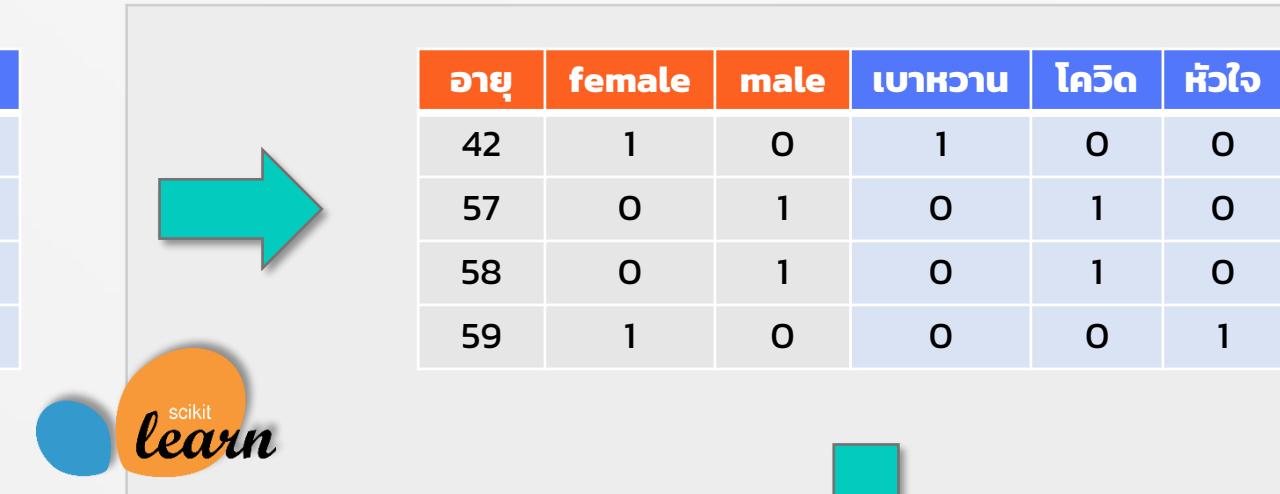
อายุ	female	male	เบาหวาน	โควิด	หัวใจ
42	1	0	1	0	0
57	0	1	0	1	0
58	0	1	0	1	0
59	1	0	0	0	1



\*\*0=เบาหวาน, 1=โควิด, 2=หัวใจ

# More about Target

อายุ	female	male	โรค
42	1	0	เบาหวาน
57	0	1	โควิด
58	0	1	โควิด
59	1	0	หัวใจ



Prediction

โรค
โควิด
เบาหวาน
โควิด
หัวใจ

Prediction

เบาหวาน	โควิด	หัวใจ
0	1	0
1	0	0
0	1	0
0	0	1

# Data

**Data Stating**



**Data Requirement**



**More about  
Target**



# Logistic Regression (Multi-Class Classification)



# Model

# Model

Assumption

Real Face of the  
Model

Cost Function and  
Cost Landscape

How to Create  
Model (Math)

How to Create  
Model (Code)

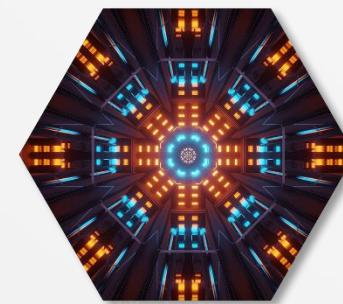
# Assumption

1. Linear Relationship
2. Normality of Residuals
3. Homoscedasticity
4. No Missing Features
5. No Multicollinearity

# Assumption



For more information



Model Improvement  
in DL101

# Model

**Assumption**



Real Face of the Model

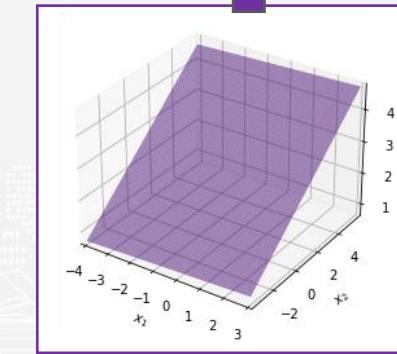
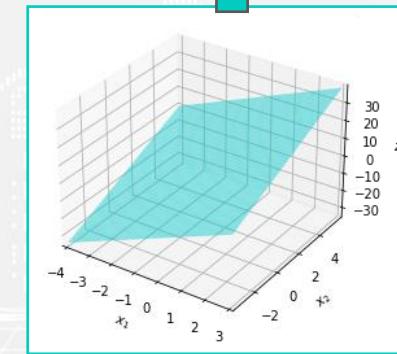
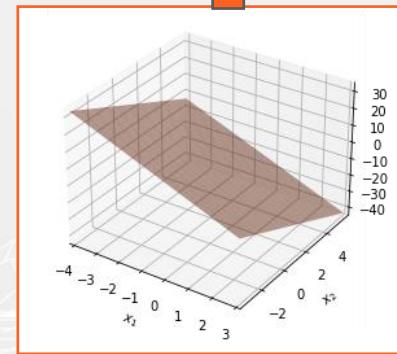
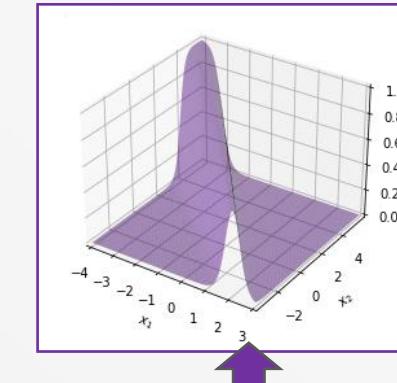
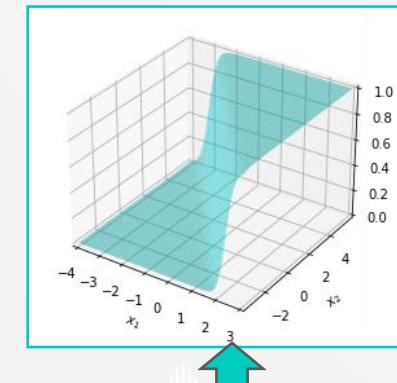
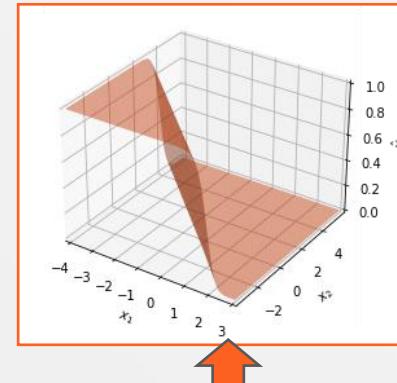
Cost Function and Cost Landscape

How to Create Model (Math)

How to Create Model (Code)

# Real Face of the Model

**Logistic regression** คือ โมเดลคณิตศาสตร์ที่ส่งค่าสมการเชิงเส้น (linear regression) ไปในช่วง  $(0,1)$  เพื่อจำแนกประเภทข้อมูล



# Real Face of the Model

**Logistic regression** คือ โมเดลคณิตศาสตร์ที่จำแนกประเภทของข้อมูล ด้วยการสร้างสมการเชิงเส้น (linear regression)

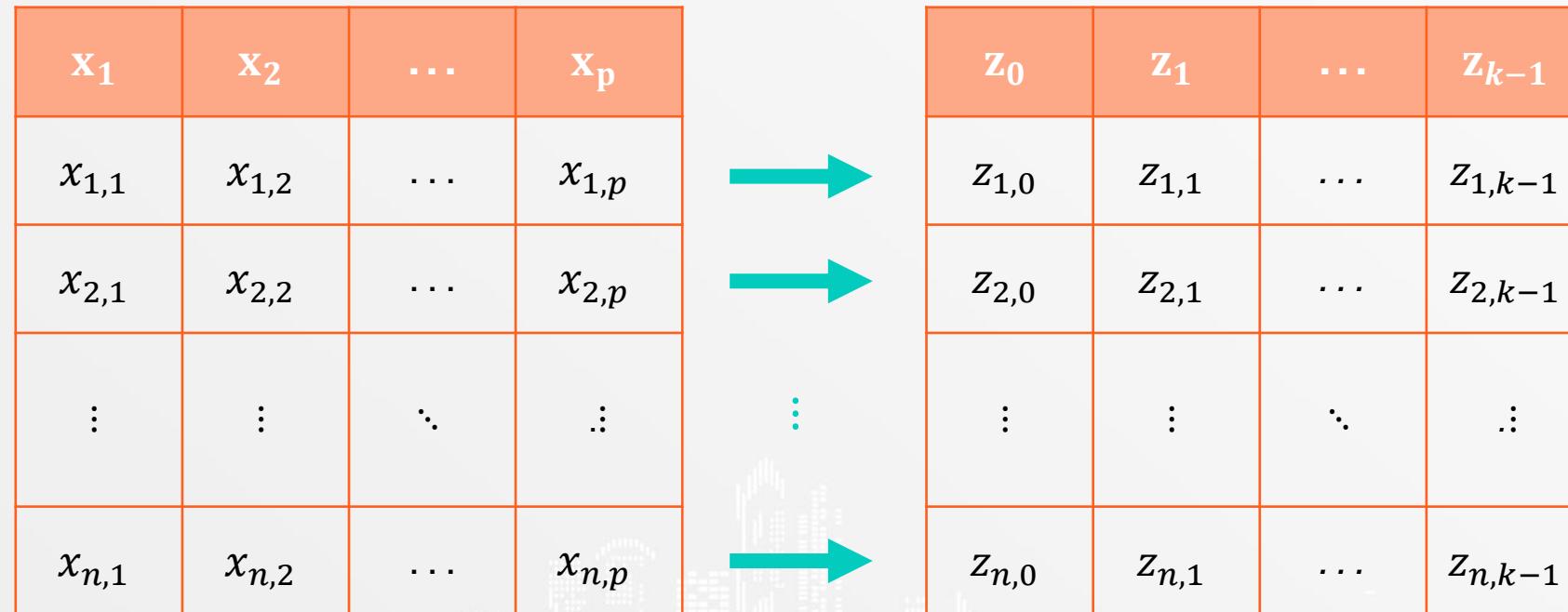
$$z_m = w_{0,m} + w_{1,m}x_1 + w_{2,m}x_2 + \cdots + w_{p,m}x_p$$

$$\hat{y}_m = \frac{e^{z_m}}{\sum_{c=0}^{k-1} e^{z_c}}$$

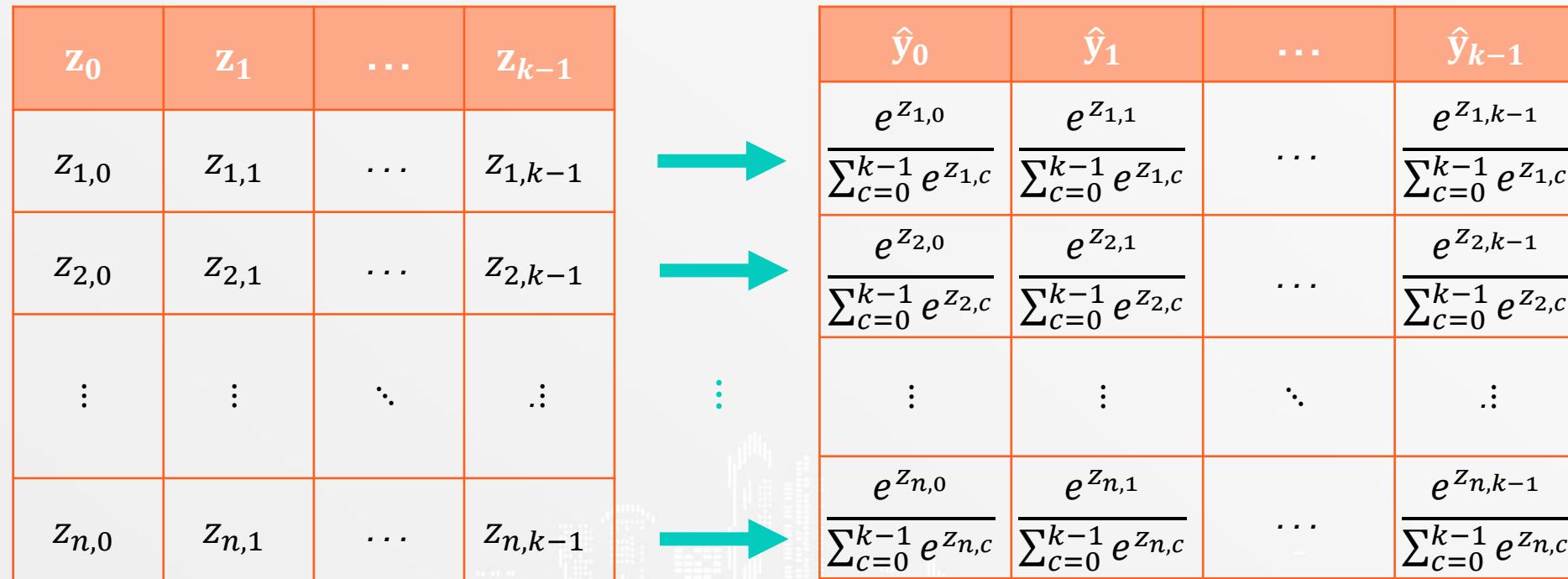
โดยที่

- ◆  $\hat{y}_m$  คือ ความน่าจะเป็นที่จะเป็น class ที่  $m$  (predicted target)
- ◆  $z_m$  คือ สมการเชิงเส้น (linear regression) ของ class ที่  $m$
- ◆  $x_1, x_2, x_3, \dots, x_p$  คือ ตัวแปรต้น (feature)
- ◆  $w_{0,m}, w_{1,m}, w_{2,m}, \dots, w_{p,m}$  คือ สัมประสิทธิ์ที่ส่งจาก feature ไป class ที่  $m$  (coefficient)
- ◆  $m \in 0, 1, 2, \dots, k - 1$

# Real Face of the Model



# Real Face of the Model



# Real Face of the Model

$$z_0 = w_{0,0} + w_{1,0}x_1 + w_{2,0}x_2 + \cdots + w_{p,0}x_p$$



$$z_{1,0} = w_{0,0} + w_{1,0}x_{1,1} + w_{2,0}x_{1,2} + \cdots + w_{p,0}x_{1,p}$$

$$z_{2,0} = w_{0,0} + w_{1,0}x_{2,1} + w_{2,0}x_{2,2} + \cdots + w_{p,0}x_{2,p}$$

⋮

$$z_{n,0} = w_{0,0} + w_{1,0}x_{n,1} + w_{2,0}x_{n,2} + \cdots + w_{p,0}x_{n,p}$$

# Real Face of the Model

$$z_1 = w_{0,1} + w_{1,1}x_1 + w_{2,1}x_2 + \cdots + w_{p,1}x_p$$



$$z_{1,1} = w_{0,1} + w_{1,1}x_{1,1} + w_{2,1}x_{1,2} + \cdots + w_{p,1}x_{1,p}$$

$$z_{2,1} = w_{0,1} + w_{1,1}x_{2,1} + w_{2,1}x_{2,2} + \cdots + w_{p,1}x_{2,p}$$

⋮

$$z_{n,1} = w_{0,1} + w_{1,1}x_{n,1} + w_{2,1}x_{n,2} + \cdots + w_{p,1}x_{n,p}$$

# Real Face of the Model

$$z_{k-1} = w_{0,k-1} + w_{1,k-1}x_1 + w_{2,k-1}x_2 + \cdots + w_{p,k-1}x_p$$



$$z_{1,k-1} = w_{0,k-1} + w_{1,k-1}x_{1,1} + w_{2,k-1}x_{1,2} + \cdots + w_{p,k-1}x_{1,p}$$

$$z_{2,k-1} = w_{0,k-1} + w_{1,k-1}x_{2,1} + w_{2,k-1}x_{2,2} + \cdots + w_{p,k-1}x_{2,p}$$

⋮

$$z_{n,k-1} = w_{0,k-1} + w_{1,k-1}x_{n,1} + w_{2,k-1}x_{n,2} + \cdots + w_{p,k-1}x_{n,p}$$

# Real Face of the Model



“ เราต้องการหา

$$w_{0,0}, w_{1,0}, w_{2,0}, \dots, w_{p,0}$$

$$w_{0,1}, w_{1,1}, w_{2,1}, \dots, w_{p,1}$$

⋮

$$w_{0,k-1}, w_{1,k-1}, w_{2,k-1}, \dots, w_{p,k-1}$$

ที่ทำให้ cost function ต่ำที่สุด ”

# Model

**Assumption**



**Real Face of the Model**



**Cost Function and Cost Landscape**

**How to Create Model (Math)**



**How to Create Model (Code)**



# Cost Function and Cost Landscape

**Cost function** กี่เราจะใช้ในการสร้าง model คือ

$$-\frac{1}{n} \sum_{i=1}^n \sum_{c=0}^{k-1} y_{i,c} \log(\hat{y}_{i,c})$$

โดยสูตรข้างต้นมีชื่อว่า **Cross Entropy**

# Cost Function and Cost Landscape



For more information



Cross Entropy

# Model

**Assumption**



**Real Face of the Model**



**Cost Function and Cost Landscape**



**How to Create Model (Math)**

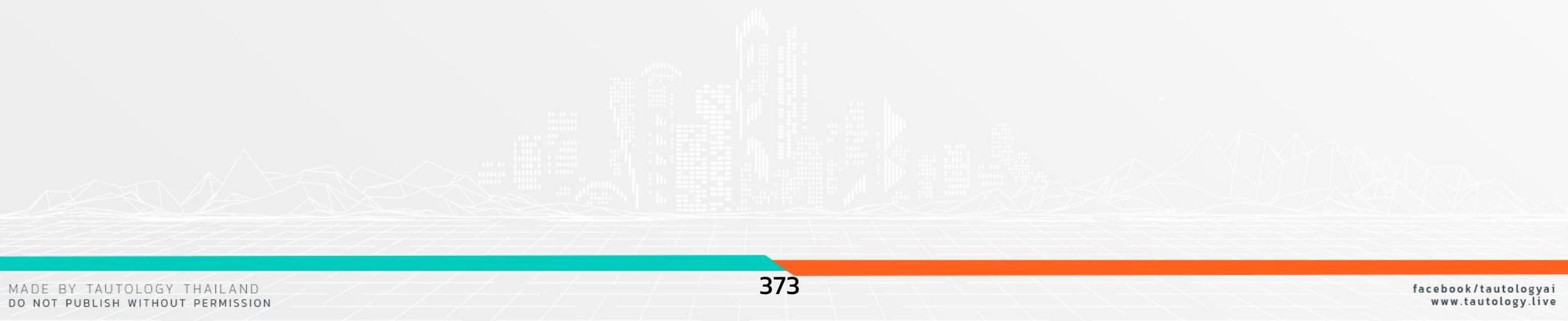


**How to Create Model (Code)**



# How to Create Model (Math)

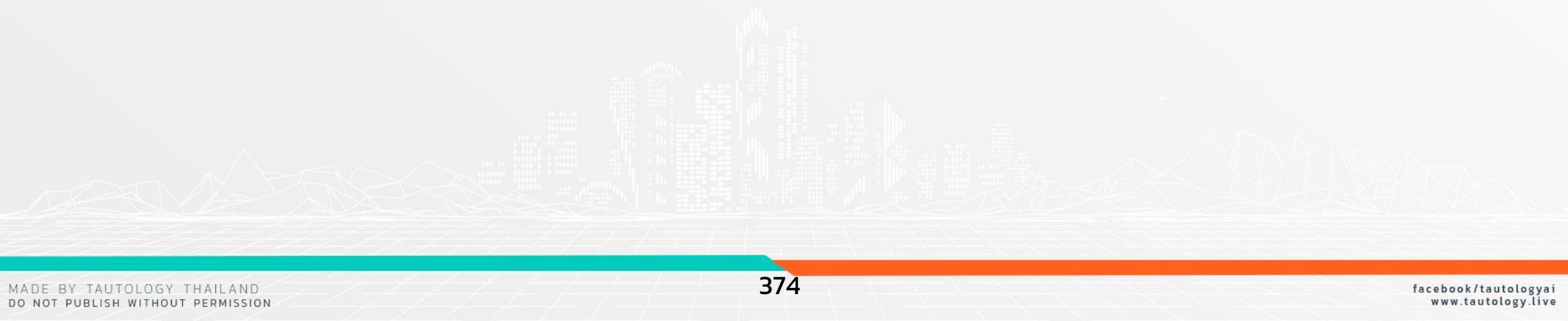
- Gradient Descent (Logistic Regression)
- Calculation Example



# Gradient Descent (Logistic Regression)

เครื่องมือที่เราจะใช้ในการหาค่าตอบคือ

**“ Gradient Descent ”**



# Gradient Descent (Logistic Regression)

## Equation of Gradient Descent

$$w_d = w_d - \alpha \frac{\partial Cost}{\partial w_d}$$

โดย ◆  $\alpha$  คือ ค่าที่ใช้ควบคุม step size ของ  $w_d$  ( $> 0$ )

# Gradient Descent (Logistic Regression)

เมื่อพิจารณา  $w$  ที่ส่งไปยัง  $z_0$  จะได้ว่า

$$w_{0,0} = w_{0,0} - \alpha \frac{\partial Cost}{\partial w_{0,0}}$$

$$w_{1,0} = w_{1,0} - \alpha \frac{\partial Cost}{\partial w_{1,0}}$$

⋮

$$w_{p,0} = w_{p,0} - \alpha \frac{\partial Cost}{\partial w_{p,0}}$$

# Gradient Descent (Logistic Regression)

เมื่อพิจารณา  $w$  ที่ส่งไปยัง  $\mathbf{z}_1$  จะได้ว่า

$$w_{0,1} = w_{0,1} - \alpha \frac{\partial Cost}{\partial w_{0,1}}$$

$$w_{1,1} = w_{1,1} - \alpha \frac{\partial Cost}{\partial w_{1,1}}$$

⋮

$$w_{p,1} = w_{p,1} - \alpha \frac{\partial Cost}{\partial w_{p,1}}$$

# Gradient Descent (Logistic Regression)

เมื่อพิจารณา  $w$  ที่ส่งไปยัง  $\mathbf{z}_{k-1}$  จะได้ว่า

$$w_{0,k-1} = w_{0,k-1} - \alpha \frac{\partial \text{Cost}}{\partial w_{0,k-1}}$$

$$w_{1,k-1} = w_{1,k-1} - \alpha \frac{\partial \text{Cost}}{\partial w_{1,k-1}}$$

⋮

$$w_{p,k-1} = w_{p,k-1} - \alpha \frac{\partial \text{Cost}}{\partial w_{p,k-1}}$$

# Gradient Descent (Logistic Regression)

เมื่อพิจารณาในทุกมิติพร้อมกัน จะได้ว่า

$$\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,k-1} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{p,0} & w_{p,1} & \cdots & w_{p,k-1} \end{bmatrix} = \begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,k-1} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{p,0} & w_{p,1} & \cdots & w_{p,k-1} \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial \text{Cost}}{\partial w_{0,0}} & \frac{\partial \text{Cost}}{\partial w_{0,1}} & \cdots & \frac{\partial \text{Cost}}{\partial w_{0,k-1}} \\ \frac{\partial \text{Cost}}{\partial w_{1,0}} & \frac{\partial \text{Cost}}{\partial w_{1,1}} & \cdots & \frac{\partial \text{Cost}}{\partial w_{1,k-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \text{Cost}}{\partial w_{p,0}} & \frac{\partial \text{Cost}}{\partial w_{p,1}} & \cdots & \frac{\partial \text{Cost}}{\partial w_{p,k-1}} \end{bmatrix}$$

# Gradient Descent (Logistic Regression)

## Equation of Gradient Descent

$$W = W - \alpha \nabla Cost$$

โดย ◆  $\alpha$  คือ ค่าที่ใช้ควบคุม step size ของ  $W$

# Gradient Descent (Logistic Regression)

## Equation of Gradient Descent for Logistic Regression

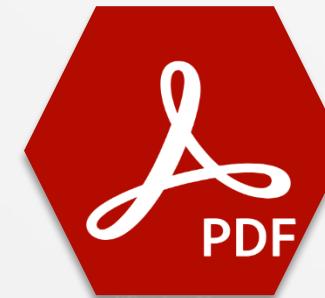
$$W = W + \frac{\alpha}{n} X_b^T (Y - \hat{Y})$$

โดย ◆  $\alpha$  คือ ค่าที่ใช้ควบคุม step size ของ  $W$

# Gradient Descent (Logistic Regression)



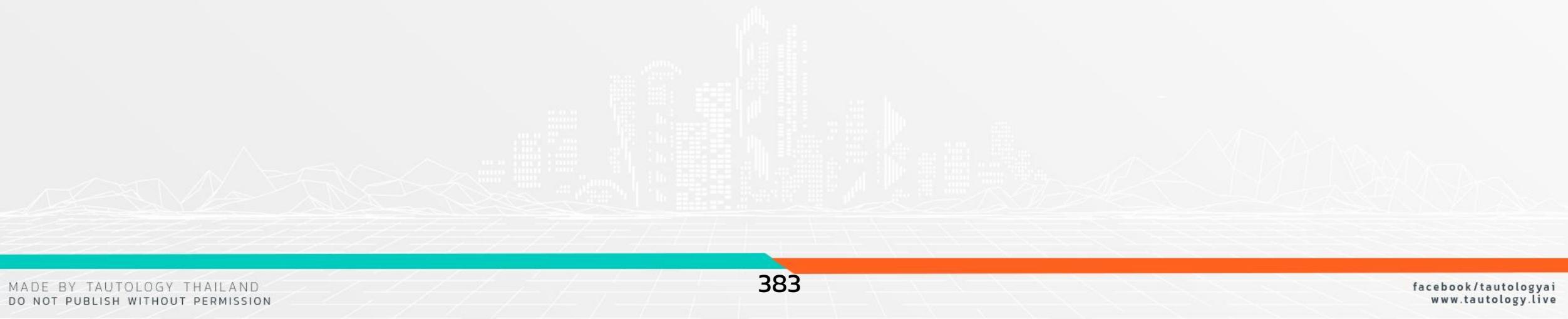
Derivation of Gradient Descent  
for Logistic Regression



Open File  
**Derive\_GD\_LoR.pdf**

# How to Create Model (Math)

- Gradient Descent (Logistic Regression)**
- Calculation Example



# Calculation Example

ตัวอย่างการคำนวณ  $W$  สำหรับ logistic regression  
ด้วย gradient descent

$x_1$	$x_2$	$y$
0	1	1
1	0	2
-1	0	3
0	-1	4

ตารางแสดง dataset

# Calculation Example

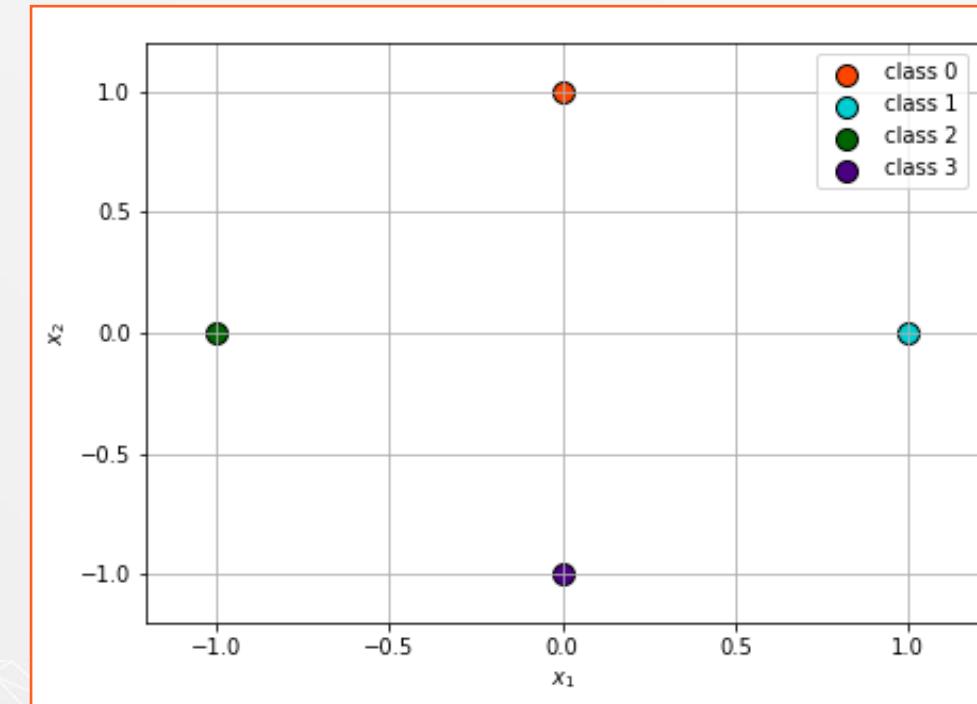
ตัวอย่างการคำนวณ  $W$  สำหรับ logistic regression  
ด้วย gradient descent

$x_1$	$x_2$	1	2	3	4
0	1	1	0	0	0
1	0	0	1	0	0
-1	0	0	0	1	0
0	-1	0	0	0	1

ตารางแสดง dataset

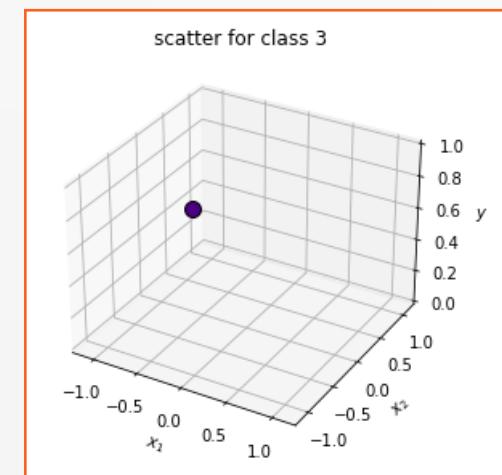
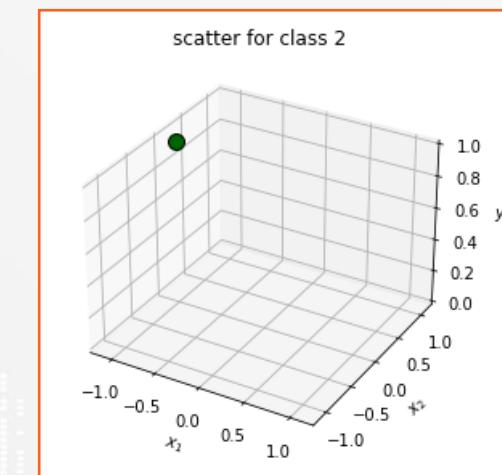
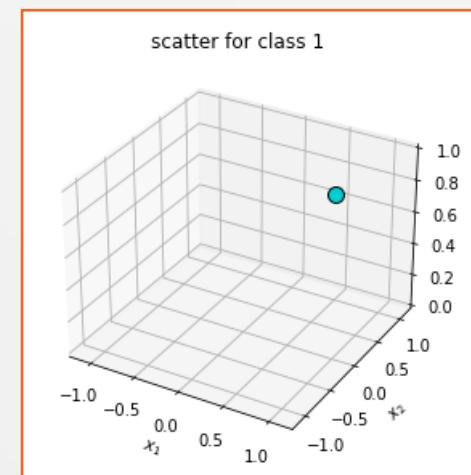
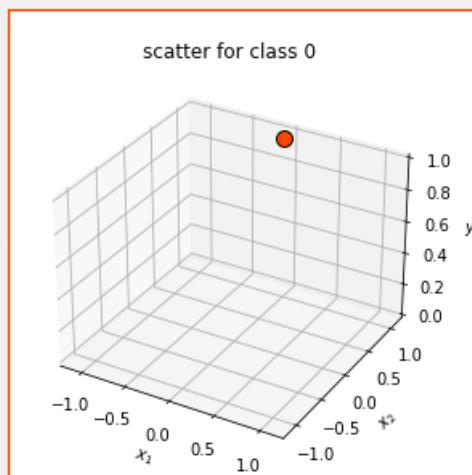
# Calculation Example

- จากข้อมูลใน dataset เราสามารถหาดีกราฟ 2 มิติ ได้ดังนี้



# Calculation Example

- จากข้อมูลใน dataset เราสามารถหาดالة 3 มิติ ได้ดังนี้



# Calculation Example

## ขั้นตอนในการคำนวณ gradient descent

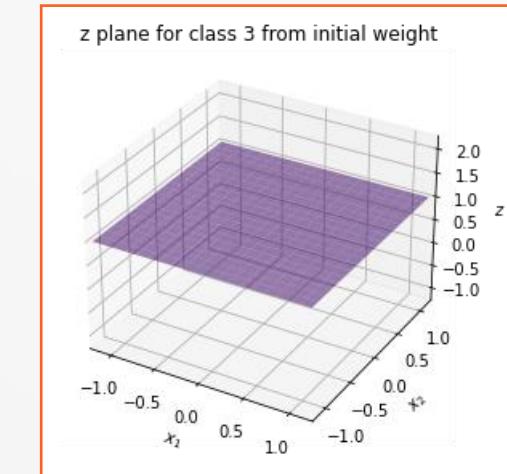
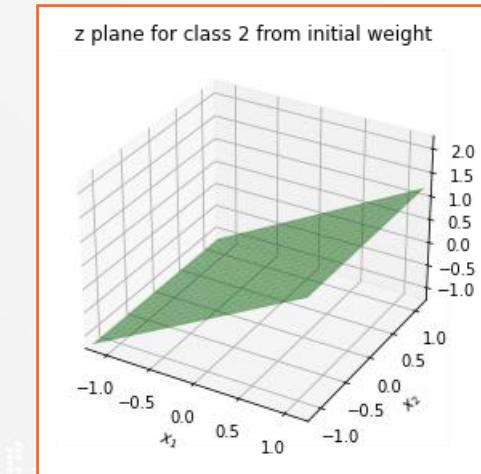
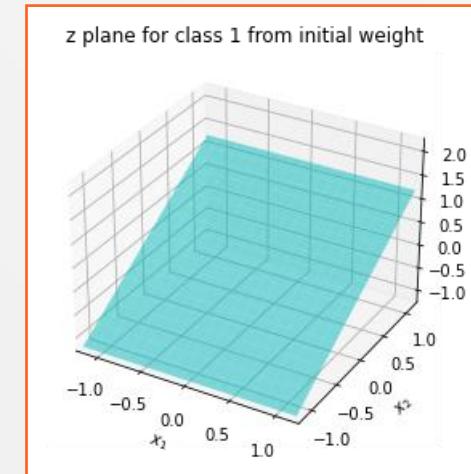
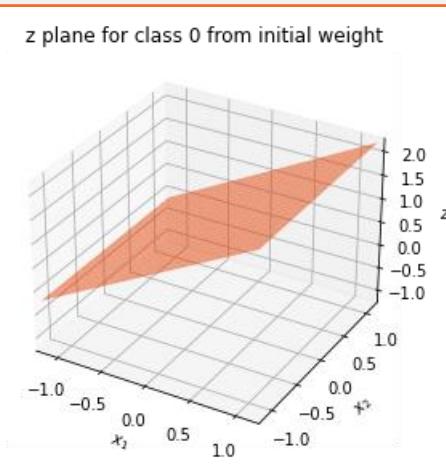
1. สุ่ม  $W$  เริ่มต้น
2. กำหนดจำนวนรอบที่จะ update  $W$  (epoch)
3. กำหนดค่า  $\alpha$
4. For loop เพื่อ update  $W$ 
  - a) คำนวณ  $Z$
  - b) คำนวณ  $\hat{Y}$
  - c) คำนวณ  $W$

# Calculation Example

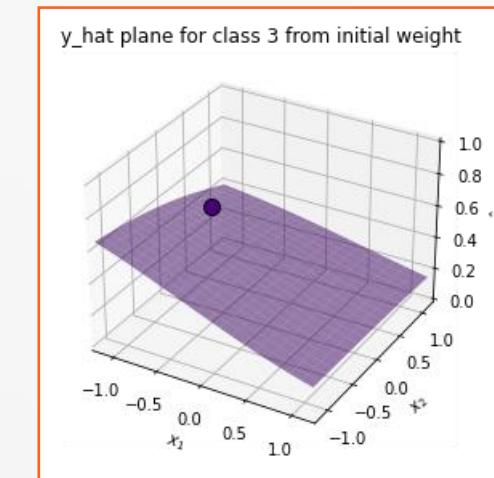
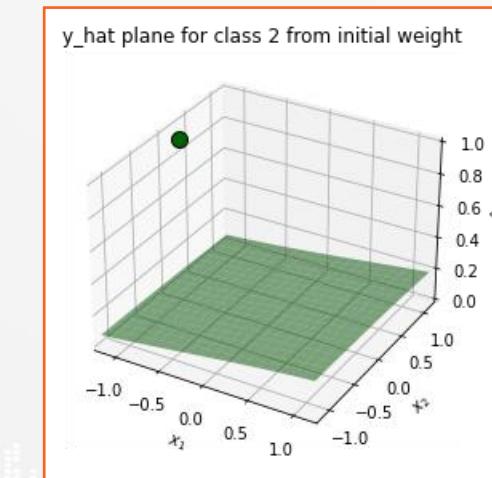
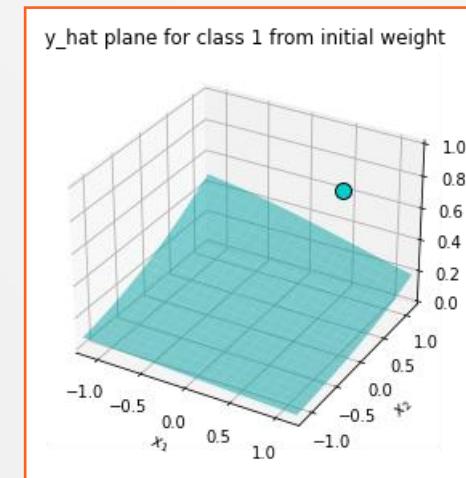
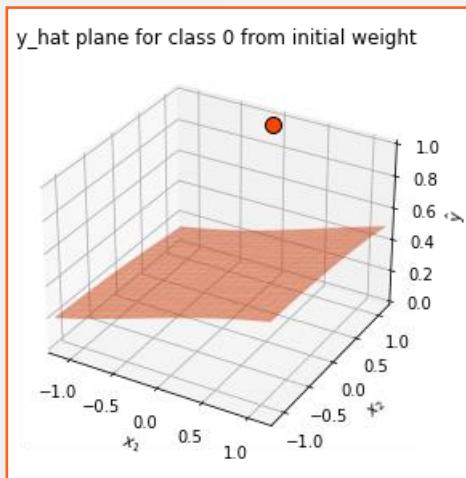
## 1. สุ่ม $W$ เริ่มต้น

$$\text{สมมติให้สุ่ม } W = \begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,2} & w_{0,3} \\ w_{1,0} & w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,0} & w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

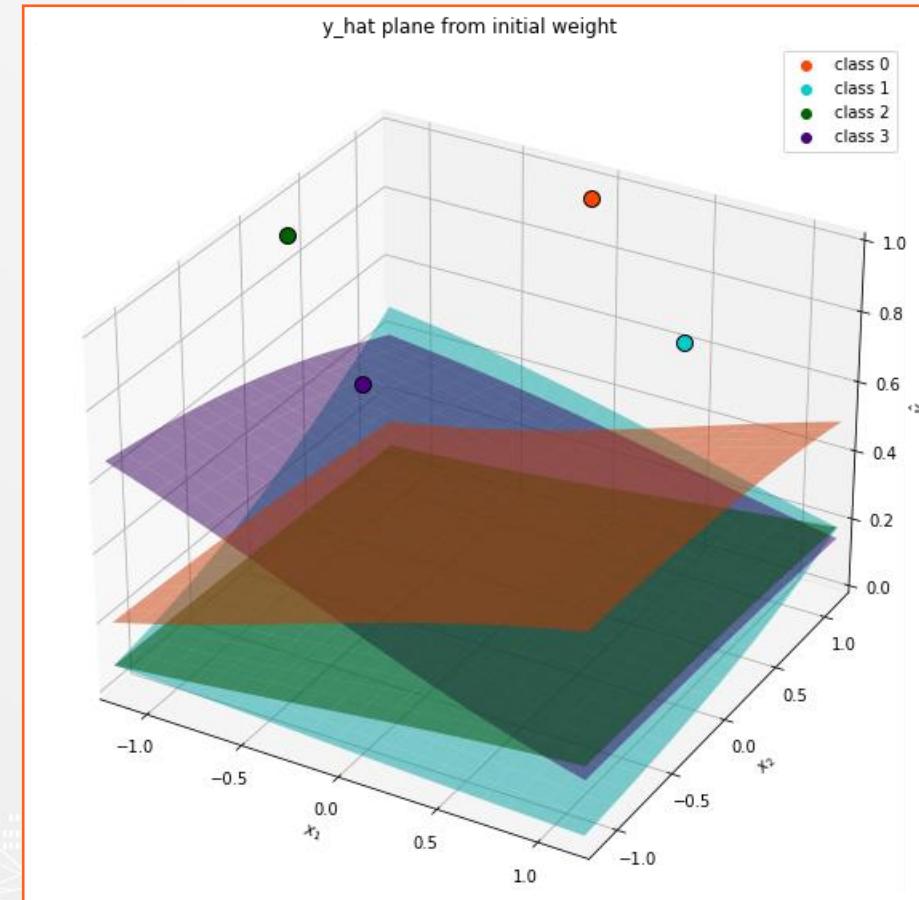
# Calculation Example



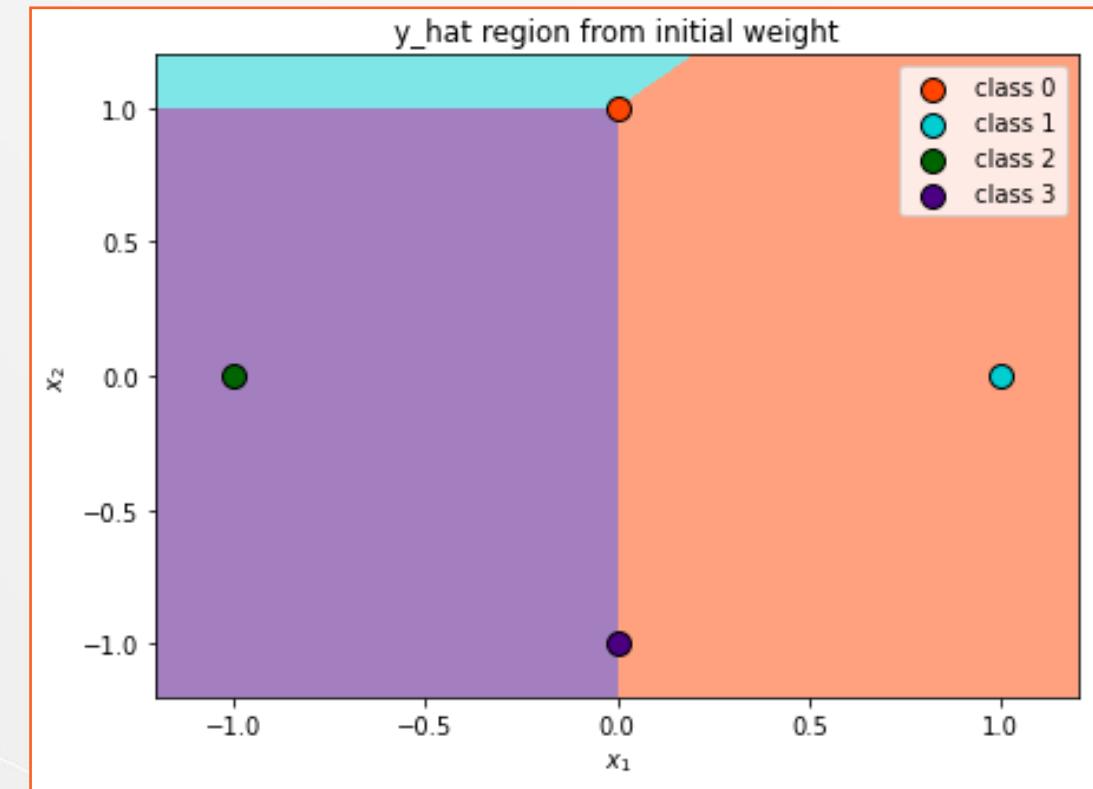
# Calculation Example



# Calculation Example



# Calculation Example



# Calculation Example

2. กำหนดจำนวนรอบที่จะ update  $W$  (epoch)

กำหนดให้ epoch = 3

# Calculation Example

## 3. กำหนดค่า $\alpha$

กำหนดให้  $\alpha = 2.5$

# Calculation Example

## 4. Update $W$

$$W = W + \frac{\alpha}{n} X_b^T (Y - \hat{Y})$$

โดย ◆  $\alpha$  คือ ค่าที่ใช้ควบคุม step size ของ  $W$

# Calculation Example

จากข้อมูลใน dataset เราสามารถเขียน  $X, Y$  และ  $X_b$  ได้ดังต่อไปนี้

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad Y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{และ} \quad X_b = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $Z$

epoch = 1

$$Z = X_b W = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 2 & 0 & 1 & 1 \\ 0 & 0 & -1 & 1 \\ 1 & -1 & 0 & 1 \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $\hat{Y}$

epoch = 1

$$\hat{Y} = \begin{bmatrix} \frac{e^{z_{1,0}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,1}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,2}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,3}}}{\sum_{c=0}^3 e^{z_{1,c}}} \\ \frac{e^{z_{2,0}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,1}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,2}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,3}}}{\sum_{c=0}^3 e^{z_{2,c}}} \\ \frac{e^{z_{3,0}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,1}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,2}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,3}}}{\sum_{c=0}^3 e^{z_{3,c}}} \\ \frac{e^{z_{4,0}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,1}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,2}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,3}}}{\sum_{c=0}^3 e^{z_{4,c}}} \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $\hat{Y}$

epoch = 1

$$\hat{Y} = \begin{bmatrix} \frac{e^{z_{1,0}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,1}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,2}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,3}}}{\sum_{c=0}^3 e^{z_{1,c}}} \\ \frac{e^{z_{2,0}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,1}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,2}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,3}}}{\sum_{c=0}^3 e^{z_{2,c}}} \\ \frac{e^{z_{3,0}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,1}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,2}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,3}}}{\sum_{c=0}^3 e^{z_{3,c}}} \\ \frac{e^{z_{4,0}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,1}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,2}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,3}}}{\sum_{c=0}^3 e^{z_{4,c}}} \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $\hat{Y}$

epoch = 1

$$\hat{Y} = \begin{bmatrix} \frac{e^{z_{1,0}}}{e^1 + e^1 + e^0 + e^1} & \frac{e^{z_{1,1}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,2}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,3}}}{\sum_{c=0}^3 e^{z_{1,c}}} \\ \frac{e^{z_{2,0}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,1}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,2}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,3}}}{\sum_{c=0}^3 e^{z_{2,c}}} \\ \frac{e^{z_{3,0}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,1}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,2}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,3}}}{\sum_{c=0}^3 e^{z_{3,c}}} \\ \frac{e^{z_{4,0}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,1}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,2}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,3}}}{\sum_{c=0}^3 e^{z_{4,c}}} \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $\hat{Y}$

epoch = 1

$$\hat{Y} = \begin{bmatrix} 0.3 & 0.3 & 0.11 & 0.3 \\ 0.53 & 0.07 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.07 & 0.53 \\ 0.4 & 0.05 & 0.15 & 0.4 \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $W$

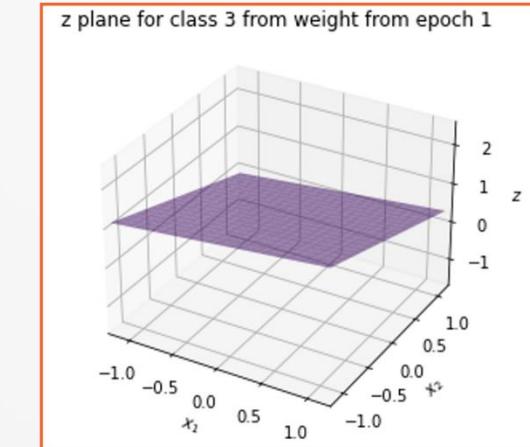
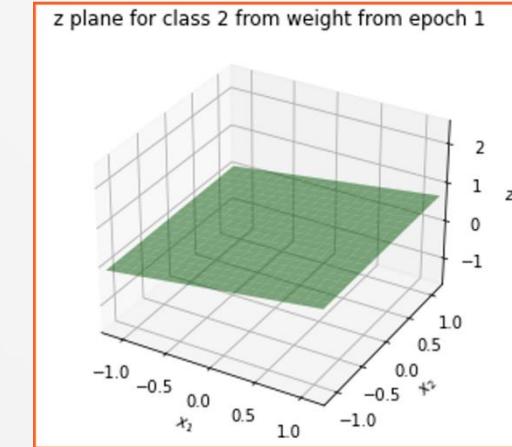
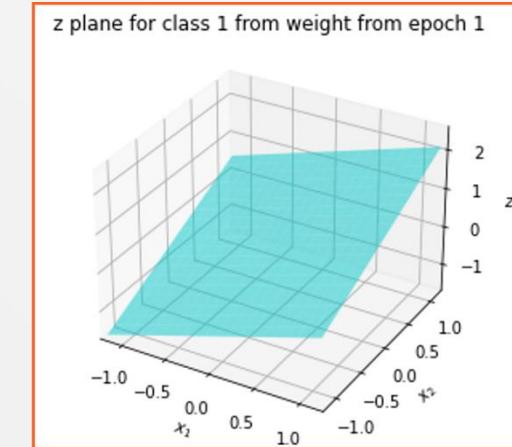
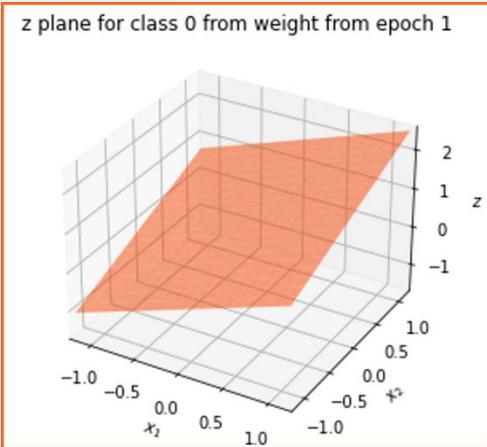
epoch = 1

$$W = W + \frac{\alpha}{n} X_b^T (Y - \hat{Y})$$

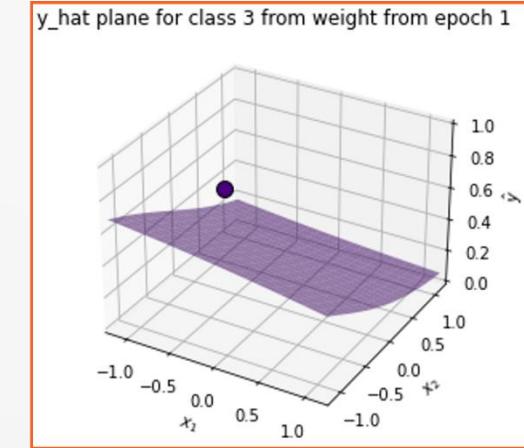
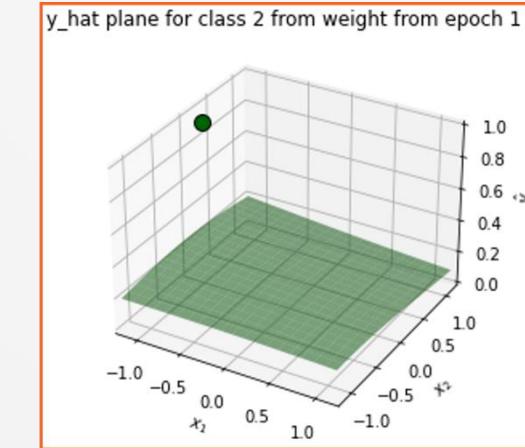
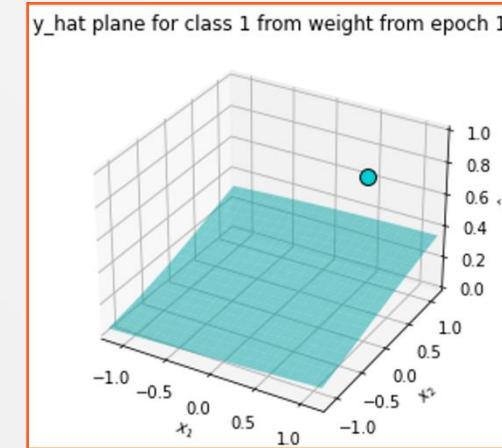
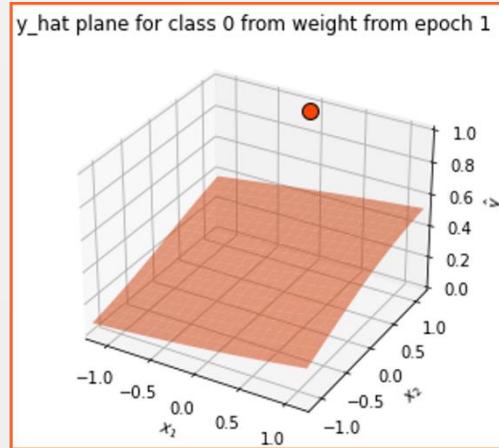
$$= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \frac{2.5}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \left( \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.3 & 0.3 & 0.11 & 0.3 \\ 0.53 & 0.07 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.07 & 0.53 \\ 0.4 & 0.05 & 0.15 & 0.4 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 0.73 & 0.24 & 0.3 & 0.73 \\ 0.79 & 0.7 & 0.3 & 0.21 \\ 0.69 & 0.85 & 0.23 & -0.56 \end{bmatrix}$$

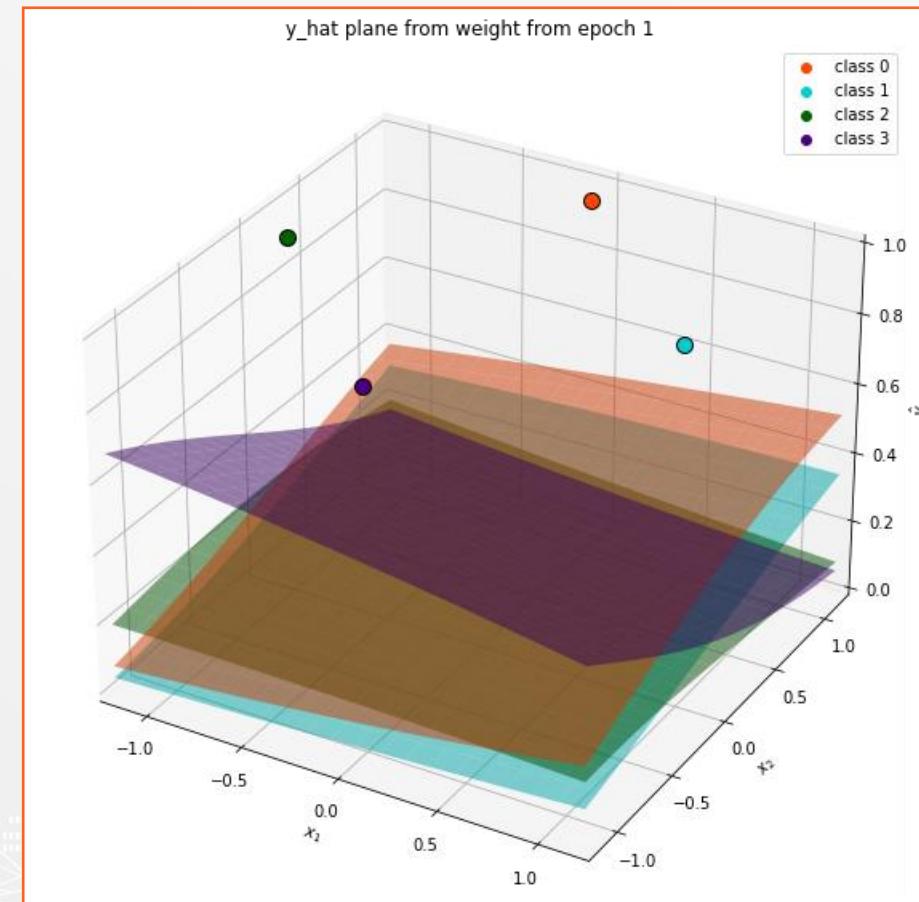
# Calculation Example



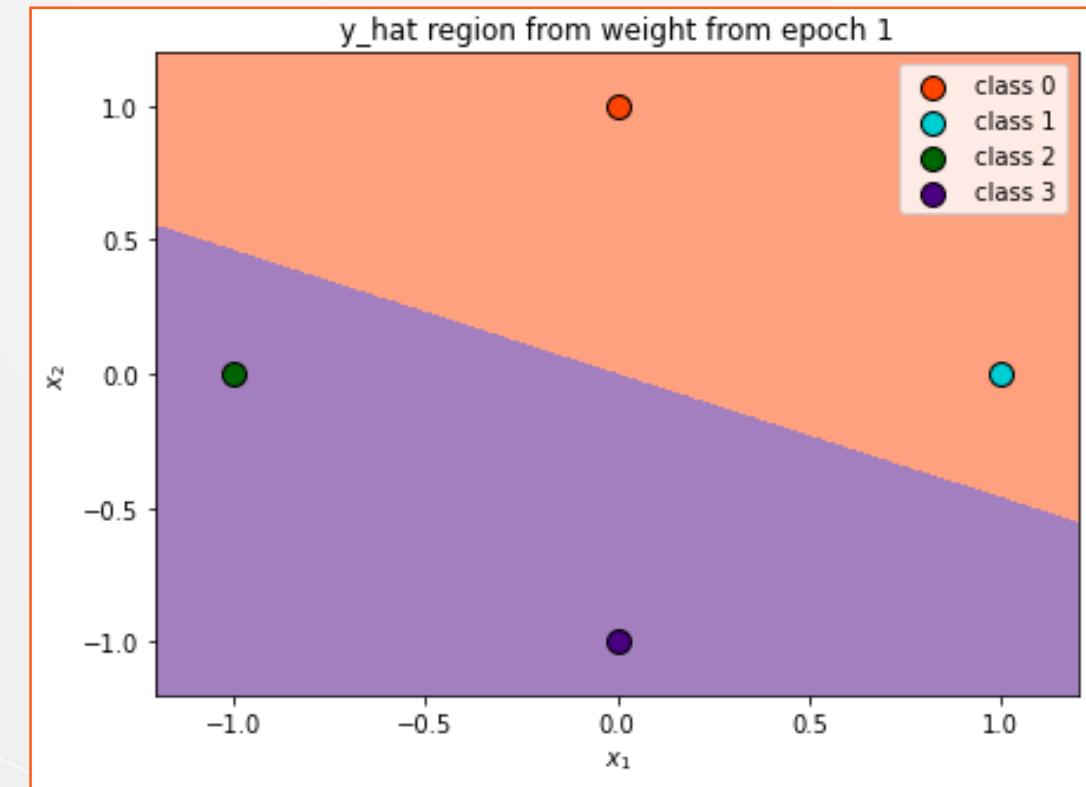
# Calculation Example



# Calculation Example



# Calculation Example



# Calculation Example

## 4. Update $W$ : คำนวณ $Z$

epoch = 2

$$\begin{aligned} Z = X_b W &= \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0.73 & 0.24 & 0.3 & 0.73 \\ 0.79 & 0.7 & 0.3 & 0.21 \\ 0.69 & 0.85 & 0.23 & -0.56 \end{bmatrix} \\ &= \begin{bmatrix} 1.42 & 1.09 & 0.32 & 0.17 \\ 1.52 & 0.94 & 0.59 & 0.94 \\ -0.056 & -0.47 & -5.33 \times 10^{-4} & 0.52 \\ 0.044 & -0.61 & 0.27 & 1.29 \end{bmatrix} \end{aligned}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $\hat{Y}$

epoch = 2

$$\hat{Y} = \begin{bmatrix} \frac{e^{z_{1,0}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,1}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,2}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,3}}}{\sum_{c=0}^3 e^{z_{1,c}}} \\ \frac{e^{z_{2,0}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,1}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,2}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,3}}}{\sum_{c=0}^3 e^{z_{2,c}}} \\ \frac{e^{z_{3,0}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,1}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,2}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,3}}}{\sum_{c=0}^3 e^{z_{3,c}}} \\ \frac{e^{z_{4,0}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,1}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,2}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,3}}}{\sum_{c=0}^3 e^{z_{4,c}}} \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $\hat{Y}$

epoch = 2

$$\hat{Y} = \begin{bmatrix} \frac{e^{1.42}}{e^{1.42} + e^{1.09} + e^{0.32} + e^{0.17}} & \frac{e^{z_{1,1}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,2}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,3}}}{\sum_{c=0}^3 e^{z_{1,c}}} \\ \frac{e^{z_{2,0}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,1}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,2}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,3}}}{\sum_{c=0}^3 e^{z_{2,c}}} \\ \frac{e^{z_{3,0}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,1}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,2}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,3}}}{\sum_{c=0}^3 e^{z_{3,c}}} \\ \frac{e^{z_{4,0}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,1}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,2}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,3}}}{\sum_{c=0}^3 e^{z_{4,c}}} \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $\hat{Y}$

epoch = 2

$$\hat{Y} = \begin{bmatrix} 0.43 & 0.31 & 0.14 & 0.12 \\ 0.4 & 0.22 & 0.16 & 0.22 \\ 0.22 & 0.15 & 0.23 & 0.4 \\ 0.16 & 0.08 & 0.2 & 0.56 \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $W$

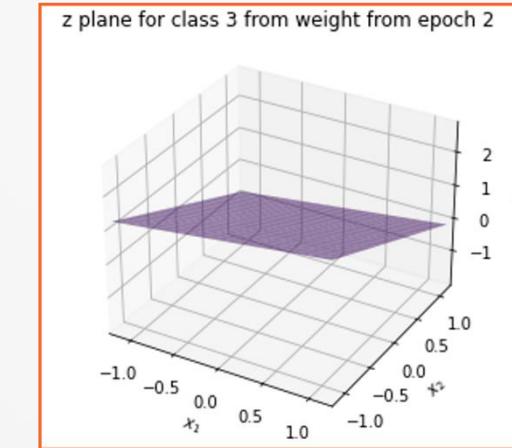
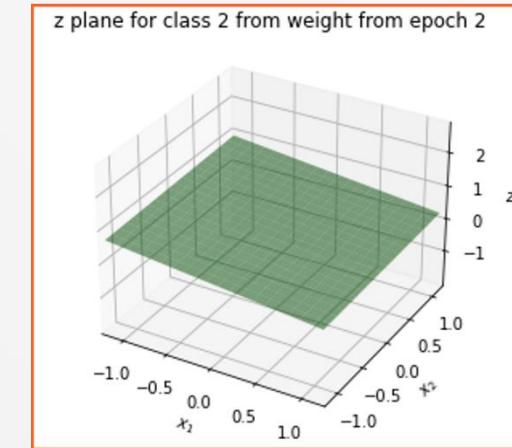
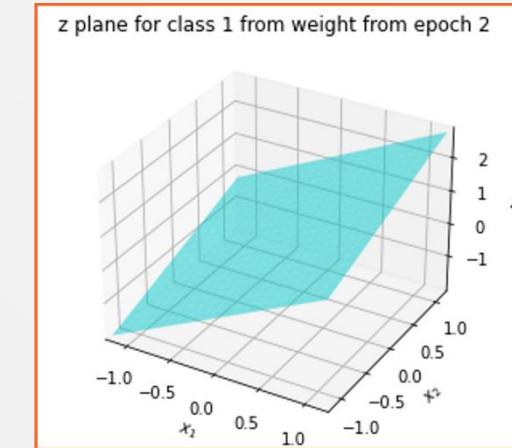
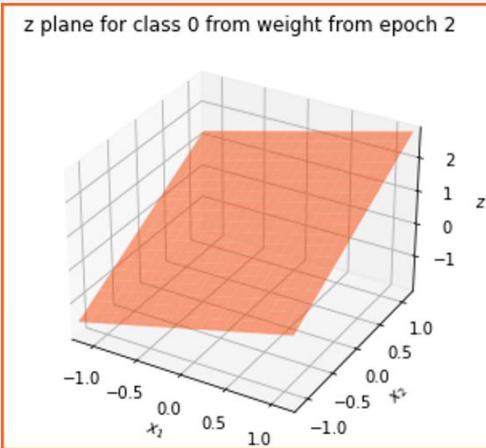
epoch = 2

$$W = W + \frac{\alpha}{n} X_b^T (Y - \hat{Y})$$

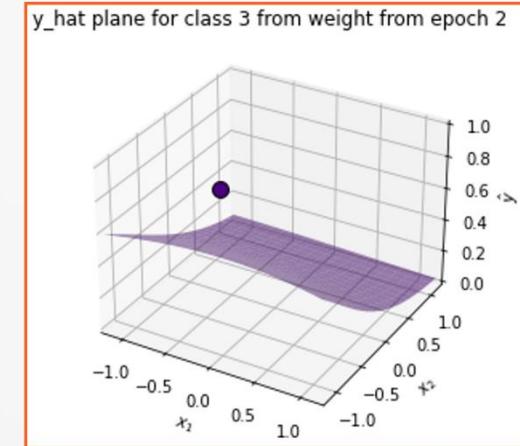
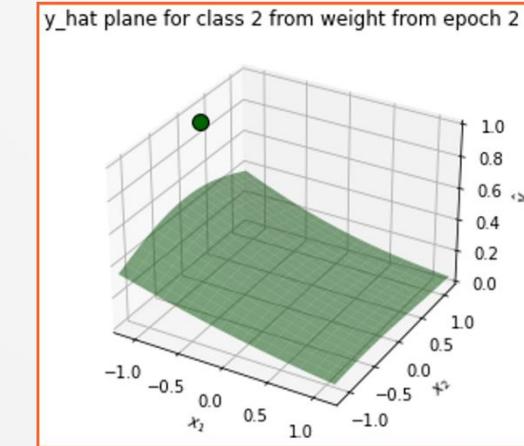
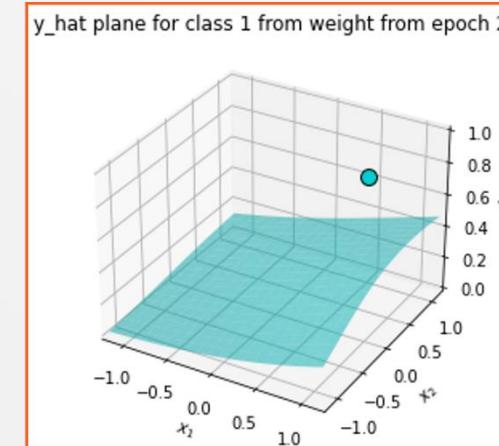
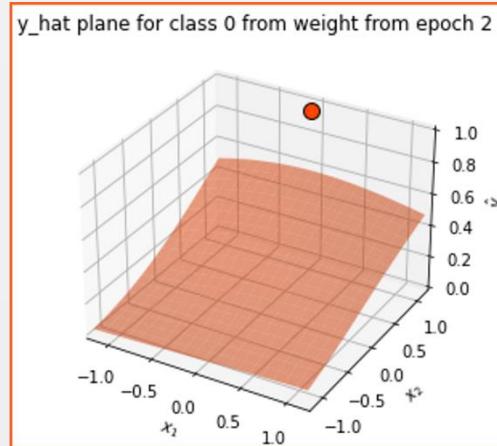
$$= \begin{bmatrix} 0.73 & 0.24 & 0.3 & 0.73 \\ 0.79 & 0.7 & 0.3 & 0.21 \\ 0.69 & 0.85 & 0.23 & -0.56 \end{bmatrix} + \frac{2.5}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \left( \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.43 & 0.31 & 0.14 & 0.12 \\ 0.4 & 0.22 & 0.16 & 0.22 \\ 0.22 & 0.15 & 0.23 & 0.4 \\ 0.16 & 0.08 & 0.2 & 0.56 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 0.6 & 0.39 & 0.46 & 0.55 \\ 0.68 & 1.28 & -0.28 & 0.32 \\ 1.15 & 0.71 & 0.06 & -0.91 \end{bmatrix}$$

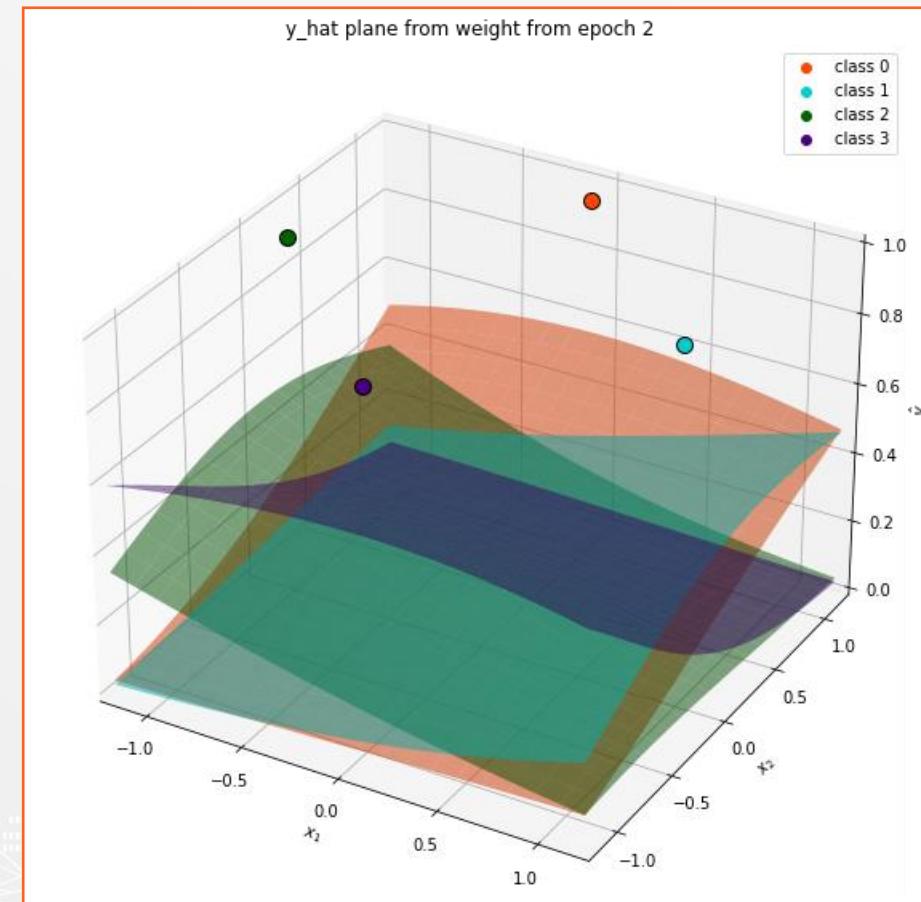
# Calculation Example



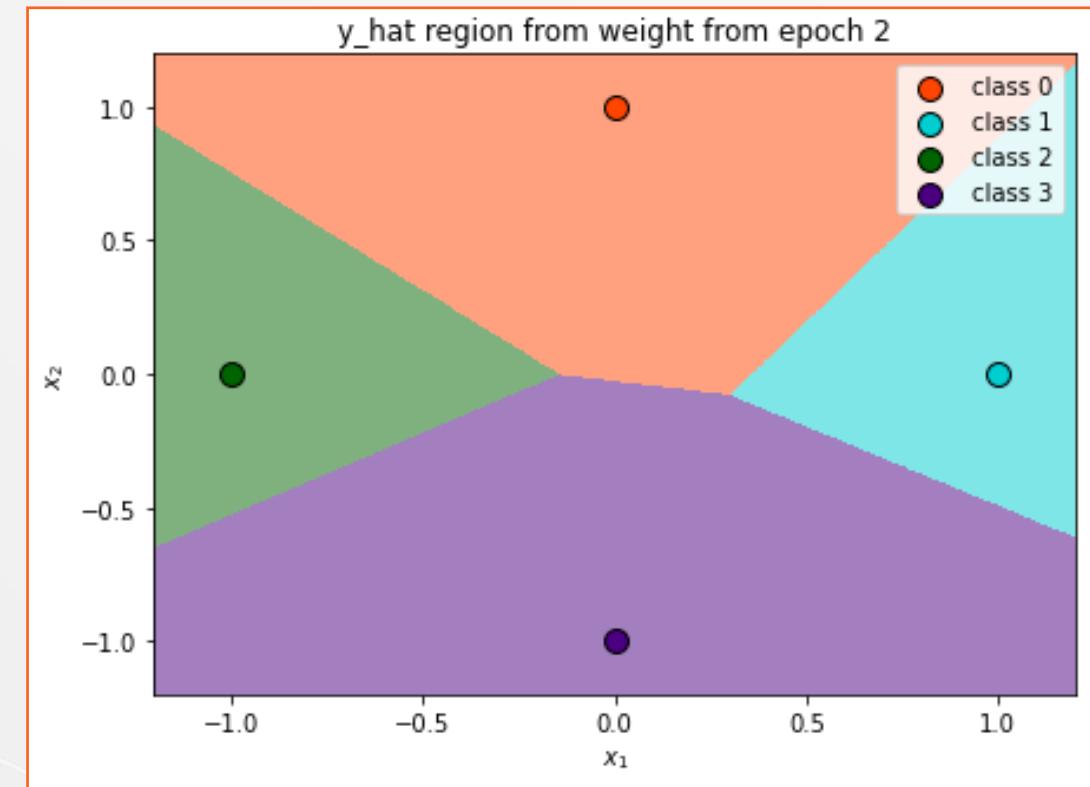
# Calculation Example



# Calculation Example



# Calculation Example



# Calculation Example

## 4. Update $W$ : คำนวณ $Z$

epoch = 3

$$Z = X_b W = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0.6 & 0.39 & 0.46 & 0.55 \\ 0.68 & 1.28 & -0.28 & 0.32 \\ 1.15 & 0.71 & 0.06 & -0.91 \end{bmatrix} = \begin{bmatrix} 1.75 & 1.1 & 0.52 & -0.37 \\ 1.28 & 1.67 & 0.18 & 0.87 \\ -0.08 & -0.89 & 0.74 & 0.23 \\ -0.54 & -0.32 & 0.4 & 1.46 \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $\hat{Y}$

epoch = 3

$$\hat{Y} = \begin{bmatrix} \frac{e^{z_{1,0}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,1}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,2}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,3}}}{\sum_{c=0}^3 e^{z_{1,c}}} \\ \frac{e^{z_{2,0}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,1}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,2}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,3}}}{\sum_{c=0}^3 e^{z_{2,c}}} \\ \frac{e^{z_{3,0}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,1}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,2}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,3}}}{\sum_{c=0}^3 e^{z_{3,c}}} \\ \frac{e^{z_{4,0}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,1}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,2}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,3}}}{\sum_{c=0}^3 e^{z_{4,c}}} \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $\hat{Y}$

epoch = 3

$$\hat{Y} = \begin{bmatrix} \frac{e^{1.75}}{e^{1.75} + e^{1.1} + e^{0.52} + e^{-0.37}} & \frac{e^{z_{1,1}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,2}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,3}}}{\sum_{c=0}^3 e^{z_{1,c}}} \\ \frac{e^{z_{2,0}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,1}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,2}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,3}}}{\sum_{c=0}^3 e^{z_{2,c}}} \\ \frac{e^{z_{3,0}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,1}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,2}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,3}}}{\sum_{c=0}^3 e^{z_{3,c}}} \\ \frac{e^{z_{4,0}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,1}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,2}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,3}}}{\sum_{c=0}^3 e^{z_{4,c}}} \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $\hat{Y}$

epoch = 3

$$\hat{Y} = \begin{bmatrix} 0.52 & 0.27 & 0.15 & 0.06 \\ 0.29 & 0.42 & 0.1 & 0.19 \\ 0.2 & 0.09 & 0.45 & 0.27 \\ 0.08 & 0.1 & 0.21 & 0.61 \end{bmatrix}$$

# Calculation Example

## 4. Update $W$ : คำนวณ $W$

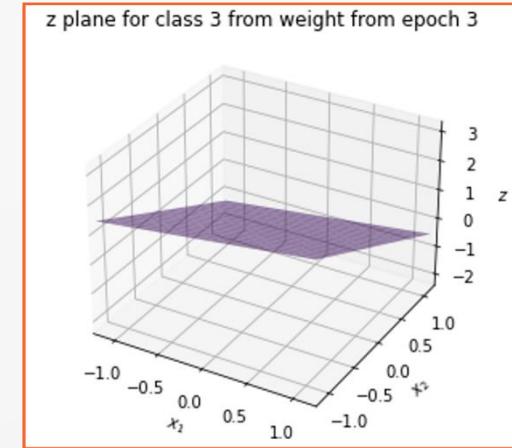
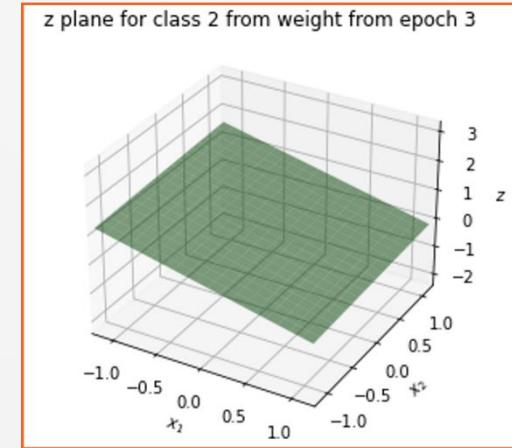
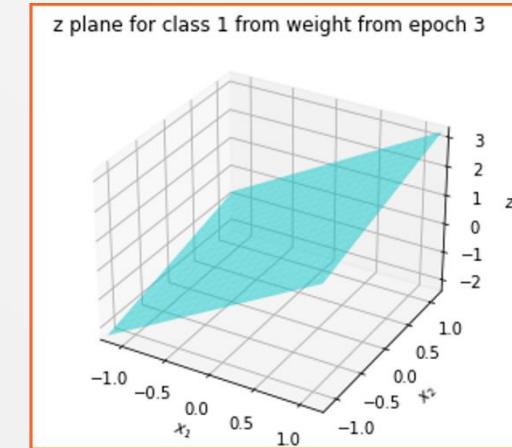
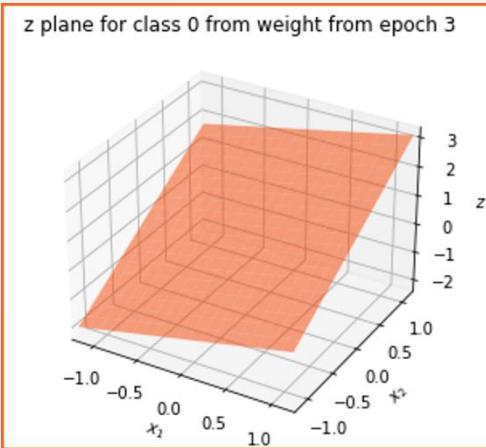
epoch = 3

$$W = W + \frac{\alpha}{n} X_b^T (Y - \hat{Y})$$

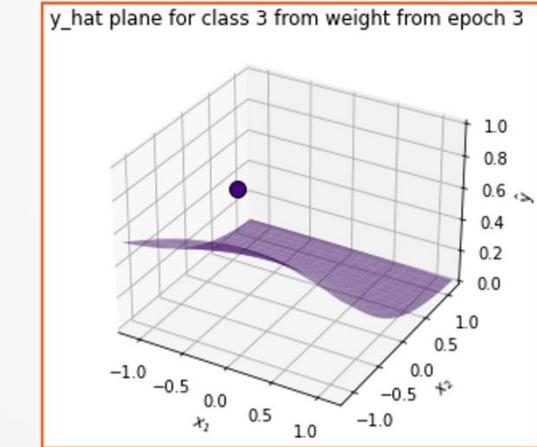
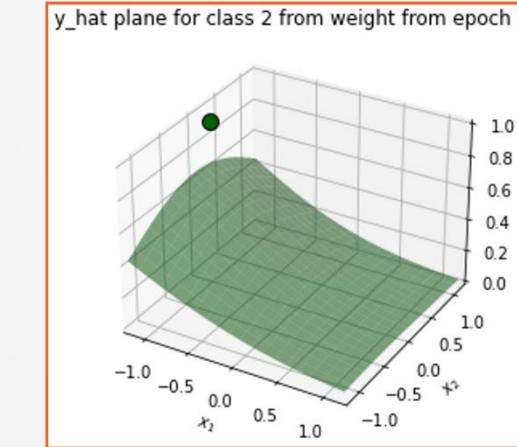
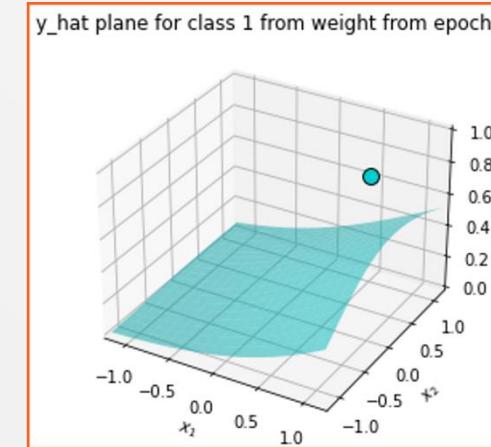
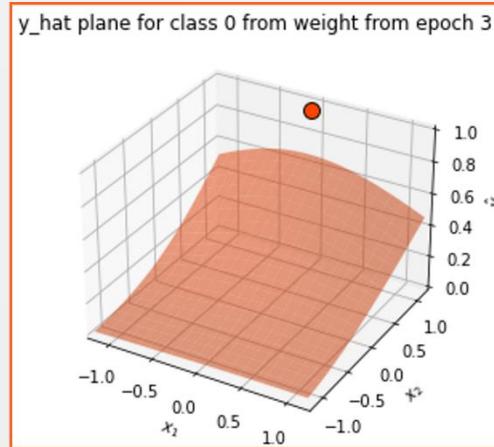
$$= \begin{bmatrix} 0.6 & 0.39 & 0.46 & 0.55 \\ 0.68 & 1.28 & -0.28 & 0.32 \\ 1.15 & 0.71 & 0.06 & -0.91 \end{bmatrix} + \frac{2.5}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} - \begin{bmatrix} 0.52 & 0.27 & 0.15 & 0.06 \\ 0.29 & 0.42 & 0.1 & 0.19 \\ 0.2 & 0.09 & 0.45 & 0.27 \\ 0.08 & 0.1 & 0.21 & 0.61 \end{bmatrix}$$

$$= \begin{bmatrix} 0.55 & 0.46 & 0.52 & 0.47 \\ 0.62 & 1.69 & -0.68 & 0.37 \\ 1.5 & 0.6 & 0.09 & -1.2 \end{bmatrix}$$

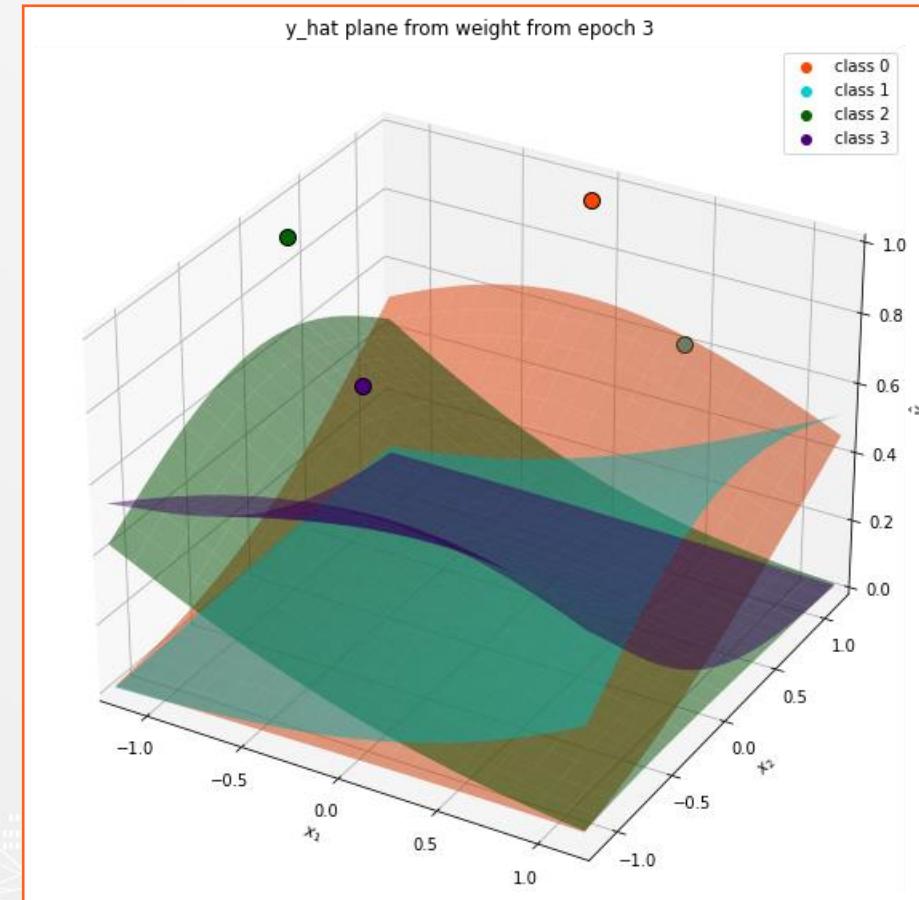
# Calculation Example



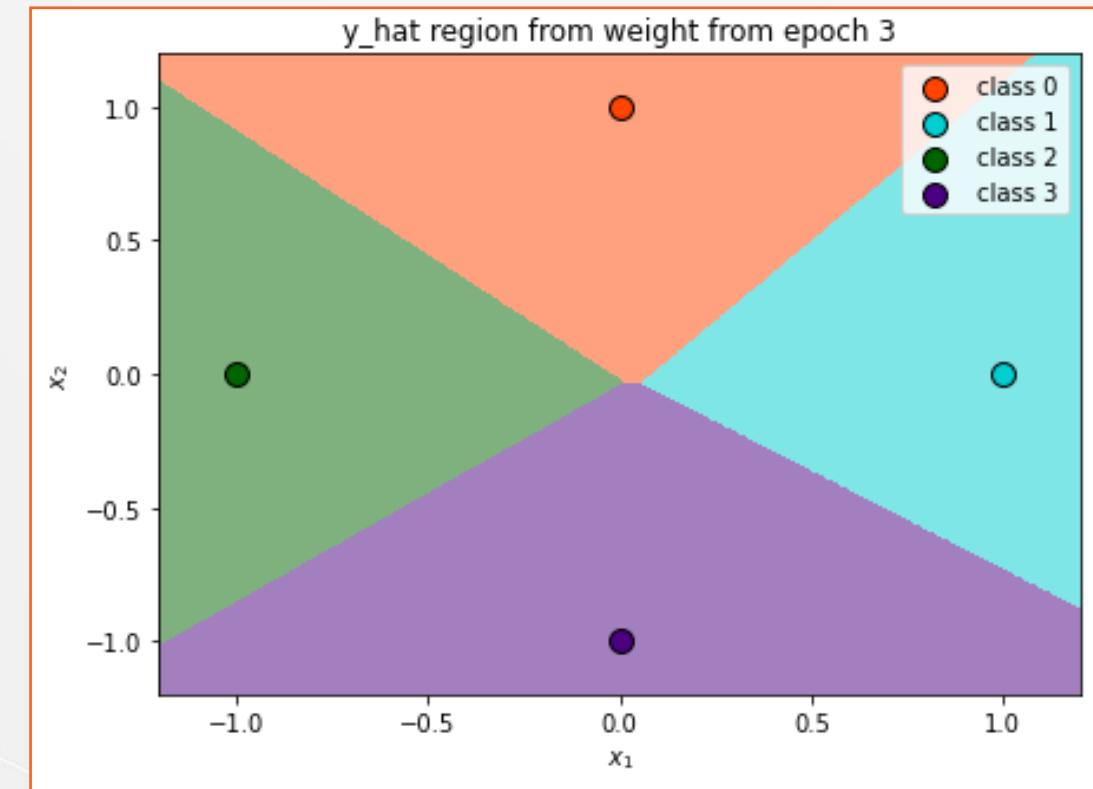
# Calculation Example



# Calculation Example

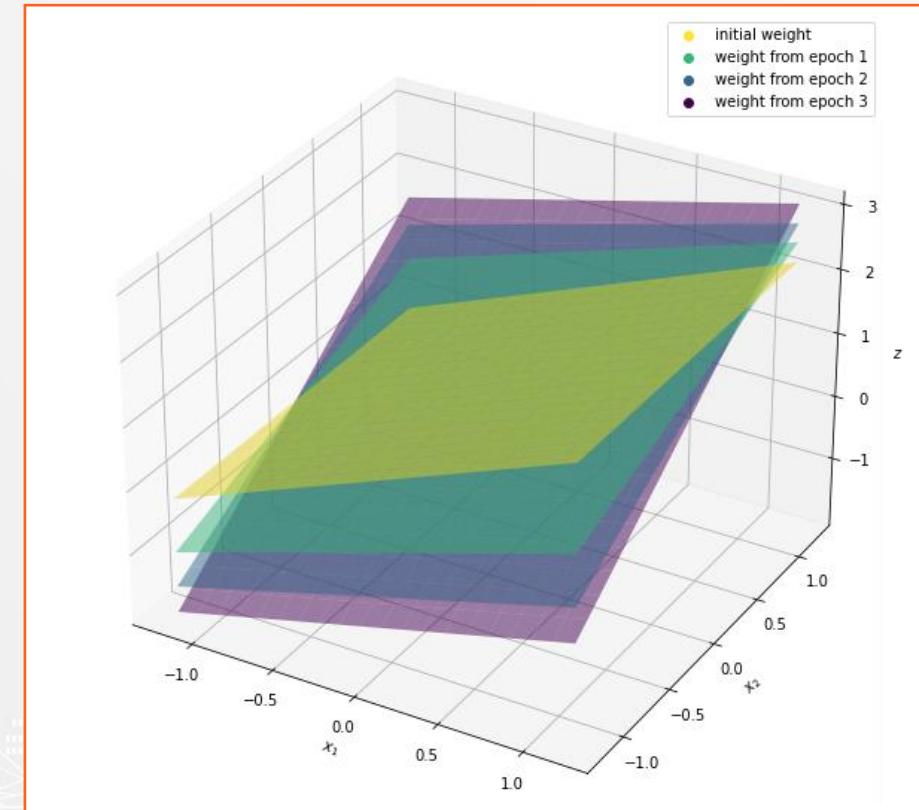


# Calculation Example



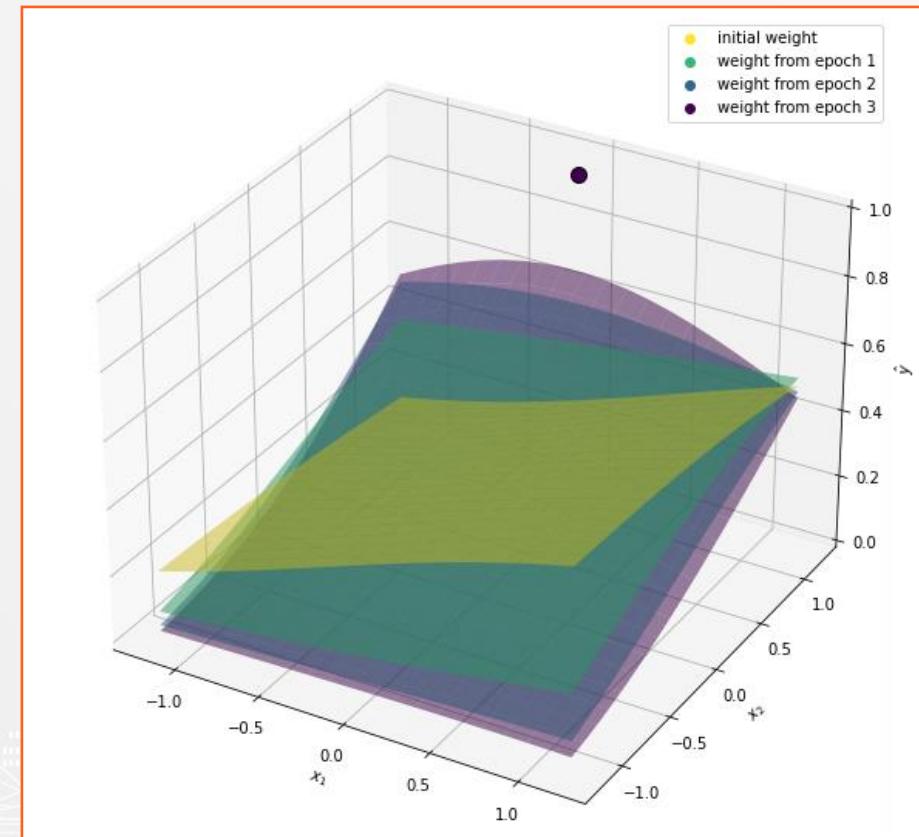
# Calculation Example

z plane ឧបនគរណៈ epoch (class 0)

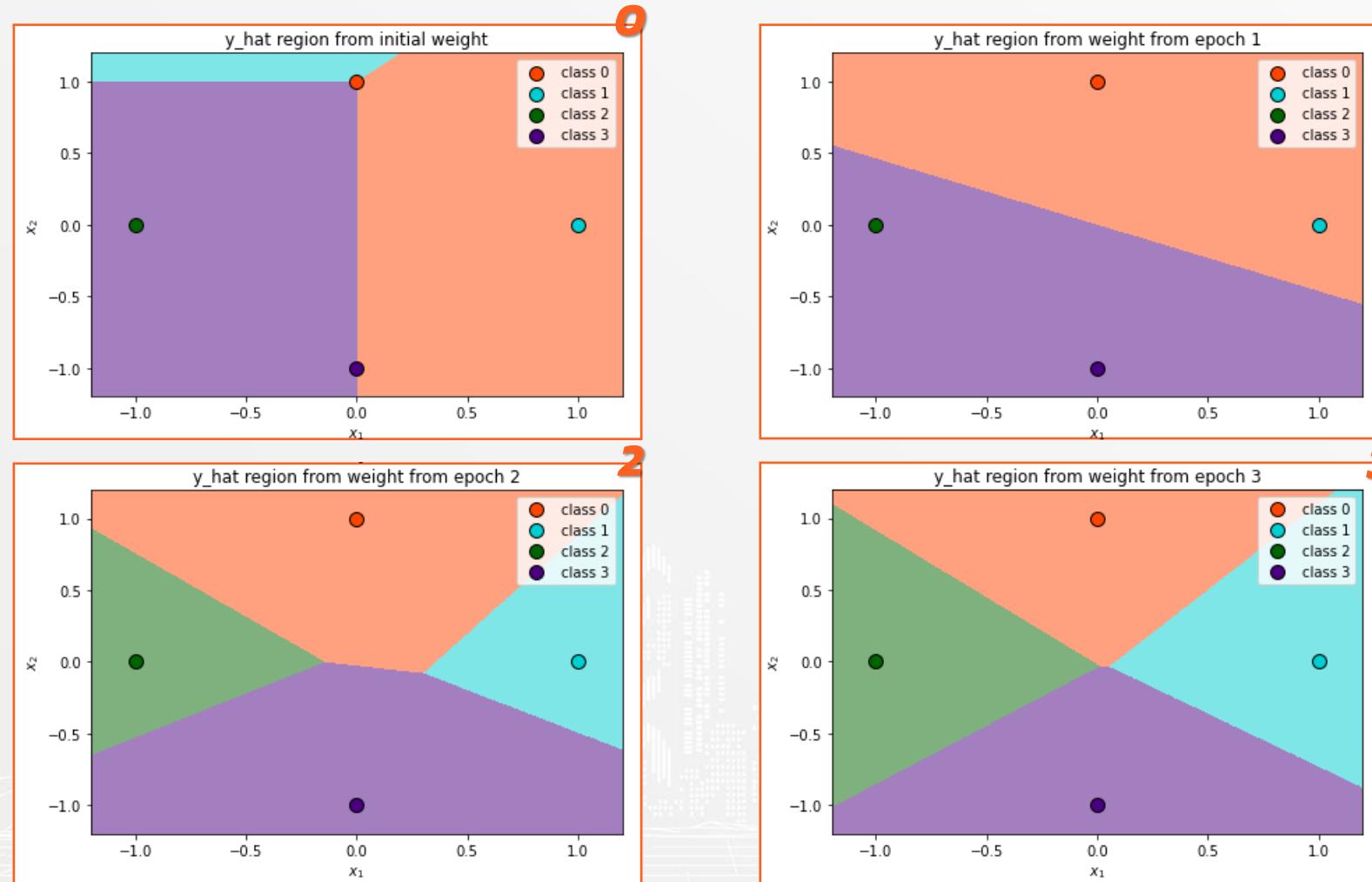


# Calculation Example

$\hat{y}$  plane នៃវត្ថុ៖ epoch (class 0)



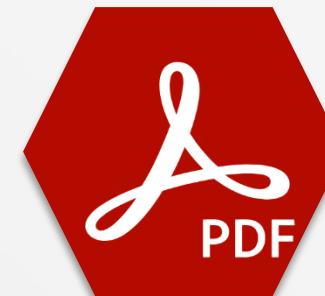
# Calculation Example



# Calculation Example



Exercise of Logistic Regression (multi-class)

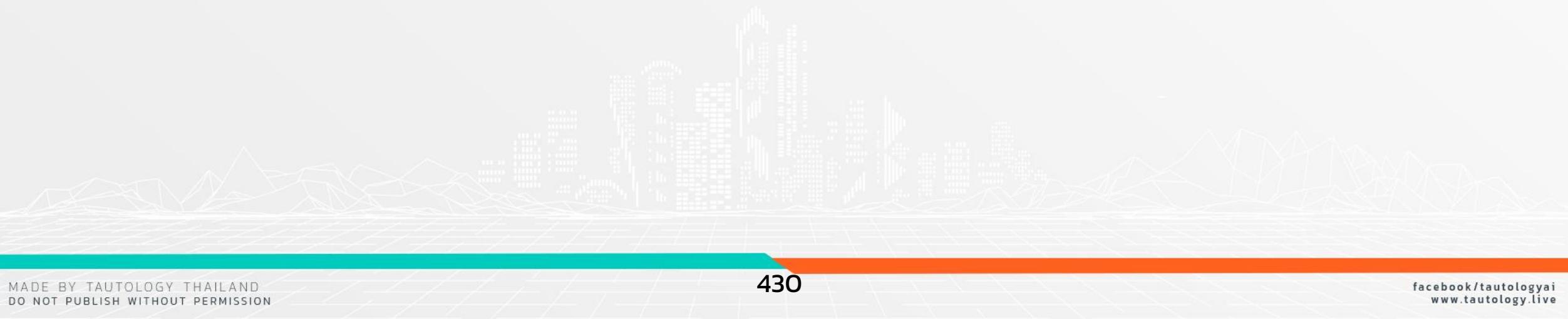


Open File

**Exercise\_LoR\_(multi-class).pdf**

# How to Create Model (Math)

- Gradient Descent (Logistic Regression)**
- Calculation Example**



# Model

**Assumption**



**Real Face of the Model**



**Cost Function and Cost Landscape**



**How to Create Model (Math)**



**How to Create Model (Code)**



# How to Create Model (Code)

ตัวอย่าง code สำหรับคำนวณ  $W$

$x_1$	$x_2$	$y$
0	1	1
1	0	2
-1	0	3
0	-1	4

ตารางแสดง dataset

# How to Create Model (Code)

- Code สำหรับสร้าง model จากข้อมูลของเราโดยที่

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \text{ และ } y = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

```
1 clf = LogisticRegression(penalty='none')
2 clf.fit(x, y)
```

```
LogisticRegression(penalty='none')
```

# How to Create Model (Code)

- ค่า  $w_0$  ถูกเก็บไว้ใน attribute ชื่อ intercept\_

```
1 clf.intercept_
```

```
array([-2.62306367e-08,  2.95335134e-08,  2.54560370e-08, -2.87561552e-08])
```

# How to Create Model (Code)

- ค่า  $w_1, \dots, w_p$  จะเก็บไว้ใน attribute ชื่อ `coef_`

```
1 clf.coef_
```

```
array([[-1.44123650e-09,  1.04728380e+01],
       [ 1.04728380e+01, -8.95380278e-10],
       [-1.04728380e+01, -8.95382812e-10],
       [-1.44123698e-09, -1.04728380e+01]])
```

# How to Create Model (Code)

ดังนั้น เราจะสามารถเขียนเป็น model ของ logistic regression สำหรับข้อมูลชุดนี้ได้ดังนี้

$$z_0 = 0 + 0x_1 + 10.47x_2$$

$$z_1 = 0 + 10.47x_1 + 0x_2$$

$$z_2 = 0 - 10.47x_1 + 0x_2$$

$$z_3 = 0 + 0x_1 - 10.47x_2$$

$$\hat{y} = \frac{e^{z_m}}{\sum_{c=0}^3 e^{z_c}}$$

# How to Create Model (Code)

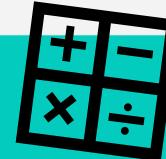


Code for this section



Open File  
**Model Creation (multi-class).ipynb**

# How to Create Model (Code)



MATH

$$W = \begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,k-1} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{p,0} & w_{p,1} & \cdots & w_{p,k-1} \end{bmatrix}$$

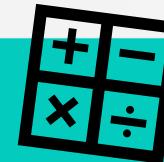


CODE

`intercept_ = [w0,0 w0,1 ... w0,k-1]`

`coef_ = [ w1,0 w2,0 ... wp,0 ]  
          [ w1,1 w2,1 ... wp,1 ]  
          [ : : ... : : ]  
          [ w1,k-1 w2,k-1 ... wp,k-1 ]`

# How to Create Model (Code)



MATH

1. สุ่ม  $W$  เริ่มต้น
2. กำหนดจำนวนรอบที่จะ update  $W$  (epoch)
3. กำหนดค่า  $\alpha$
4. for i in range(epoch):  
    update  $W$

scikit  
learn

CODE

```
1 clf = LogisticRegression(penalty='none')
2 clf.fit(X, y)

LogisticRegression(penalty='none')

1 clf.intercept_
array([-2.62306367e-08,  2.95335134e-08,  2.54560370e-08, -2.87561552e-08])

1 clf.coef_
array([[-1.44123650e-09,  1.04728380e+01],
       [ 1.04728380e+01, -8.95380278e-10],
       [-1.04728380e+01, -8.95382812e-10],
       [-1.44123698e-09, -1.04728380e+01]])
```

# Model

**Assumption**



**Real Face of the Model**



**Cost Function and Cost Landscape**



**How to Create Model (Math)**



**How to Create Model (Code)**



# Logistic Regression (Multi-Class Classification)



# Prediction

# Prediction

**Logistic regression** คือ โมเดลคณิตศาสตร์ที่จำแนกประเภทของข้อมูล ด้วยการสร้างสมการเชิงเส้น (linear regression)

$$z_m = w_{0,m} + w_{1,m}x_1 + w_{2,m}x_2 + \cdots + w_{p,m}x_p$$

$$\hat{y}_m = \frac{e^{z_m}}{\sum_{c=0}^{k-1} e^{z_c}}$$

โดยที่

- ◆  $\hat{y}_m$  คือ ความน่าจะเป็นที่จะเป็น class ที่  $m$  (predicted target)
- ◆  $z_m$  คือ สมการเชิงเส้น (linear regression) ของ class ที่  $m$
- ◆  $x_1, x_2, x_3, \dots, x_p$  คือ ตัวแปรตัว (feature)
- ◆  $w_{0,m}, w_{1,m}, w_{2,m}, \dots, w_{p,m}$  คือ สัมประสิทธิ์ที่ส่งจาก feature ไป class ที่  $m$  (coefficient)
- ◆  $m \in 0, 1, 2, \dots, k - 1$

# Prediction

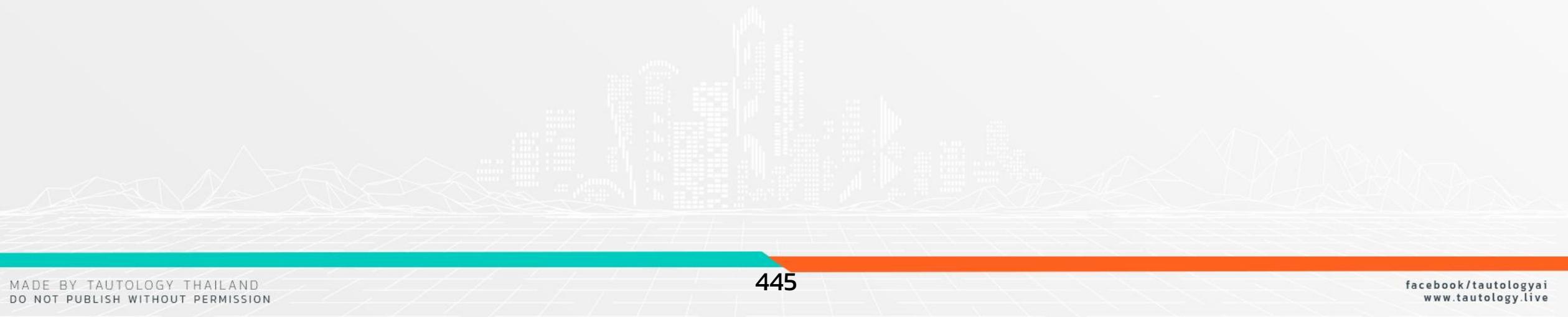
$$W = \begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,k-1} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{p,0} & w_{p,1} & \cdots & w_{p,k-1} \end{bmatrix}$$

# Prediction

1-Sample

Multi-Sample

Code



# 1-Sample

ตัวอย่างการคำนวณ  $\hat{y}$

$x_1$	$x_2$
0	1.5



$\hat{y}$
?

# 1-Sample

สมมติว่า  $W$  ของปัญหานี้ที่เราหามาได้คือ

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 10.47 & -10.47 & 0 \\ 10.47 & 0 & 0 & -10.47 \end{bmatrix}$$

# 1-Sample

ซึ่งทำให้สามารถเขียนสมการ  $\hat{y}$  ได้ดังต่อไปนี้

- ✿  $z_0 = 0 + 0 \cdot x_1 + 10.47x_2 = 10.47 \times 1.5 = 15.705$
- $z_1 = 0 + 10.47x_1 + 0 \cdot x_2 = 10.47 \times 0 = 0$
- $z_2 = 0 - 10.47x_1 + 0 \cdot x_2 = 10.47 \times 0 = 0$
- $z_3 = 0 + 0 \cdot x_1 - 10.47x_2 = 10.47 \times 1.5 = -15.705$

# 1-Sample

ซึ่งทำให้สามารถเขียนสมการ  $\hat{y}$  ได้ดังต่อไปนี้

$$\star \quad \hat{y}_0 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3} + e^{z_4}} = \frac{e^{15.705}}{e^{15.705} + e^0 + e^0 + e^{-15.705}} \approx 1 \quad \rightarrow \quad 0$$

$$\hat{y}_1 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3} + e^{z_4}} = \frac{e^0}{e^{15.705} + e^0 + e^0 + e^{-15.705}} \approx 0$$

$$\hat{y}_2 = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3} + e^{z_4}} = \frac{e^0}{e^{15.705} + e^0 + e^0 + e^{-15.705}} \approx 0$$

$$\hat{y}_3 = \frac{e^{z_4}}{e^{z_1} + e^{z_2} + e^{z_3} + e^{z_4}} = \frac{e^{-15.705}}{e^{15.705} + e^0 + e^0 + e^{-15.705}} \approx 0$$

# 1-Sample

ดังนั้น เราจะได้  $\hat{y}$  ของข้อมูลชุดนี้คือ



# Prediction

**1-Sample**



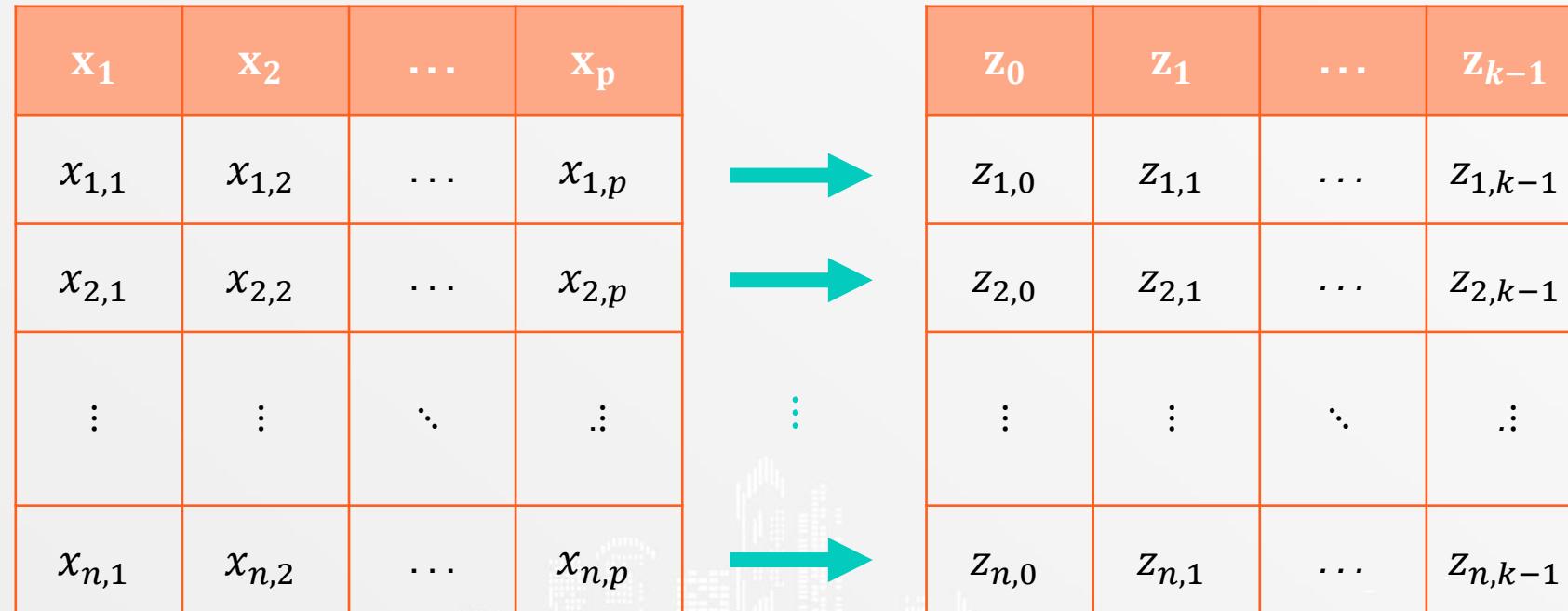
Multi-Sample



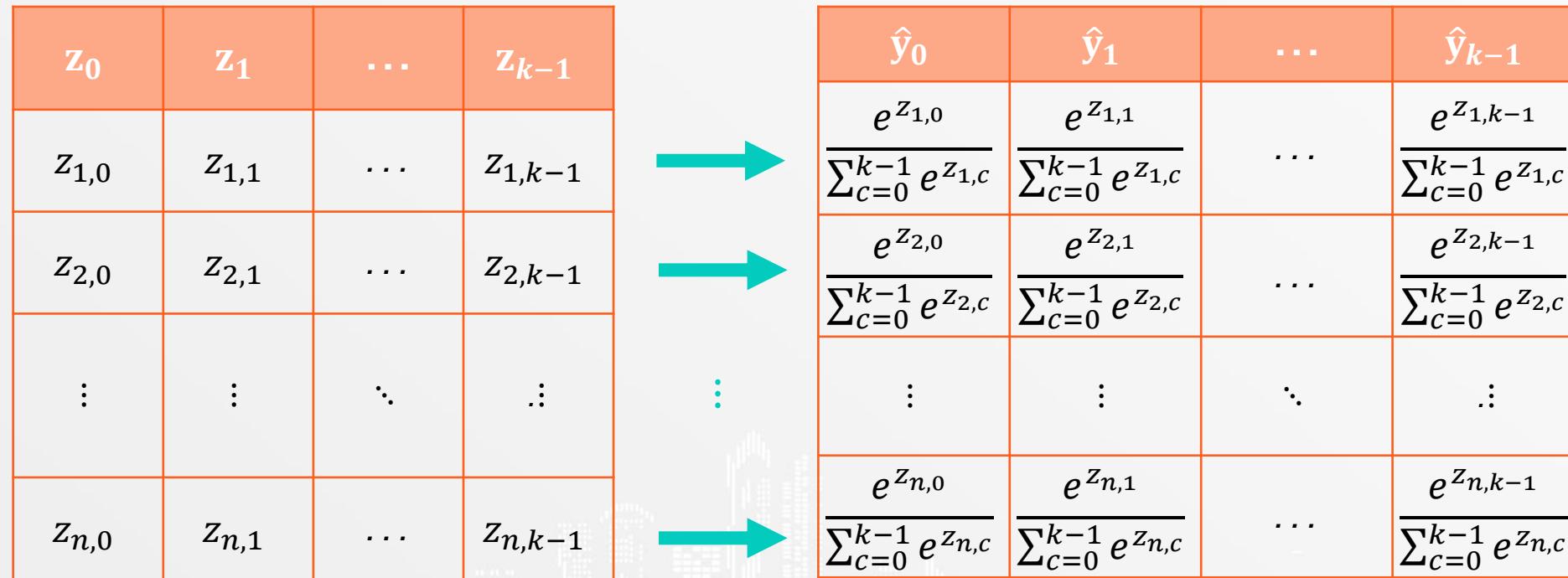
Code



# Multi-Sample



# Multi-Sample



# Multi-Sample

ตัวอย่างการคำนวณ  $\hat{y}$

$x_1$	$x_2$
0	1.5
1.5	0
-1.5	0
0	-1.5



$\hat{y}$
?
?
?
?

# Multi-Sample

- สมมติว่า  $W$  ของปัญหานี้ที่เราหามาได้คือ

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 10.47 & -10.47 & 0 \\ 10.47 & 0 & 0 & -10.47 \end{bmatrix}$$

- และจากข้อมูลใน dataset เราสามารถเขียน  $X_b$  ได้ดังต่อไปนี้

$$X_b = \begin{bmatrix} 1 & 0 & 1.5 \\ 1 & 1.5 & 0 \\ 1 & -1.5 & 0 \\ 1 & 0 & -1.5 \end{bmatrix}$$

# Multi-Sample

เราคำนวณค่า  $Z$  ได้จาก  $Z = X_b W$

$$Z = \begin{bmatrix} 1 & 0 & 1.5 \\ 1 & 1.5 & 0 \\ 1 & -1.5 & 0 \\ 1 & 0 & -1.5 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 10.47 & -10.47 & 0 \\ 10.47 & 0 & 0 & -10.47 \end{bmatrix}$$
$$= \begin{bmatrix} 15.705 & 0 & 0 & -15.705 \\ 0 & 15.705 & -15.705 & 0 \\ 0 & -15.705 & 15.705 & 0 \\ -15.705 & 0 & 0 & 15.705 \end{bmatrix}$$

# Multi-Sample

จาก  $\hat{y}_m = \frac{e^{z_m}}{\sum_{c=1}^k e^{z_c}}$  จะได้ว่า

$$\hat{Y} = \begin{bmatrix} \frac{e^{15.705}}{e^{15.705} + e^0 + e^0 + e^{-15.705}} & \frac{e^{z_{1,1}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,2}}}{\sum_{c=0}^3 e^{z_{1,c}}} & \frac{e^{z_{1,3}}}{\sum_{c=0}^3 e^{z_{1,c}}} \\ \frac{e^{z_{2,0}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,1}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,2}}}{\sum_{c=0}^3 e^{z_{2,c}}} & \frac{e^{z_{2,3}}}{\sum_{c=0}^3 e^{z_{2,c}}} \\ \frac{e^{z_{3,0}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,1}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,2}}}{\sum_{c=0}^3 e^{z_{3,c}}} & \frac{e^{z_{3,3}}}{\sum_{c=0}^3 e^{z_{3,c}}} \\ \frac{e^{z_{4,0}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,1}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,2}}}{\sum_{c=0}^3 e^{z_{4,c}}} & \frac{e^{z_{4,3}}}{\sum_{c=0}^3 e^{z_{4,c}}} \end{bmatrix}$$

# Multi-Sample

จาก  $\hat{y}_m = \frac{e^{z_m}}{\sum_{c=0}^{k-1} e^{z_c}}$  จะได้ว่า

$$\hat{Y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Multi-Sample

จาก  $\hat{y}_m = \frac{e^{z_m}}{\sum_{c=0}^{k-1} e^{z_c}}$  จะได้ว่า

$$\hat{Y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \hat{\mathbf{y}} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

# Multi-Sample

ดังนั้น เราจะได้  $\hat{y}$  สำหรับข้อมูลชุดนี้คือ

$x_1$	$x_2$
0	1.5
1.5	0
-1.5	0
0	-1.5



$\hat{y}$
0
1
2
3

# Prediction

**1-Sample**



**Multi-Sample**



**Code**



# Code

ตัวอย่าง code สำหรับหา  $\hat{y}$

$x_1$	$x_2$
0	1
1	0
-1	0
0	-1



$\hat{y}$
?
?
?
?

# Code

- Code สำหรับหา  $\hat{y}$  จากข้อมูลของเราโดยที่  $X = \begin{bmatrix} 0 & 1.5 \\ 1.5 & 0 \\ -1.5 & 0 \\ 0 & -1.5 \end{bmatrix}$

```
1 clf.predict(X)
```

```
array([0, 1, 2, 3], dtype=int64)
```

# Code

- Code สำหรับหาค่า probability ของ sample ที่จะถูกพยากรณ์ในแต่ละ class

```
1 clf.predict_proba(X)
```

```
array([[9.9999699e-01, 1.50506847e-07, 1.50506847e-07, 2.26523154e-14],  
       [1.50506824e-07, 9.9999699e-01, 2.26523135e-14, 1.50506824e-07],  
       [1.50506826e-07, 2.26523137e-14, 9.9999699e-01, 1.50506825e-07],  
       [2.26523155e-14, 1.50506848e-07, 1.50506848e-07, 9.9999699e-01]])
```

# Code

ดังนั้น เราจะได้  $\hat{y}$  สำหรับข้อมูลชุดนี้คือ

$x_1$	$x_2$
0	1
1	0
-1	0
0	-1



$\hat{y}$
1
2
3
4

# Code

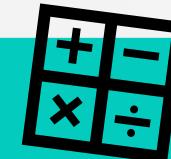


Code for this section



Open File  
**Model Creation (multi-class).ipynb**

# Code



MATH

$$Z = X_b W$$

$$\hat{y}_m = \frac{e^{z_m}}{\sum_{c=0}^{k-1} z_c}$$



CODE

```
1 clf.predict(X)
```

```
array([0, 1, 2, 3], dtype=int64)
```

# Prediction

**1-Sample**



**Multi-Sample**



**Code**



# Logistic Regression (Multi-Class Classification)



# Improvement

# Improvement

Imbalanced class

Regularization

# Imbalance Class

Bootstrapping

SMOTE

ADASYN

Balanced  
Class Weight

# Imbalance Class

Bootstrapping

SMOTE

ADASYN

Balanced  
Class Weight

# Imbalance Class

```
1 clf = LogisticRegression(penalty='none', class_weight='balanced')
2 clf.fit(X, y)
```

# Improvement

Imbalanced class



Regularization



# Regularization

**L2**  
**Regularization**

**L1**  
**Regularization**

**Elastic Net**



# Regularization

## L2 Regularization

```
1 alphas = np.logspace(-3, 3, num=100)
```

```
1 clf = LogisticRegressionCV(penalty='l2', Cs=alphas, cv=5)
2 clf.fit(X, y)
```

# Regularization

## L1 Regularization

```
1 alphas = np.logspace(-3, 3, num=100)
```

```
1 clf = LogisticRegressionCV(penalty='l1', solver='saga', Cs=alphas, cv=5)
2 clf.fit(X, y)
```

# Regularization

## Elastic Net

```
1 alphas = np.logspace(-3, 3, num=100)
2 l1_ratios = [0.01, 0.05, 0.1, 0.5, 0.7, 0.9, 0.95, 0.99, 1]
```

```
1 clf = LogisticRegressionCV(penalty='elasticnet', solver='saga',
2                             l1_ratios=l1_ratios, Cs=alphas, cv=5)
3 clf.fit(X, y)
```

# Regularization



Code for this section



Open File  
**Regularization (multi-class).ipynb**

# Improvement

**Imbalanced class**



**Regularization**



# Logistic Regression (Multi-Class Classification)



# DL102 : Logistic Regression

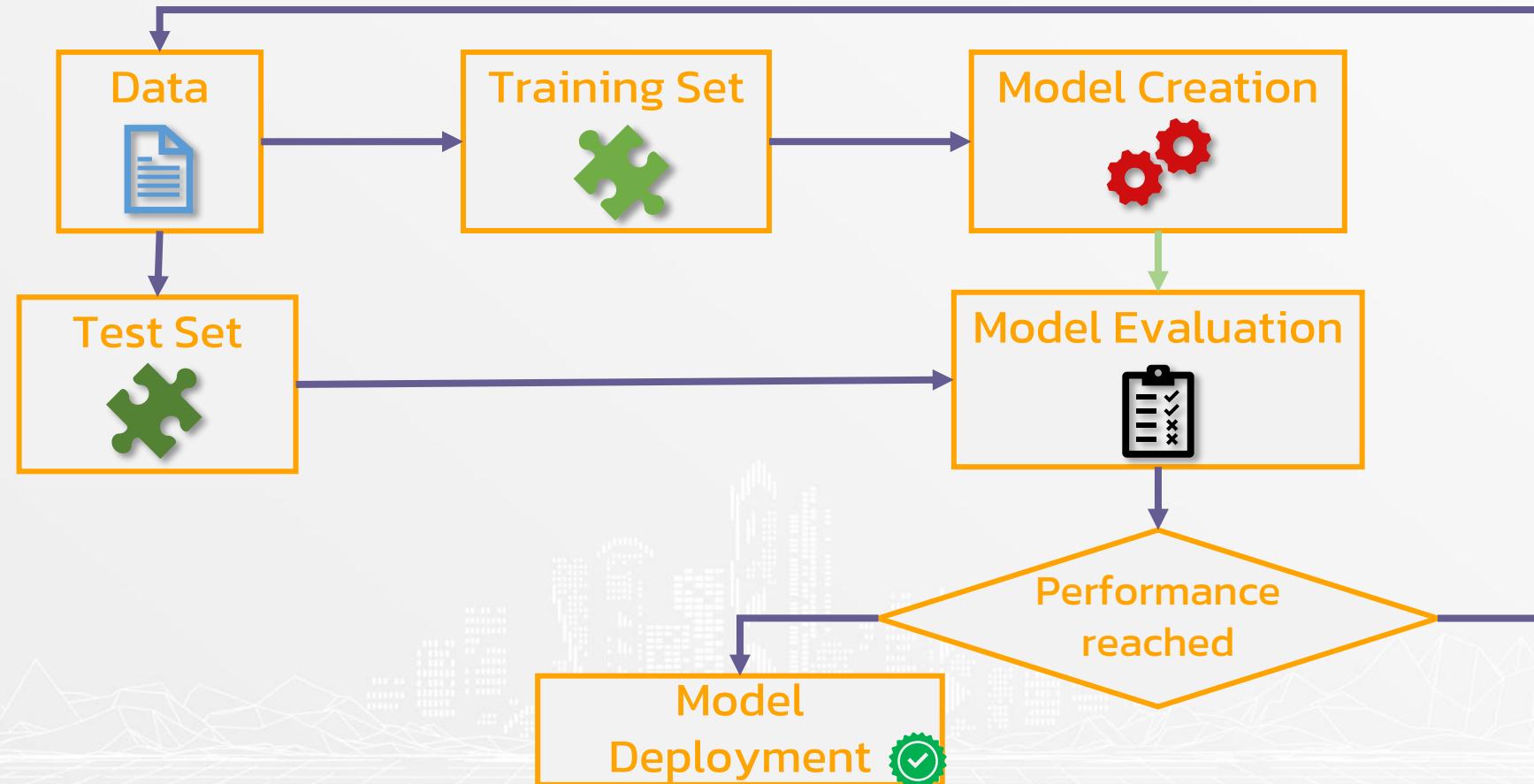




# LOGISTIC REGRESSION WORKSHOP

BY TAU<sup>O</sup>LOGY

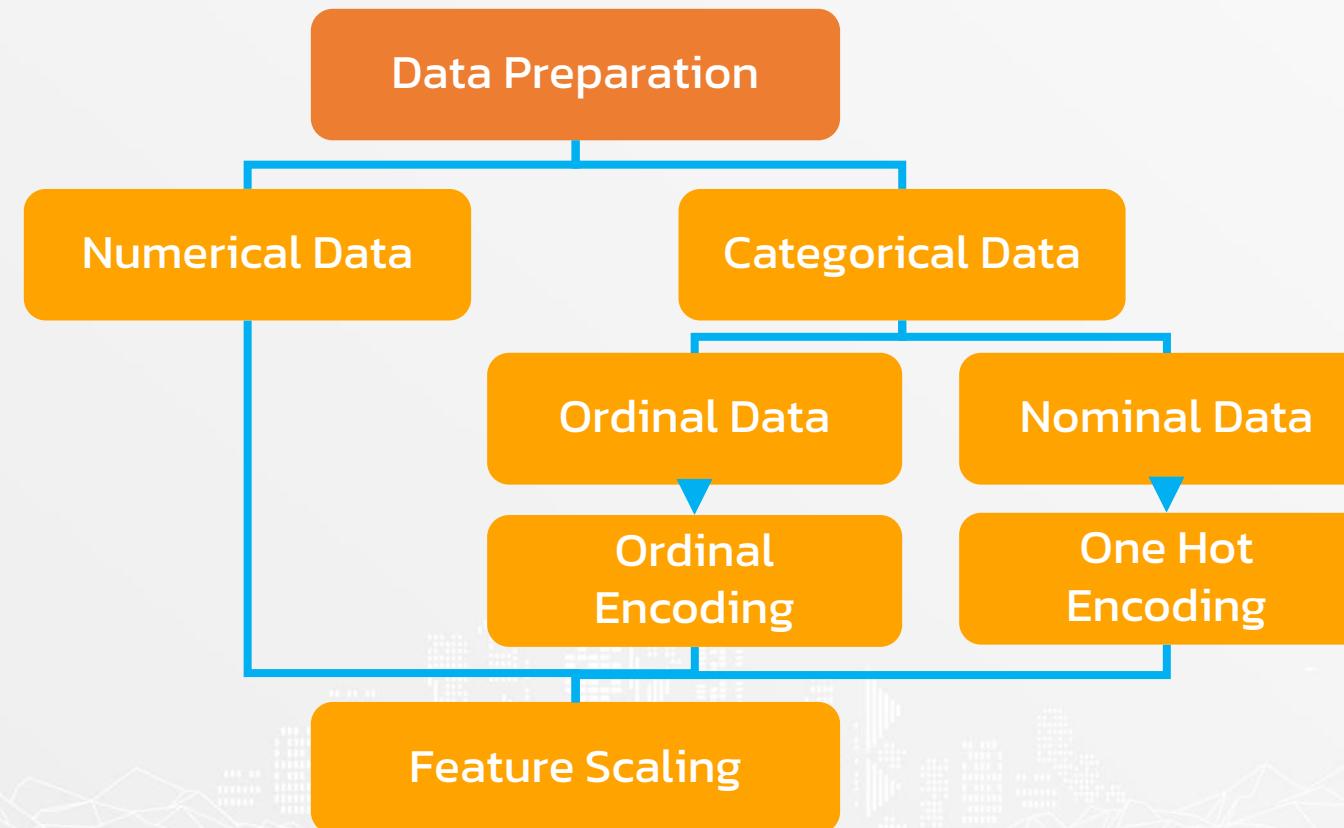
# Supervised Learning Workflow



# Code Pipeline



# Data Preparation





# AI in Fruit Industry

# Abstract

สร้าง model เพื่อจำแนกผลไม้สด และผลไม้เสีย สำหรับ apple, banana และ orange โดยพิจารณาจากรูปผลไม้



# Why this project important?



- ◆ สามารถสร้างระบบคัดแยกผลผลิตที่ทำงานได้อย่างมีประสิทธิภาพ
- ◆ สามารถนำความรู้ไปต่อยอดเพื่อสร้าง smart farm
- ◆ สามารถนำไปต่อยอดเพื่อจำแนกผลไม้ หรือ สับค้า ชนิดอื่น

# Who this project is for?

- ◆ เกษตรกรที่สนใจ AI กับการเกษตร
- ◆ ผู้ควบคุมสายการผลิต
- ◆ นักวิเคราะห์ข้อมูล



# Fruit Dataset



Dataset

## Fruits fresh and rotten for classification

Apples Oranges Bananas

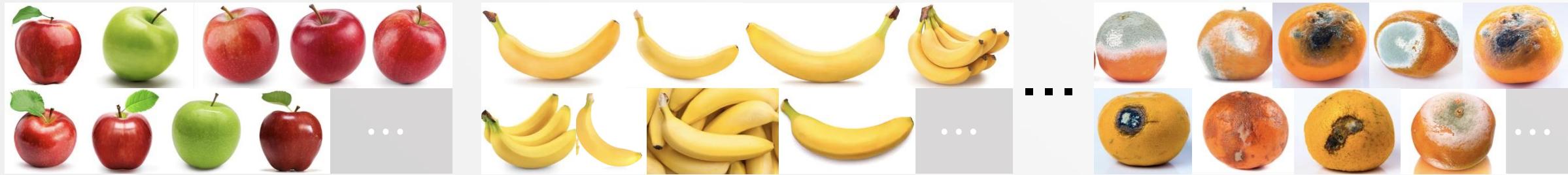


Sriram Reddy Kalluri • updated 3 years ago (Version 1)

<https://www.kaggle.com/sriramr/fruit-fresh-and-rotten-for-classification>

# Fruit Dataset

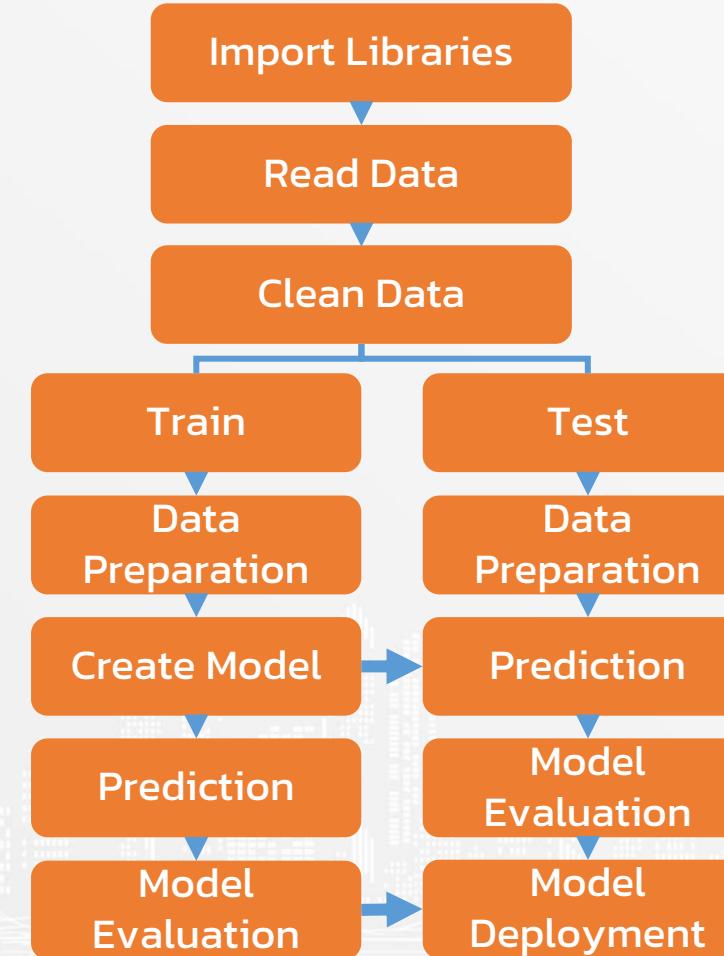
## Feature



## Target

- target : **freshapples**, **freshbanana**, **freshoranges**, **rottenapples**, **rottenbanana**, **rottenoranges**

# What we learn from this project?



# Import Libraries

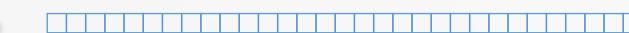
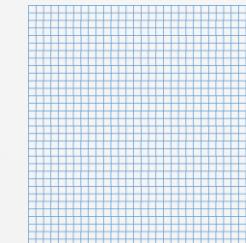
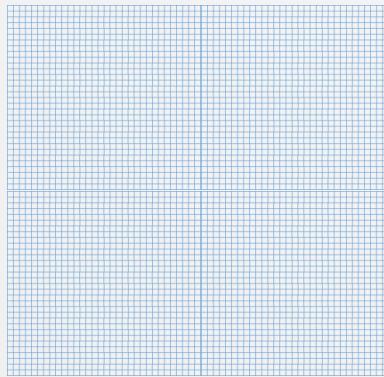
```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 from glob import glob
6 from PIL import Image
7 import cv2
8 from tqdm.auto import tqdm
9 from sklearn.model_selection import train_test_split
10 from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder, StandardScaler, MinMaxScaler
11 from sklearn.linear_model import LogisticRegression, LogisticRegressionCV
12 from sklearn.metrics import (
13     plot_confusion_matrix,
14     classification_report
15 )
16
17 import warnings
18 warnings.filterwarnings('ignore')
19
20 np.random.seed(12345)
```

# Read Data

```
1 classes = ['freshapples', 'freshbanana', 'freshoranges',  
2             'rottenapples', 'rottenbanana', 'rottenoranges']
```

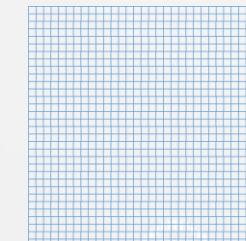
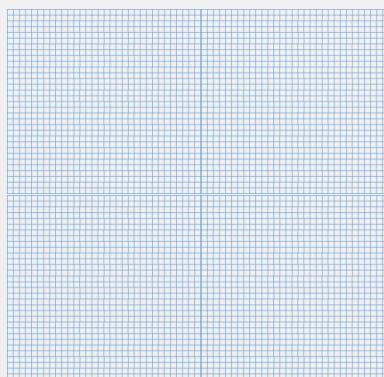
```
1 X = np.empty([0, 32*32*3])  
2 y = np.empty([0, 1])  
3  
4 for _class in tqdm(classes):  
5     img_path = glob('dataset/' + _class + '*')  
6     for path in tqdm(img_path):  
7         img = Image.open(path)  
8         img = img.resize((32, 32))  
9         img = np.array(img)  
10        if img.shape[2] == 4:  
11            img = cv2.cvtColor(img, cv2.COLOR_BGRA2BGR)  
12        img = img.reshape(1, -1)  
13        X = np.vstack([X, img])  
14        y = np.vstack([y, _class])
```

# Read Data



**freshapples**

**32x32**



**freshbanana**

**32x32**

# Read Data

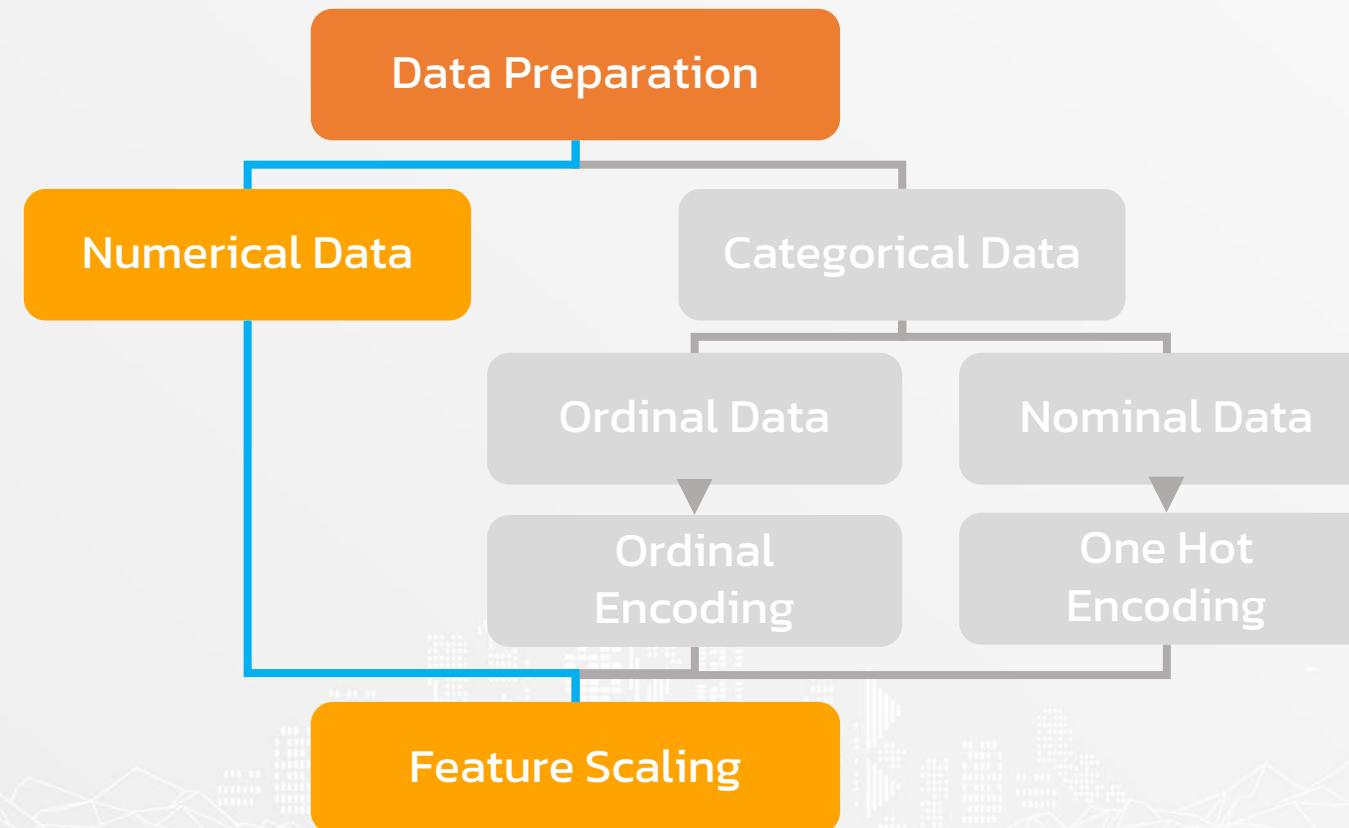
$x_1$	$x_2$	$x_3$	...	$x_{3072}$
0.0	0.0	0.0	...	0.0
0.0	0.0	0.0	...	0.0
0.0	0.0	0.0	...	0.0
:	:	:	:	:
255.0	255.0	255.0	...	255.0

 $X$ 

$y$
freshapples
freshapples
freshapples
:
rottenoranges

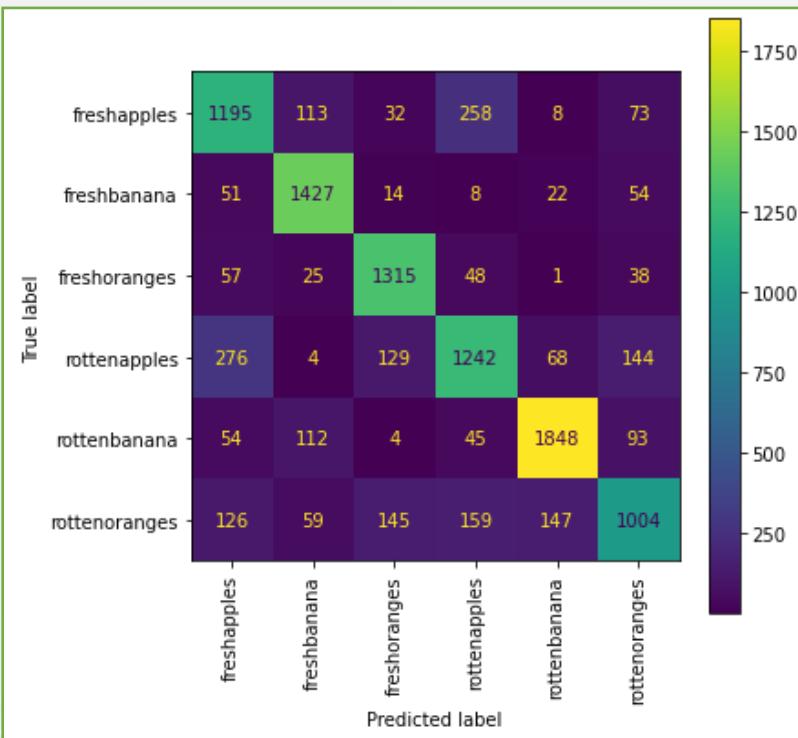
 $y$

# Data Preparation

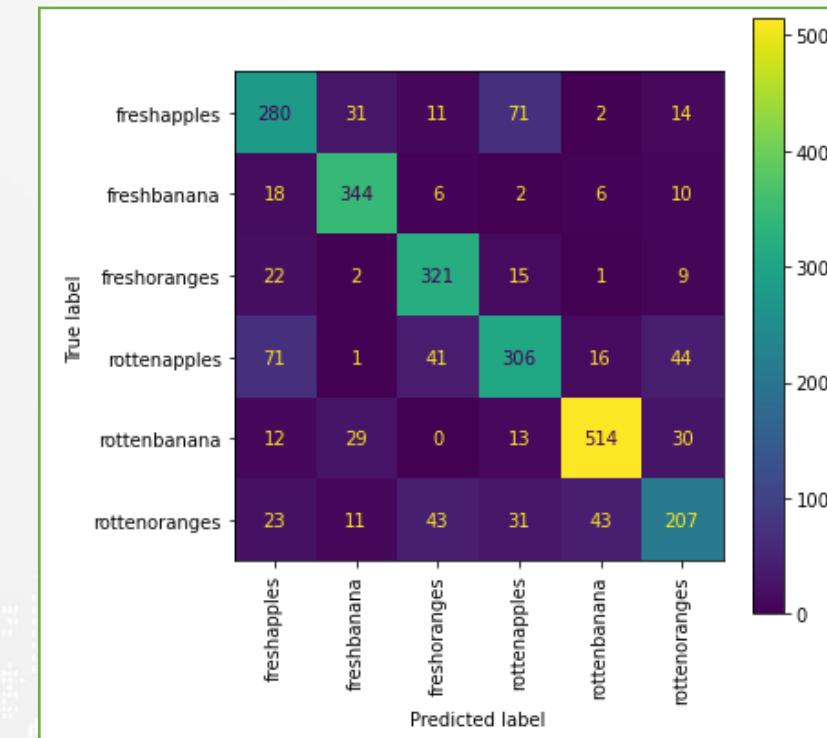


# Model Evaluation

- Confusion Matrix for **training set**



- Confusion Matrix for **test set**



# Code

- Confusion Matrix for **training set**

```
1 fig, ax = plt.subplots(figsize=(6, 6))
2 plot_confusion_matrix(clf, X_train_scaled, y_train, ax=ax)
3 plt.xticks(rotation=90)
4 plt.show()
```

- Confusion Matrix for **test set**

```
1 fig, ax = plt.subplots(figsize=(6, 6))
2 plot_confusion_matrix(clf, X_test_scaled, y_test, ax=ax)
3 plt.xticks(rotation=90)
4 plt.show()
```

# Model Evaluation

- Scoring for **training set**

	precision	recall	f1-score	support
<b>freshapples</b>	0.679363	0.711733	0.695172	1679.00000
<b>freshbanana</b>	0.820115	0.905457	0.860676	1576.00000
<b>freshoranges</b>	0.802318	0.886119	0.842139	1484.00000
<b>rottenapples</b>	0.705682	0.666667	0.685620	1863.00000
<b>rottenbanana</b>	0.882521	0.857143	0.869647	2156.00000
<b>rottenoranges</b>	0.714083	0.612195	0.659225	1640.00000
<b>accuracy</b>	0.772360	0.772360	0.772360	0.77236
<b>macro avg</b>	0.767347	0.773219	0.768746	10398.00000
<b>weighted avg</b>	0.770561	0.772360	0.770028	10398.00000

# Code

- Scoring for **training set**

```
1 report = classification_report(y_train, y_pred_train, output_dict=True)
```

```
1 print('accuracy =', report['accuracy'])
```

```
1 pd.DataFrame.from_dict(report).T
```

# Model Evaluation

- Scoring for **test set**

	precision	recall	f1-score	support
<b>freshapples</b>	0.657277	0.684597	0.670659	409.000000
<b>freshbanana</b>	0.822967	0.891192	0.855721	386.000000
<b>freshoranges</b>	0.760664	0.867568	0.810606	370.000000
<b>rottenapples</b>	0.698630	0.638831	0.667394	479.000000
<b>rottenbanana</b>	0.883162	0.859532	0.871186	598.000000
<b>rottenoranges</b>	0.659236	0.578212	0.616071	358.000000
<b>accuracy</b>	0.758462	0.758462	0.758462	0.758462
<b>macro avg</b>	0.746989	0.753322	0.748606	2600.000000
<b>weighted avg</b>	0.756430	0.758462	0.756053	2600.000000

# Code

- Scoring for **test set**

```
1 report = classification_report(y_test, y_pred_test, output_dict=True)
```

```
1 print('accuracy =', report['accuracy'])
```

```
1 pd.DataFrame.from_dict(report).T
```



## 05. FRUIT INDUSTRY





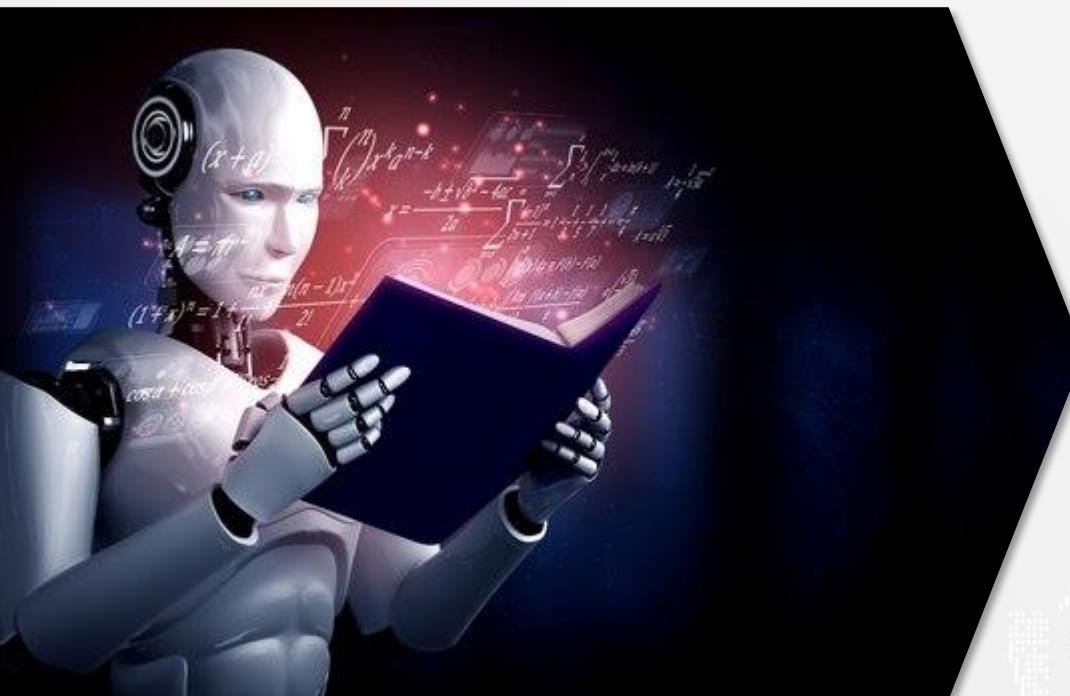
# AI in News Categorization

# Abstract

สร้าง model เพื่อจำแนกหมวดหมู่ข่าว โดยพิจารณาจากหัวข้อข่าว และเนื้อหาข่าว



# Why this project important?



- ◆ สามารถสร้างระบบจำแนกหมวดหมู่ข่าว
- ◆ สามารถสร้างระบบคัดกรองข่าวเฉพาะหมวดหมู่ที่สนใจ
- ◆ สามารถนำไปต่อยอดสร้างระบบสรุปใจสำคัญจากข่าว/บทความ/หนังสือ

# Who this project is for?

- ◆ บุคลากรด้านสื่อสารมวลชน
- ◆ นักภาษาศาสตร์
- ◆ นักวิเคราะห์ข้อมูล



# News Dataset



<https://github.com/PyThaiNLP/prachathai-67k>

# News Dataset

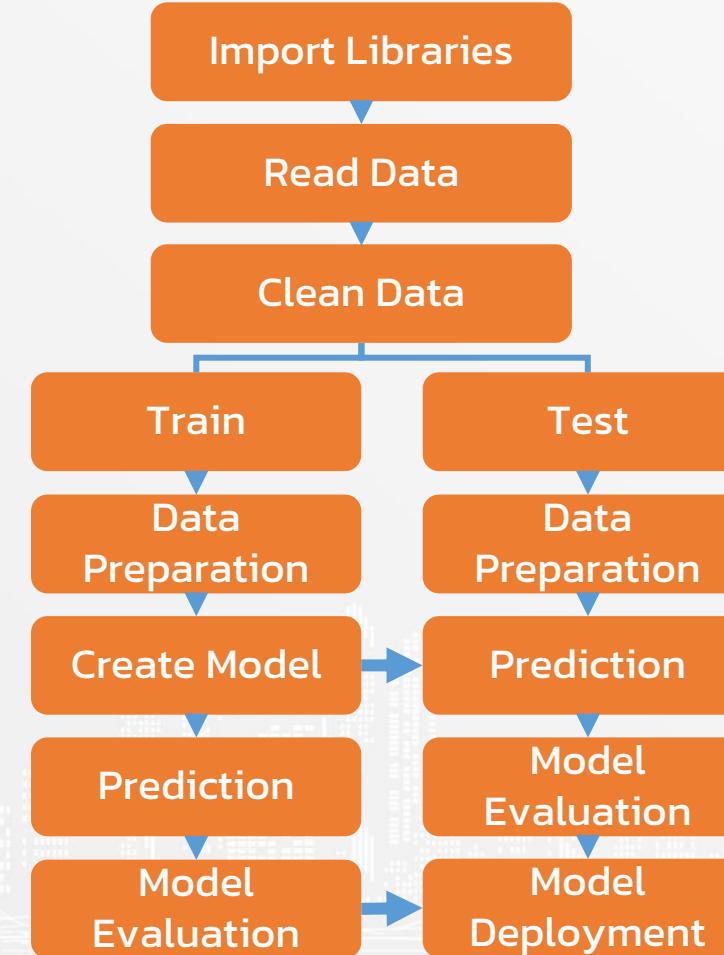
## Feature

- title : หัวข้อข่าว
- body\_text : เนื้อหาข่าว

## Target

- target : politics, human\_right, quality\_of\_life, international, social, environment, economy, culture, labor, stability, ict, education

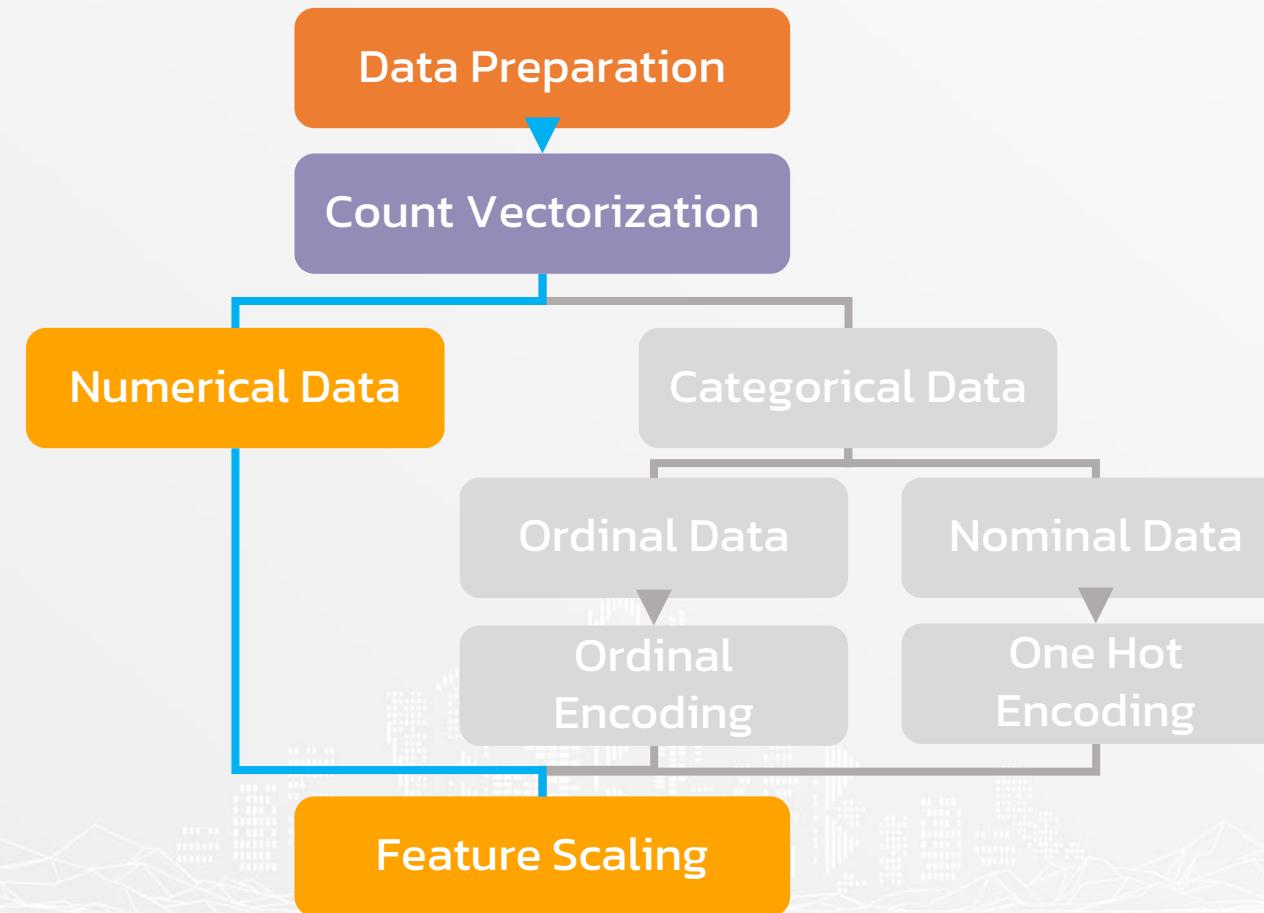
# What we learn from this project?



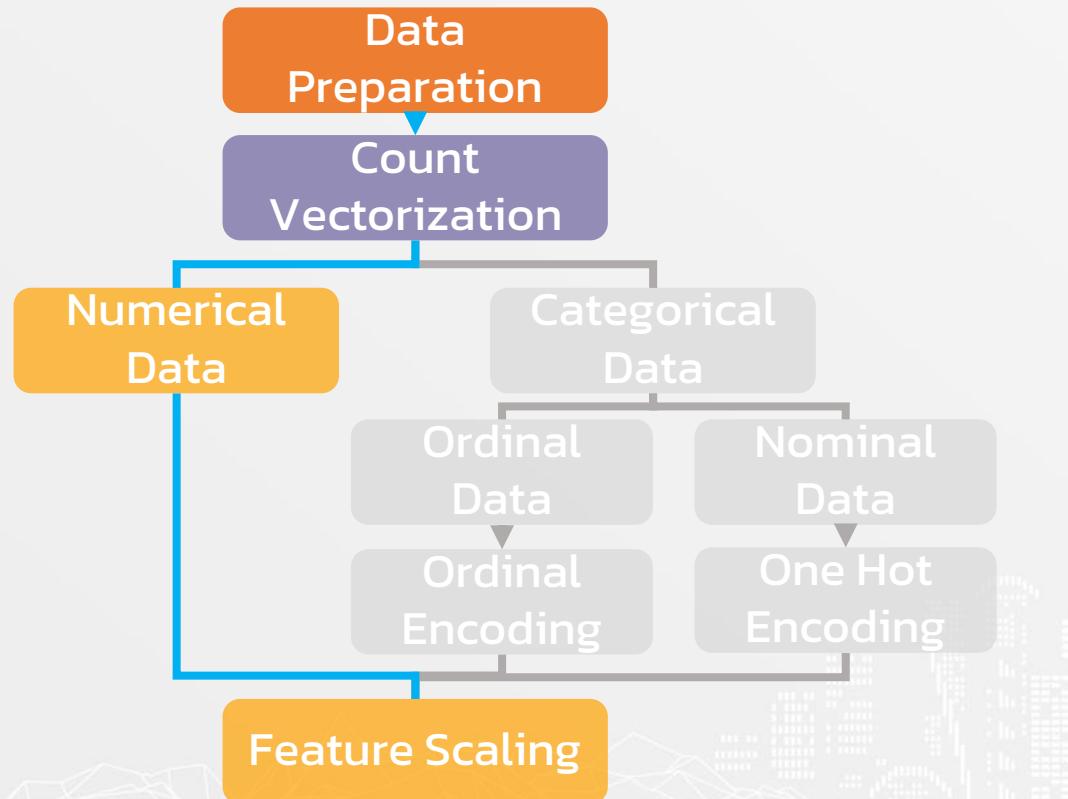
# Import Libraries

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 from sklearn.model_selection import train_test_split
6 from sklearn.feature_extraction.text import CountVectorizer
7 from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder, StandardScaler, MinMaxScaler
8 from sklearn.linear_model import LogisticRegression, LogisticRegressionCV
9 from sklearn.metrics import plot_confusion_matrix, classification_report
10
11 import warnings
12 warnings.filterwarnings('ignore')
13
14 np.random.seed(12345)
```

# Data Preparation



# Data Preparation



## Count vectorization

สร้าง feature ใหม่ โดยการหา unique word จากข้อความทั้งหมดใน dataset จากนั้นให้พิจารณาว่าแต่ละข้อความประกอบด้วย unique word อะไรบ้าง และจำนวนกี่ครั้ง

	'ก'	'เข'	'ด'	'ม'	'ส'	'สั'	'แอ'
'แอปเปิลมีสี แดง'	0	0	1	1	1	0	1
'ก็ว'-มีสีเขียว'	1	1	0	1	1	0	0
'ส้มมีสีส้ม'	0	0	0	1	1	2	0

# Count Vectorization

	cnt_title_-	cnt_title_-	cnt_title_-!	cnt_title_-"	...	cnt_title_-ໃຂນ	cnt_title_-ໃຂວ	...
เพรະເຄຍປຸກຮະດມເອາໄວ້ ຈນກຳໃໝ່ເກີດຄວາມເກລີຍດ ໜັງ...	1	0	0	2	...	0	0	...
ສຖານການຄົນສມນຕີ : ສາມາຄມສ່ວນເສີມວັດນຮຣມ ແລກກັນພູ...	1	0	0	0	...	0	0	...
ກຽງໂຮມ, 17 ພຸດທະນາ – ໜັງກາພບໜ້າວໜ້າວອົຕາລີກໍ່ໄ...	2	0	0	0	...	0	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

# Code

```
1 # tokenize thai word
2 from pythainlp.tokenize import syllable_tokenize
```

# Code

- Count vectorization for **training set**

```
1 corpus_train = X_train['title'].tolist()
2 title_vectorizer = CountVectorizer(max_features=1000, tokenizer=syllable_tokenize)
3 title_vectorizer.fit(corpus_train)
4 title_cnt_vec_train = title_vectorizer.transform(corpus_train).toarray()
```

```
1 title_cnt_vec_feature_name = [
2     'cnt_title_' + feature for feature in title_vectorizer.get_feature_names()
3 ]
```

```
1 X_train[title_cnt_vec_feature_name] = title_cnt_vec_train
2 X_train.drop('title', axis=1, inplace=True)
```

# Code

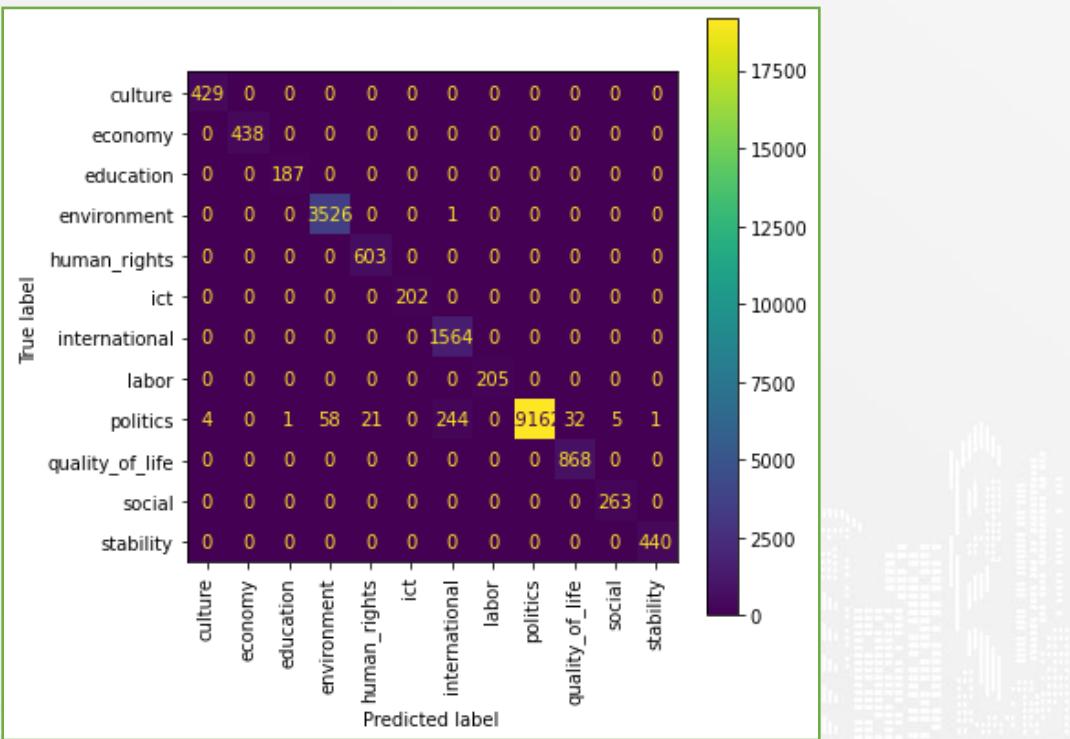
- Count vectorization for **test set**

```
1 corpus_test = X_test['title'].tolist()  
2 title_cnt_vec_test = title_vectorizer.transform(corpus_test).toarray()
```

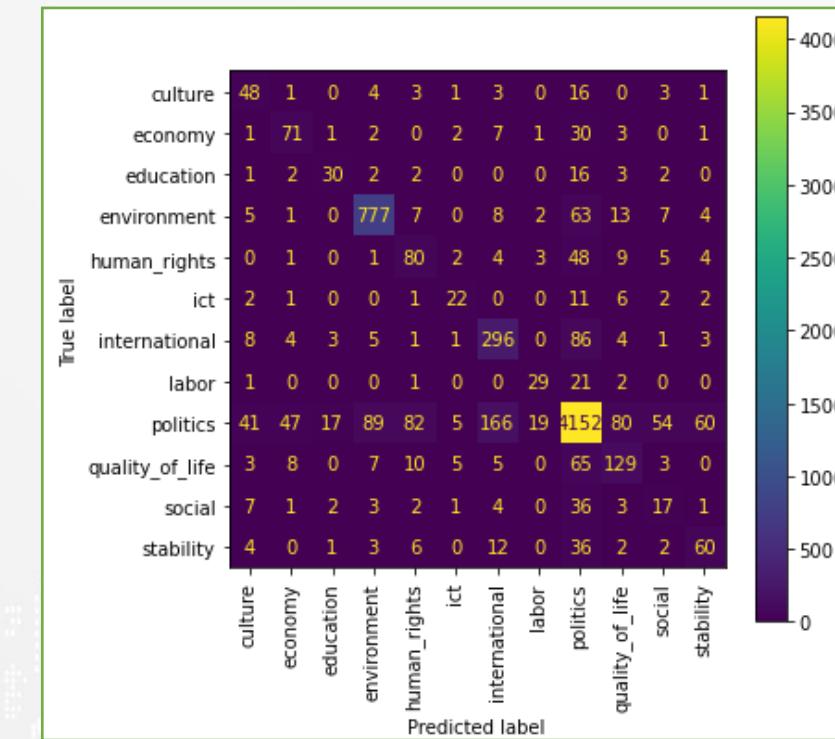
```
1 X_test[title_cnt_vec_feature_name] = title_cnt_vec_test  
2 X_test.drop('title', axis=1, inplace=True)
```

# Model Evaluation

- Confusion Matrix for **training set**



- Confusion Matrix for **test set**



# Code

- Confusion Matrix for **training set**

```
1 fig, ax = plt.subplots(figsize=(6, 6))
2 plot_confusion_matrix(clf, X_train_scaled, y_train, ax=ax)
3 plt.xticks(rotation=90)
4 plt.show()
```

- Confusion Matrix for **test set**

```
1 fig, ax = plt.subplots(figsize=(6, 6))
2 plot_confusion_matrix(clf, X_test_scaled, y_test, ax=ax)
3 plt.xticks(rotation=90)
4 plt.show()
```

# Model Evaluation

- Scoring for **training set**

	precision	recall	f1-score	support
<b>culture</b>	0.990762	1.000000	0.995360	429.000000
<b>economy</b>	1.000000	1.000000	1.000000	438.000000
<b>education</b>	0.994681	1.000000	0.997333	187.000000
<b>environment</b>	0.983817	0.999716	0.991703	3527.000000
<b>human_rights</b>	0.966346	1.000000	0.982885	603.000000
<b>ict</b>	1.000000	1.000000	1.000000	202.000000
<b>international</b>	0.864566	1.000000	0.927364	1564.000000
<b>labor</b>	1.000000	1.000000	1.000000	205.000000

	precision	recall	f1-score	support
<b>politics</b>	1.000000	0.981258	0.990540	19528.000000
<b>quality_of_life</b>	0.964444	1.000000	0.981900	868.000000
<b>social</b>	0.981343	1.000000	0.990584	263.000000
<b>stability</b>	0.997732	1.000000	0.998865	440.000000
<b>accuracy</b>	0.987011	0.987011	0.987011	0.987011
<b>macro avg</b>	0.978641	0.998415	0.988045	28254.000000
<b>weighted avg</b>	0.988288	0.987011	0.987291	28254.000000

# Code

- Scoring for **training set**

```
1 report = classification_report(y_train, y_pred_train, output_dict=True)
```

```
1 print('accuracy =', report['accuracy'])
```

```
1 pd.DataFrame.from_dict(report).T
```

# Model Evaluation

- Scoring for **test set**

	precision	recall	f1-score	support
<b>culture</b>	0.396694	0.600000	0.477612	80.000000
<b>economy</b>	0.518248	0.596639	0.554688	119.000000
<b>education</b>	0.555556	0.517241	0.535714	58.000000
<b>environment</b>	0.870101	0.875986	0.873034	887.000000
<b>human_rights</b>	0.410256	0.509554	0.454545	157.000000
<b>ict</b>	0.564103	0.468085	0.511628	47.000000
<b>international</b>	0.586139	0.718447	0.645583	412.000000
<b>labor</b>	0.537037	0.537037	0.537037	54.000000

	precision	recall	f1-score	support
<b>politics</b>	0.906550	0.862843	0.884157	4812.000000
<b>quality_of_life</b>	0.507874	0.548936	0.527607	235.000000
<b>social</b>	0.177083	0.220779	0.196532	77.000000
<b>stability</b>	0.441176	0.476190	0.458015	126.000000
<b>accuracy</b>	0.808465	0.808465	0.808465	0.808465
<b>macro avg</b>	0.539235	0.577645	0.554679	7064.000000
<b>weighted avg</b>	0.822440	0.808465	0.814192	7064.000000

# Code

- Scoring for **test set**

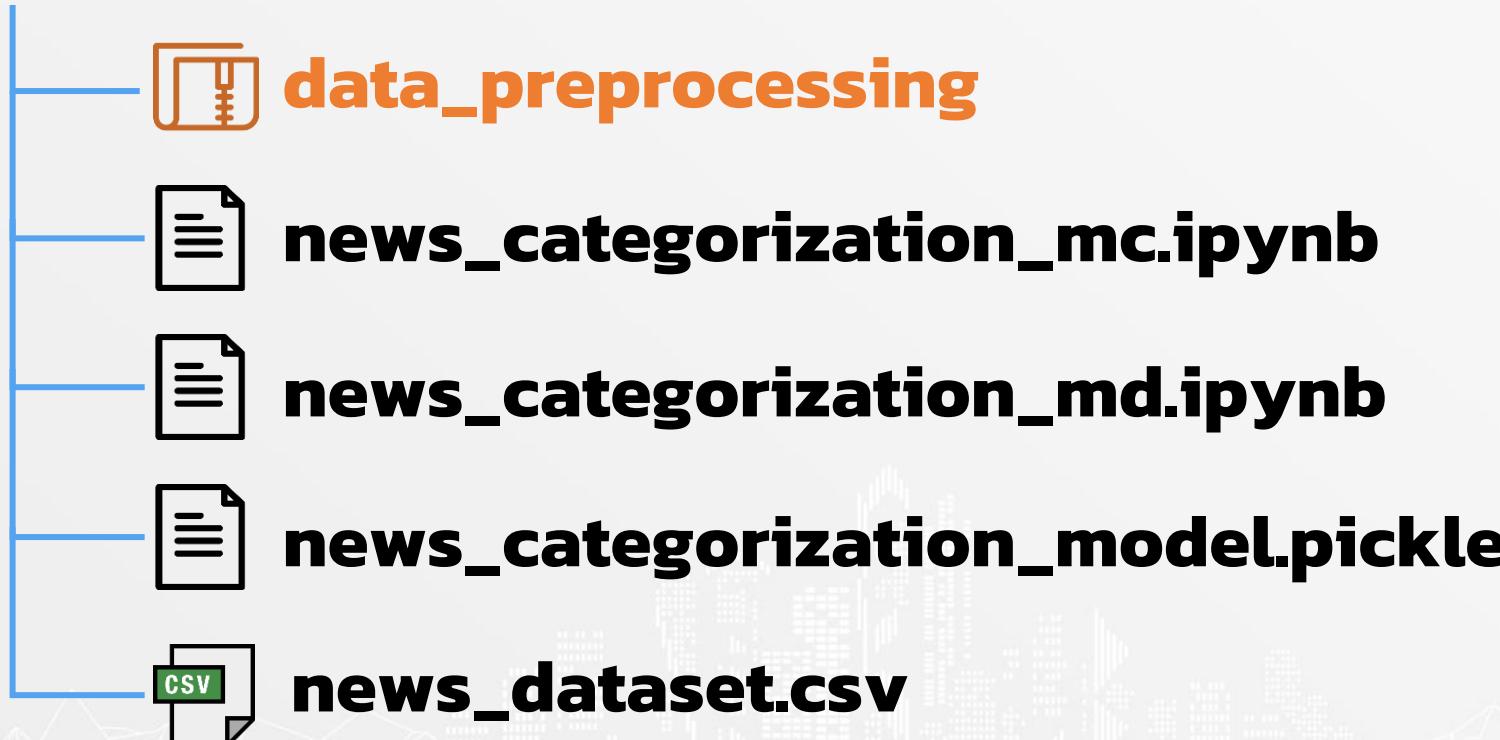
```
1 report = classification_report(y_test, y_pred_test, output_dict=True)
```

```
1 print('accuracy =', report['accuracy'])
```

```
1 pd.DataFrame.from_dict(report).T
```



## 06. NEWS CATEGORIZATION



# DL102 : Logistic Regression



# THANK YOU !

*We hope you enjoy our course*