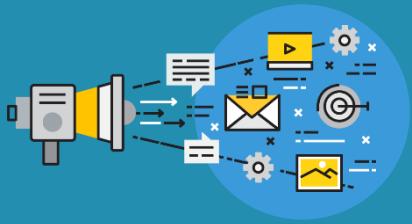


Algorithms and Data Analysis

-演算法與資料分析-

Machine Learning II

授課教師：張珀銀 老師



# Outline

- 
- ① 機器學習簡介
  - ② 機器學習類型
  - ③ 資料集的切割方法
  - ④ 機器學習模型的驗證模估
  - ⑤ 支持向量機
  - ⑥ 集成學習
  - ⑦ 最近鄰居法



1

# 機器學習簡介

# 什麼是機器學習？

- 電腦從資料中進行學習的科學（和藝術）。
- 廣義概念：
- 機器學習是讓計算機具有學習的能力，無需進行明確程式設計。——亞瑟·薩繆爾，1959
- 工程性概念：
- 計算機程式利用經驗E學習任務T，性能是P，如果針對任務T的性能P隨著經驗E不斷增長，則稱為機器學習。——湯姆·米切爾，1997

# 什麼是機器學習？

例1：垃圾郵件篩檢程式

根據垃圾郵件和普通郵件學習標記垃圾郵件。

訓練集：用來進行學習的資料樣本

任務T：標記新郵件是否是垃圾郵件

經驗E：訓練資料

性能P：正確分類的比例準確率

例2：下載維基百科的網站

電腦雖然有了很多資料，但不會變聰明

這不是機器學習

# 電腦解決問題的模式

AI、機器學習方法：電腦程式仰賴資料，從中學習解決問題。

優點：一般化能力高，針對資料，從中學習、建立模型。

缺點：需足夠數量及品質的資料。

傳統方法：研究問題的特性、找出解答、然後實做解答

優點：針對問題，計算效率高

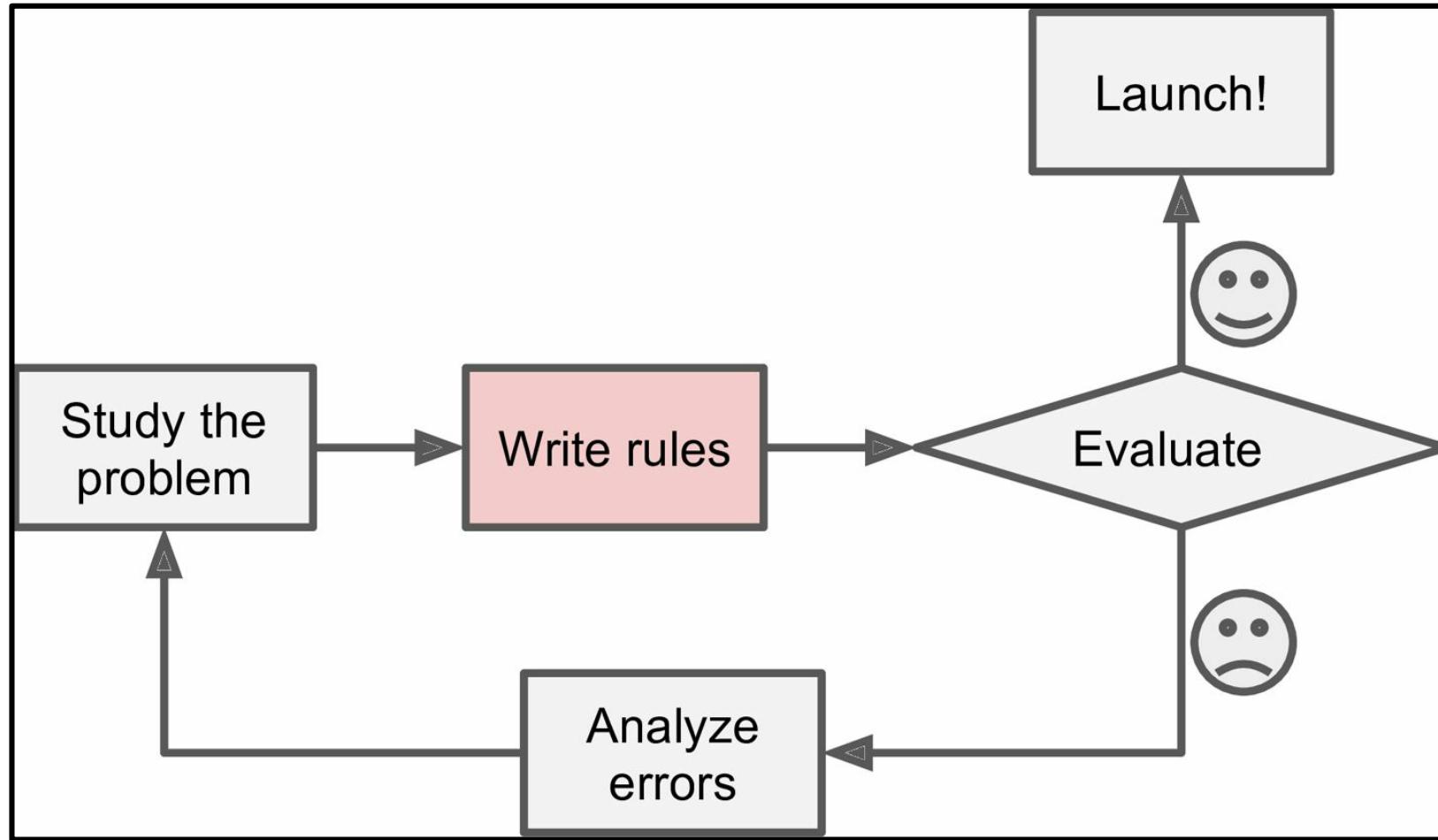
缺點：固定的解決方案無法解決各種狀況。

## 傳統方法

- 觀察垃圾郵件發現規則：
- 比如4U、credit card、free、amazing) 在郵件主題中頻繁出現
- 發件人名字、郵件正文的格式，等等。
- 為觀察到的規律寫了一個檢測算法，如果檢測到了這些規律，程序就會標記郵件為垃圾郵件。
- 測試程序，重複第1步和第2步，直到滿足要求。

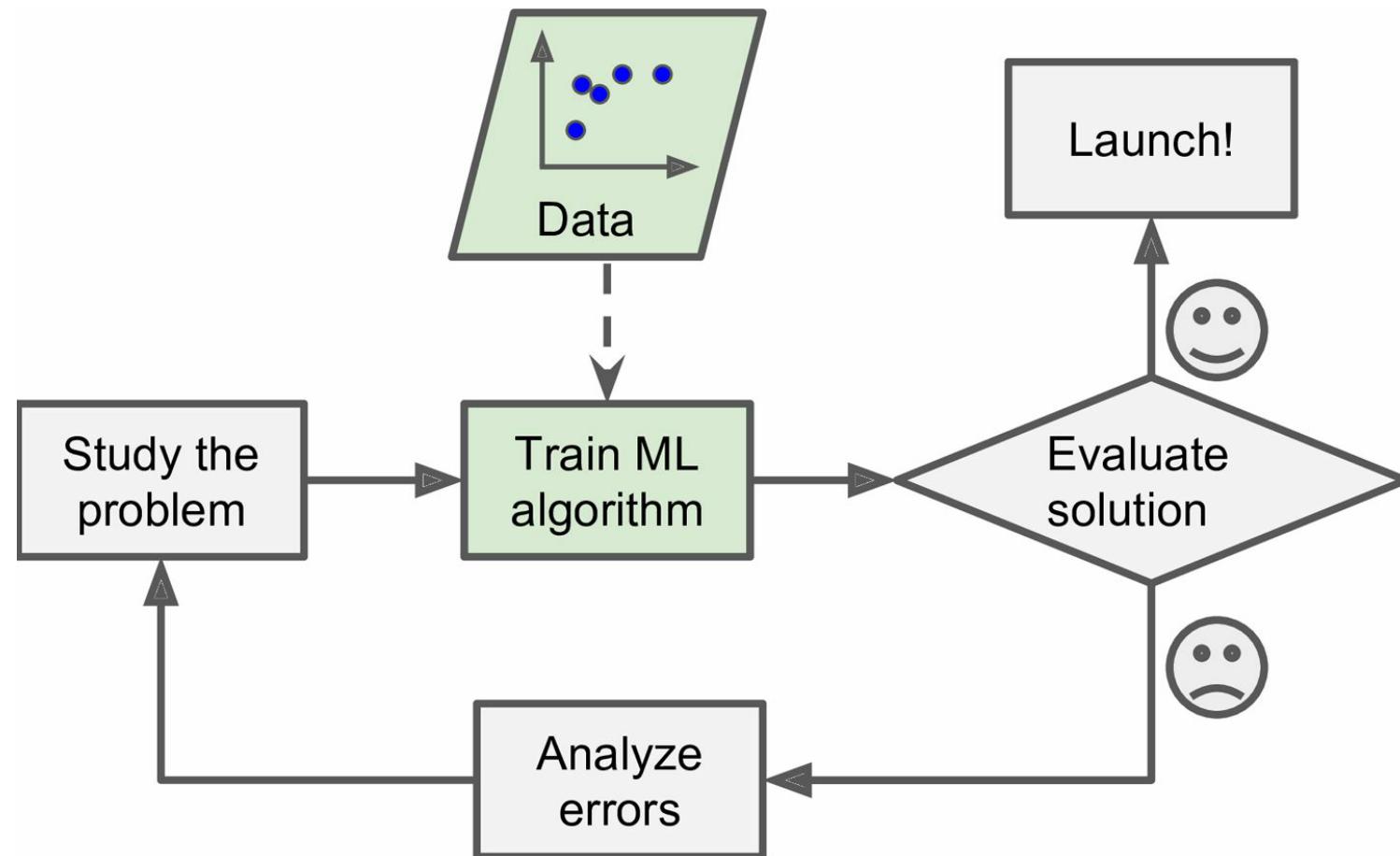
# 傳統方法

- 程式是複雜的規則，維護困難。



# 機器學習方法

- 機器學習技術自動學習哪個詞和短語是垃圾郵件的特徵，通過與普通郵件比較，檢測垃圾郵件

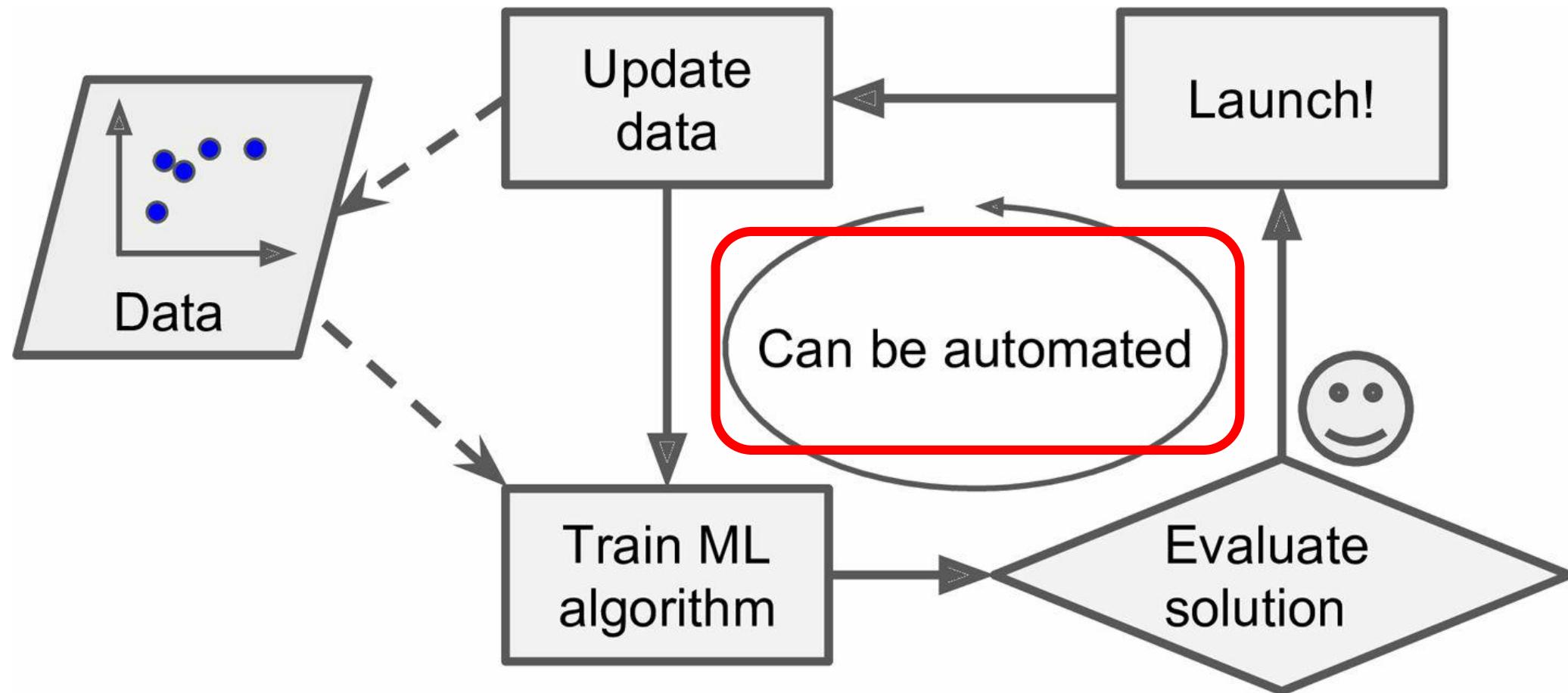


## 機器學習方法

- 傳統方法：垃圾郵件持續更改規則，就需要被動地不停地寫入新規則。
- 機器學習方法：關鍵字特徵異常頻繁性令機器學習方法發現新規則，自動標記垃圾郵件，無需干預

# 機器學習方法

- 機器學習方法可自動適應改變



## 機器學習方法另一優點

善於處理對傳統方法太複雜，或沒有演算法的問題。

例：

想寫一個可以識別語音“one”和“two”的簡單程式。

傳統方法：無法應用至嘈雜環境下的數百萬人的數千詞彙、數十種語言。

機器學習：根據大量單詞的錄音，寫一個可以自我學習的算法。

## 機器學習的強項

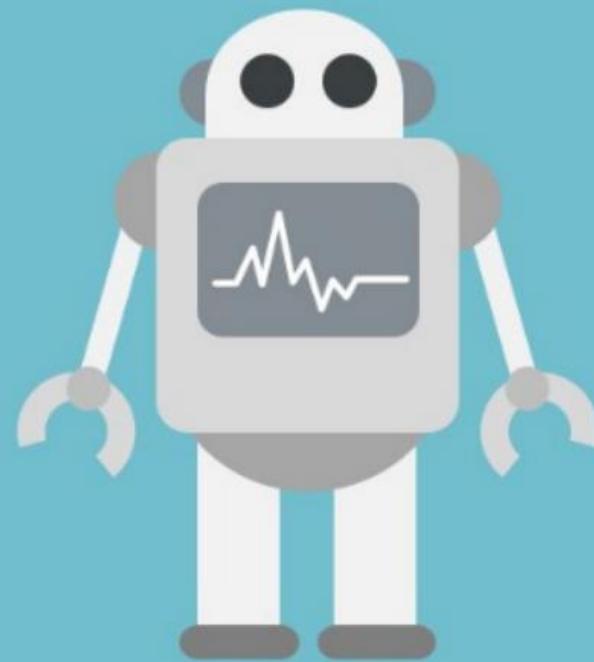
- 需要進行大量手工調整或需要擁有長串規則才能解決的問題：機器學習算法通常可以簡化代碼、提高性能。
- 問題複雜，傳統方法難以解決：最好的機器學習方法可以找到解決方案。
- 環境有波動：機器學習算法可以適應新數據。
- 洞察複雜問題和大量數據。

# 機器學習 (Machine learning)

- 機器學習 (ML) 是對演算法和統計模型的科學研究，電腦系統使用這些演算法和統計模型來執行特定任務，而無需使用明確的指令，而是依靠模式和推理。它被視為人工智慧的子集。
- 機器學習演算法基於樣本資料（稱為“訓練資料”）建立數學模型，以便進行預測或決策而無需明確地程式設計以執行任務。
- 因為在開發各種應用程式的情境中，例如電子郵件過濾和電腦視覺，難以開發出有效執行任務的規則式演算法。

# Artificial Intelligence

人工智能



1950's

# Machine Learning

機器學習



1980's

# Deep Learning

深度學習



2010's

2

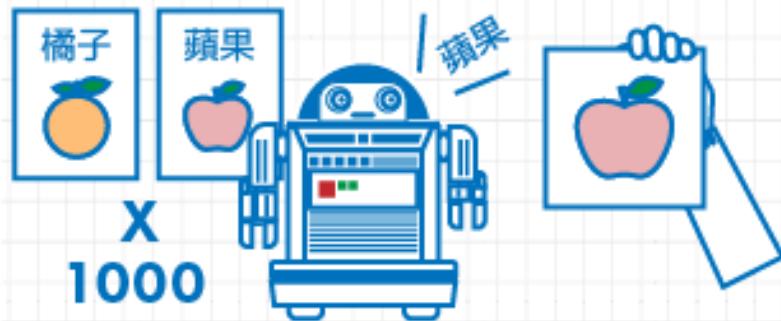
# 機器學習類型

# 機器學習類型 (1/2)

## 監督式學習

Supervised Learning

給予「有標籤」的資料

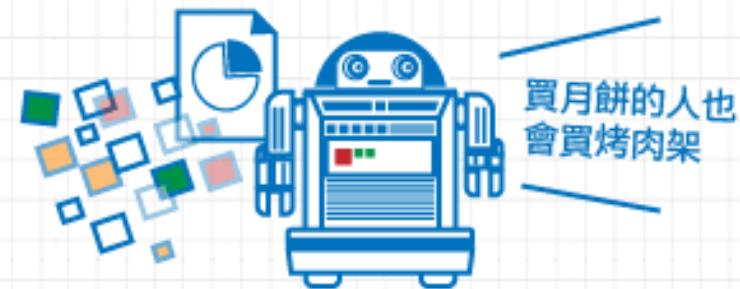


給機器各看了 1000 張標籤為蘋果和橘子的照片後、詢問機器新的一張照片中是蘋果還是橘子

## 非監督式學習

Unsupervised Learning

給予「無標籤」的資料，機器會自動中找出潛在的規則



依據資料的分布、找到資料間的相似性；機器可能找出「買月餅的人也會買烤肉架」這個關聯

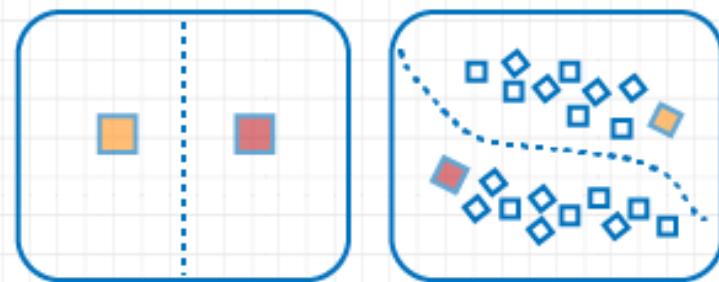
資料來源：股感知知識庫

## 機器學習類型 (2/2)

### 半監督學習

Semi-supervised learning

少部分資料有標籤，而大部分資料沒有標籤

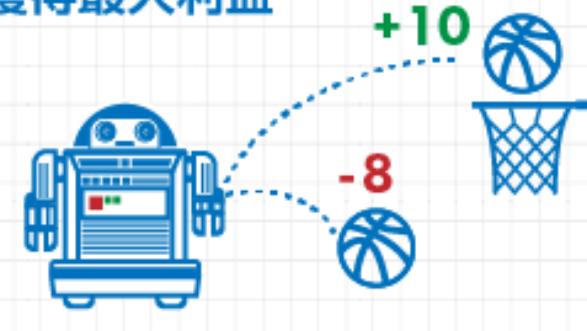


先使用有標籤過的資料先切出一條分界線，再利用剩下無標籤資料的整體分布，調整出兩大類別的新分界。如此降低標籤資料的成本

### 增強學習

reinforcement learning

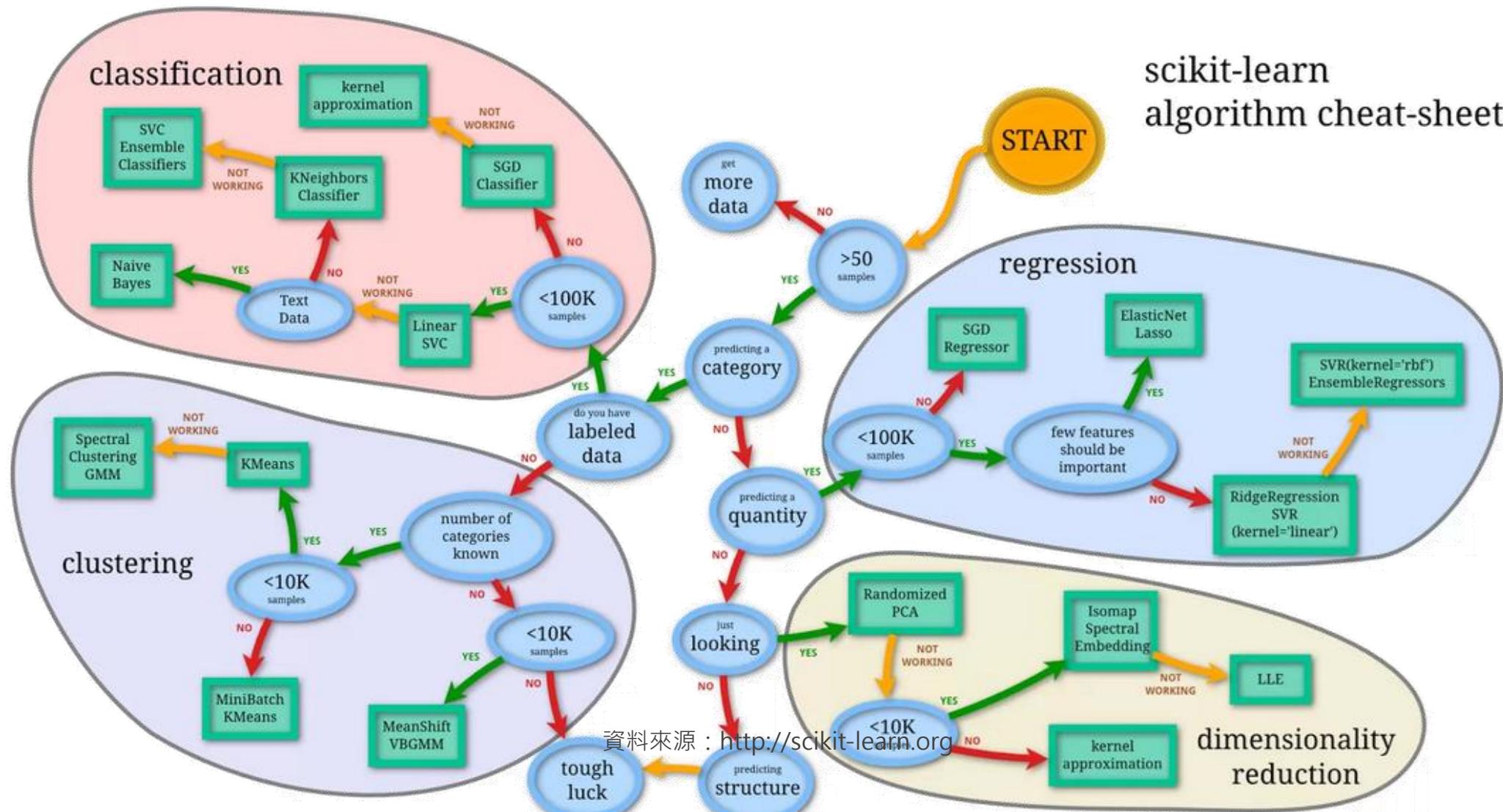
透過觀察環境而行動，並會隨時根據新進來的資料逐步修正、以獲得最大利益



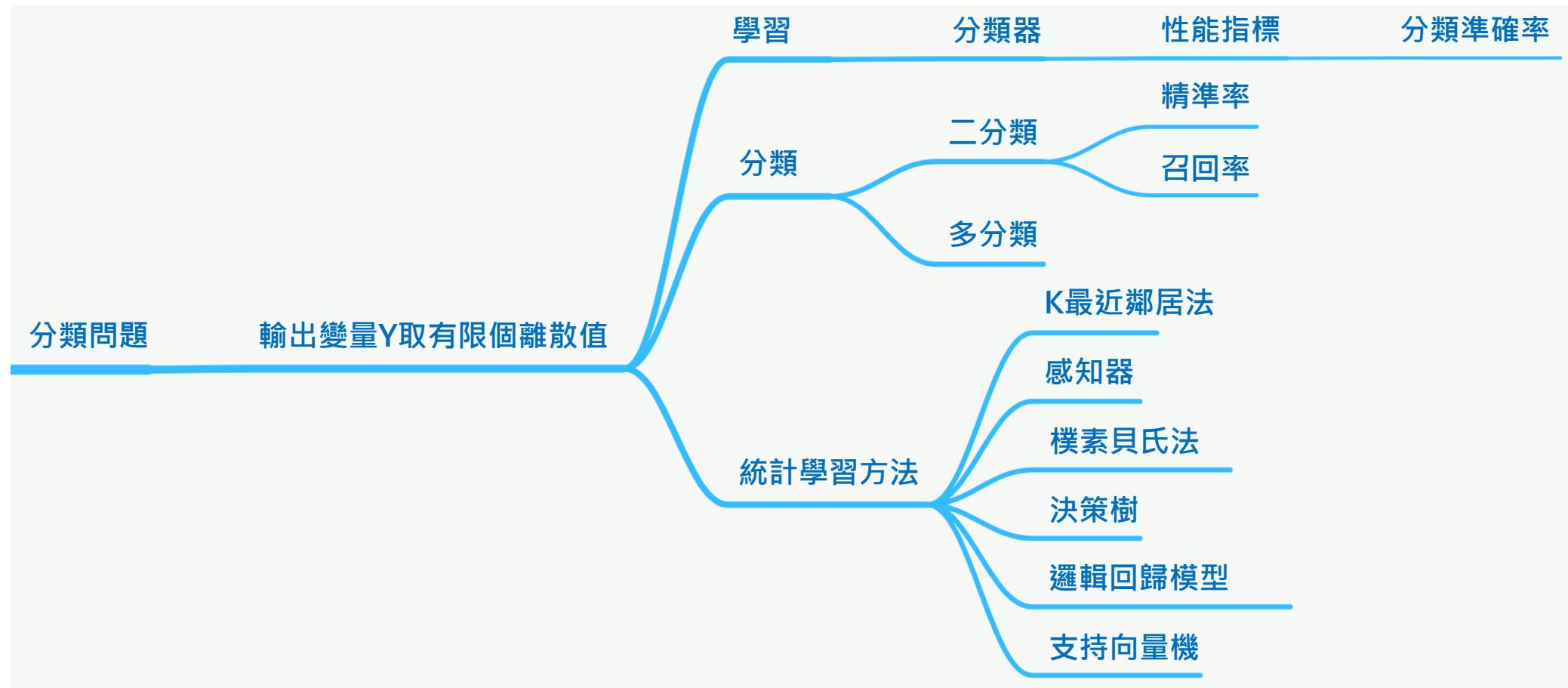
機器人投籃，根據反饋的好壞，機器會自行逐步修正、最終得到正確的結果

資料來源：股感知知識庫

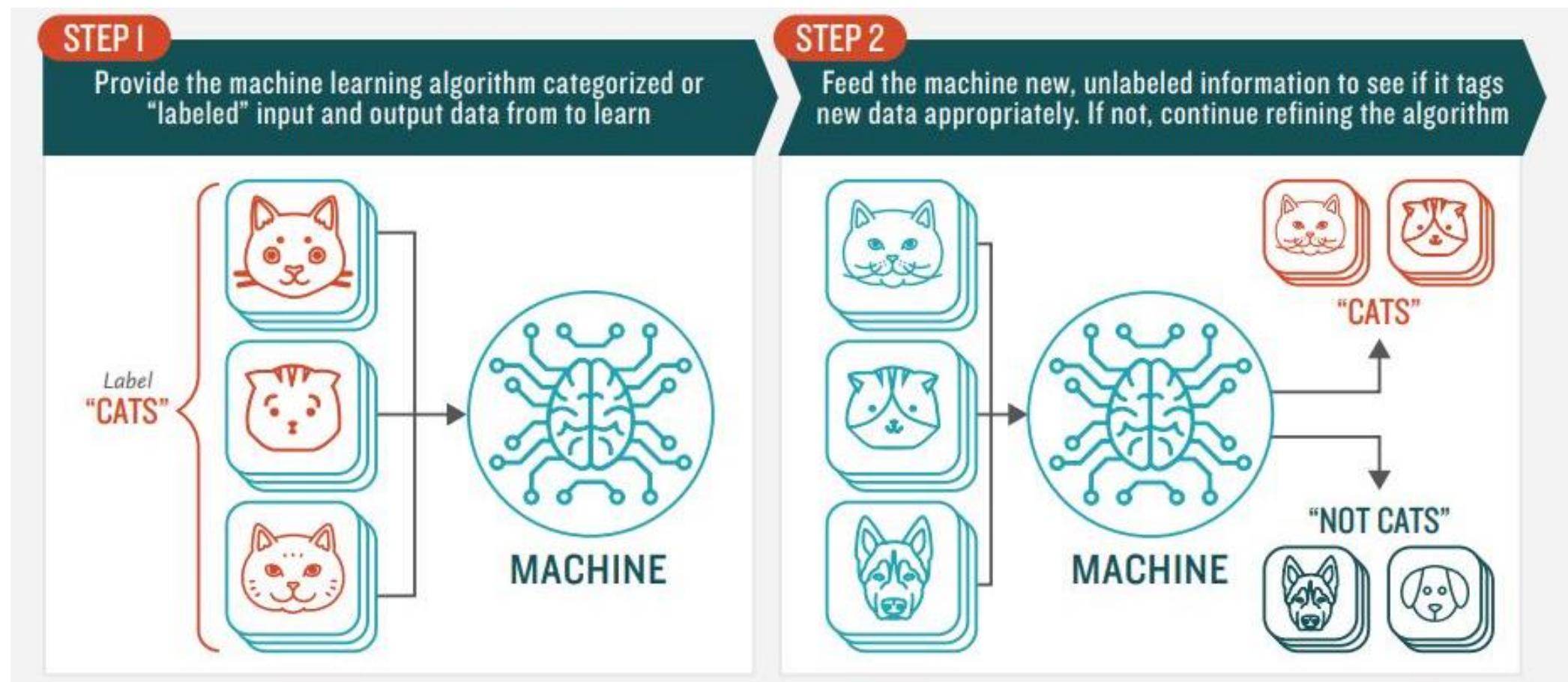
# scikit-learn toolkit 快捷列表



# 分類方法思维導圖



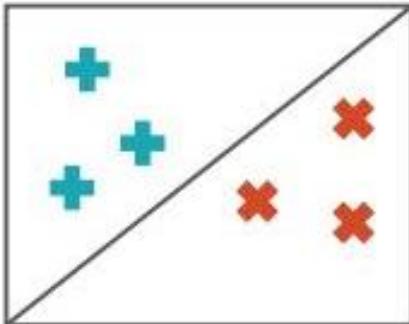
# 監督式學習概念 (1/2)



資料來源：<https://www.newtechdojo.com/list-machine-learning-algorithms/how-supervised-machine-learning-works/>

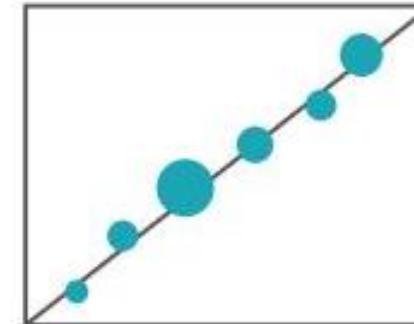
# 監督式學習概念 (2/2)

## TYPES OF PROBLEMS TO WHICH IT'S SUITED



### 類別型問題 CLASSIFICATION

Sorting items  
into categories



### 連續型問題 REGRESSION

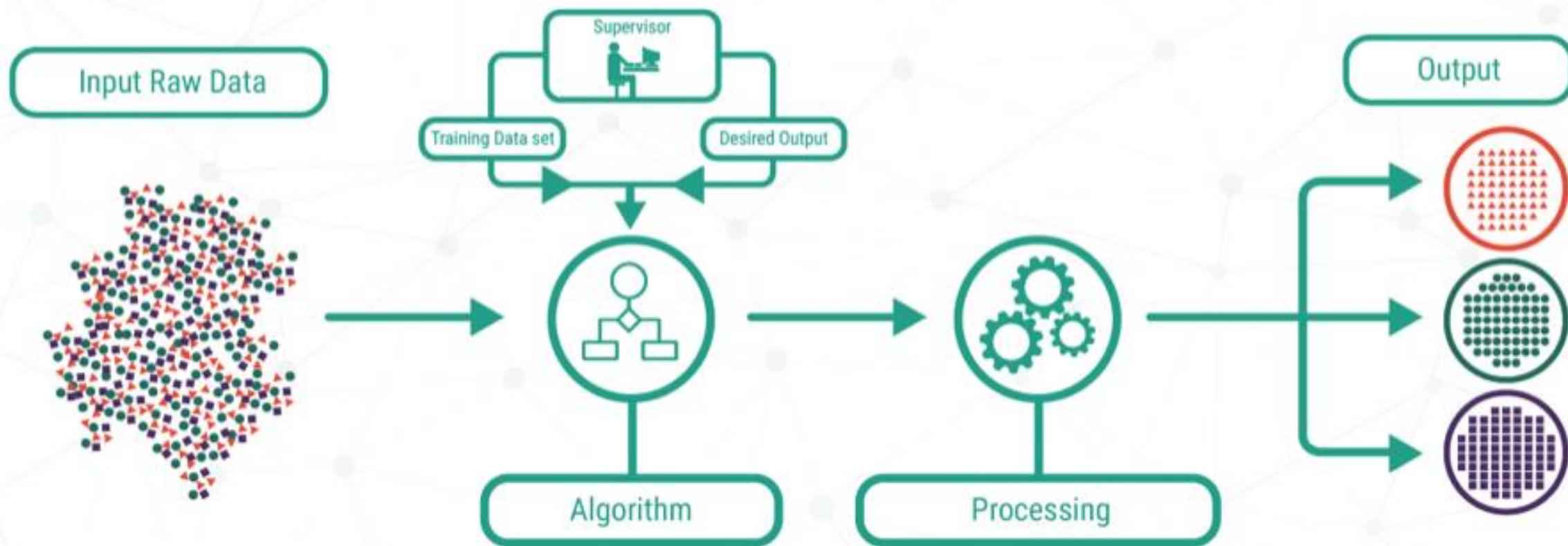
Identifying real values  
(dollars, weight, etc.)

如：貓、狗

如：明天氣溫28°C

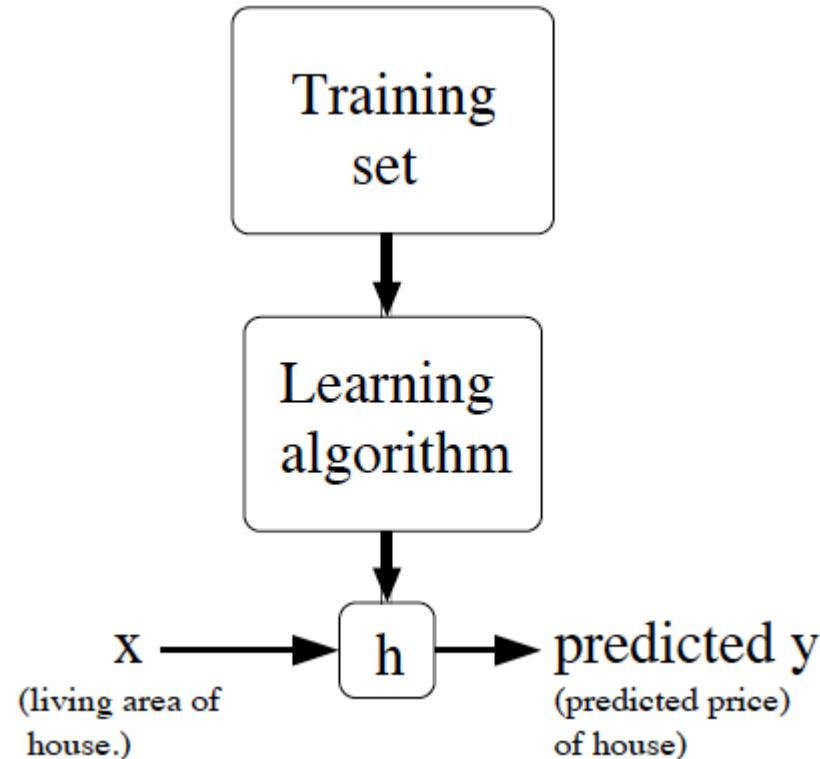
資料來源：<https://www.newtechdojo.com/list-machine-learning-algorithms/how-supervised-machine-learning-works/>

# 監督式學習流程架構



資料來源：<http://bigdata-madesimple.com/machine-learning-explained-understanding-supervised-unsupervised-and-reinforcement-learning/>

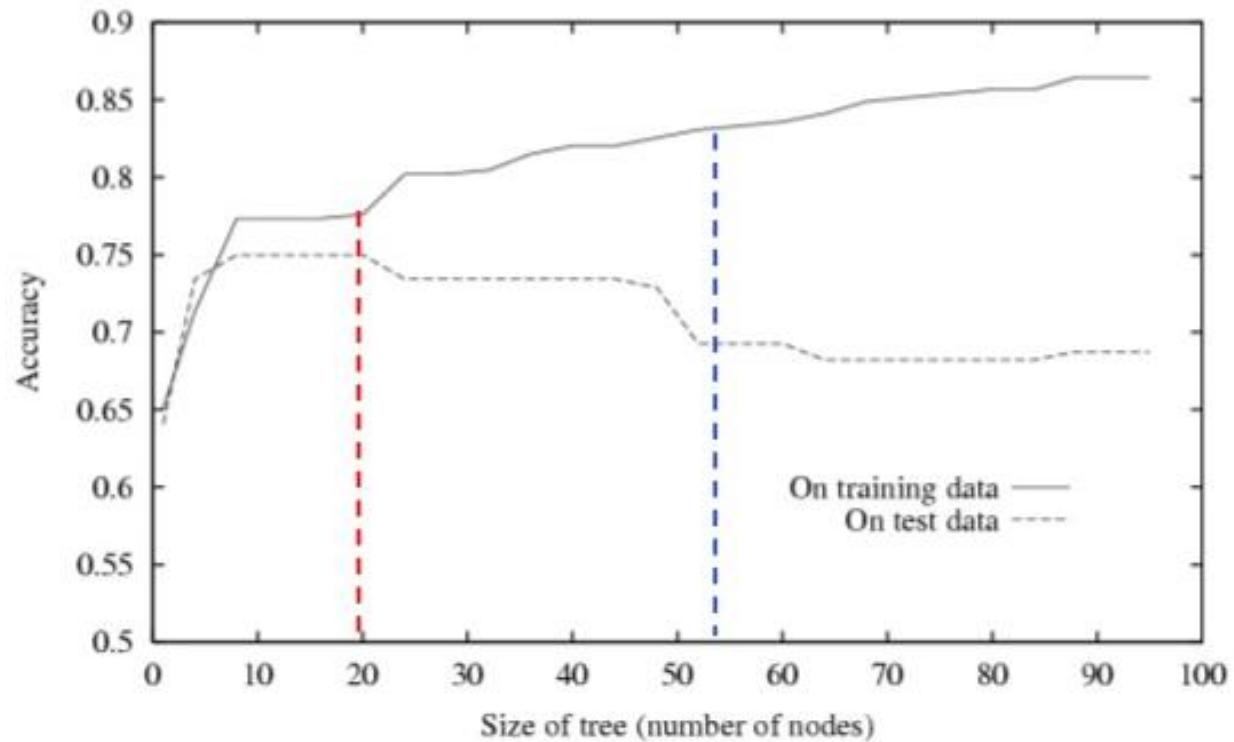
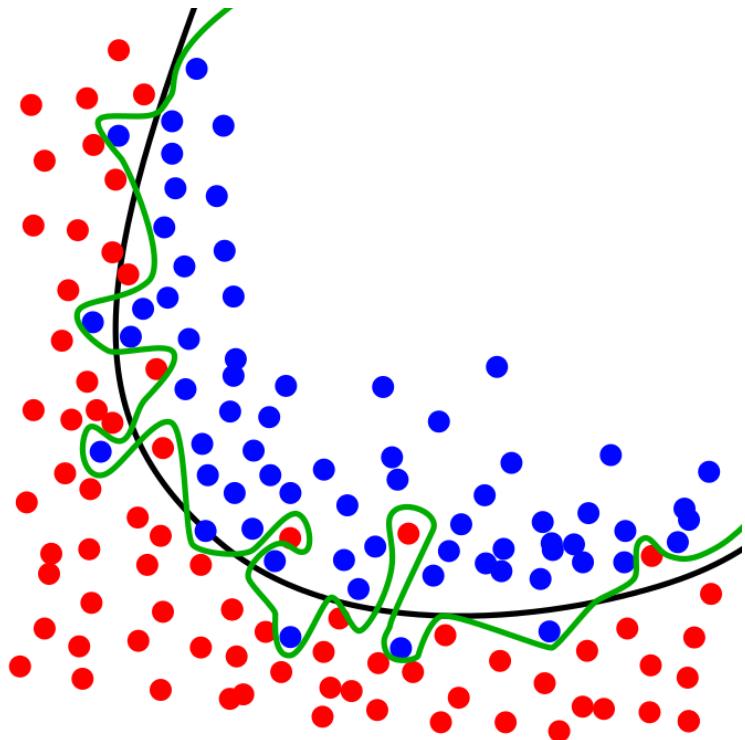
# 監督式學習數學模型



- $x^{(i)}$ ：表示輸入的資料（input variable），也稱為輸入特徵（input features），在這個例子中就是房屋大小
- $y^{(i)}$ ：表示輸出的資料（output variable），也稱為目標變數（target variable），也就是我們想要估計的東西，在這裡就是指房價
- $(x^{(i)}, y^{(i)})$ ：將一個  $x^{(i)}$  與  $y^{(i)}$  配對之後，稱為一組 training example
- $\{(x^{(i)}, y^{(i)}): i = 1, \dots, m\}$ ：所有的 training examples 合起來稱為 training set，也就是指整個用來學習的資料庫
- $X$ ：代表輸入資料的空間（space）
- $Y$ ：代表輸出資料的空間，在這個例子中， $X$  與  $Y$  都是  $\mathbb{R}$

資料來源：<http://mropengate.blogspot.tw/2015/05/ai-supervised-learning.html>

# 過度擬合 (overfitting)

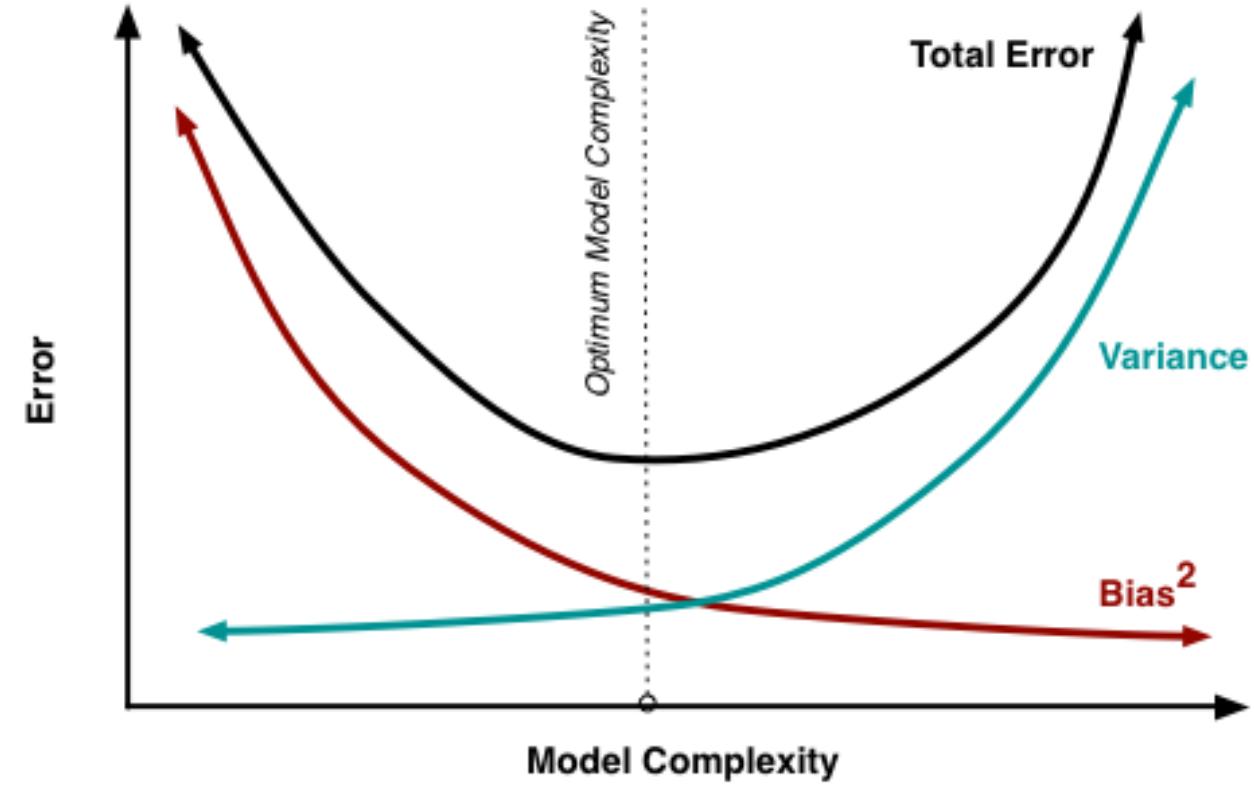
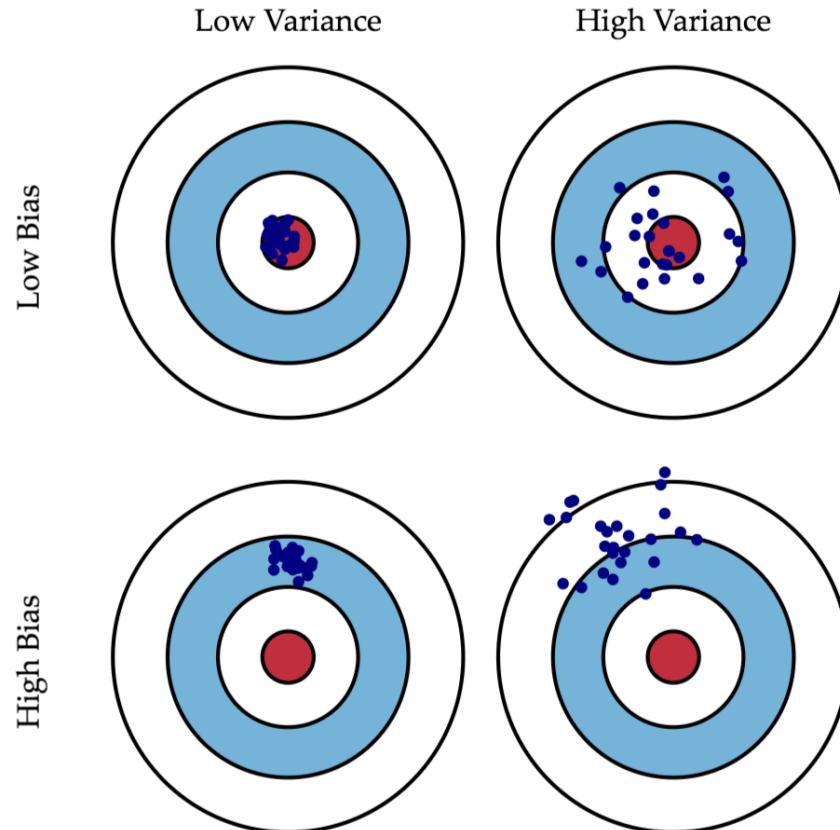


奧卡姆剃刀原則：若無必要，勿增實體

如果對於同一現象有兩種不同的假說，我們應該採取比較簡單的那一種

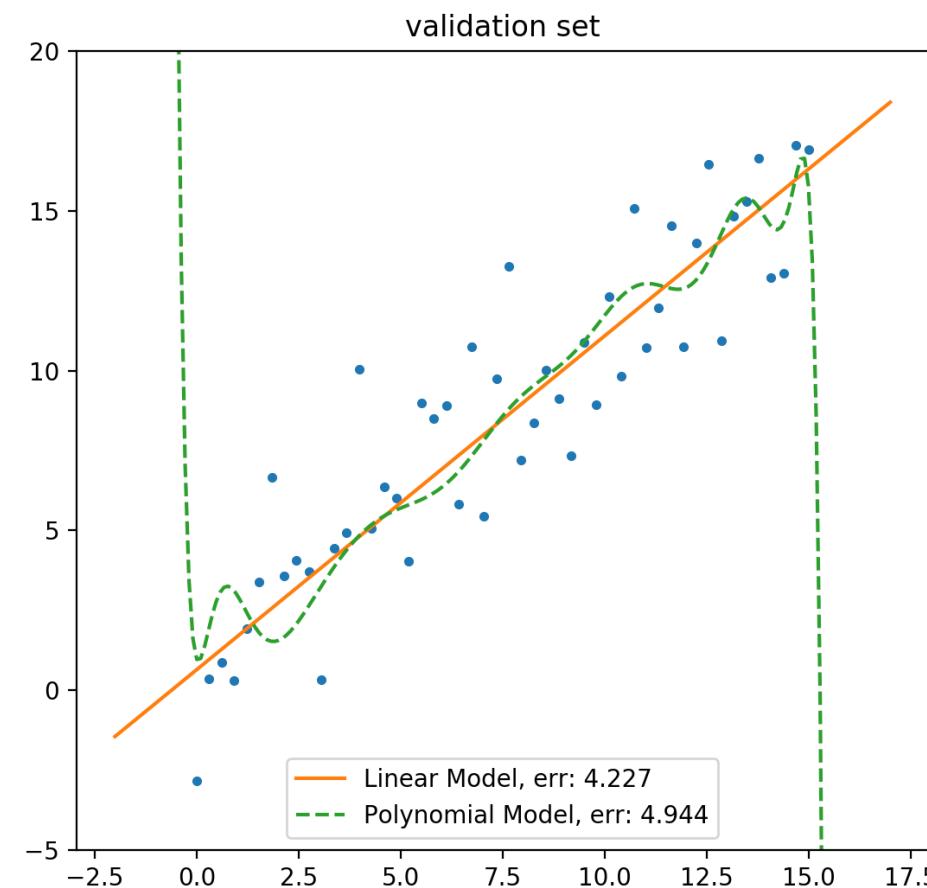
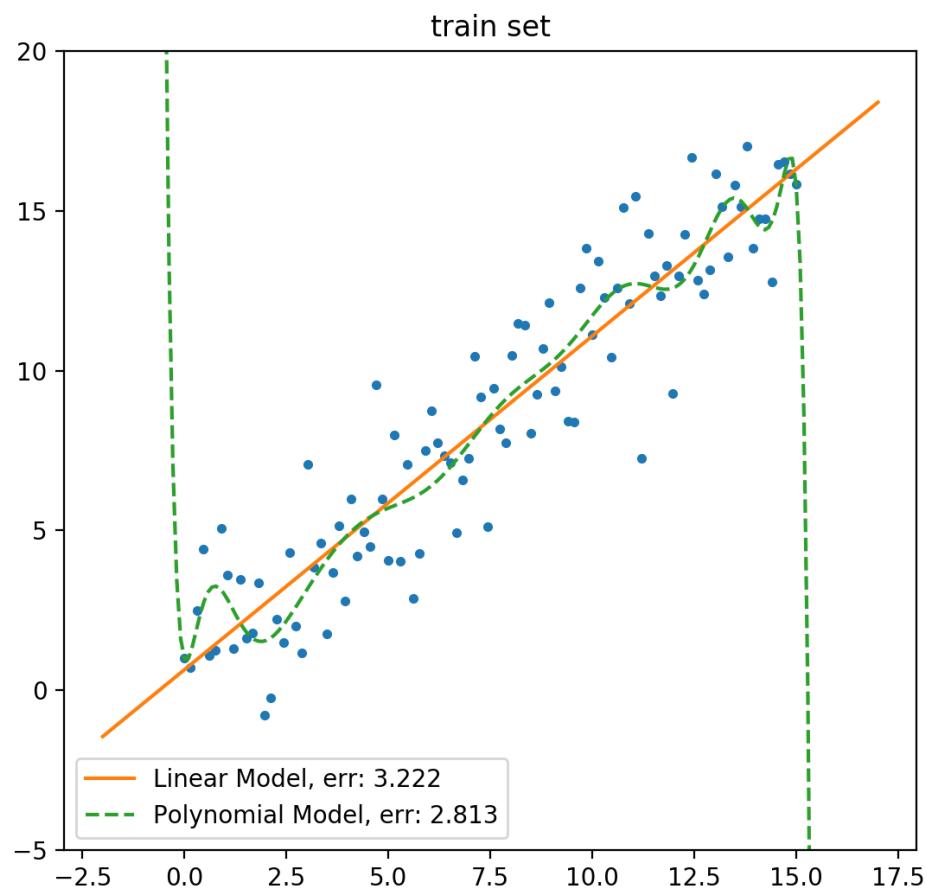
資料來源：<http://mropengate.blogspot.tw/2015/05/ai-supervised-learning.html>

# 偏差和變異之權衡 (Bias-Variance Tradeoff)



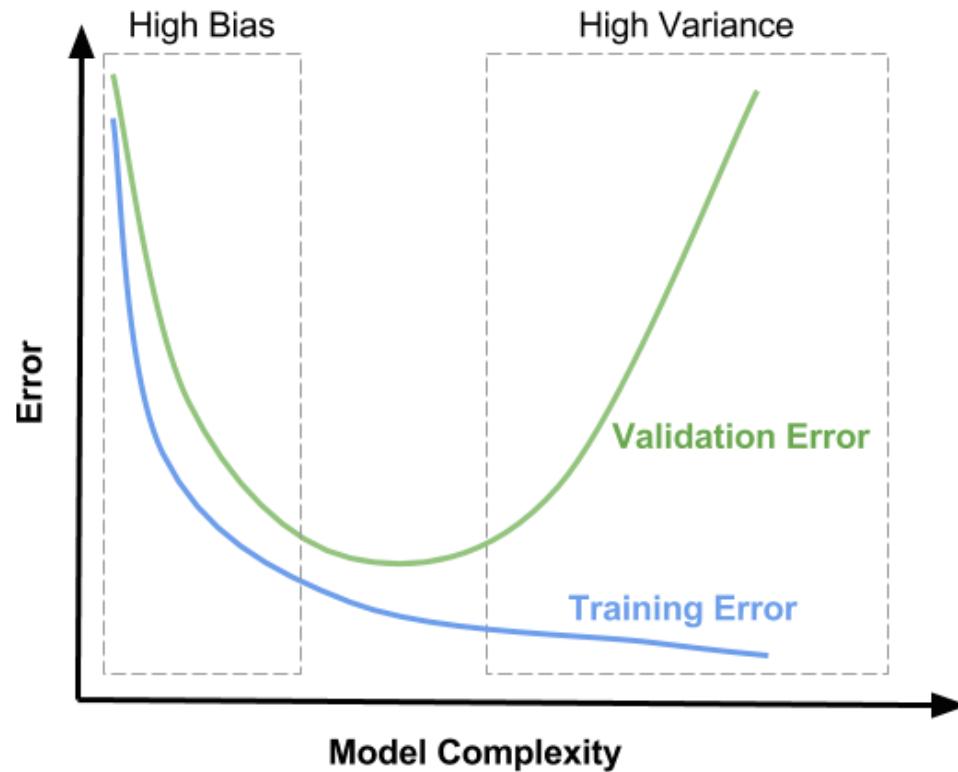
資料來源：<https://liam0205.me/2017/03/25/bias-variance-tradeoff/>

# 偏差和變異之權衡 - Example

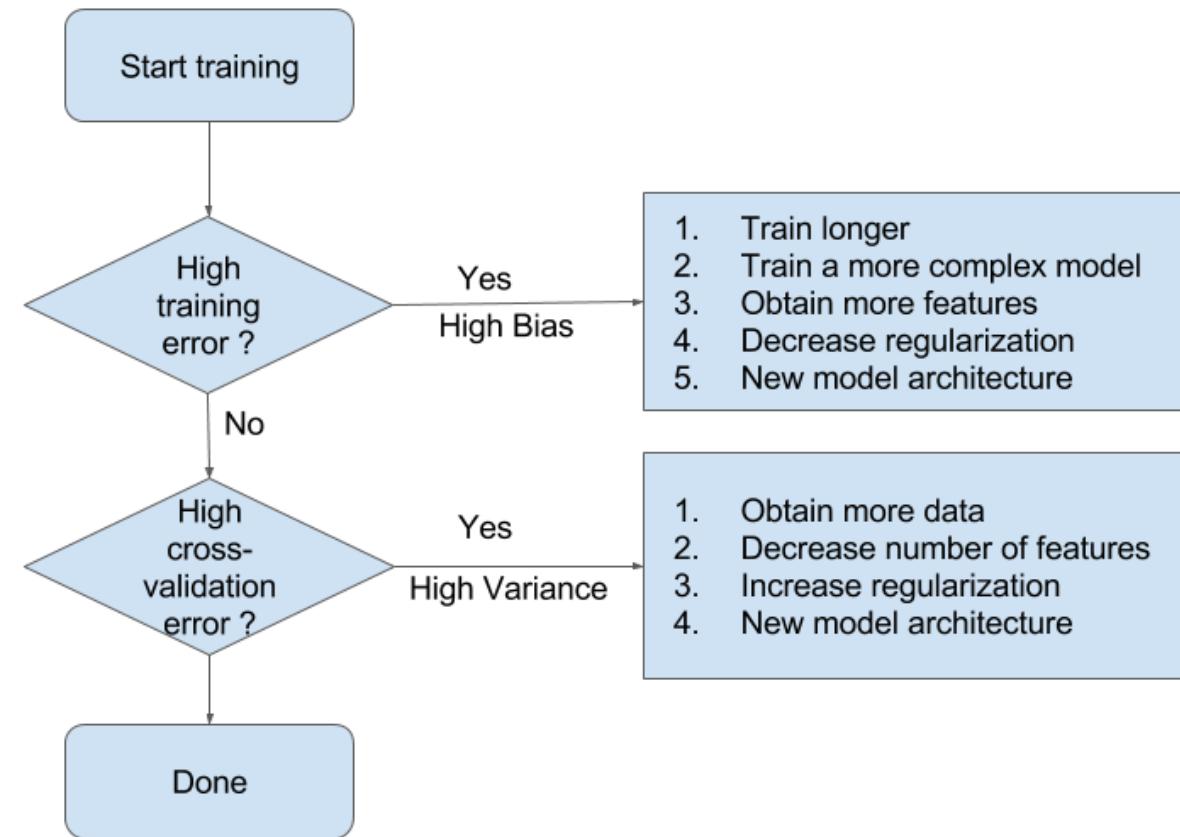


資料來源 : <https://liam0205.me/2017/03/25/bias-variance-tradeoff/>

# 偏差和變異之權衡 - 處理



過擬合與乏擬合表現



過擬合與乏擬合之處理方式

資料來源：<https://liam0205.me/2017/03/25/bias-variance-tradeoff/>

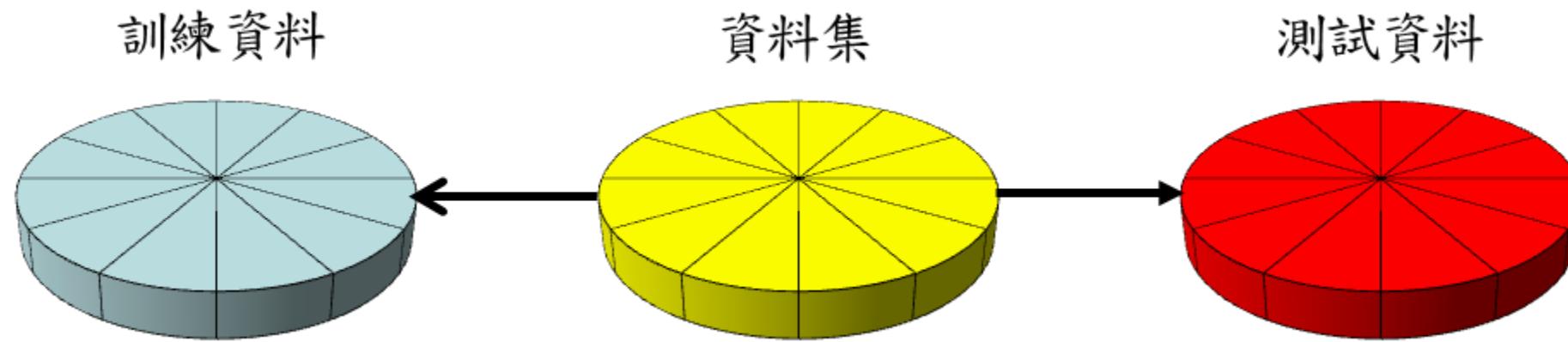
3

# 資料集的切割方法

# 資料集的測試評估種類

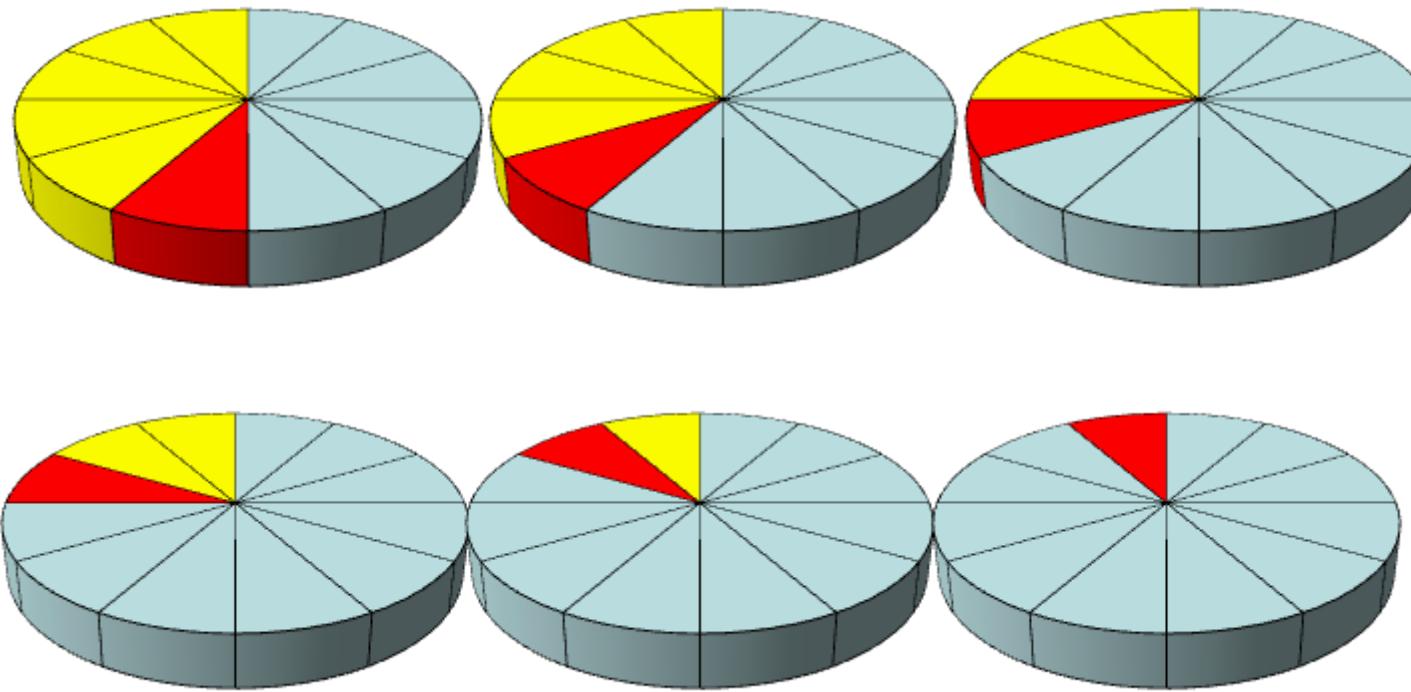
- 換置測試法 (Resubstitution)
- 漸進確認法 (Progressive Validation)
- 拔靴確認法 (Bootstrap Validation)

# 換置測試法 (Resubstitution)

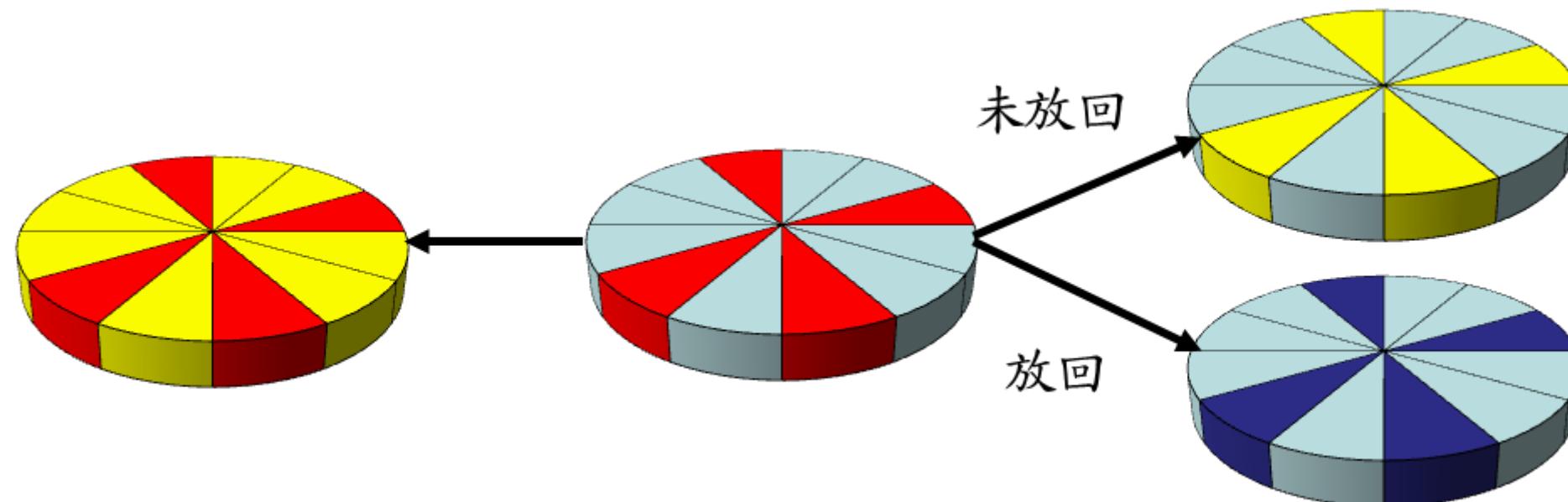


資料來源：如何確保大數據分析的品質：淺談監督式機器學習的測試評估方法

# 漸進確認法 (Progressive Validation)

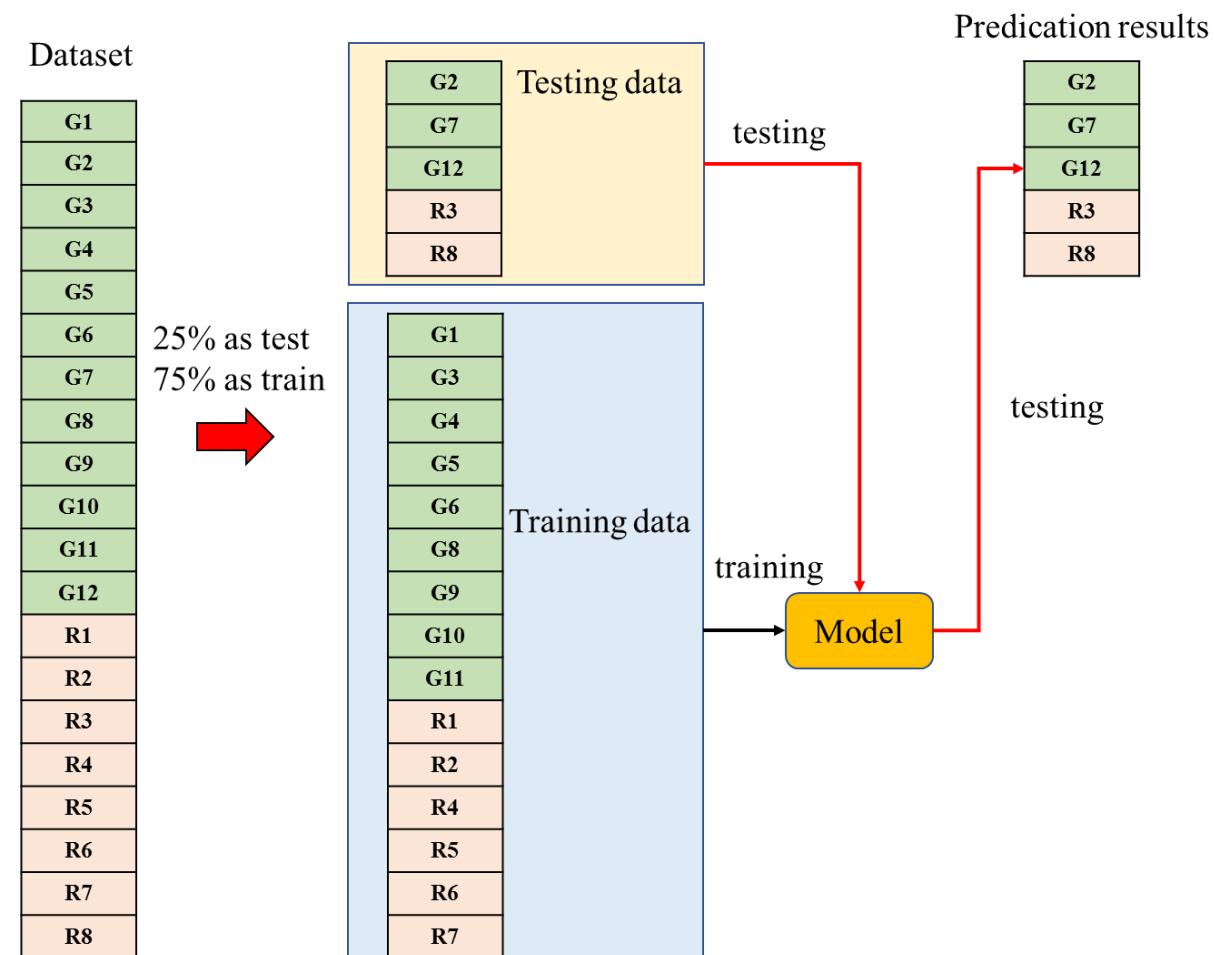


# 拔靴確認法 (Bootstrap Validation)



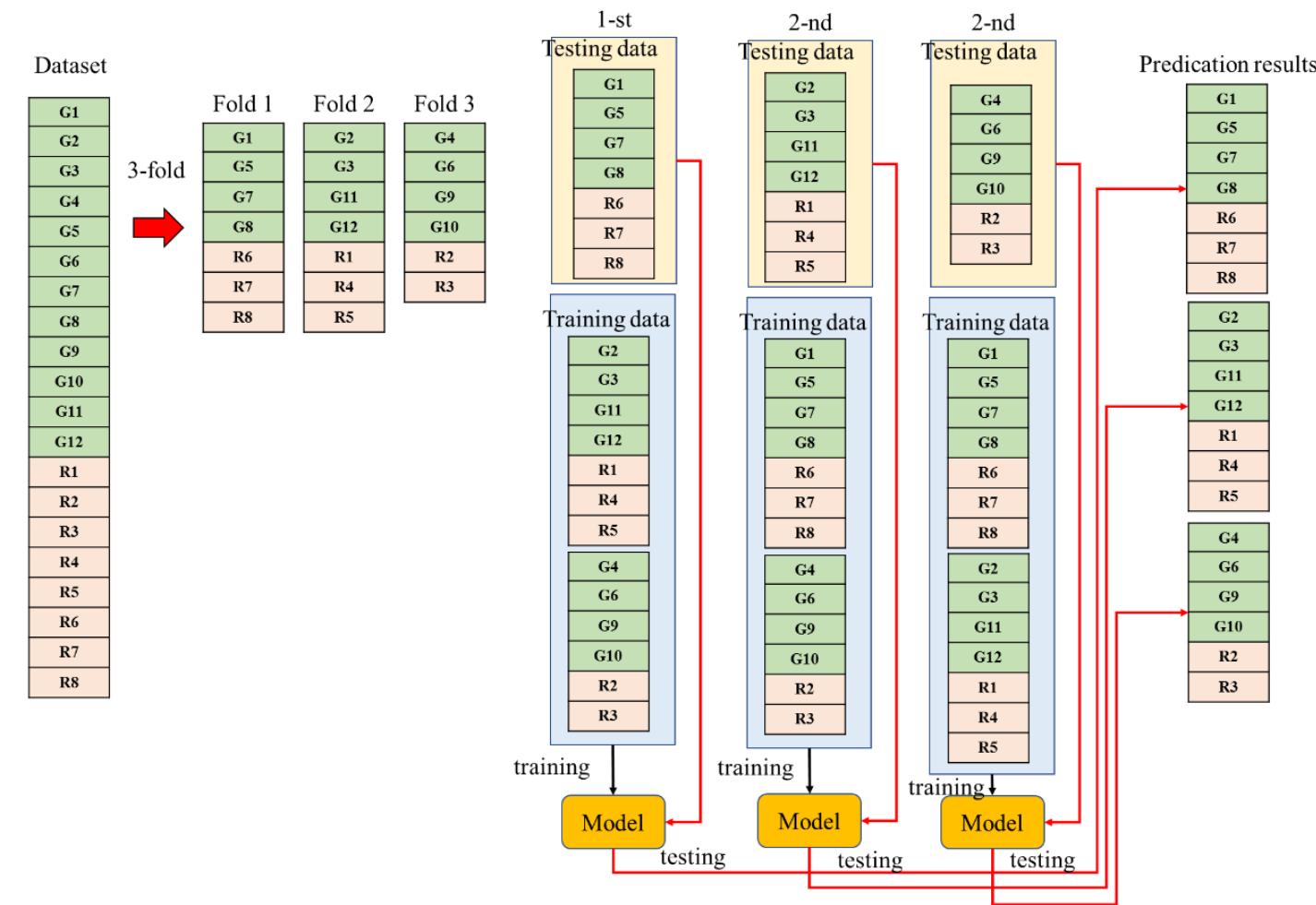
# 截留確認法 (Holdout Validation)

- 從資料集中隨機取得 $p\%$ 資料當作「訓練資料(Training data)」和剩下的 $(1-p)\%$ 當做「測試資料(Testing data)」。



# 交叉確認法 (Cross Validation)

- 將資料隨機平均分成 $k$ 個集合，然後將某一個集合當做「測試資料 (Testing data)」，剩下的 $k-1$ 個集合做為「訓練資料 (Training data)」，如此重複進行直到每一個集合都被當做「測試資料(Testing data)」為止

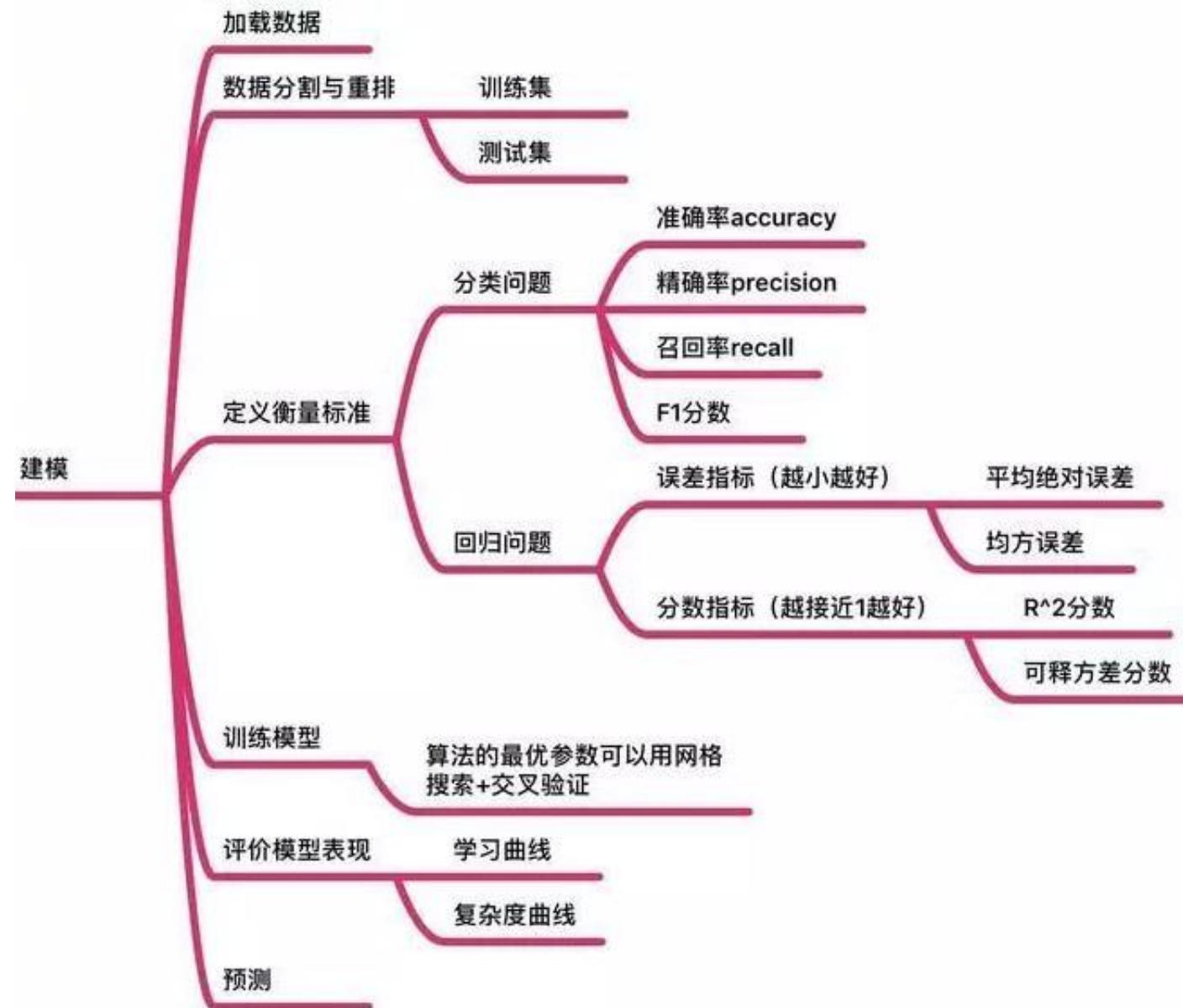




4

# 機器學習模型的驗證模估

# 分類模型評估思維導圖

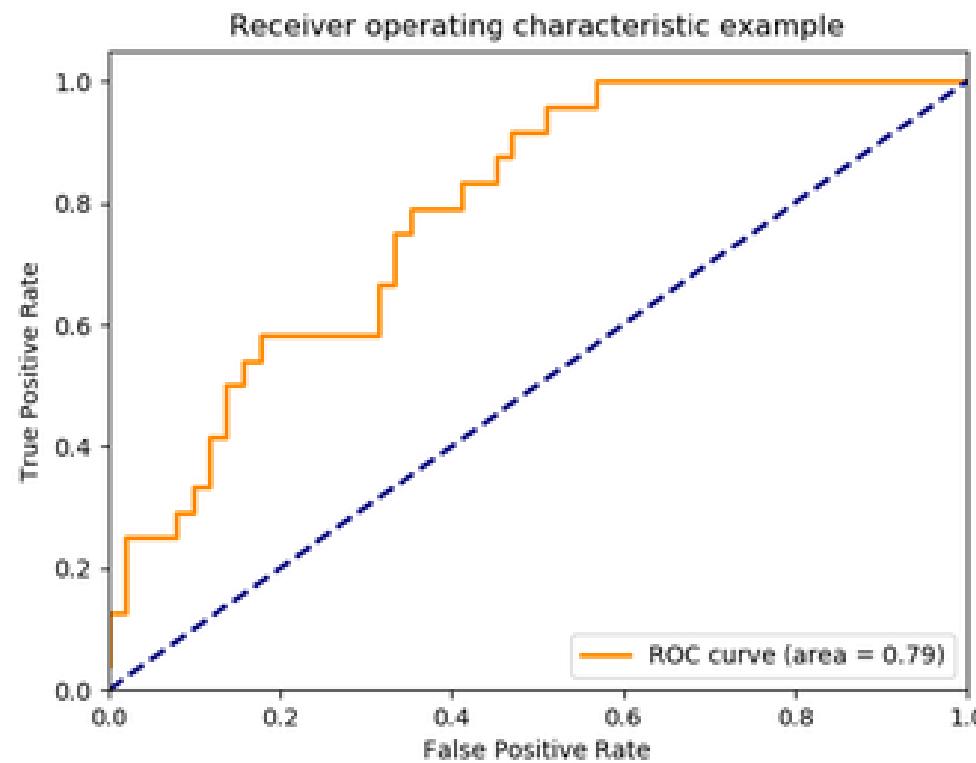


# 混淆矩陣計算實例

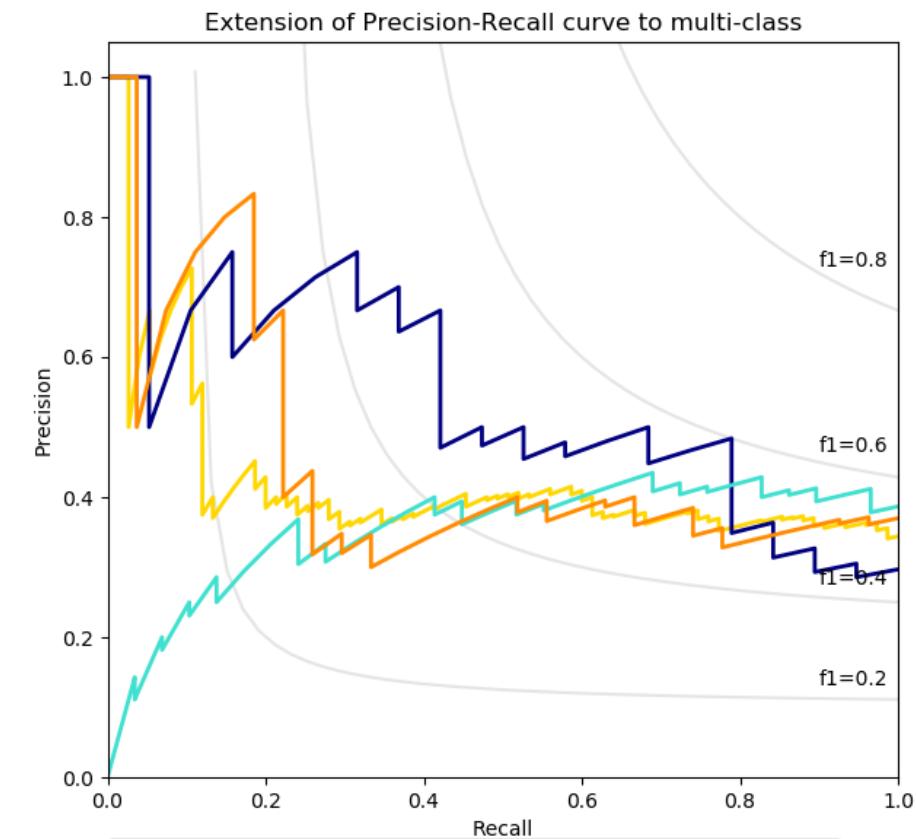
		Patients with bowel cancer (as confirmed on endoscopy)		
		Condition positive	Condition negative	
Fecal occult blood screen test outcome	Test outcome positive	True positive (TP) = 20	False positive (FP) = 180	Positive predictive value $= TP / (TP + FP)$ $= 20 / (20 + 180)$ $= 10\%$
	Test outcome negative	False negative (FN) = 10	True negative (TN) = 1820	Negative predictive value $= TN / (FN + TN)$ $= 1820 / (10 + 1820)$ $\approx 99.5\%$
		Sensitivity $= TP / (TP + FN)$ $= 20 / (20 + 10)$ $\approx 67\%$	Specificity $= TN / (FP + TN)$ $= 1820 / (180 + 1820)$ $= 91\%$	

資料來源 : wikipedia

# 圖形化評估方式 (1/2)



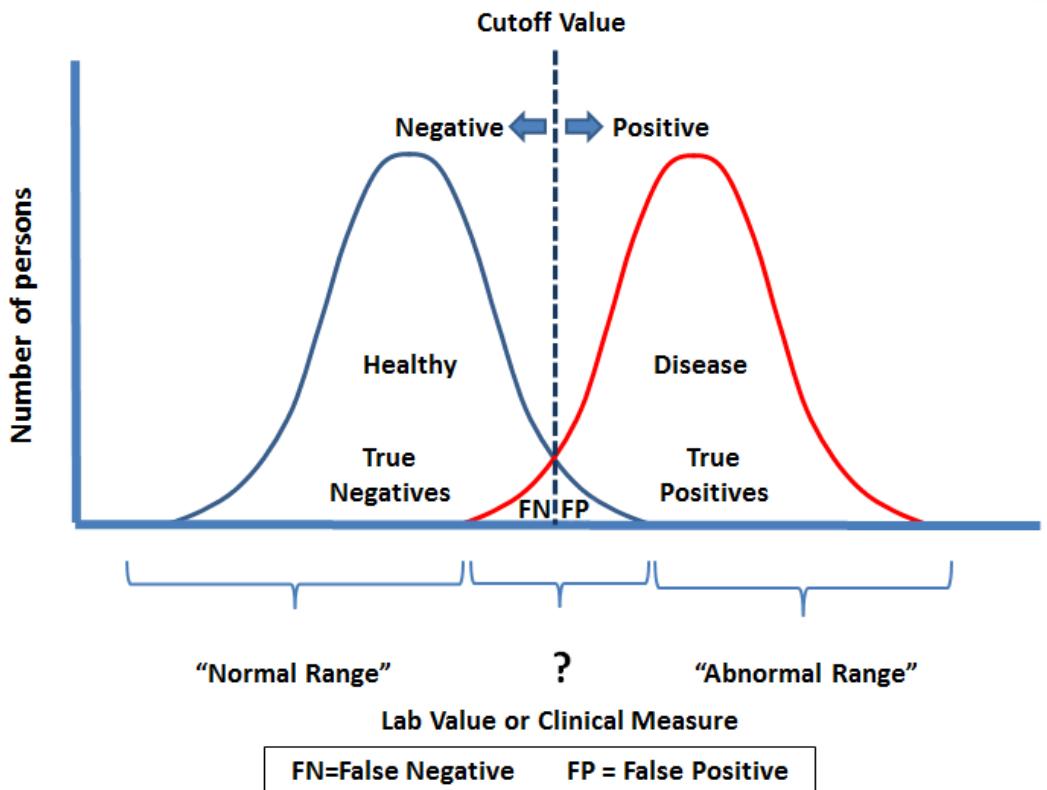
ROC曲線圖  
(ROC Curves)



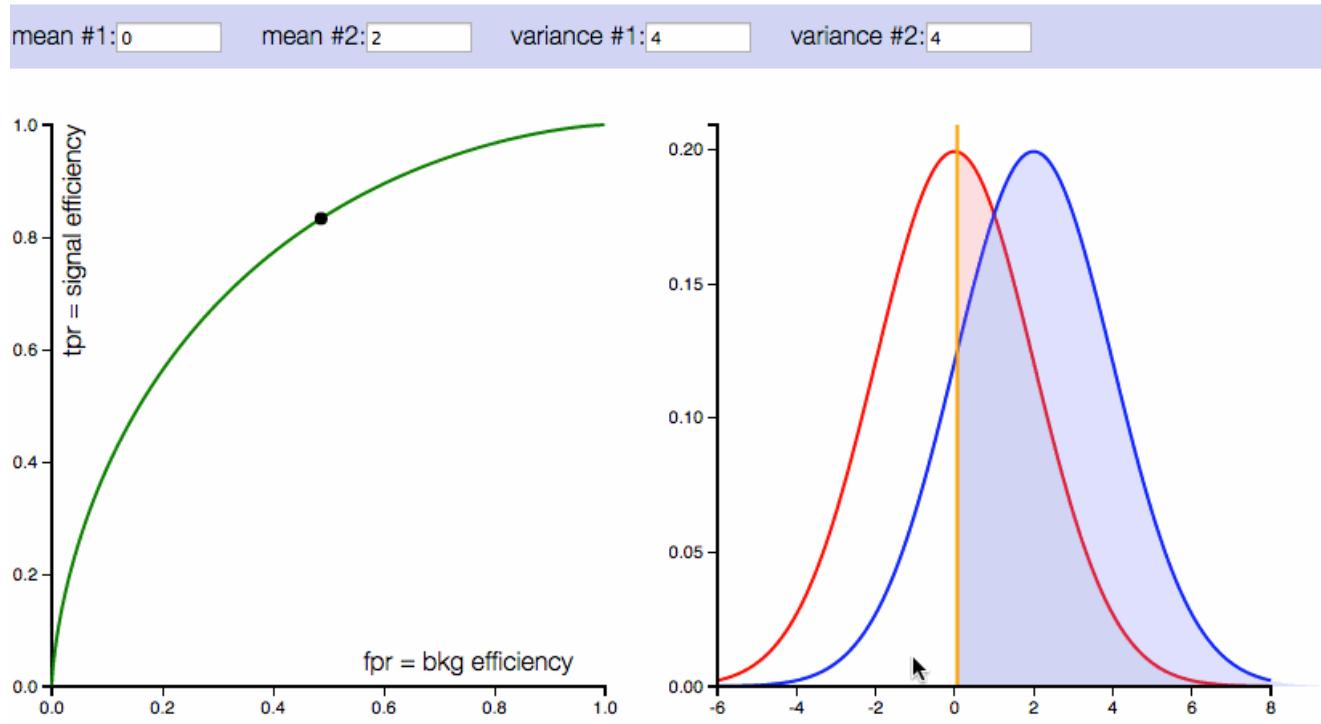
AUP曲線圖  
(Precision Recall Curves)

資料來源：如何確保大數據分析的品質：淺談監督式機器學習的測試評估方法

# 圖形化評估方式 (2/2)



ROC curve demo



資料來源：<http://arogozhnikov.github.io/2015/10/05/roc-curve.html>



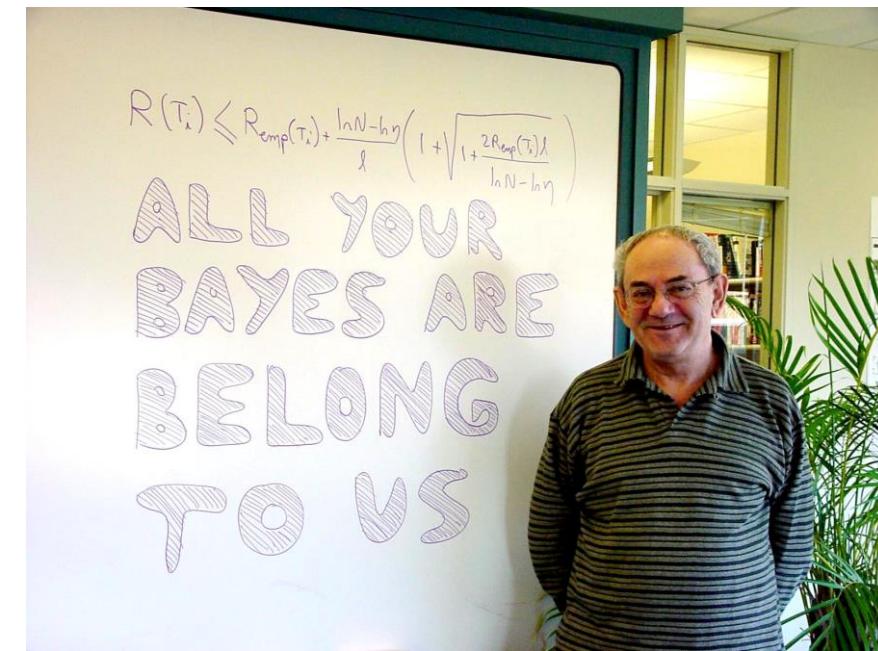
4

# 機器學習模型介紹

# Support Vector Machine 支持向量機

# What's Support Vector Machine?

- 支持向量機 SVM (Vapnik et al., 1963) 是一種基於統計學習理論的機器學習方法，也是監督式學習方法中的一種
- 是一種可用於分類與迴歸的方法
- 透過學習訓練資料後獲得模型
- 利用模型預測測試資料所屬類別



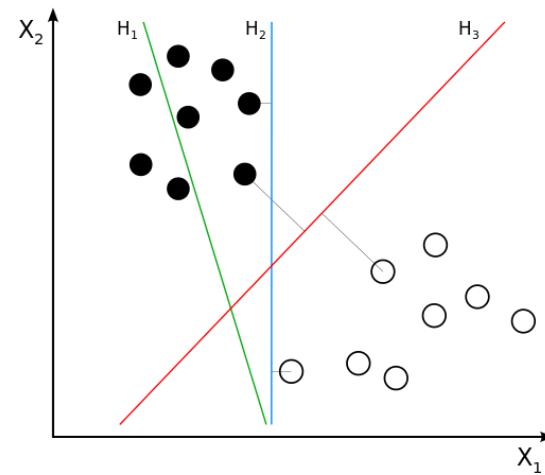
# 支持向量機的優點

- 在高維空間中有效
- 在維數大於樣本數的情況下仍然有效
- 在決策函數中使用訓練點的子集（稱為支援向量），因此它也具有存儲效率。
- 多功能：可以為決策功能指定不同的內核(Kernel)功能。提供了通用內核，但是也可以指定自訂內核。

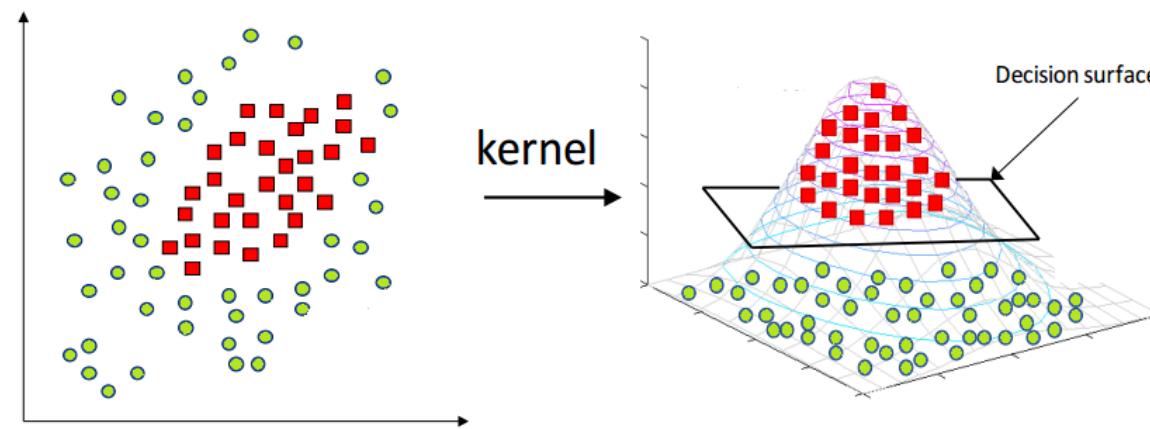
## 支持向量機的缺點

- 如果特徵數量遠大於樣本數量，請避免在選擇內核函數時過度擬合，並且正則項至關重要。
- SVM不直接提供概率估計，而是使用昂貴的五倍交叉驗證計算

# What's unique about SVM?

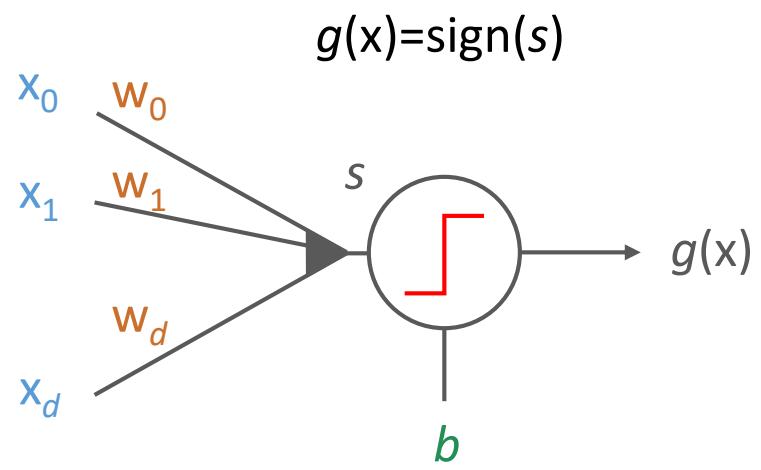


(1) 超平面

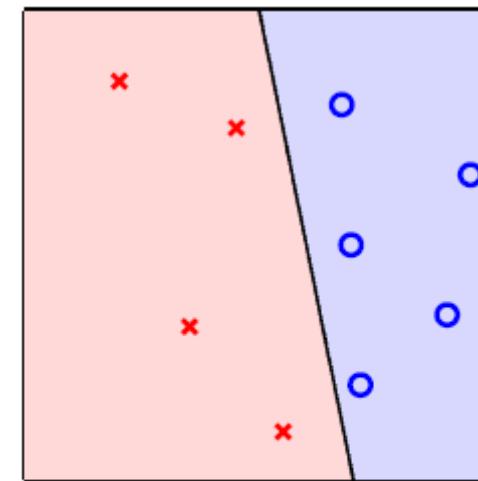


(2) 核技巧

# Linear Classification



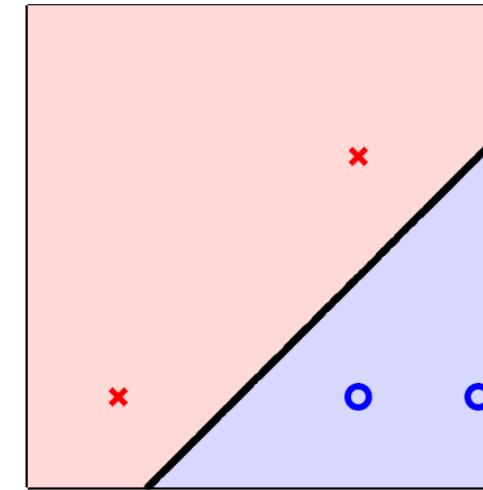
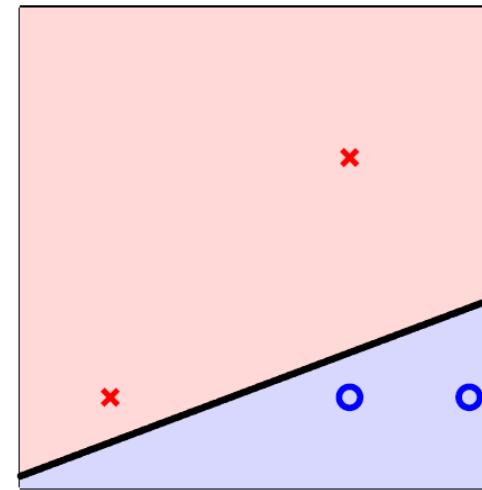
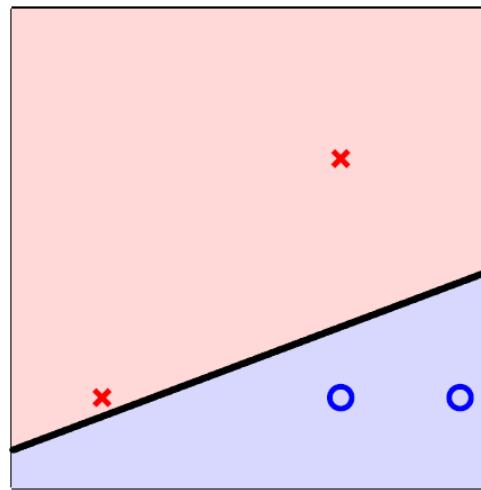
線性分類器:  
 $g(x)=\text{sign}(\mathbf{w}^T \mathbf{x} + b)$  (hyperplane)



(linear separable)

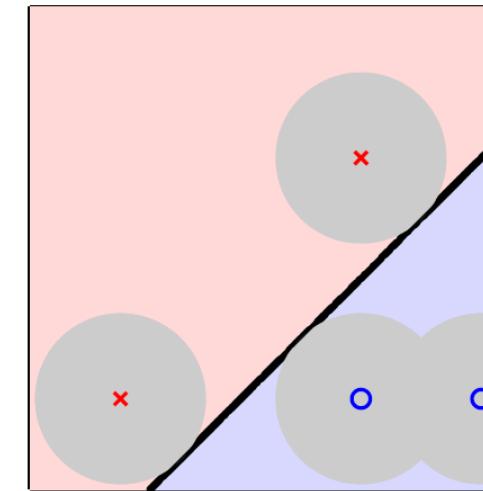
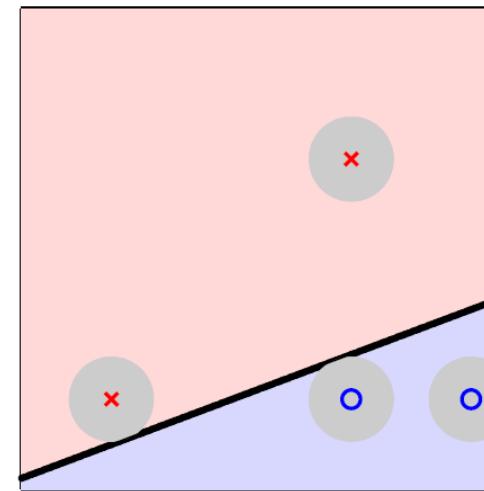
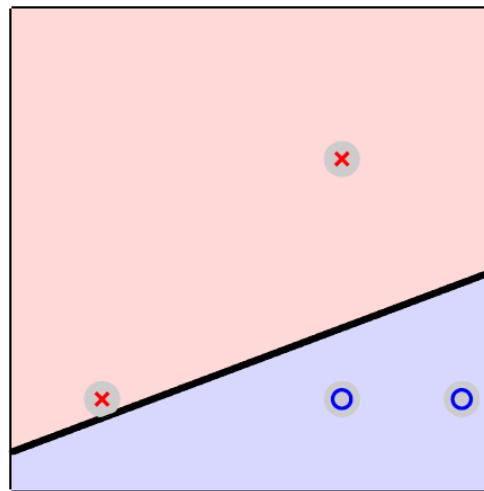
# Which Line is Best?

哪一條線分得最好？



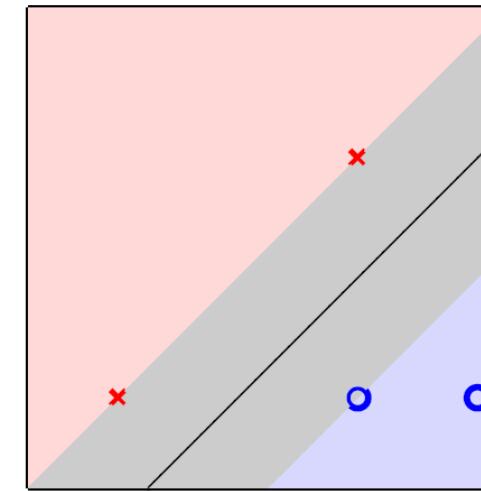
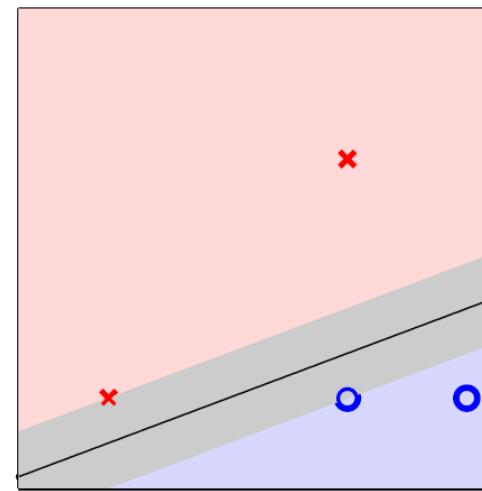
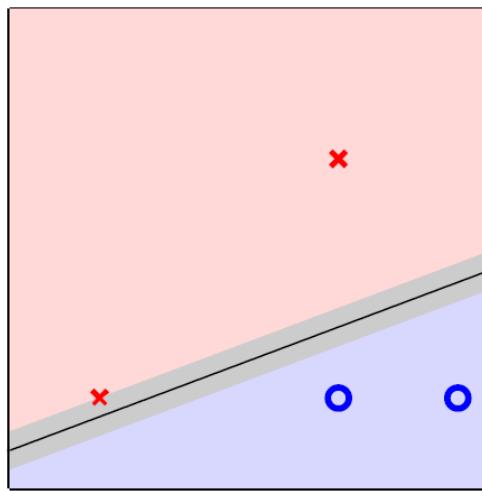
# Why Rightmost Hyperplane?

為什麼最右邊的線分得最好？



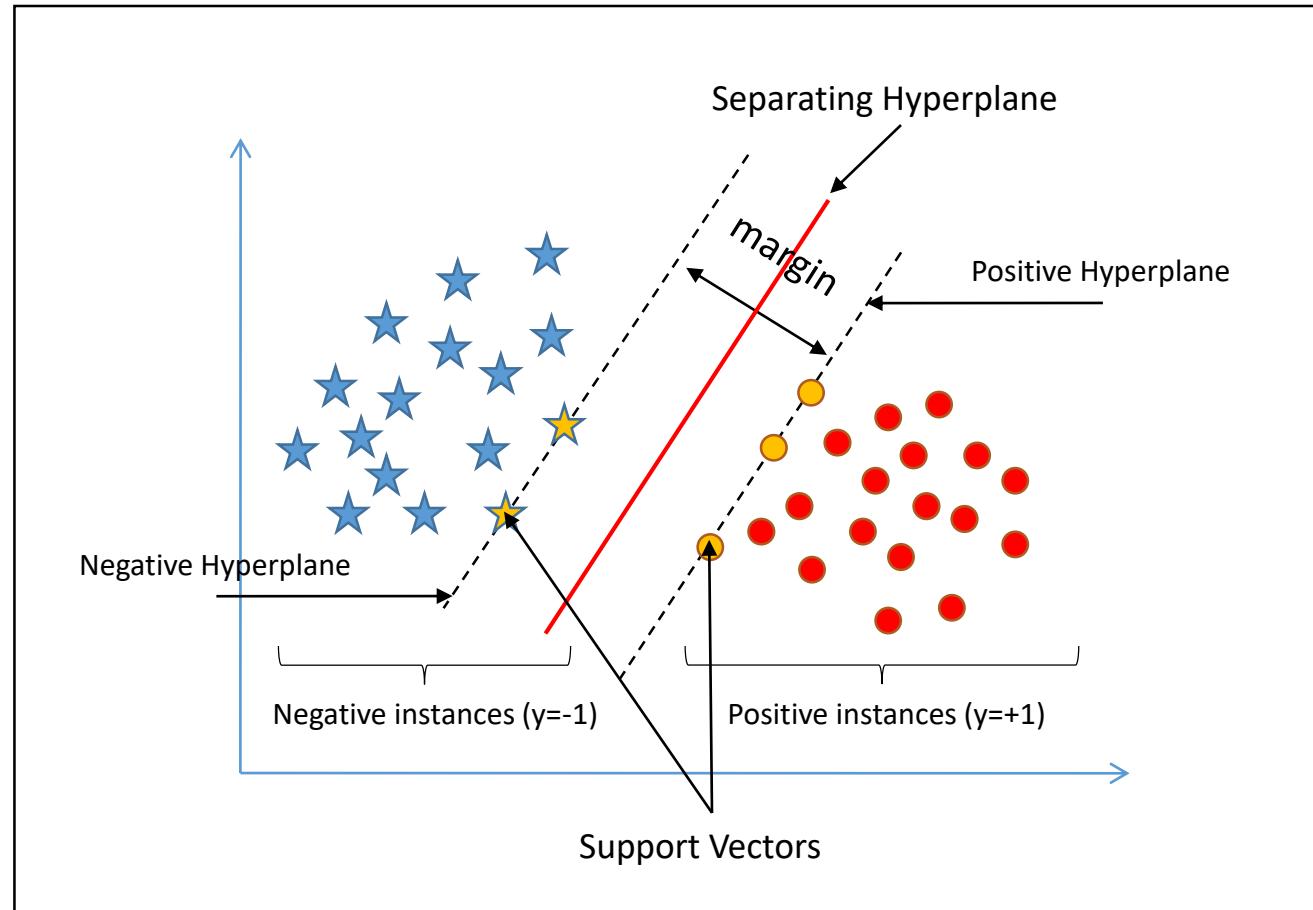
# Fat Hyperplane

SVM目標: 找到最大的分隔超平面

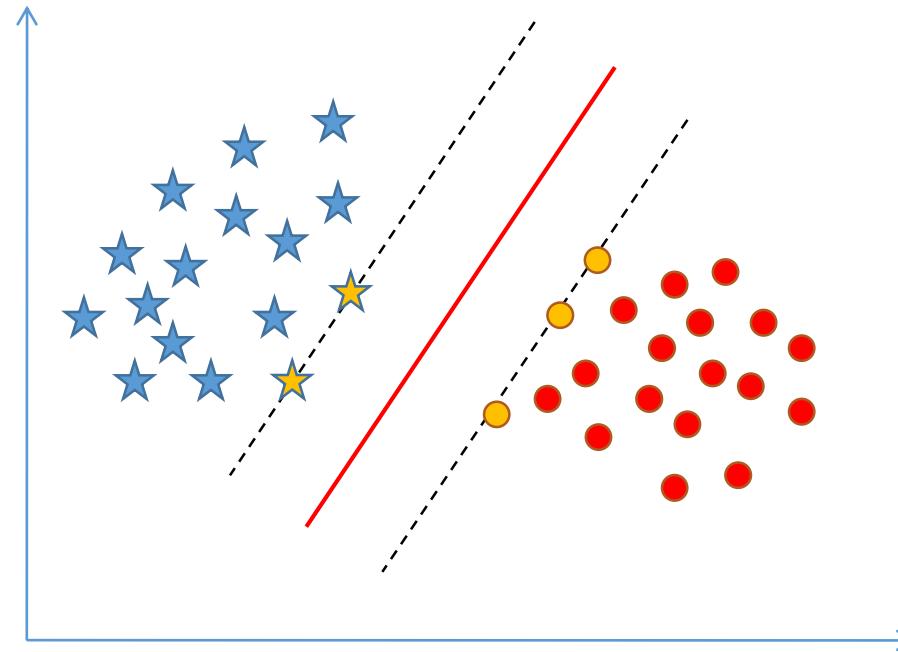


# Basic concept of SVM

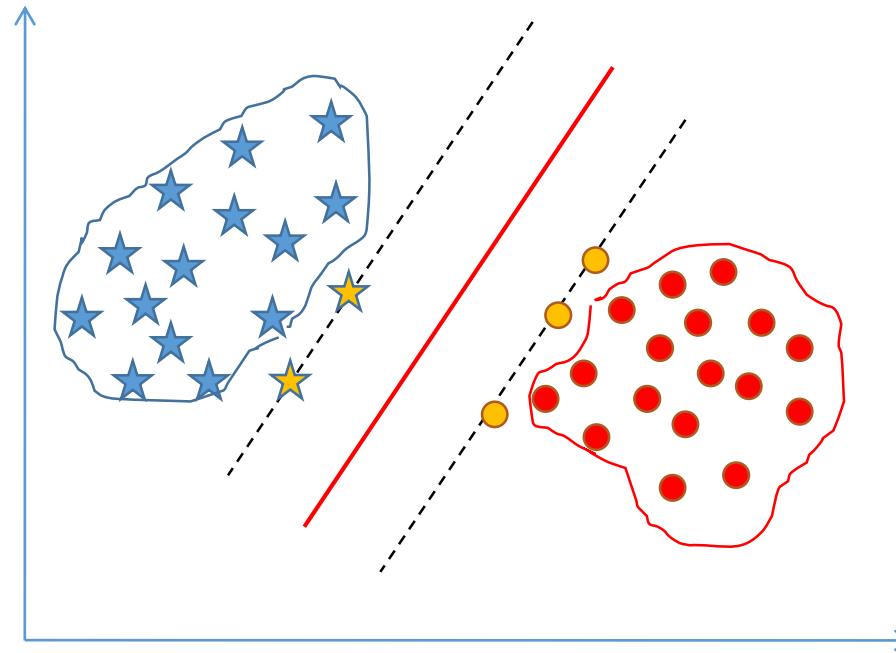
- 找出一個能將不同類別的資料完美的分開並保持**最大距離** (margin) 的**分隔超平面** (separating hyperplane)
- 在超平面邊界上的點又稱為**支持向量** (support vectors)



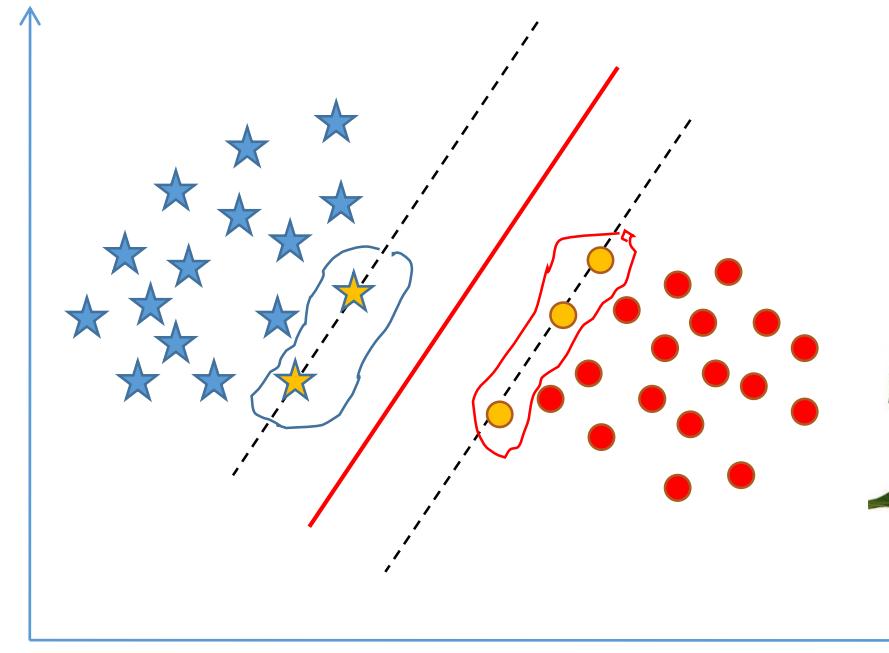
# Basic concept of SVM



# Basic concept of SVM



# Basic concept of SVM



# Hard-margin SVM

**Standard Large-Margin Hyperplane Problem:**

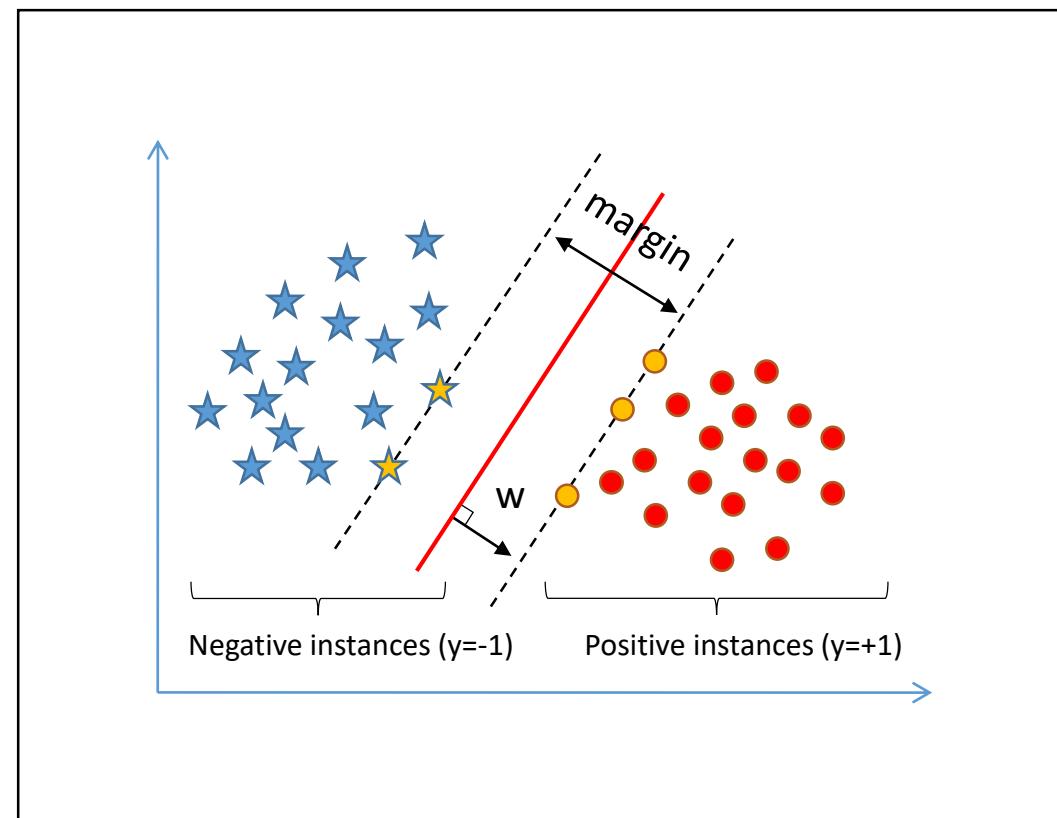
$$\max_{b,w} \text{margin}(b, w)$$

$$\text{s. t. } \text{every } y_n(w^T x_n + b) > 0$$

$$\text{margin}(b, w) = \min_{n=1,2,\dots,N} \text{distance}(x_n, b, w)$$

$$\text{margin} = \frac{2}{\|w\|}, \|w\| \text{越小距離越大}$$

目標: 找到最大分隔超平面



# Primal SVM

Standard Large-Margin Hyperplane Problem:

$$g_{SVM}(x) = ?$$

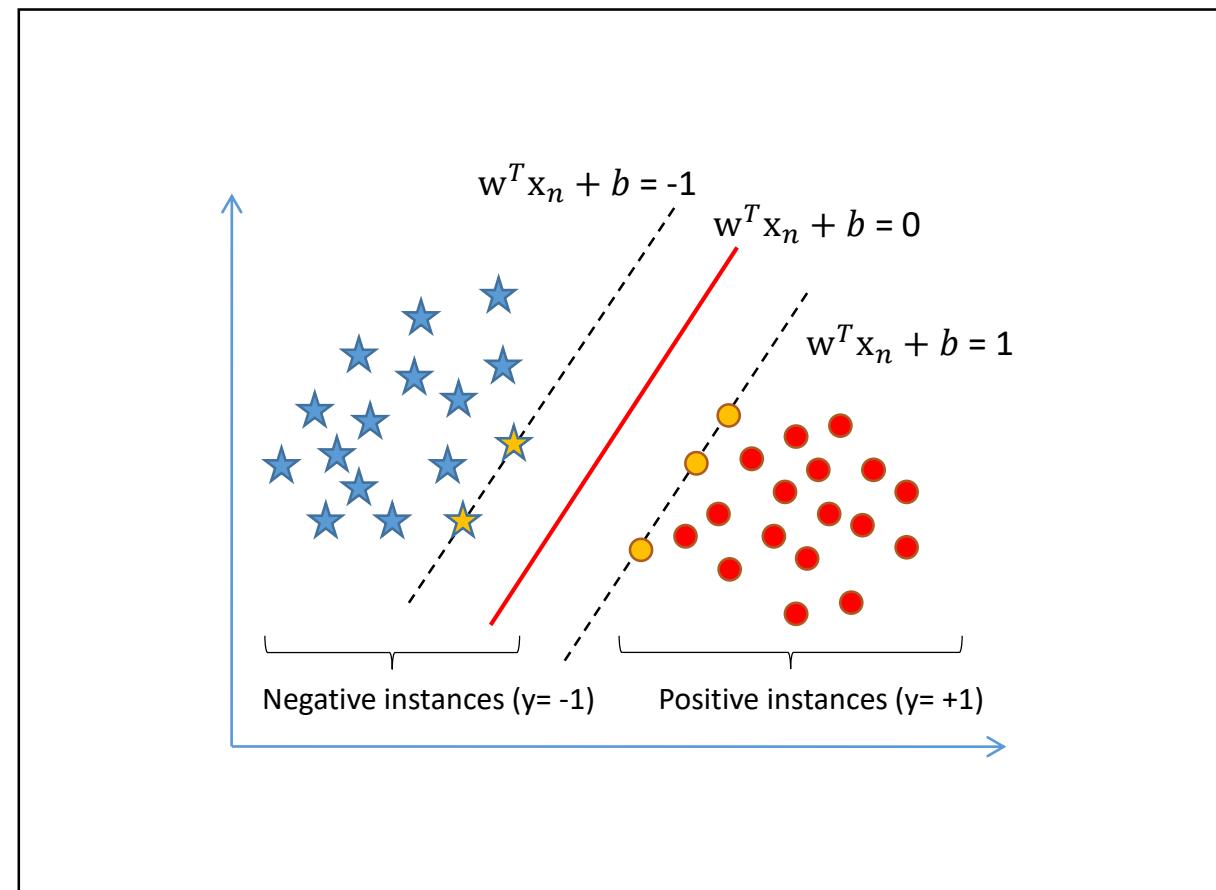
primal formula:

$$\min_{b,w} \frac{1}{2} w^T w$$

Objective function

$$\text{s. t. } y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N$$

Constraints



# 展示簡化Primal SVM 解題步驟流程

1. 將離交界處最近的兩類別的 3~4 個點  
帶入條件限制公式
2. 利用代入後的公式解聯立求  $w_1$  與  $w_2$   
之極值
3. 利用求得的  $w_1$  與  $w_2$  之極值再求得  $b$  值
4. 利用  $w_1$ ,  $w_2$  與  $b$  值求得  $g_{SVM}(x)$ ,  
 $g_{SVM}(x) = w^T x + b = 0$   
最後可透過  $\frac{1}{2}w^T w = \frac{1}{2}(w_1^2 + w_2^2)$   
求得最小值

SVM目標函數與條件限制:

$$\min_{b,w} \quad \frac{1}{2} w^T w$$

$$\text{s. t. } y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N$$

$x$ : 該筆資料的數值

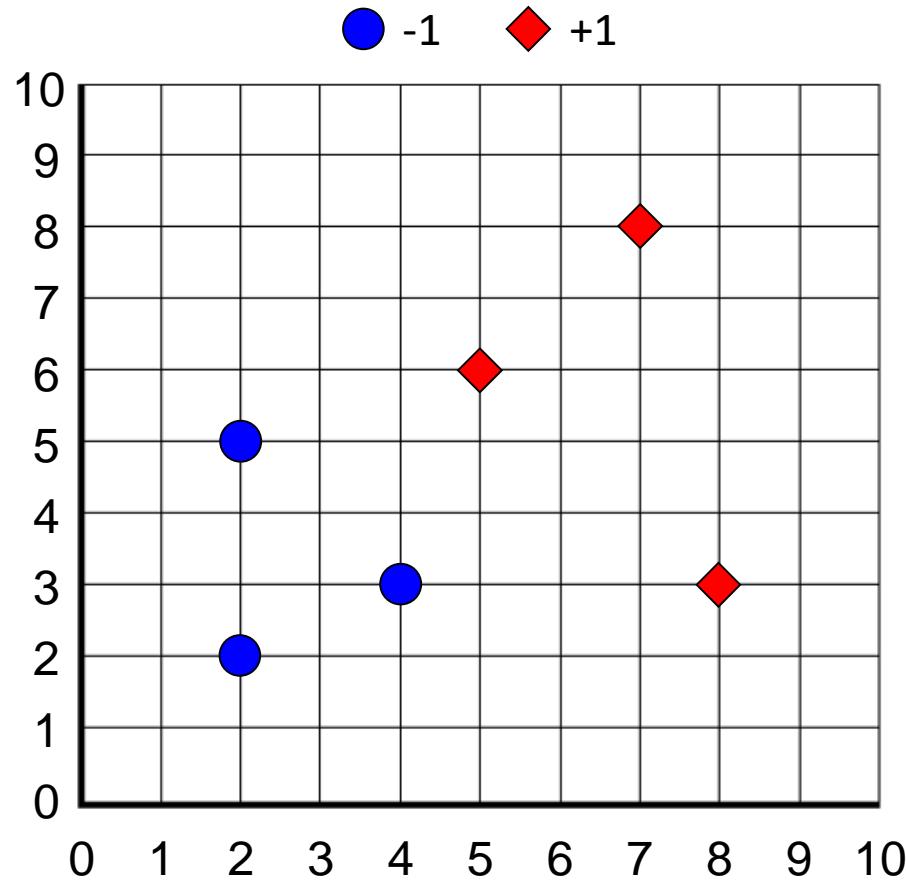
$y$ : 該筆資料所屬的類別

$w$ : 代表資料中某個維度的權重，  
為垂直於SVM的法向量

$b$ : 偏差值

$N$ : 資料集中樣本的總數

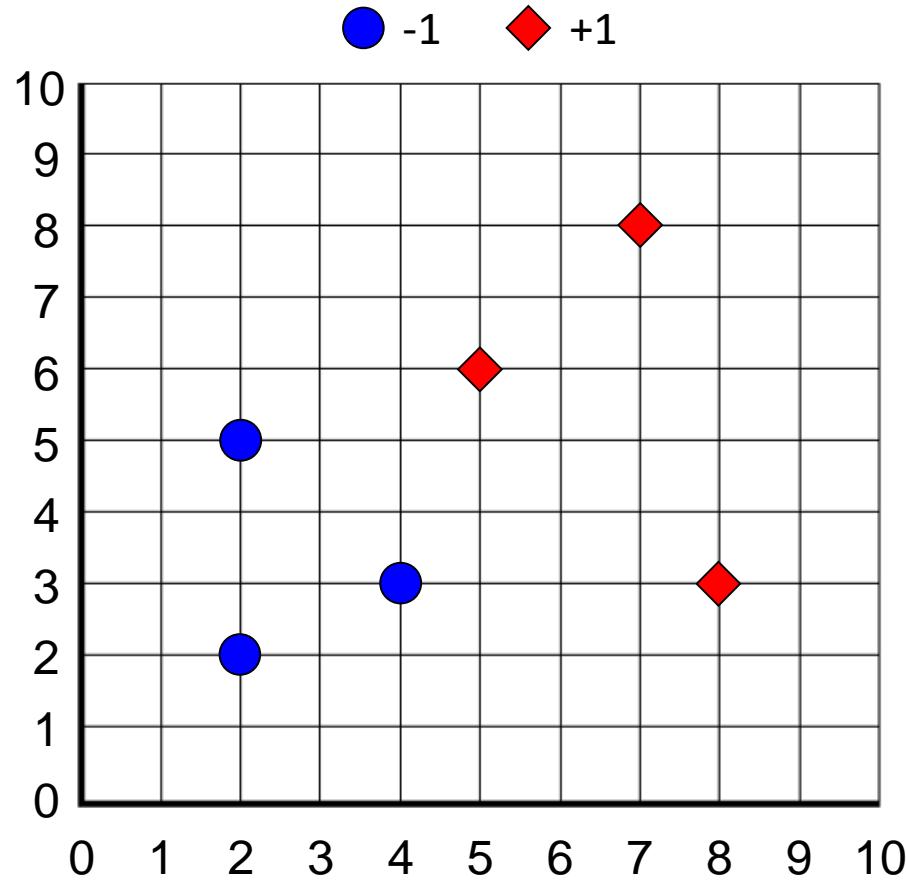
# 範例一



$$\begin{aligned} \min_{b,w} \quad & \frac{1}{2} w^T w \\ \text{s. t.} \quad & y_n (w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N \end{aligned}$$

例題一：  
如何求出  $g_{\text{SVM}}(x)$  ?

# 範例一

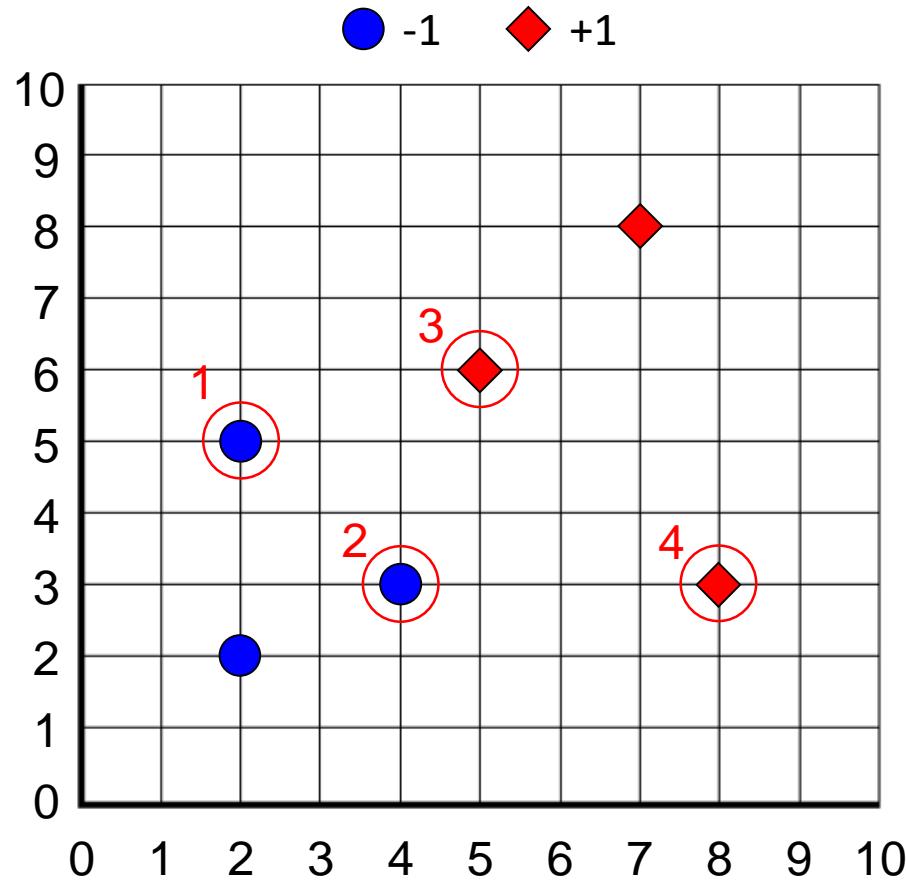


$$\begin{aligned} \min_{b,w} \quad & \frac{1}{2} w^T w \\ \text{s. t.} \quad & y_n (w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N \end{aligned}$$

步驟一

將離交界處最近的點帶入條件限制公式

# 範例一

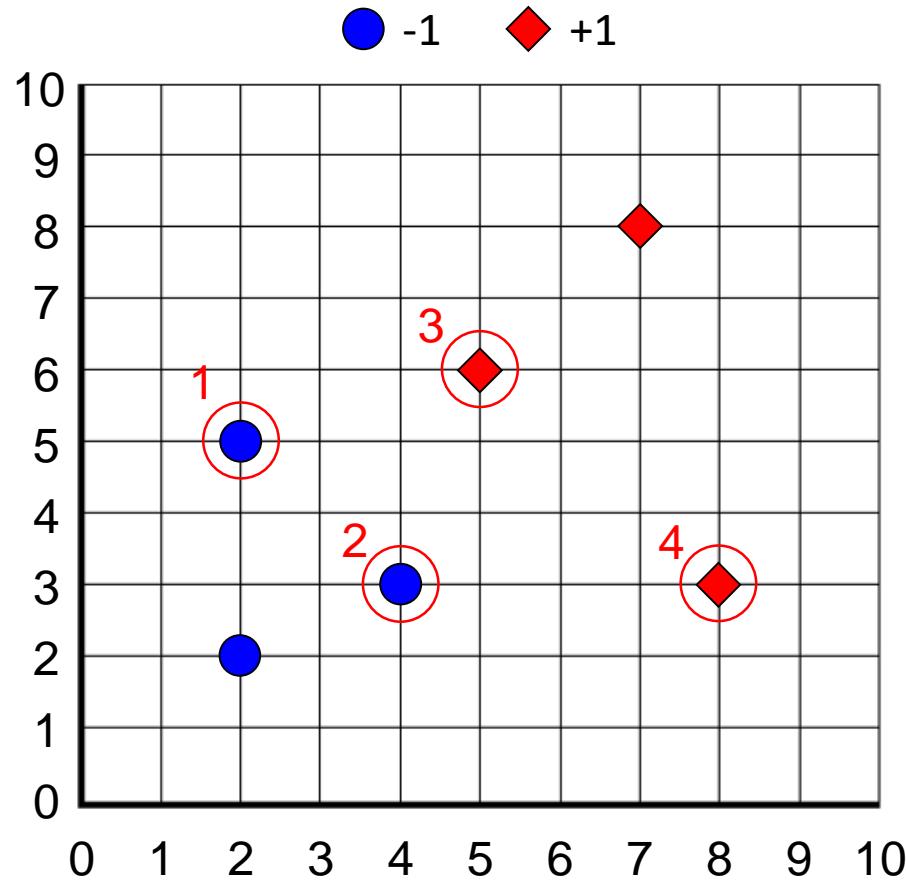


$$\begin{aligned} & \min_{b,w} \quad \frac{1}{2} w^T w \\ \text{s. t.} \quad & y_n (w^T x_n + b) \geq 1, \quad \text{for } n = 1, 2, \dots, N \end{aligned}$$

步驟一  
將離交界處最近的點帶入公式

1.  $x = (2, 5) y = -1$
2.  $x = (4, 3) y = -1$
3.  $x = (5, 6) y = +1$
4.  $x = (8, 3) y = +1$

# 範例一



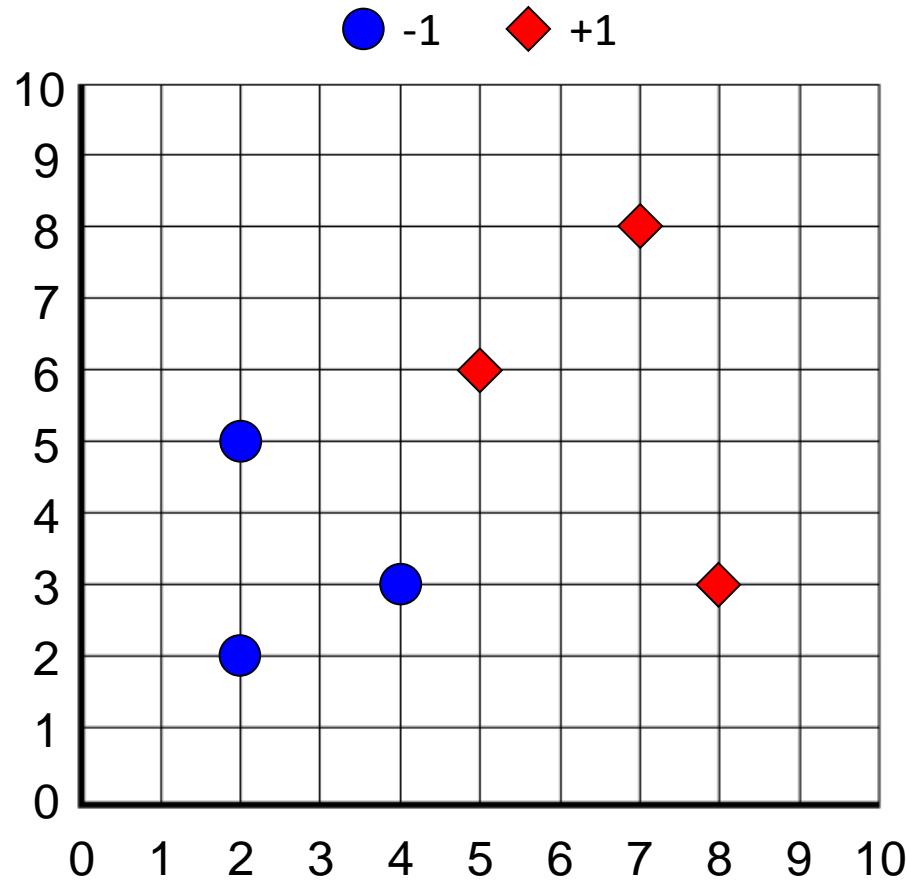
$$\begin{aligned} \min_{b,w} \quad & \frac{1}{2} w^T w \\ \text{s. t.} \quad & y_n (w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N \end{aligned}$$

步驟一

將離交界處最近的點帶入公式

1.  $x = (2, 5) y = -1 \quad -2w_1 - 5w_2 - b \geq 1 \dots (1)$
2.  $x = (4, 3) y = -1 \quad -4w_1 - 3w_2 - b \geq 1 \dots (2)$
3.  $x = (5, 6) y = +1 \quad 5w_1 + 6w_2 + b \geq 1 \dots (3)$
4.  $x = (8, 3) y = +1 \quad 8w_1 + 3w_2 + b \geq 1 \dots (4)$

# 範例一



$$\begin{aligned} \min_{b,w} \quad & \frac{1}{2} w^T w \\ \text{s. t.} \quad & y_n (w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N \end{aligned}$$

步驟二

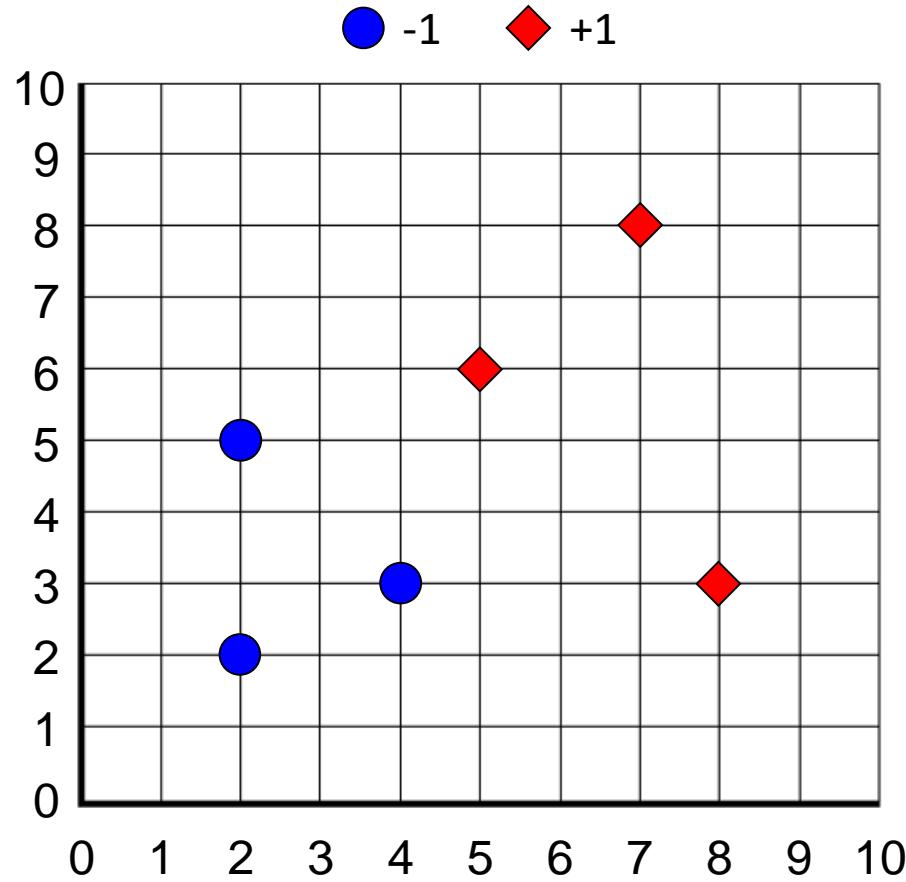
利用代入後的公式求  $w_1$  與  $w_2$  之極值

$$-4w_1 - 3w_2 - b \geq 1 \dots (2)$$

$$8w_1 + 3w_2 + b \geq 1 \dots (4)$$

$$(2) + (4) \rightarrow 4w_1 \geq 2 \rightarrow w_1 \geq \frac{1}{2}$$

# 範例一



$$\min_{b,w} \frac{1}{2} w^T w$$

$$\text{s. t. } y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N$$

步驟二

利用代入後的公式求  $w_1$  與  $w_2$  之極值

$$-2w_1 - 5w_2 - b \geq 1 \dots (1)$$

$$5w_1 + 6w_2 + b \geq 1 \dots (3)$$

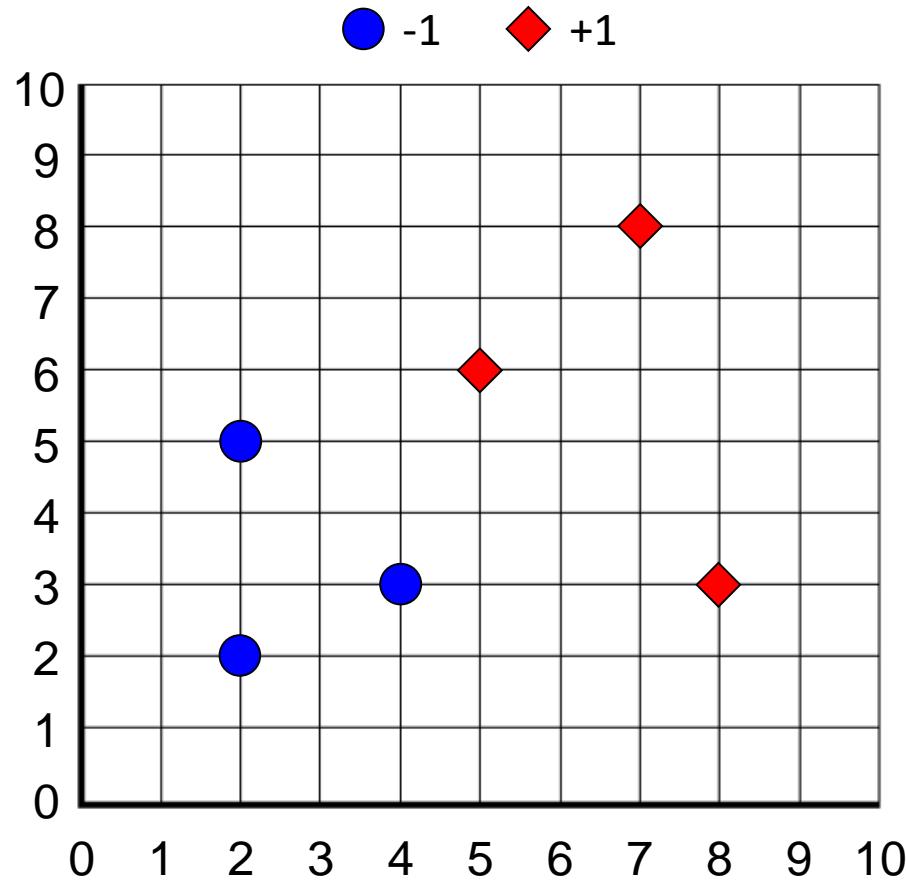
$$8w_1 + 3w_2 + b \geq 1 \dots (4)$$

$$(1) + (3) \rightarrow 3w_1 + w_2 \geq 2 \dots (i)$$

$$(4) - (3) \rightarrow 3w_1 - 3w_2 \geq 0 \dots (ii)$$

$$(i) - (ii) \rightarrow 4w_2 \geq 2 \rightarrow w_2 \geq \frac{1}{2}$$

# 範例一



$$\min_{b,w} \frac{1}{2} w^T w$$

$$\text{s. t. } y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N$$

步驟二

利用代入後的公式求  $w_1$  與  $w_2$  之極值

整理前述步驟，得知  $w_1$  與  $w_2$

$w_1$  為：

$$(2) + (4) \rightarrow 4w_1 \geq 2 \rightarrow w_1 \geq \frac{1}{2}$$

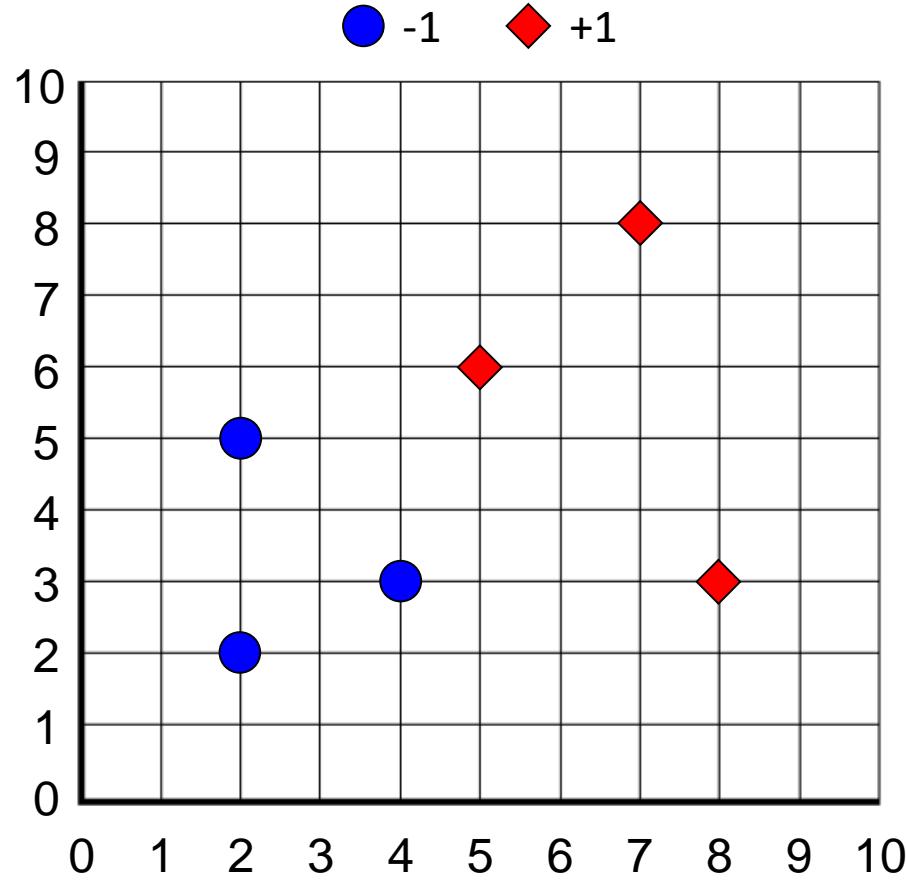
$w_2$  為：

$$(1) + (3) \rightarrow 3w_1 + w_2 \geq 2 \dots (\text{i})$$

$$(4) - (3) \rightarrow 3w_1 - 3w_2 \geq 0 \dots (\text{ii})$$

$$(\text{i}) - (\text{ii}) \rightarrow 4w_2 \geq 2 \rightarrow w_2 \geq \frac{1}{2}$$

# 範例一



$$\min_{b,w} \frac{1}{2} w^T w$$

$$\text{s. t. } y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N$$

步驟三

將求得的  $w_1$  與  $w_2$  之極值代入(1)~(4)中  
任意公式求  $b$  值

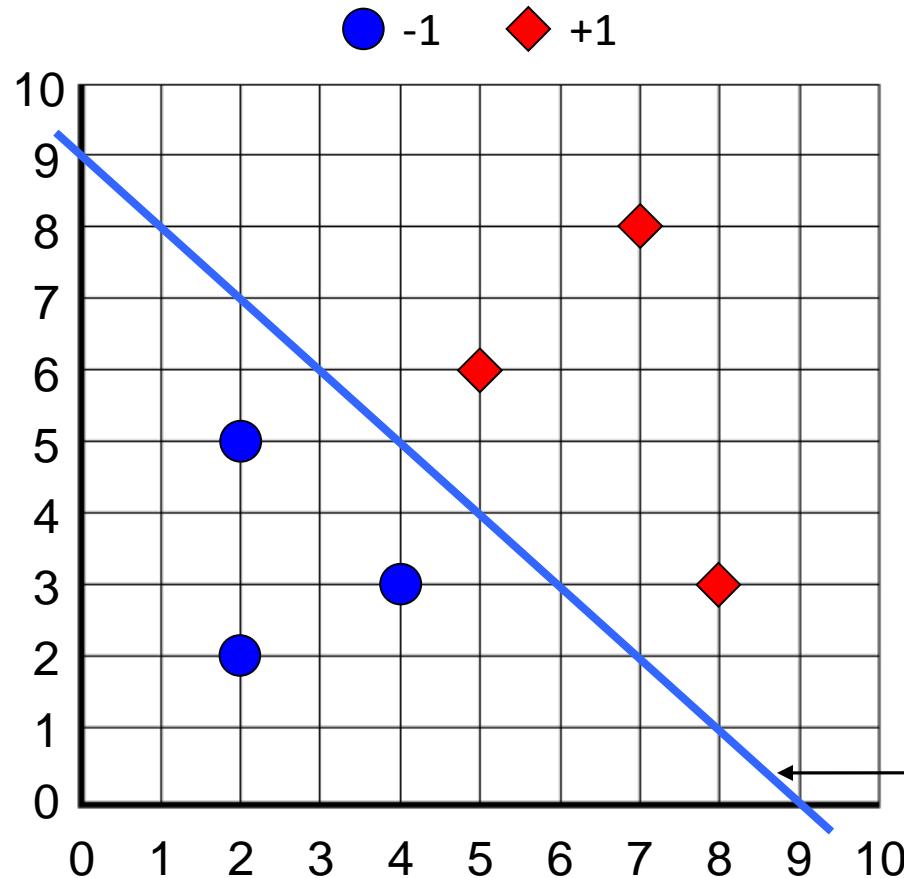
$$w_1 \geq \frac{1}{2}, w_2 \geq \frac{1}{2}$$

將  $w_1$  &  $w_2$  極值代入 (4)

$$8w_1 + 3w_2 + b \geq 1 \dots (4)$$

$$4 + \frac{3}{2} + b \geq 1 \rightarrow b \geq -\frac{9}{2}$$

# 範例一



$$\min_{b,w} \frac{1}{2} w^T w$$

$$\text{s. t. } y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N$$

步驟四

最後得到  $w_1, w_2$  與  $b$  值, 並求得  $g_{SVM}(x)$

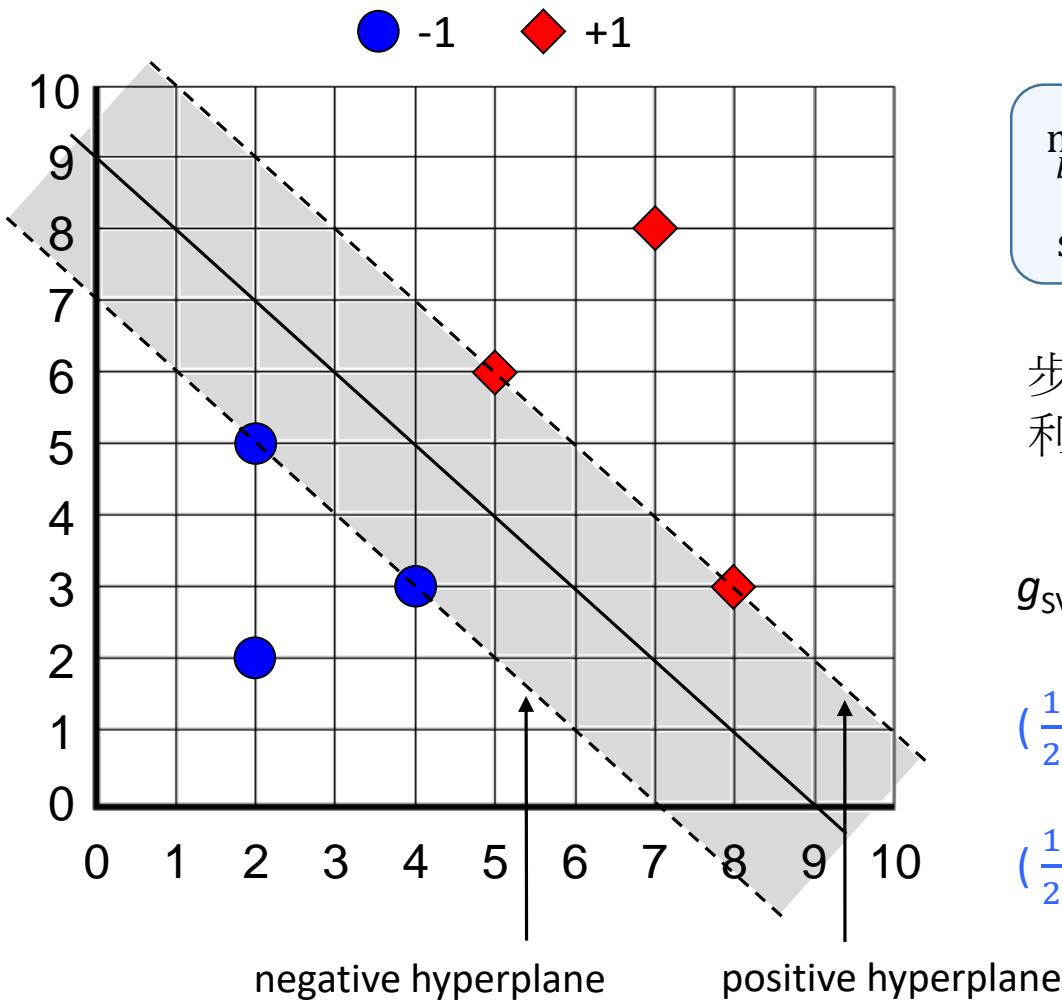
透過  $\frac{1}{2} w^T w = \frac{1}{2} (w_1^2 + w_2^2)$  求得最小值

$$(w_1 = \frac{1}{2}, w_2 = \frac{1}{2}, b = -\frac{9}{2} \text{ at lower bound})$$

$$g_{SVM}(x) = \frac{1}{2}x_1 + \frac{1}{2}x_2 - \frac{9}{2} = 0 \quad \frac{1}{2} w^T w \geq \frac{1}{4}$$

$$g_{SVM}(x)$$

# 範例一



$$\begin{aligned} \min_{b,w} \quad & \frac{1}{2} w^T w \\ \text{s. t.} \quad & y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N \end{aligned}$$

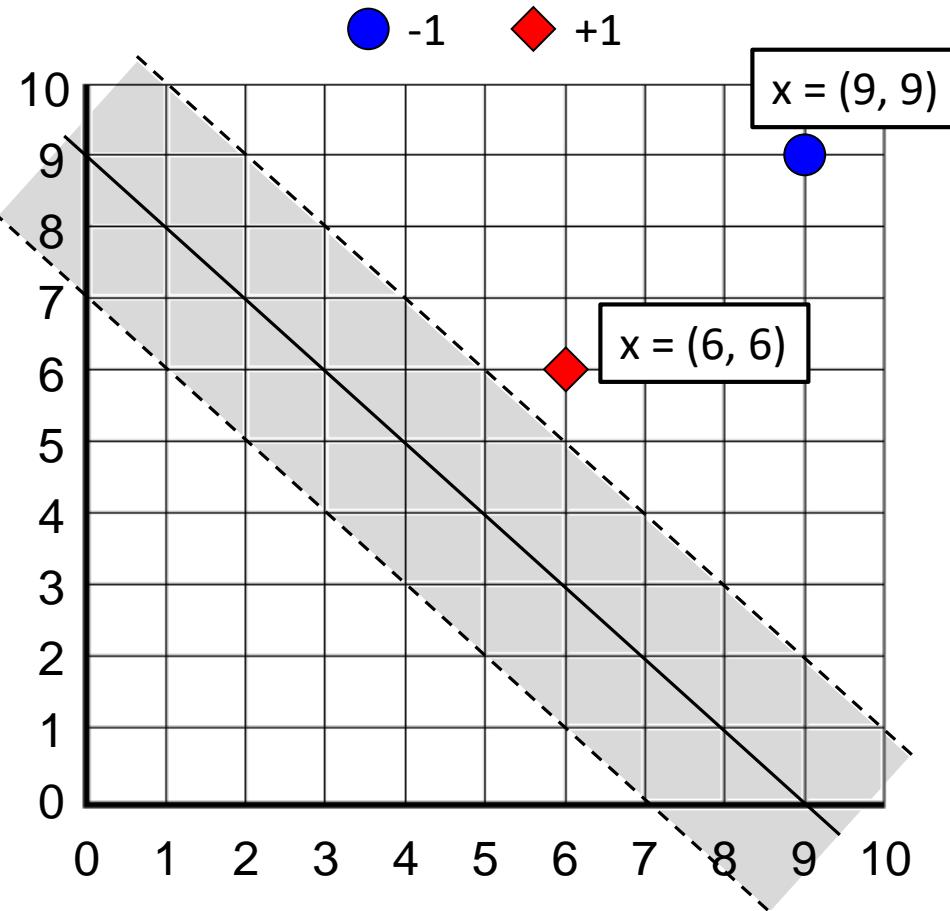
步驟五  
利用  $g_{\text{SVM}}(x)$  求出正負 hyperplane

$$g_{\text{SVM}}(x) = \frac{1}{2}x_1 + \frac{1}{2}x_2 - \frac{9}{2} = 0$$

$$\left( \frac{1}{2}x_1 + \frac{1}{2}x_2 - \frac{9}{2} \right) = 1 \dots \text{positive hyperplane}$$

$$\left( \frac{1}{2}x_1 + \frac{1}{2}x_2 - \frac{9}{2} \right) = -1 \dots \text{negative hyperplane}$$

# 範例一



$$\begin{aligned} & \min_{b,w} \frac{1}{2} w^T w \\ \text{s. t. } & y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N \end{aligned}$$

步驟六  
驗證測試資料

$$g_{\text{SVM}}(x) = \text{sign}\left(\frac{1}{2}x_1 + \frac{1}{2}x_2 - \frac{9}{2}\right)$$

●  $x = (9, 9), y = -1$

$$\frac{1}{2} \times 9 + \frac{1}{2} \times 9 - \frac{9}{2} = \frac{9}{2} \rightarrow \text{sign} = +1$$

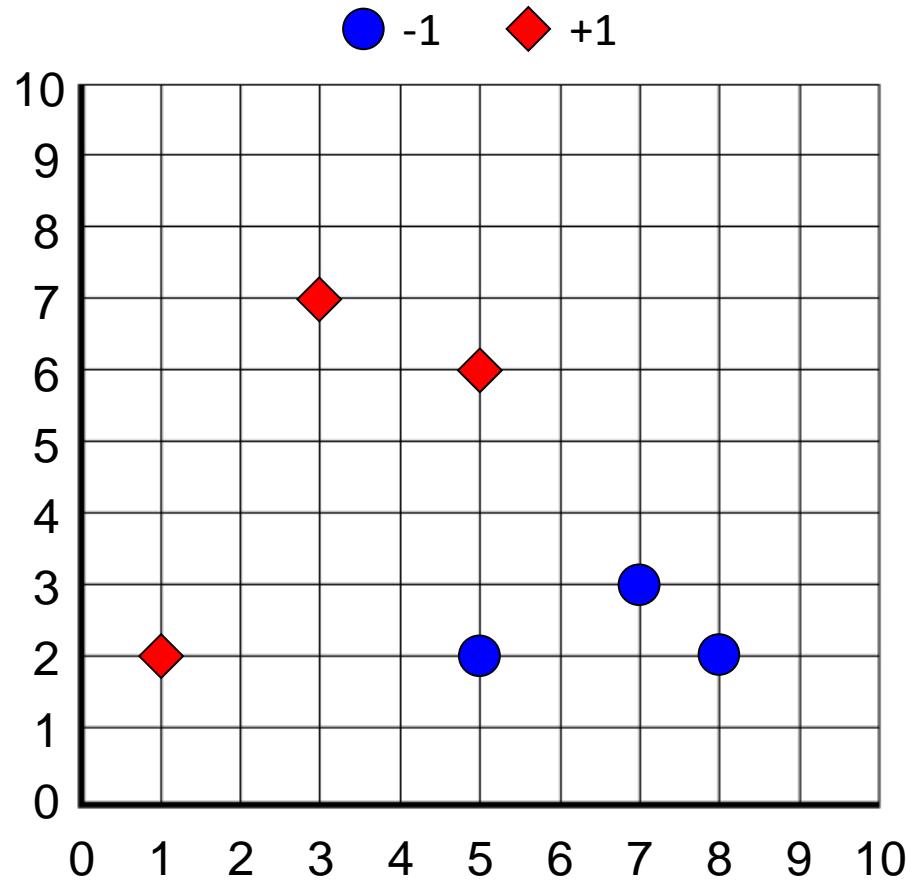
分類錯誤

◆  $x = (6, 6), y = +1$

$$\frac{1}{2} \times 6 + \frac{1}{2} \times 6 - \frac{9}{2} = \frac{3}{2} \rightarrow \text{sign} = +1$$

分類正確

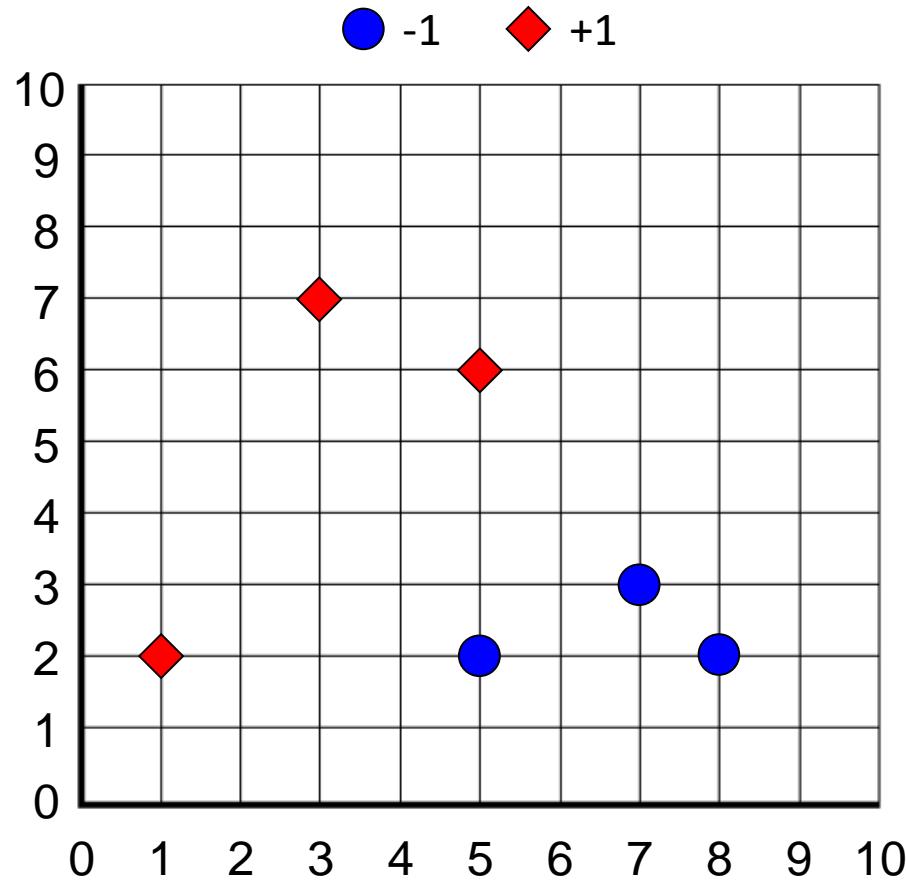
## 範例二



$$\begin{aligned} \min_{b,w} \quad & \frac{1}{2} w^T w \\ \text{s. t.} \quad & y_n (w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N \end{aligned}$$

例題二：  
請求出  $g_{\text{SVM}}(x) = \text{sign}(w^T x + b)$

## 範例二

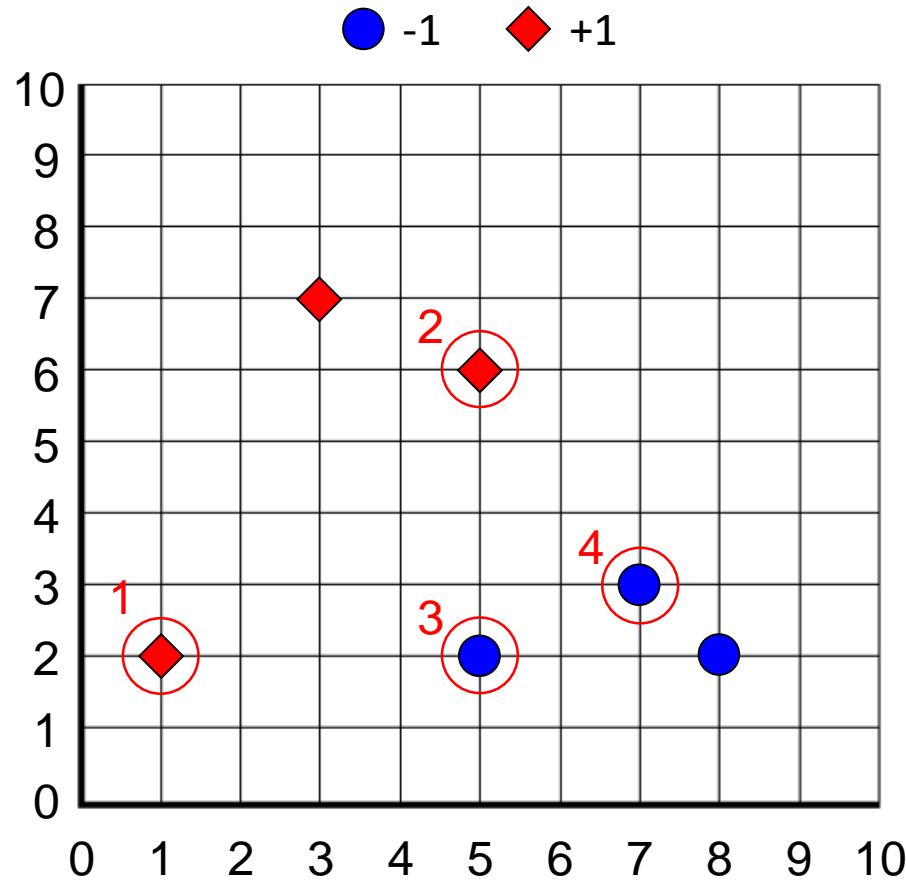


$$\begin{aligned} \min_{b,w} \quad & \frac{1}{2} w^T w \\ \text{s. t.} \quad & y_n (w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N \end{aligned}$$

步驟一

將離交界處最近的點帶入條件限制公式

## 範例二



$$\min_{b,w} \frac{1}{2} w^T w$$

$$\text{s. t. } y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N$$

步驟一

找出 support vectors 並帶入公式

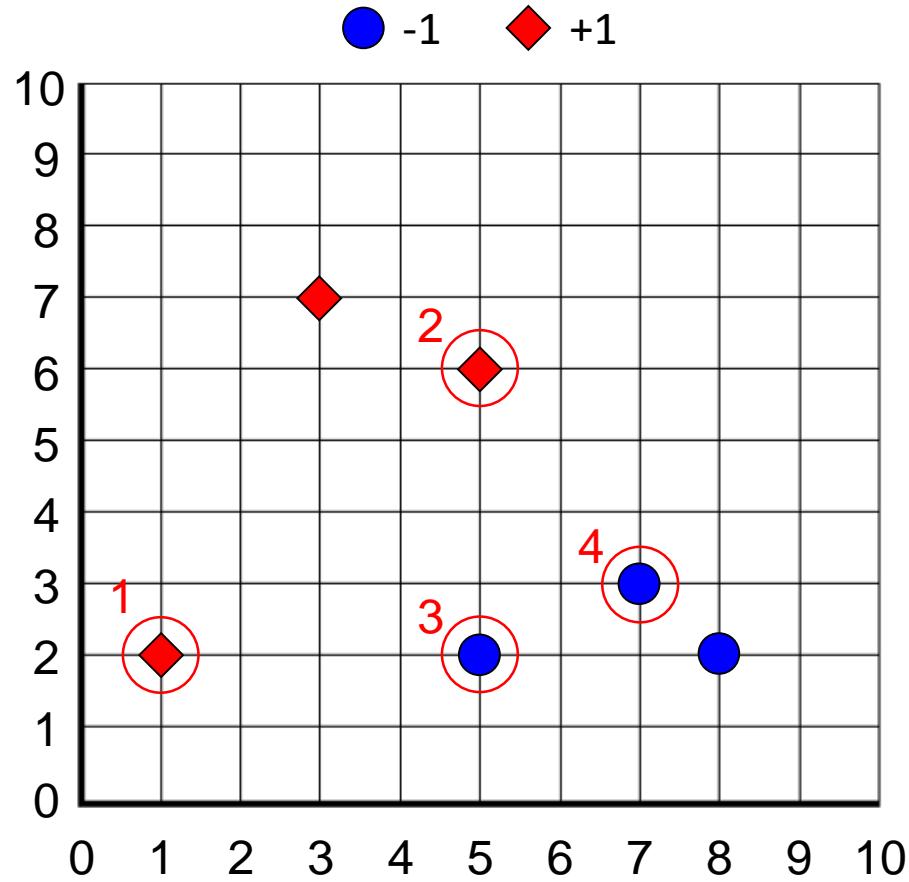
1.  $x = (1, 2) y = +1$

2.  $x = (5, 6) y = +1$

3.  $x = (5, 2) y = -1$

4.  $x = (7, 3) y = -1$

## 範例二



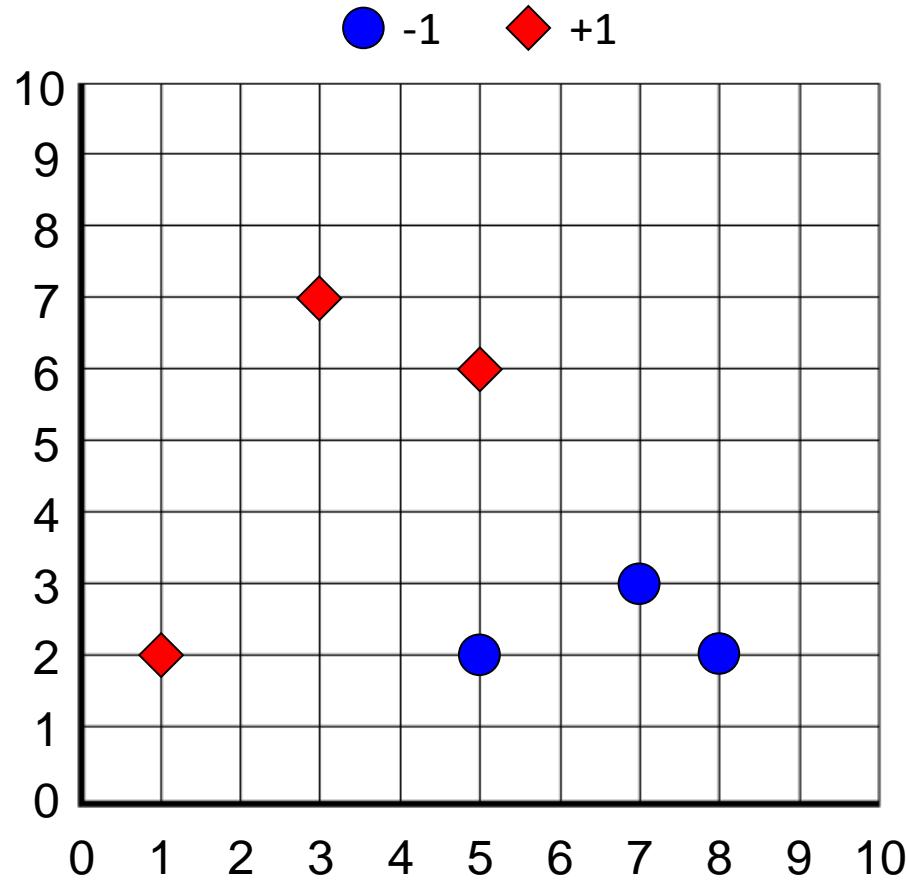
$$\begin{aligned} \min_{b,w} \quad & \frac{1}{2} w^T w \\ \text{s. t.} \quad & y_n (w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N \end{aligned}$$

步驟一

找出 support vectors 並帶入公式

1.  $x = (1, 2) y = +1 \quad 5w_1 + 6w_2 + b \geq 1 \dots (1)$
2.  $x = (5, 6) y = +1 \quad 5w_1 + 6w_2 + b \geq 1 \dots (2)$
3.  $x = (5, 2) y = -1 \quad -5w_1 - 2w_2 - b \geq 1 \dots (3)$
4.  $x = (7, 3) y = -1 \quad -7w_1 - 3w_2 - b \geq 1 \dots (4)$

## 範例二



$$\min_{b,w} \frac{1}{2} w^T w$$

$$\text{s. t. } y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N$$

步驟二

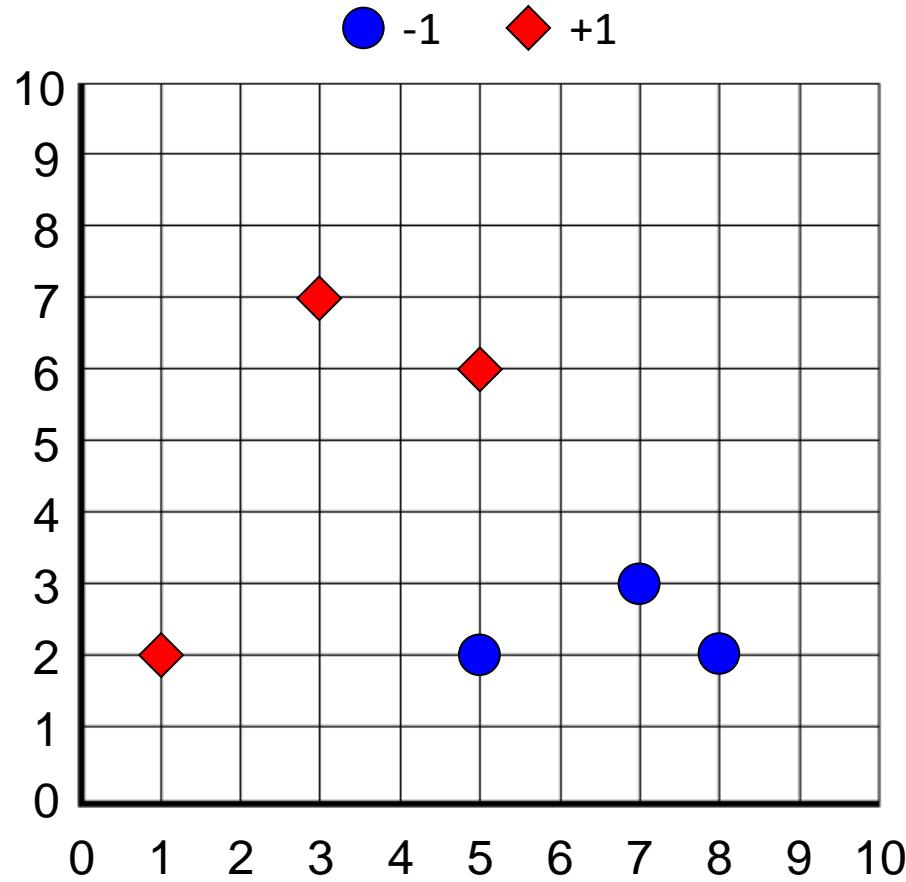
利用代入後的公式求  $w_1$  與  $w_2$  之極值

$$w_1 + 2w_2 + b \geq 1 \dots (1)$$

$$-5w_1 - 2w_2 - b \geq 1 \dots (3)$$

$$(1) + (3) \rightarrow -4w_1 \geq 2 \rightarrow w_1 \leq -\frac{1}{2}$$

## 範例二



$$\min_{b,w} \frac{1}{2} w^T w$$

$$\text{s. t. } y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N$$

步驟二

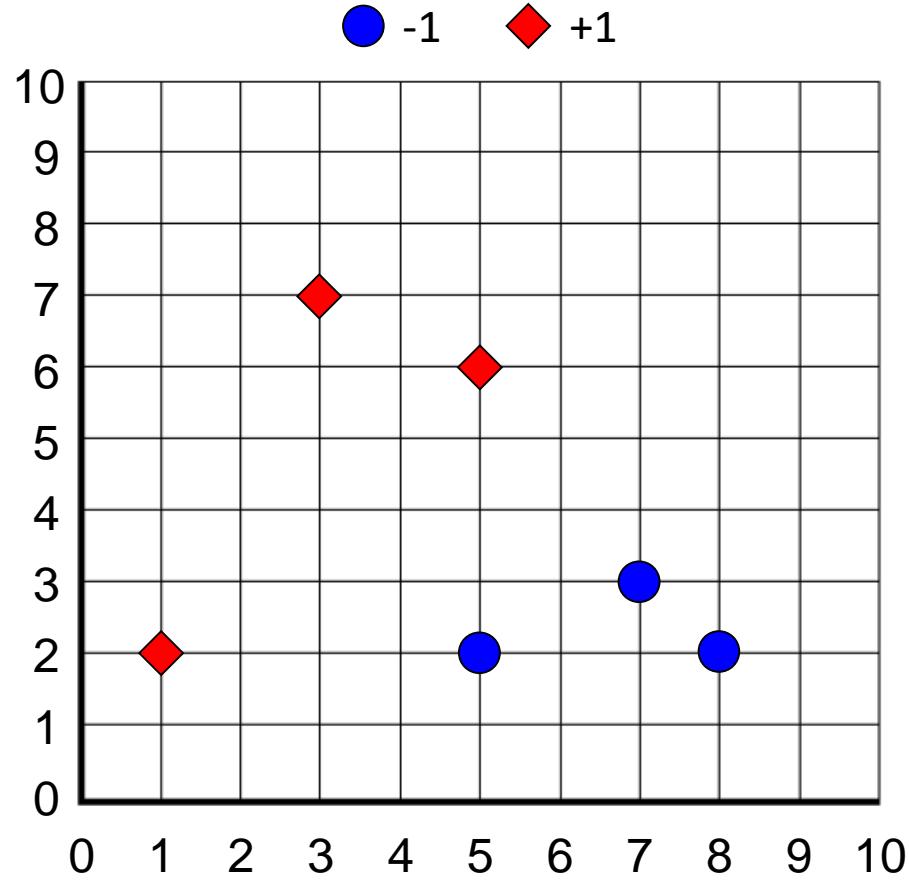
利用代入後的公式求  $w_1$  與  $w_2$  之極值

$$5w_1 + 6w_2 + b \geq 1 \dots (2)$$

$$-5w_1 - 2w_2 - b \geq 1 \dots (3)$$

$$(2) + (3) \rightarrow 4w_2 \geq 2 \rightarrow w_2 \geq \frac{1}{2}$$

## 範例二



$$\min_{b,w} \frac{1}{2} w^T w$$

$$\text{s. t. } y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N$$

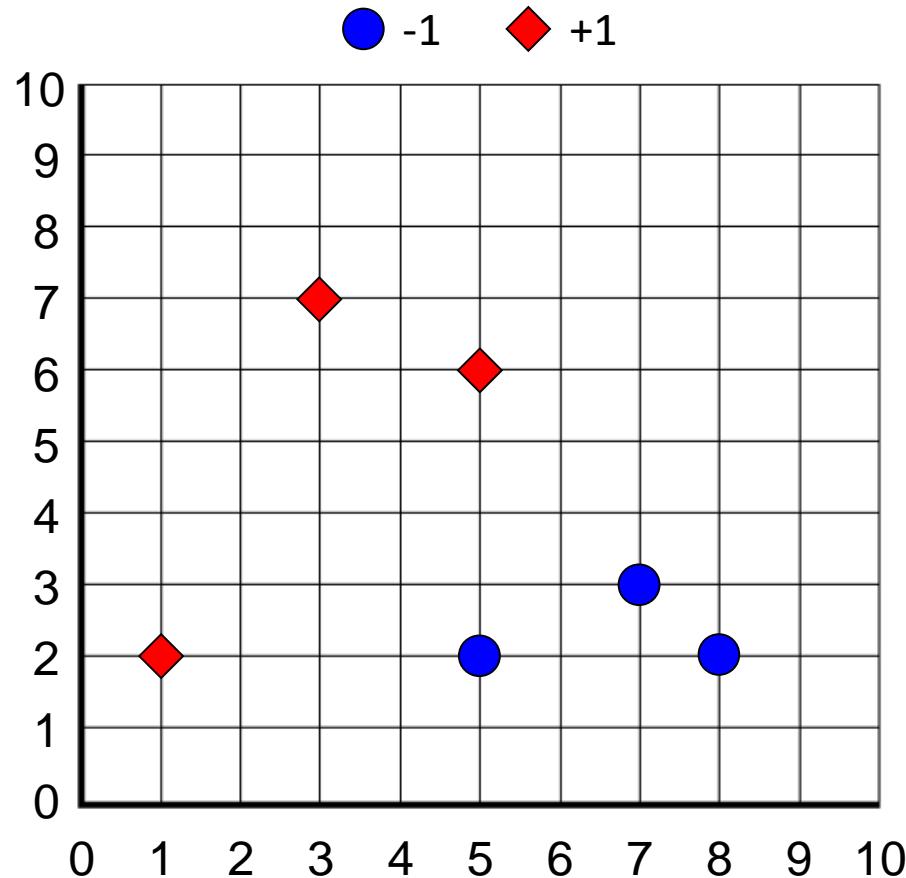
步驟二

利用代入後的公式求  $w_1$  與  $w_2$  之極值

$$(1) + (3) \rightarrow -4w_1 \geq 2 \rightarrow w_1 \leq -\frac{1}{2}$$

$$(2) + (3) \rightarrow 4w_2 \geq 2 \rightarrow w_2 \geq \frac{1}{2}$$

## 範例二



$$\min_{b,w} \frac{1}{2} w^T w$$

$$\text{s. t. } y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N$$

步驟三

將求得的  $w_1$  與  $w_2$  之極值代入(1)~(3)中  
(因為使用 (4) 故暫時不考慮)

任意公式求  $b$  值

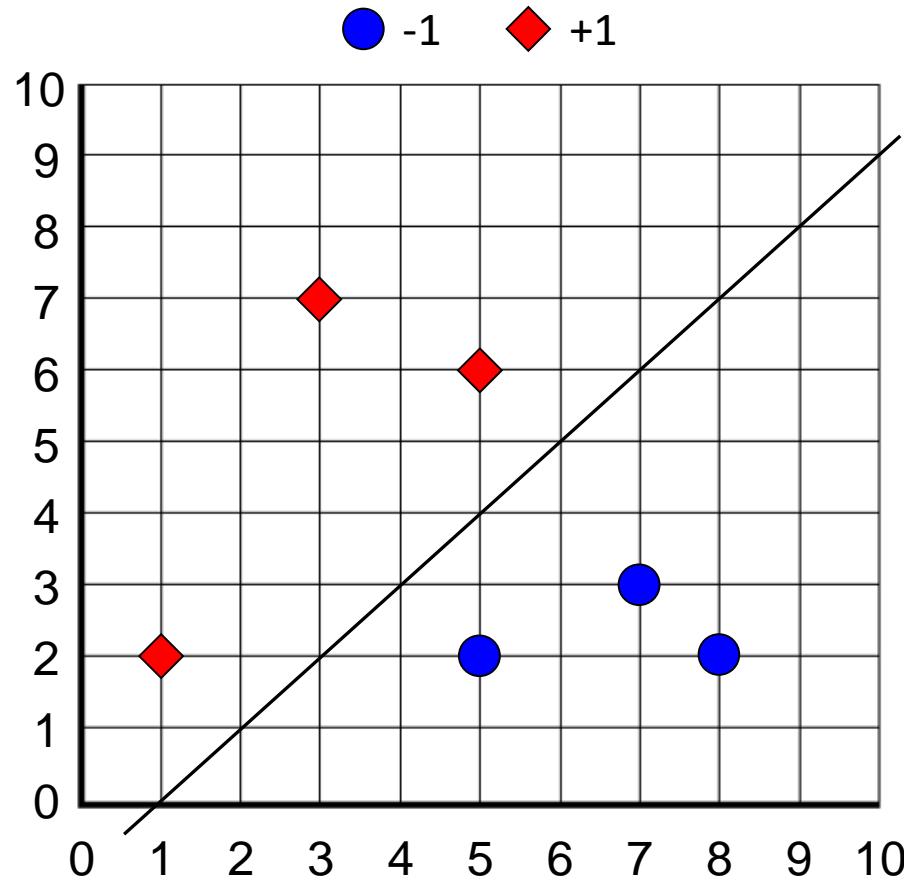
$$w_1 \leq -\frac{1}{2}, w_2 \geq \frac{1}{2}$$

將  $w_1$  &  $w_2$  極值代入 (1)

$$w_1 + 2w_2 + b \geq 1 \dots (1)$$

$$-\frac{1}{2} + 1 + b \geq 1 \rightarrow b \geq \frac{1}{2}$$

## 範例二



$$\min_{b,w} \frac{1}{2} w^T w$$

$$\text{s. t. } y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N$$

步驟四

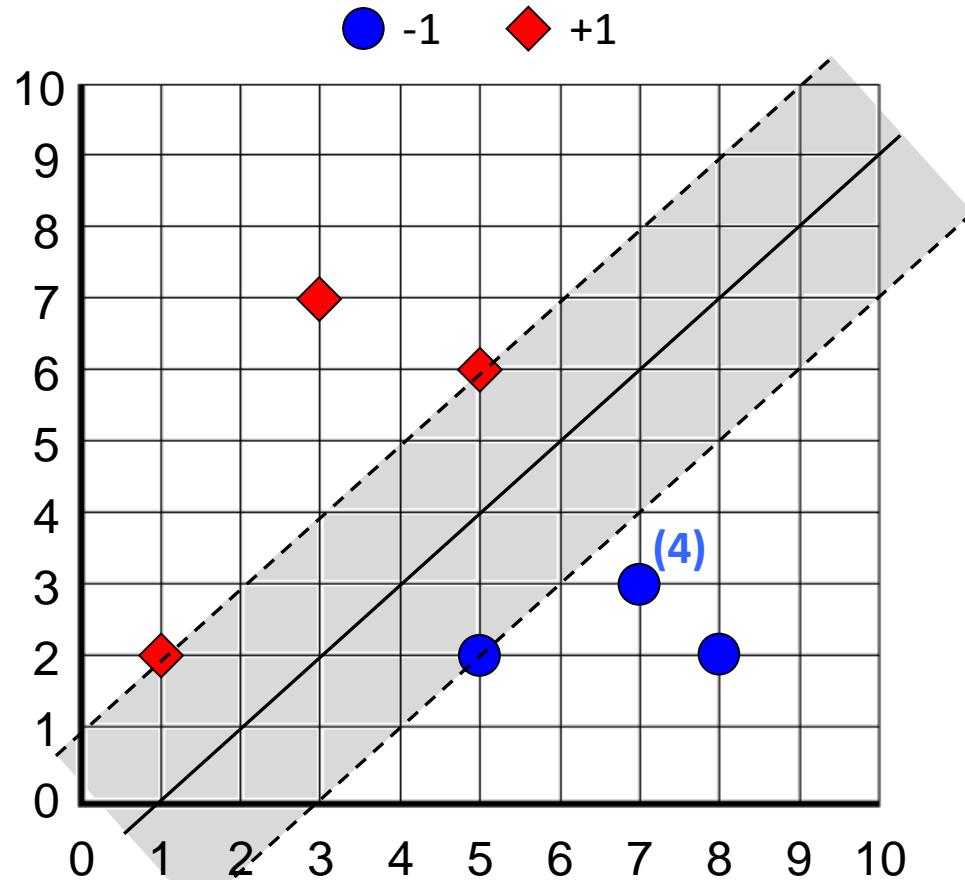
最後得到  $w_1, w_2$  與  $b$  值, 並求得  $g_{\text{SVM}}(x)$

透過  $\frac{1}{2} w^T w = \frac{1}{2} (w_1^2 + w_2^2)$  求得最小值

$(w_1 = -\frac{1}{2}, w_2 = \frac{1}{2}, b = \frac{1}{2} \text{ at lower bound})$

$$g_{\text{SVM}}(x) = -\frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{2} = 0 \quad \frac{1}{2} w^T w \geq \frac{1}{4}$$

## 範例二



$$\begin{aligned} & \min_{b,w} \frac{1}{2} w^T w \\ \text{s. t. } & y_n(w^T x_n + b) \geq 1, \text{ for } n = 1, 2, \dots, N \end{aligned}$$

步驟五  
利用  $g_{SVM}(x)$  求出正負 hyperplane

$$g_{SVM}(x) = -\frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{2} = 0$$

$$(-\frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{2}) \geq 1 \dots \text{positive hyperplane}$$

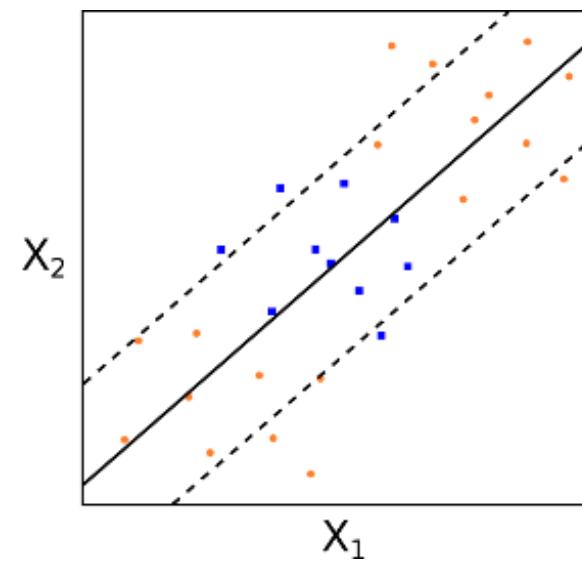
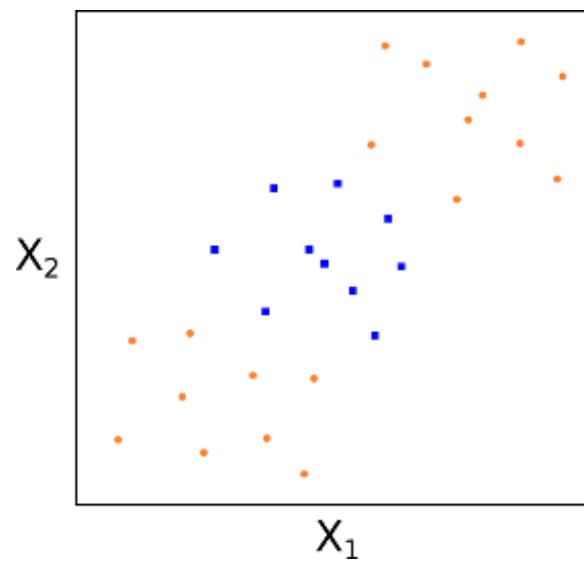
$$(-\frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{2}) \leq -1 \dots \text{negative hyperplane}$$

將點(4)代入:  $-\frac{7}{2} + \frac{3}{2} + \frac{1}{2} = -1.5 < -1$       不是 support vector

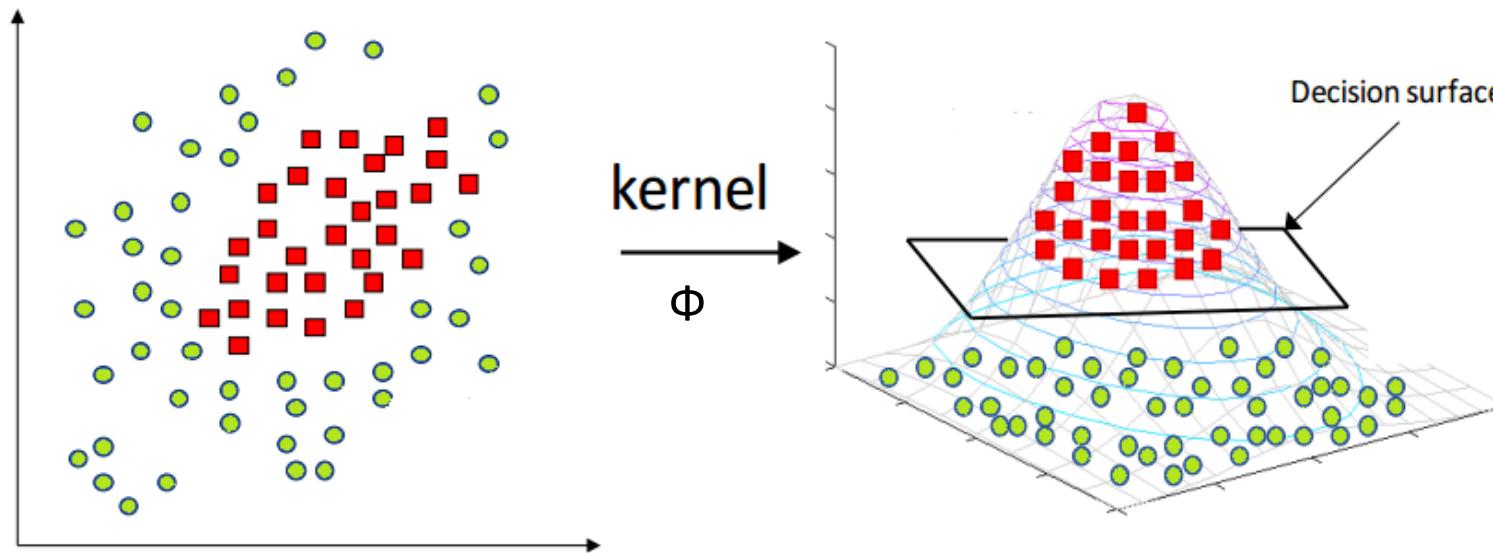
僅利用三個點就能找出  $g_{SVM}$

# Kernel Trick

問題: 沒有一條直線能將不同類別的資料分開



# Kernel Trick



在原始的輸入空間中  
為線性不可分資料

透過 Kernel 函式  
對特徵空間的轉換  
將資料在高維空間中  
轉換為線性可分

# Kernel: Transform + Inner Product

需要加快對  $z_n^T z_m = \Phi(x_n)^T (\Phi(x_m))$  的運算

$$\Phi_2(x) = (1, x_1, x_2, \dots, x_d, x_1^2, x_1 x_2, \dots, x_1 x_d, x_2 x_1, x_2^2, \dots, x_2 x_d, \dots, x_d x_1, \dots, x_d^2)$$

kernel function:  $K_{\Phi}(x, x') \equiv \Phi(x)^T \Phi(x')$

透過 kernel function 避免掉對  $z_n^T z_m = \Phi(x_n)^T (\Phi(x_m))$  的複雜運算

$$\text{transform } \Phi_2 \Leftrightarrow K_{\Phi_2}(x, x') = 1 + (x^T x') + (x^T x')^2 \quad \leftarrow \text{相對簡單許多}$$

$$\text{quadratic coefficient } q_{n,m} = y_n y_m z_n^T z_m = y_n y_m K(x_n, x_m)$$

相對複雜

非常複雜

# Gaussian Kernel (RBF Kernel)

Gaussian Kernel

可使用不同的核Kernel

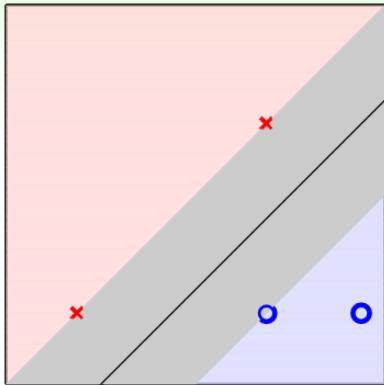
$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \text{ with } \gamma > 0$$

Gaussian Kernel SVM

代入  $K(x, x')$

$$g_{\text{SVM}} = \text{sign} \left( \sum_{\text{SV}} \alpha_n y_n K(x_n, x) + b \right) = \text{sign} \left( \sum_{\text{SV}} \alpha_n y_n \exp(-\gamma \|x - x'\|^2) + b \right)$$

# Pros and Cons - Linear Kernel

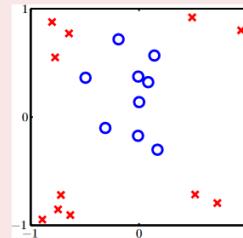


$$K(x, x') = \mathbf{x}^T \mathbf{x}'$$

線性核的缺點  
可能無法成功分類

## Cons

- restricted  
—**not always separable?**!



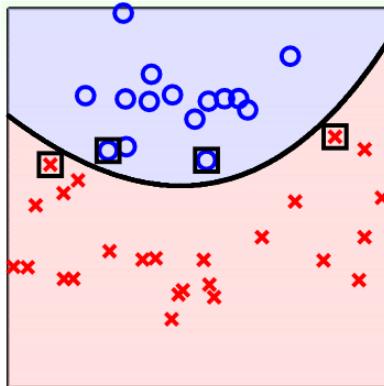
## Pros

- safe—**linear first, remember? :-)**
- fast—with **special QP solver** in primal
- very explainable—**w and SVs** say something

線性核的優點  
快速、解釋力高

linear kernel: an important basic tool

# Pros and Cons - Polynomial Kernel



$$K(\mathbf{x}, \mathbf{x}') = (\zeta + \gamma \mathbf{x}^T \mathbf{x}')^Q$$

## 多項式核的缺點

1. 數值分析困難
2. 3個參數與求解

## Cons

- numerical difficulty for large  $Q$ 
  - $|\zeta + \gamma \mathbf{x}^T \mathbf{x}'| < 1: K \rightarrow 0$
  - $|\zeta + \gamma \mathbf{x}^T \mathbf{x}'| > 1: K \rightarrow \text{big}$
- three parameters ( $\gamma, \zeta, Q$ )  
—more difficult to select

## Pros

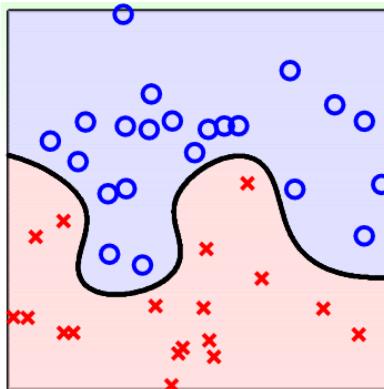
- less restricted than linear
- strong physical control  
—‘knows’ degree  $Q$

## 線性多项式核的優點：

1. 更少限制
2. 強力的物理控制參數  $Q$

polynomial kernel: perhaps small-Q only

# Pros and Cons - Gaussian Kernel



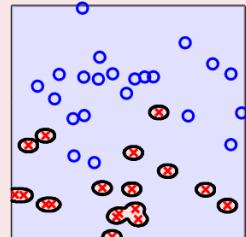
$$K(x, x') = \exp(-\gamma \|x - x'\|^2)$$

Gaussian kernel: one of most popular but shall be used with care

RBF核的缺點  
1. 分析速度較慢  
2. 需注意易過度適配

## Cons

- mysterious—no  $w$
- slower than linear
- too powerful?!

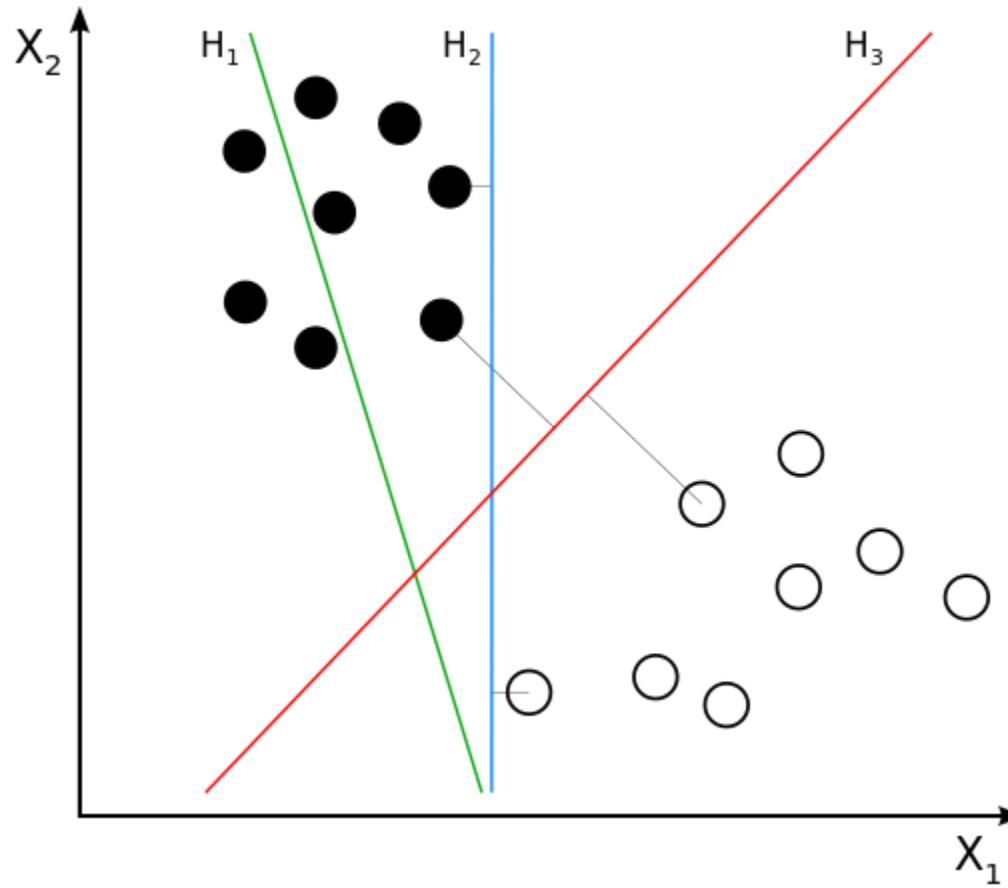


## Pros

- more powerful than linear/poly.
- bounded—less numerical difficulty than poly.
- one parameter only—easier to select than poly.

RBF核的優點：  
1. 比多項式核更強力的分類能力  
2. 數值分析較多項式核容易  
3. 參數較易於選擇

# SVM 直觀理解



- 線性可分SVM

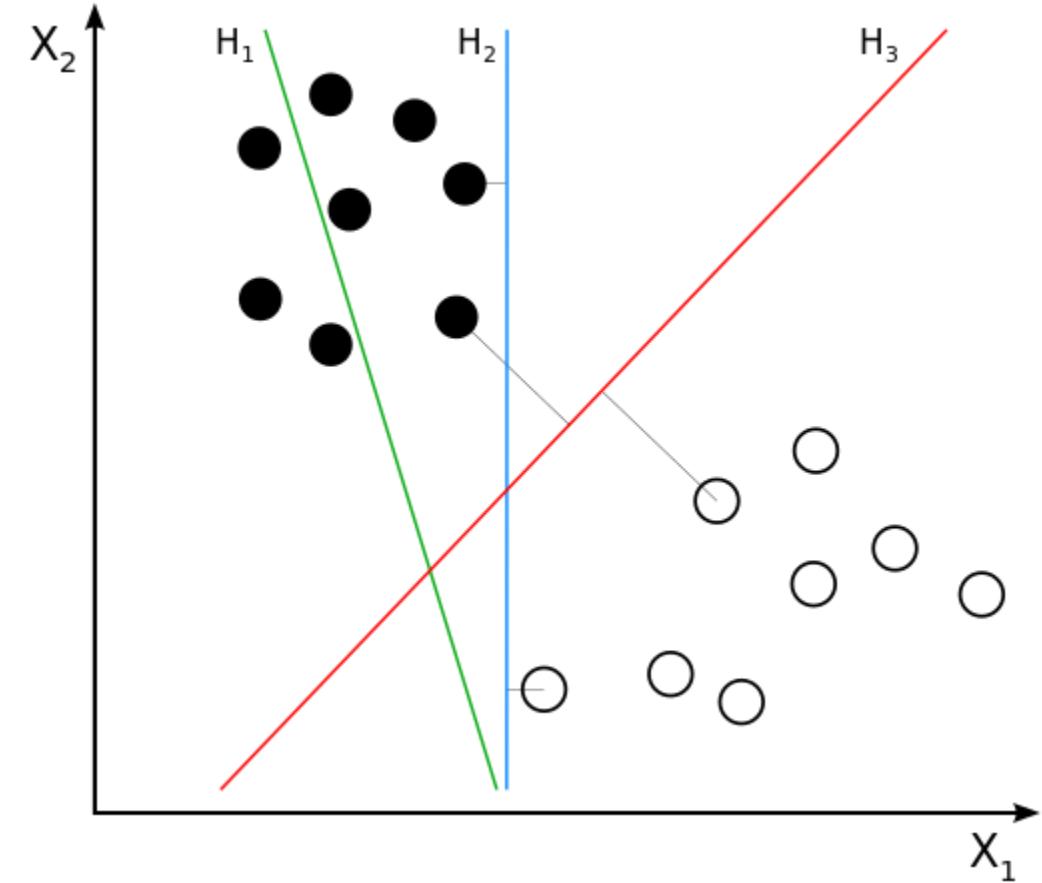
當訓練數據線性可分時，通過硬間隔(hard margin)最大化可以學習得到一個線性分類器，即硬間隔SVM，如圖的H<sub>3</sub>。

- 線性SVM

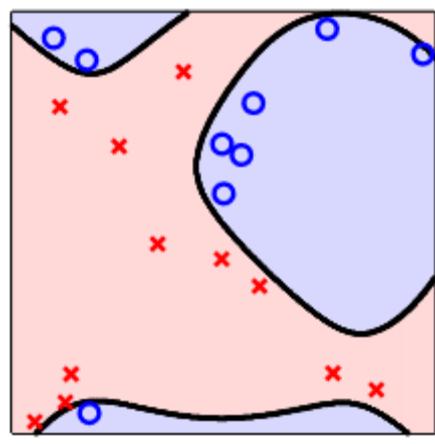
當訓練數據不能線性可分但是可以近似線性可分時，通過軟間隔(soft margin)最大化也可以學習到一個線性分類器，即軟間隔SVM。

- 非線性SVM

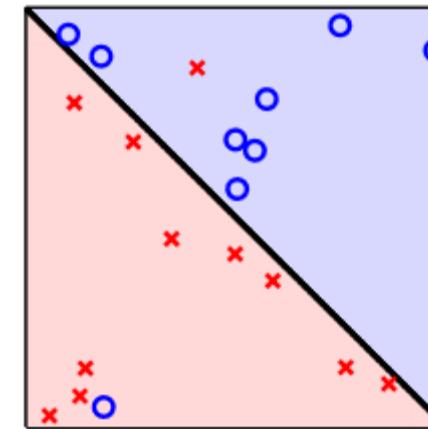
當訓練數據線性不可分時，通過使用核技巧(kernel trick)和軟間隔最大化，可以學習到一個非線性SVM。



## Cons of Hard-Margin SVM



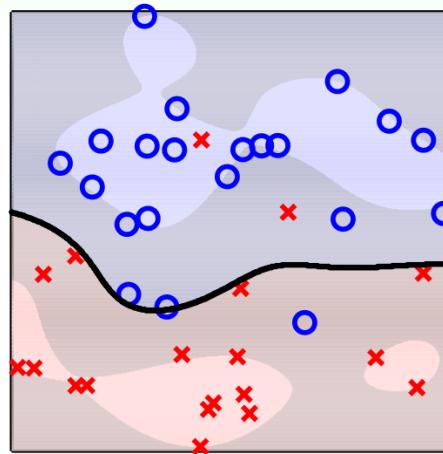
$\Phi_4$   
Hard-Margin SVM



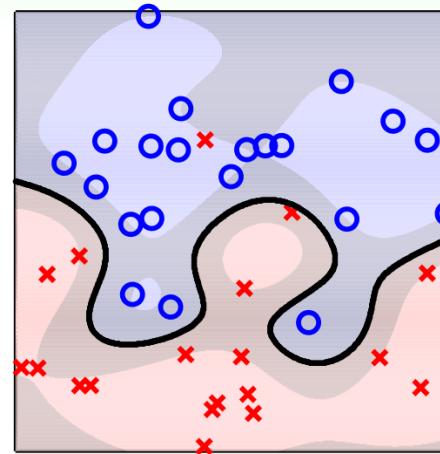
$\Phi_1$   
Soft-Margin SVM

- 訓練數據線性可分 -> 硬間隔支持向量機
- 訓練數據近似線性可分 -> 軟間隔支持向量機
- 訓練數據線性不可分 -> 非線性支持向量機 (核技巧)

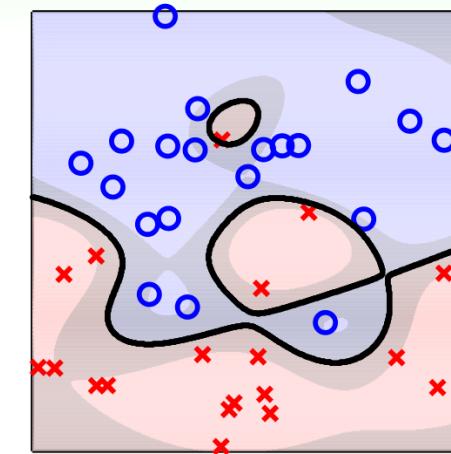
# Soft-Margin Gaussian SVM in Action



$$C = 1$$



$$C = 10$$



$$C = 100$$

large  $C \Rightarrow$  less noise tolerance  $\Rightarrow$  still maybe ‘overfit’!



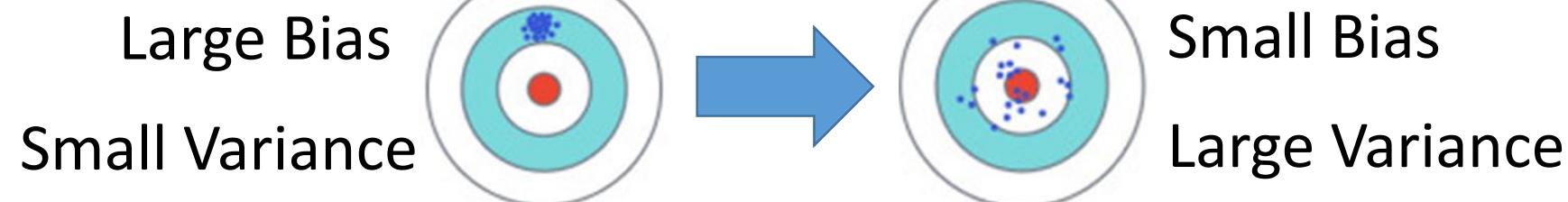
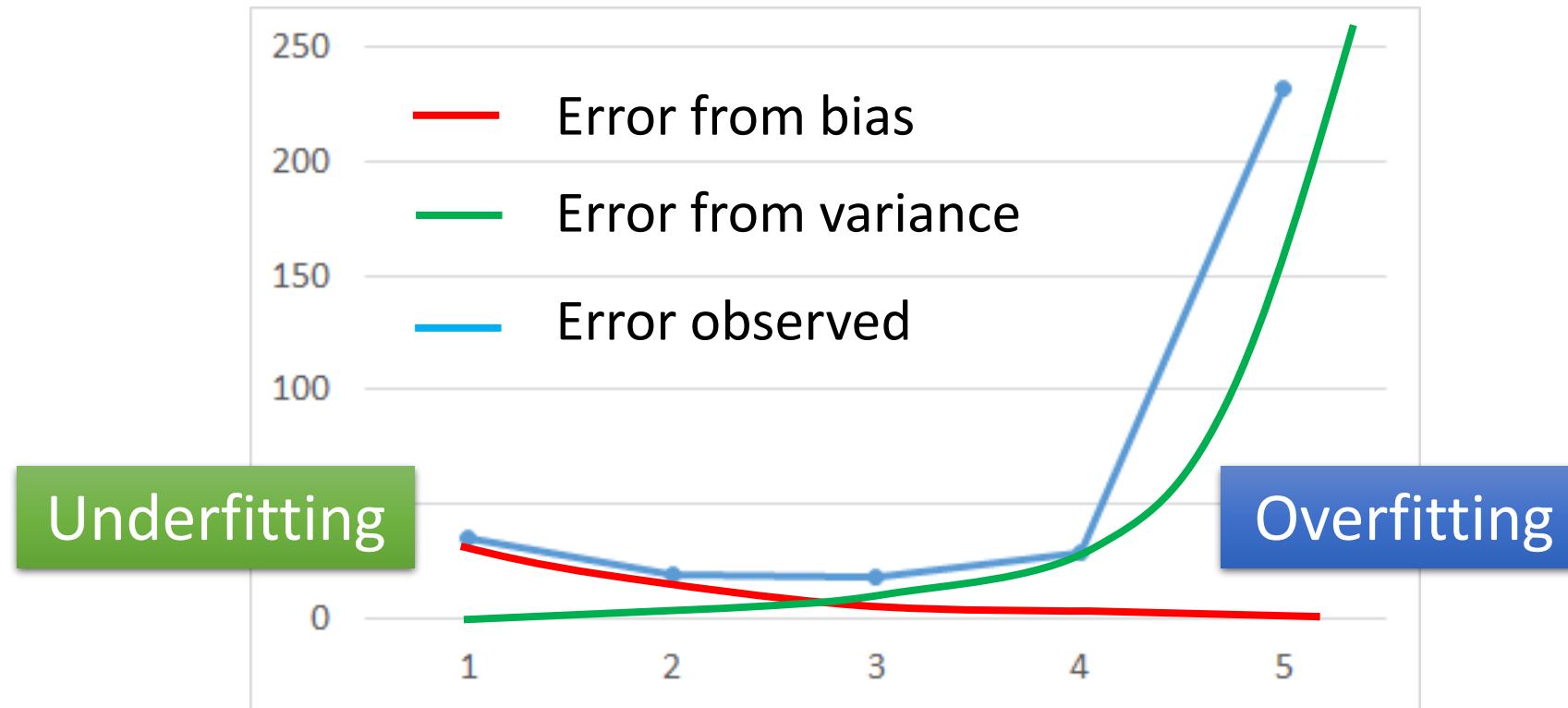
# Ensemble Learning

集成學習

## 集成學習(Ensemble learning)

- 基於Ensemble Method（集成方法）的想法是，如果單個分類器表現OK，那麼將多個分類器組合起來，其表現會優於單個分類器。也就是基於「人多力量大，三個臭皮匠勝過一個諸葛亮。」
- 滿足集成方法的條件
  - 各個分類器之間須具有差異性
  - 每個分類器的準確度必須大於0.5
- 常見的集成方法：
  - Bagging、Boosting、AdaBoost、Gradient Boost

# Review: Bias v.s. Variance



# Framework of Ensemble

- Get a set of classifiers
  - $f_1(x), f_2(x), f_3(x), \dots$

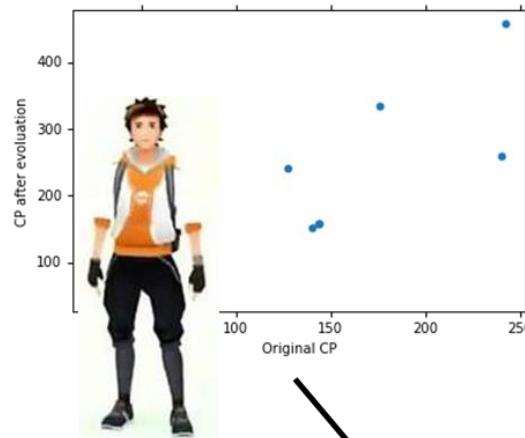
坦 補 DD

They should be diverse.

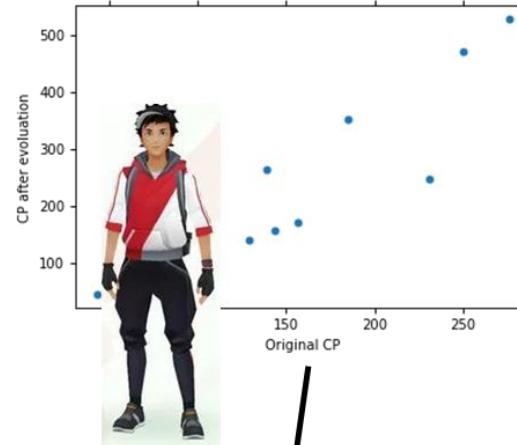
- Aggregate the classifiers (*properly*)
  - 在打王時每個人都有該站的位置

來源：<http://speech.ee.ntu.edu.tw/~tlkagk/courses.html>

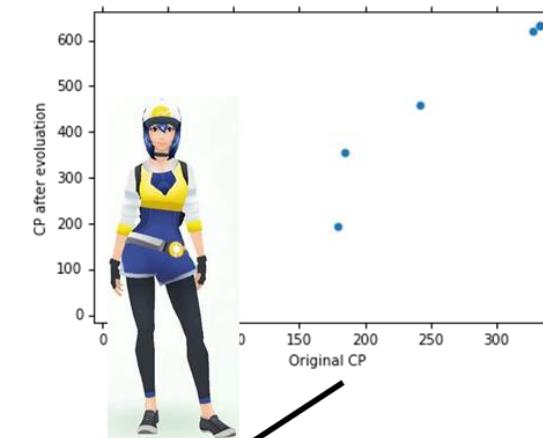
### Universe 1



### Universe 2



### Universe 3

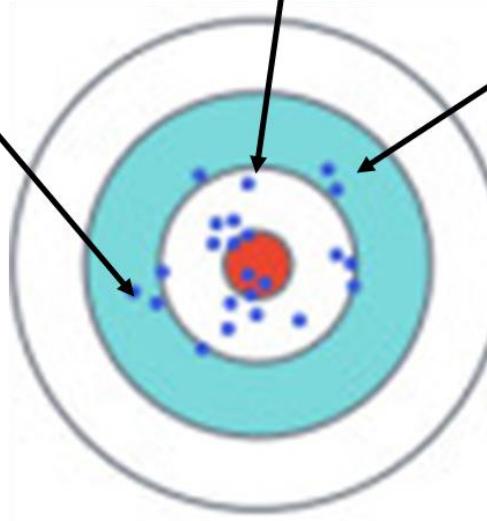


A complex model will have large variance.

We can average complex models to reduce variance.

If we average all the  $f^*$ , is it close to  $\hat{f}$

$$E[f^*] = \hat{f}$$

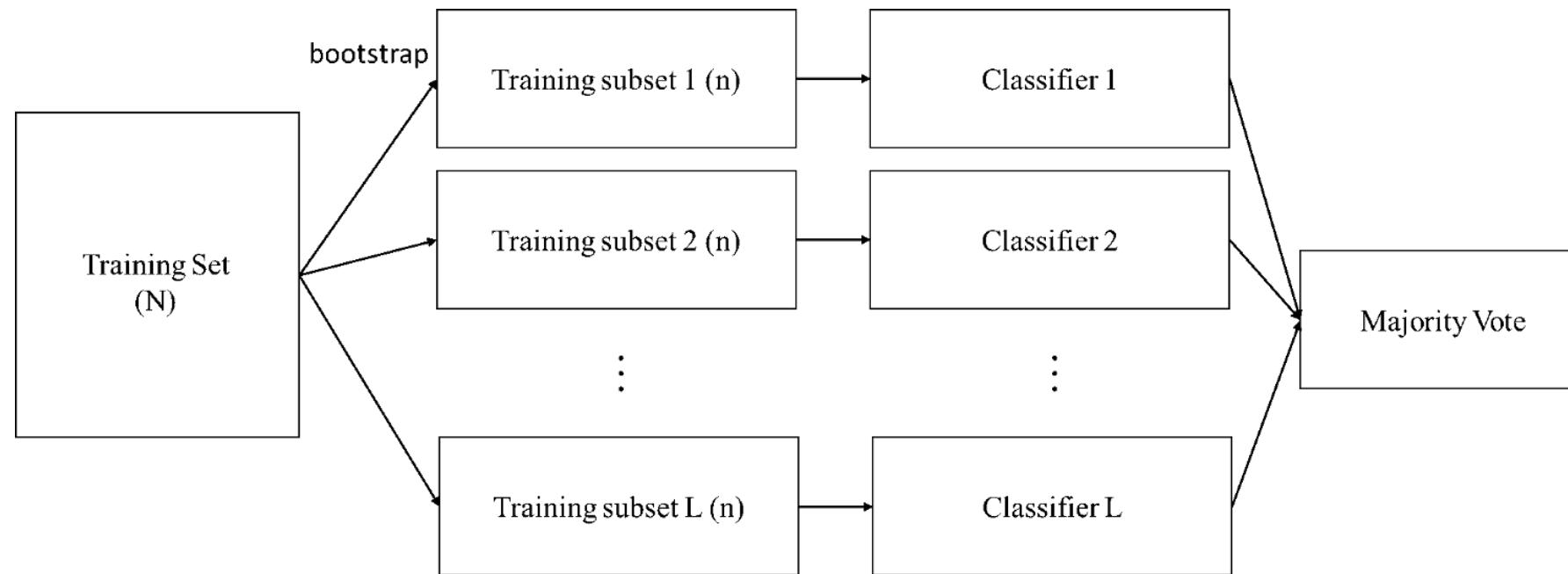


## Bagging (1/3)

- 1996年由Breiman提出
  - (Bagging是Bootstrap aggregating的縮寫)
- 透過Bagging，可讓模型從資料本身的差異中得到更好的訓練
- 從Training dataset中取出K個樣本，再從這K個樣本訓練出K個分類器（在此為tree）。每次取出的K個樣本皆會再放回母體，因此這個K個樣本之間會有部份資料重複，不過由於每顆樹的樣本還是不同，因此訓練出的分類器（樹）之間是具有差異性的

## Bagging (2/3)

- 從訓練資料中隨機抽取(取出後放回,  $n < N$ )樣本訓練多個分類器(要多少個分類器自己設定), 每個分類器的權重一致最後用投票方式(Majority vote)得到最終結果



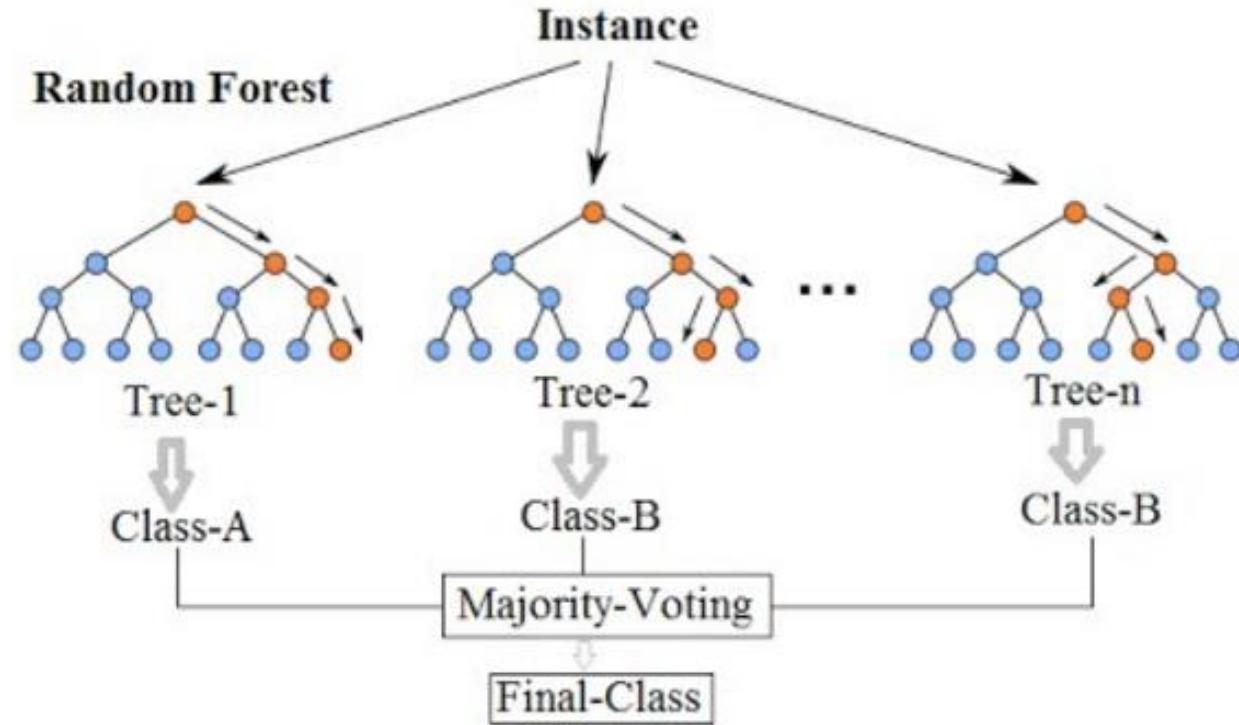
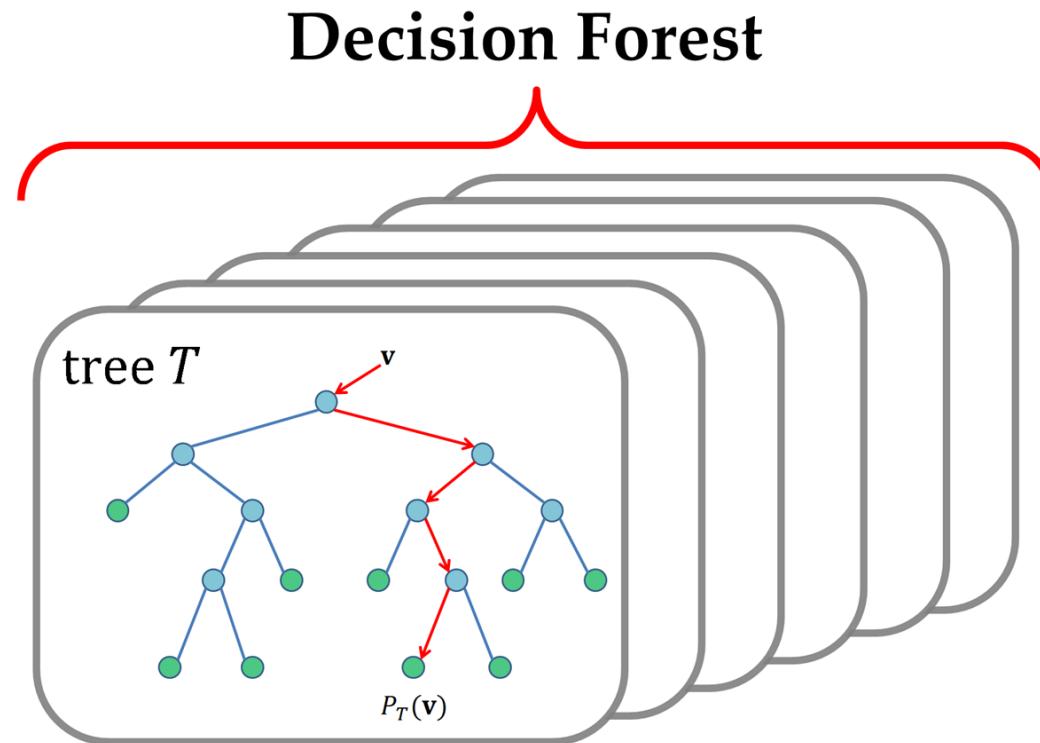
## Bagging (3/3)

- Bagging的精神在於從樣本中抽樣
  - 如果模型不是分類問題而是預測(連續變數)的問題，分類器部份也可以改成迴歸(regression)，最後投票方式改成算平均數即可
- 使用Bagging，則應選用效能比較好的分類器
- Bagging優點在於，若原始訓練樣本中有噪聲資料(不好的資料)，透過Bagging抽樣就有機會不讓有噪聲資料被訓練到，可以降低模型的不穩定性。

## 隨機森林演算法 (1/2)

- 結合**多顆CART樹** (CART樹為使用GINI演算法的決策樹) , 並加入**隨機分配的訓練資料**, 以大幅增進最終的運算結果

## 隨機森林演算法 (2/2)



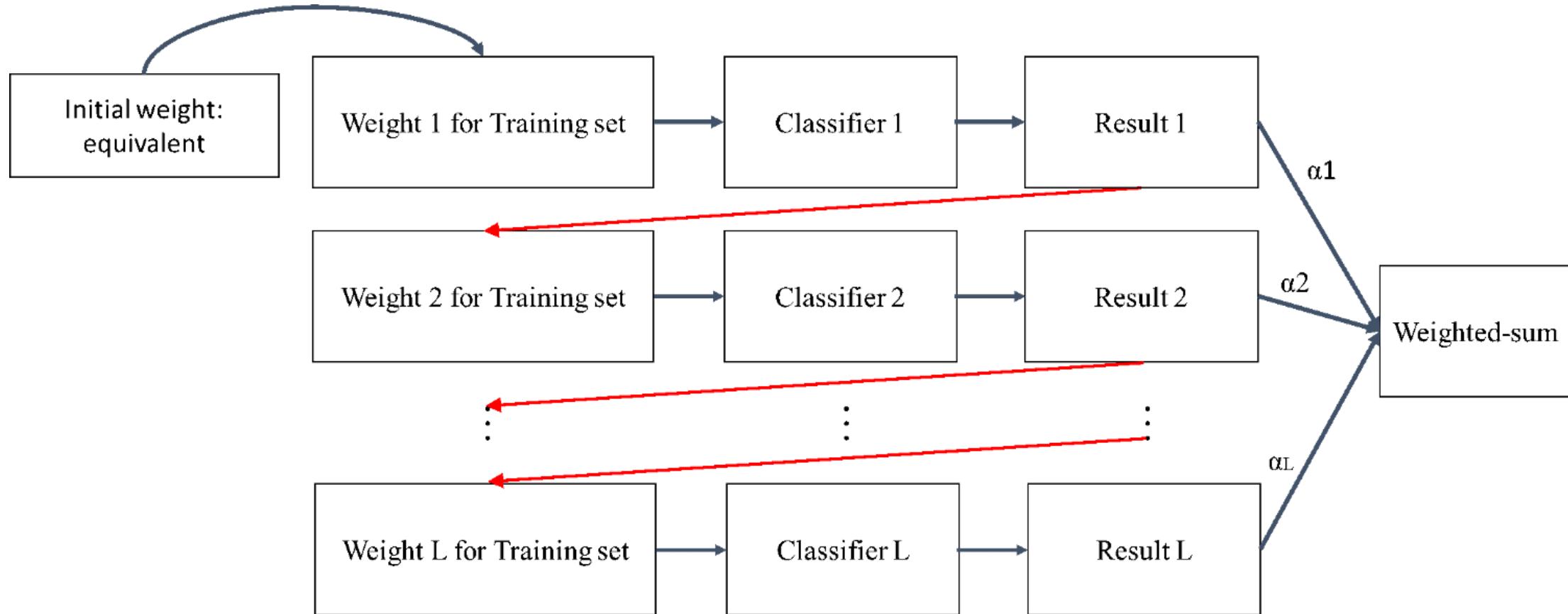
# Random Forest優點

- 對於很多種資料，它可以產生高準確度的分類器
- 它可以處理大量的輸入變數
- 它可以在決定類別時，評估變數的重要性
- 在建造森林時，它可以在內部對於一般化後的誤差產生不偏差的估計
- 它包含一個好方法可以估計遺失的資料，並且，如果有很大一部分的資料遺失，仍可以維持準確度
- 它提供一個實驗方法，可以去偵測variable interactions
- 對於不平衡的分類資料集來說，它可以平衡誤差
- 它計算各例中的親近度，對於數據挖掘、**偵測偏離者** (outlier) 和將資料視覺化非常有用
- 可被延伸應用在未標記的資料上，這類資料通常是使用非監督式聚類。也可偵測偏離者和觀看資料
- **學習過程快速**

## Boosting

- Boosting演算法是將很多個弱的分類器(weak classifier)進行合成變成一個強分類器(strong classifier)
- 和Bagging不同的是分類器之間是有關聯性的，是透過將舊分類器的錯誤資料權重提高，然後再訓練新的分類器，新分類器就會學習到錯誤分類資料(misclassified data)的特性，進而提升分類結果。

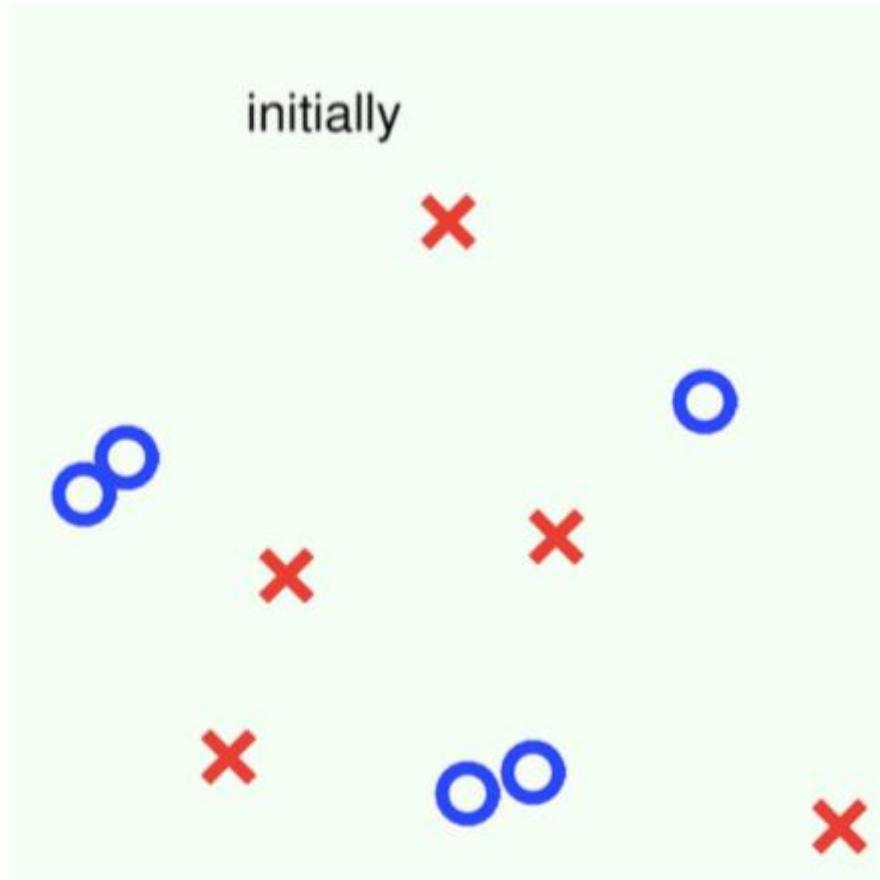
# Boosting



## Adaptive Boosting

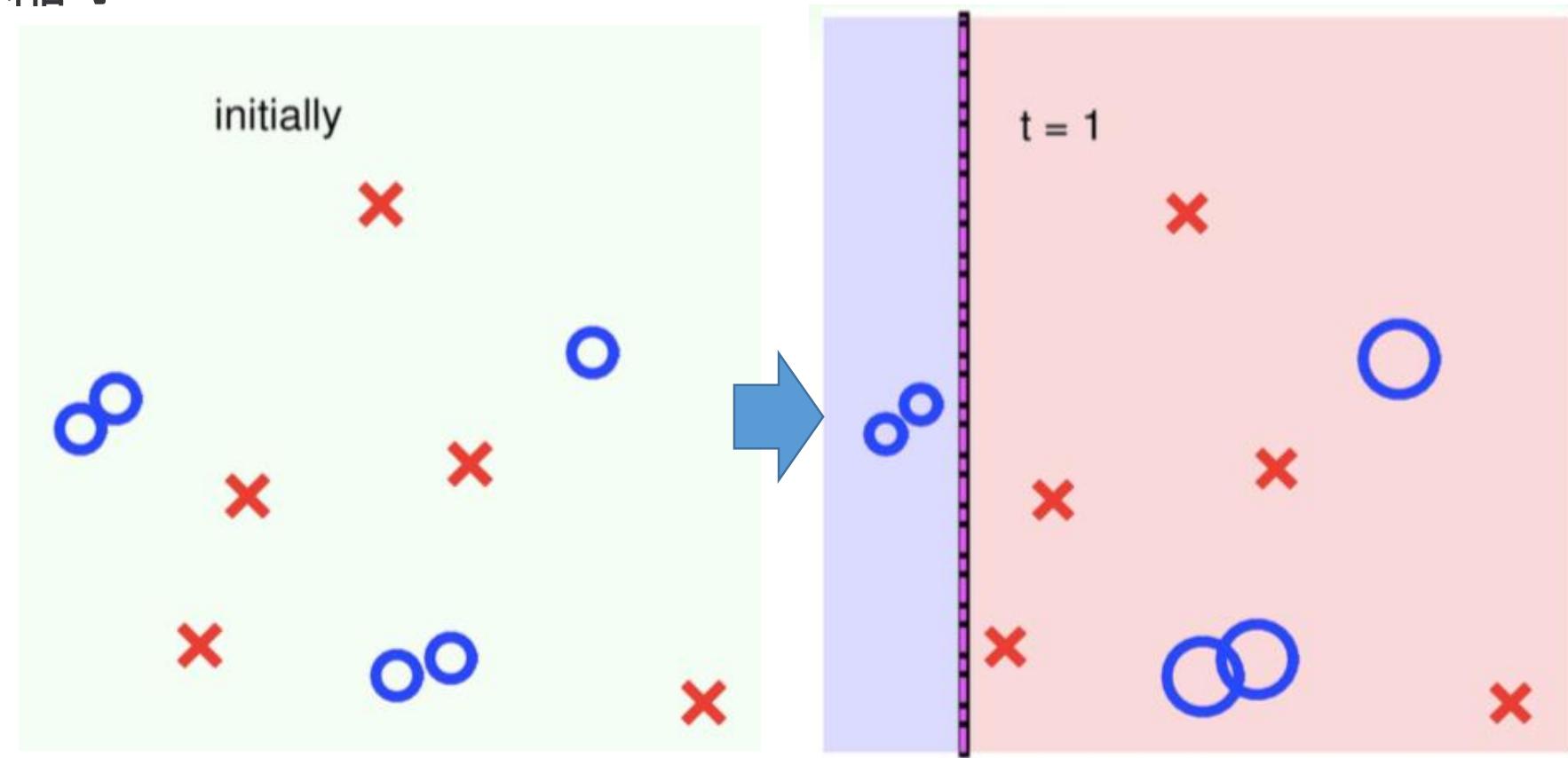
- 1995年由Yoav Freund和Robert Schapire提出
  - 1. 實施分類演算法
  - 2. 分錯的數據，增加其權重
  - 3. 再繼續實施分類演算法
- 這使得誤差大幅增加，使得分界線大幅靠近分錯的數據，進而迅速減少分錯的數據
- 不容易出現過度擬合(Over-fitting)

# AdaBoost 的演算過程



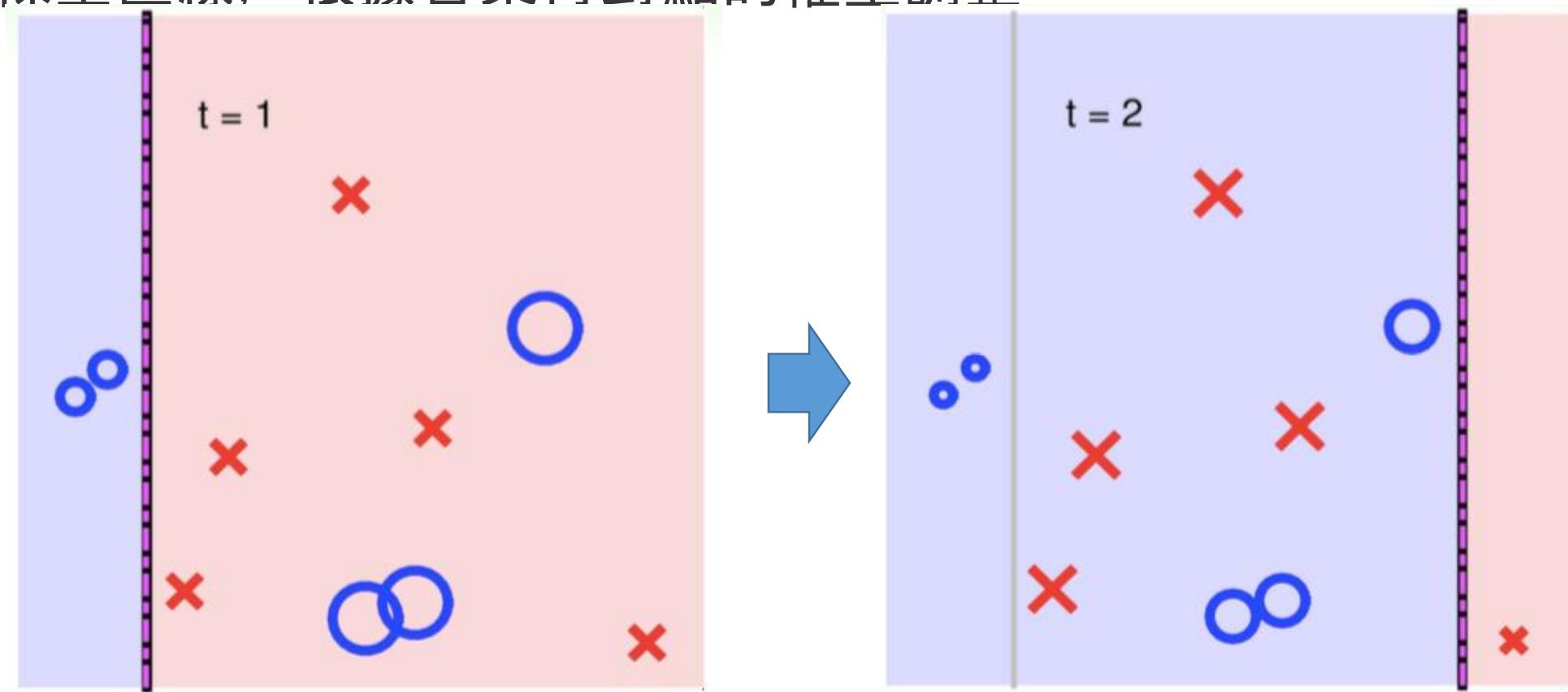
# Adaptive Boosting

- 第一輪先學出一個 learner 切一個垂直線，犯錯的點會放大、答對的點縮小



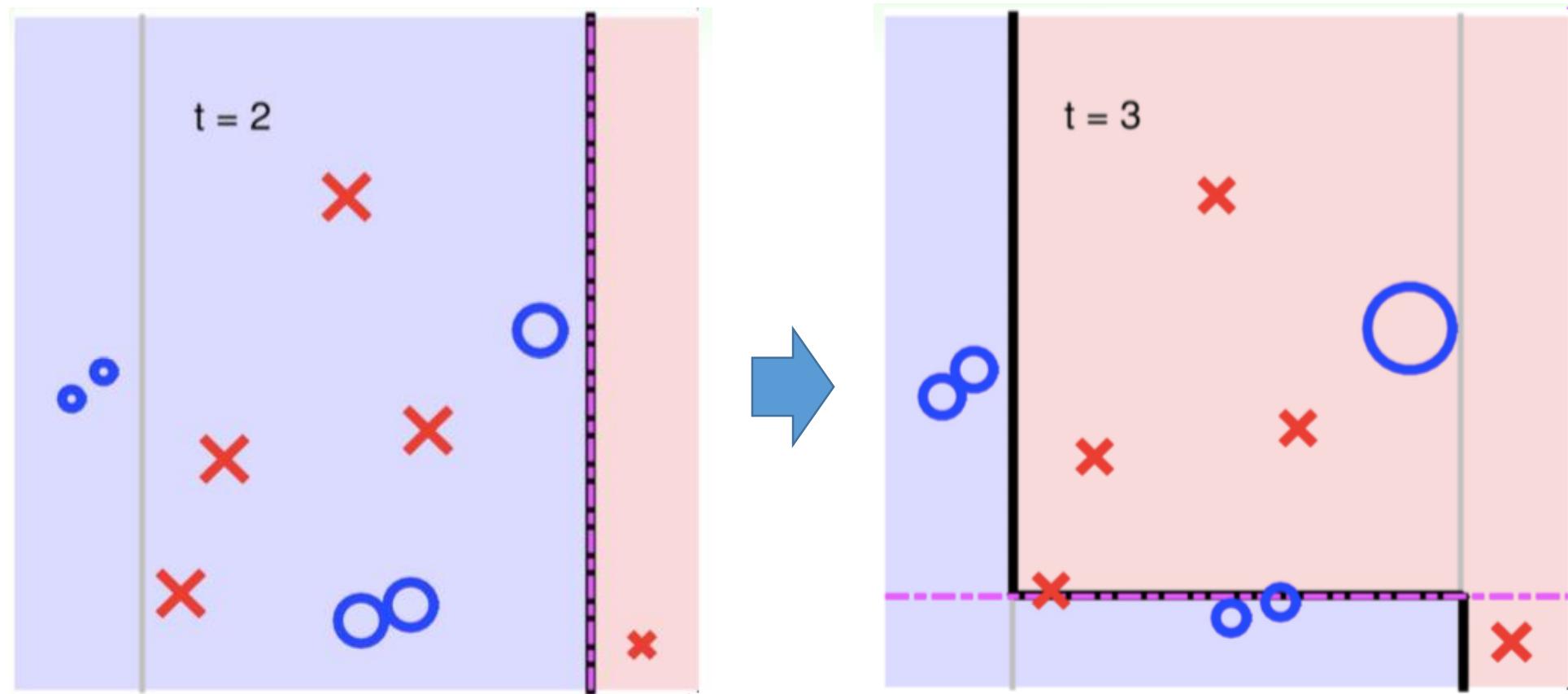
## Adaptive Boosting

- 第二輪根據犯錯放大的權重，再去學出一個不同觀點的 learner 切一條垂直線，根據答案再對點的權重調整



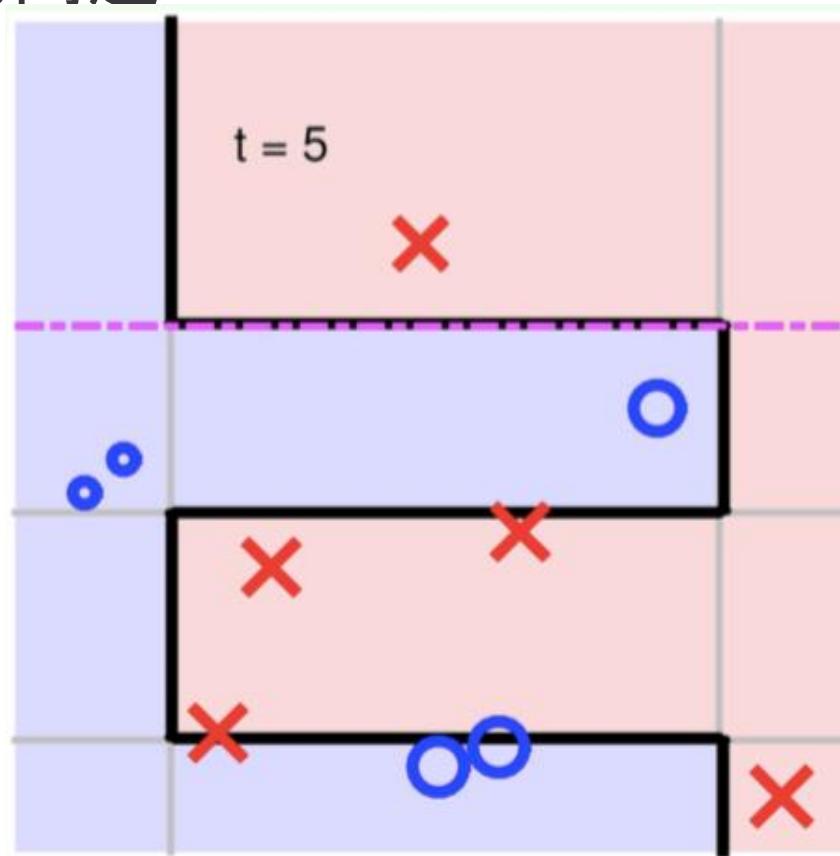
## Adaptive Boosting

- 第三輪又學出了一個不同觀點的 learner 切了一條水準線，現在已經可以看出分界線慢慢變得複雜了



## Adaptive Boosting

- AdaBoost 不斷地得出不同的 classifier，綜合 classifier的答案便可以回答一些較複雜的問題

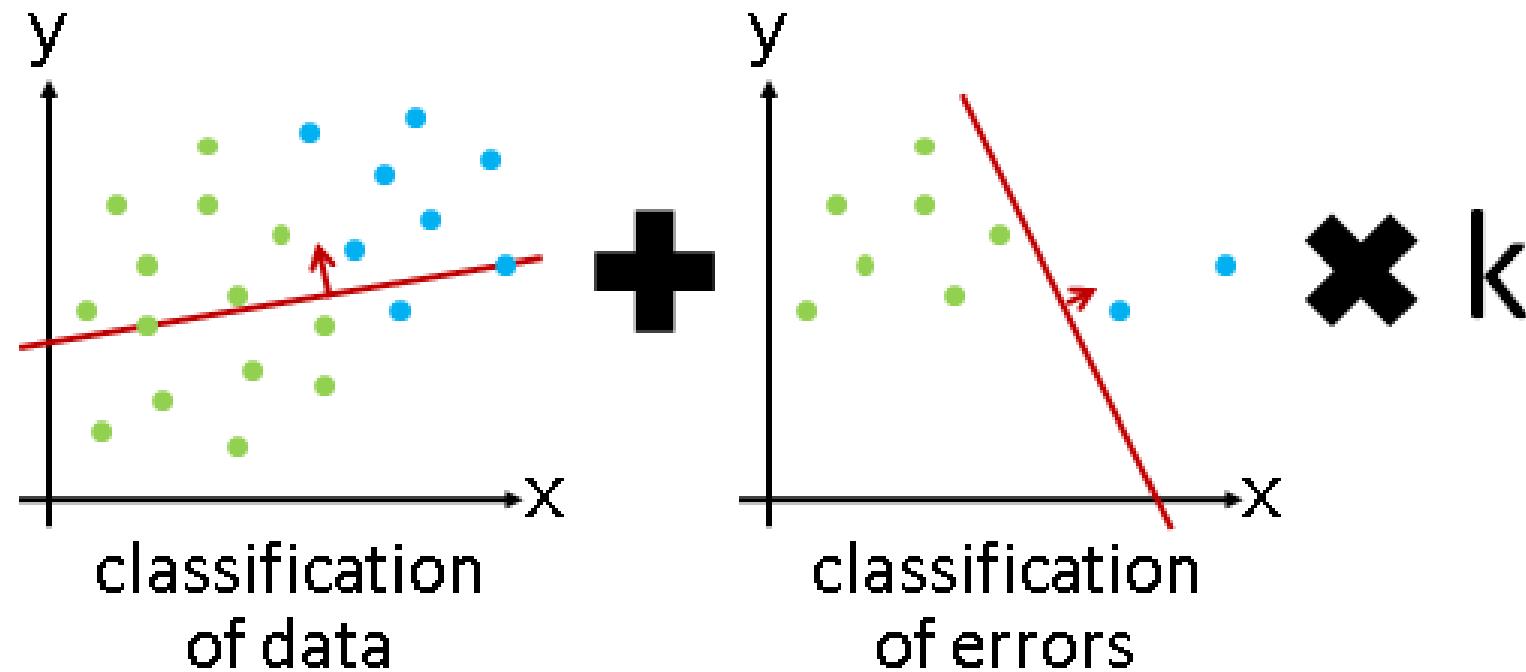


# Bagging與Boosting的區別

- 訓練樣本:
  - Bagging: 每一次的訓練集是隨機抽取(每個樣本權重一致), 抽出可放回, 以獨立同分佈選取的訓練樣本子集訓練弱分類器。
  - Boosting: 每一次的訓練集不變, 訓練集之間的選擇不是獨立的, 每一次選擇的訓練集都是依賴上一次學習得結果, 根據錯誤率(給予訓練樣本不同的權重)取樣。
- 分類器:
  - Bagging: 每個分類器的權重相等, 每個分類器可以並行生成。
  - Boosting: 每個弱分類器都有相應的權重, 對於分類誤差小的分類器會有更大的權重。每個弱分類器只能依賴上一次的分類器順序生成。

# Gradient Boosting

- 實施分類演算法，得到分界線。然後**不斷微調分界線**
- **挑出分錯的數據**，另外實施分類演算法，得到微調用的分界線。  
當前分界線，加上微調用的分界線，完成一次**微調**



## 其他集成學習方法

- 將不同的分類器進行合成提高單一分類器的效果
  - 例如SVM+k-NN+MLP等。
- 很多個SVM合成，方式為每個SVM給不同的kernel function或是kernel參數。
- Random subspace: 又稱feature bagging，從特徵中去抽樣，然後訓練多個分類器做合成，通常用在非常高維度的資料中。當然也有衍生出來的feature Adaboosting。

## 其他集成學習方法

- 近期流行的Decision tree集成學習方法：
  - Random Forest : Bagging + Decision tree
  - Boosting Tree : AdaBoost + Decision tree
  - GBDT : Gradient Boost + Decision tree

## XGBoost (eXtreme Gradient Boosting)

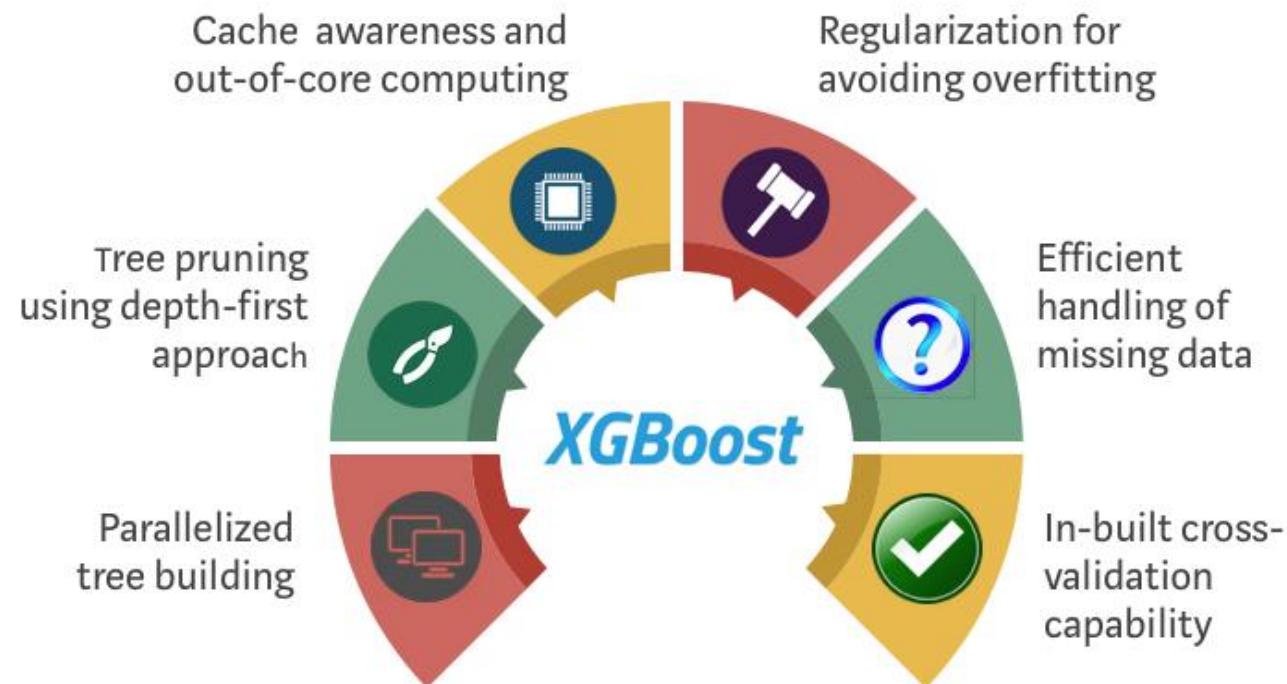
- Kaggle競賽神器：在Kaggle許多比賽的第一名都使用了XGboost，在2015年的時候29個Kaggle冠軍隊伍中有17隊使用了Xgboost。
- XGBoost是一個開源軟體函式庫，提供**極限梯度提升**框架，採用C++, Java, Python, R和Julia
- 它的目的在於提供一個**“可擴展的、可攜式和可分佈的梯度提高(GBM,GBRT,GBDT)函式庫”**。
- 支援分散式計算框架 Apache Hadoop, Apache Spark, Apache Flink。

## XGBoost

- XGBoost的聯合創始人之一陳天奇(Tianqi Chen)在2016年宣佈，XGBoost的創新系統特性和演算法優化使其速度**比大多數機器學習解決方案快10倍**。
- XGBoost是一個**優化的分散式梯度增強函式庫**，具有高效、靈活和可攜性。在梯度增強框架下實現了機器學習演算法
- XGBoost提供了一種**並行樹增強(也稱為GBDT、GBM)**，可以快速、準確地解決許多資料科學問題。相同的代碼在主要的分散式環境(Hadoop、SGE、MPI)上運行，可以解決**數十億個示例**之外的問題。

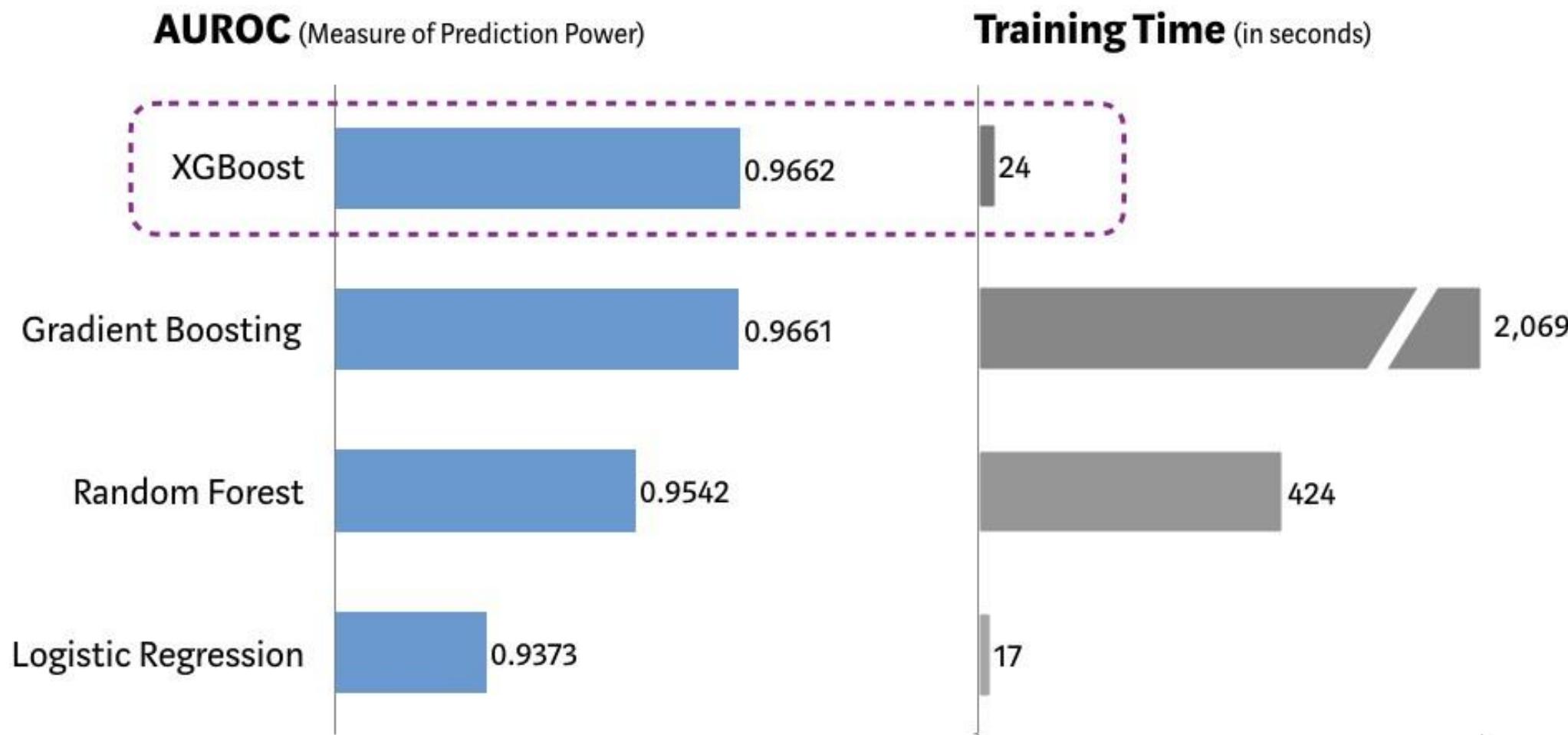
# XGBoost表現出色的原因

- XGBoost和梯度增強機(GBMs)都是集成樹方法，利用梯度下降體系結構實現了對弱學習器的增強。
- 然而，XGBoost通過系統優化和演算法增強對基本GBM框架進行了改進。



# Performance Comparison using SKLearn's 'Make\_Classification' Dataset

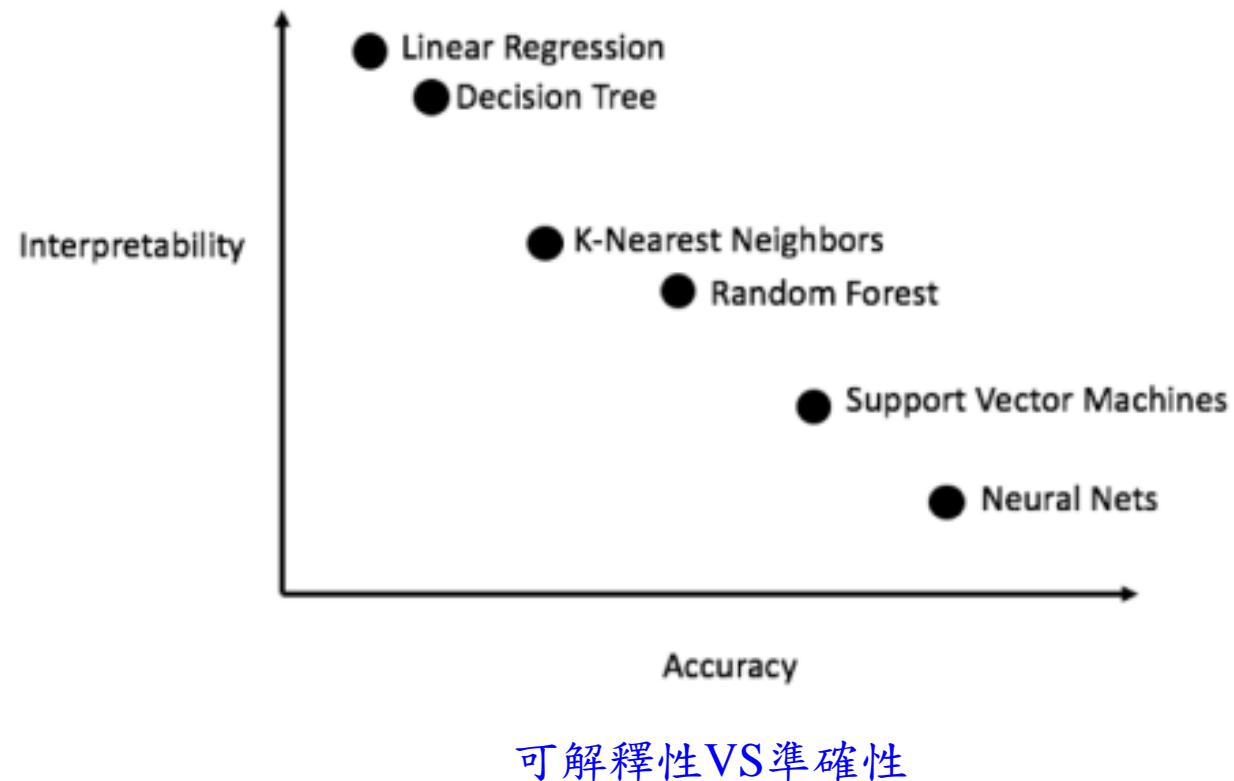
(5 Fold Cross Validation, 1MM randomly generated data sample, 20 features)



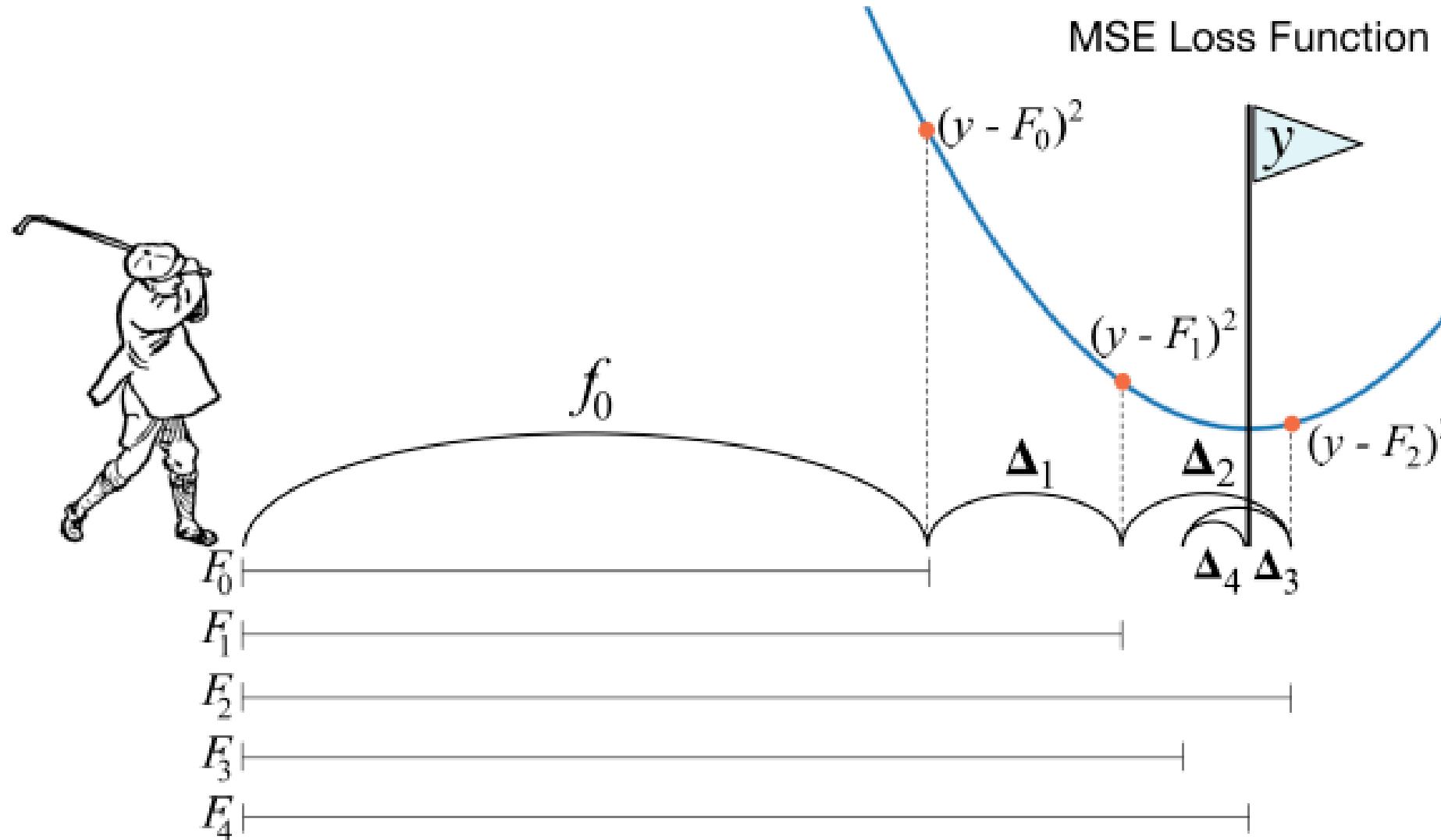
<https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d>

# Gradient Boosting Machine (GBM)的發展近況

- GBM將決策樹的精度提高到接近神經網路的精度水準。
- 它的方法是單一決策樹不夠強大，但可應用至基於前一輪誤差的決策樹演算法，越來越接近甚或超過神經網路的精度。
- 重要關鍵：**結果可以解釋。**



# 打高爾夫球很接近GBM的運作概念



# k-Nearest Neighbors

## 最近鄰居法

# 最近鄰居法 k-Nearest Neighbors

晚上俊傑騎著機車正在回家的路上，不小心捲進了一群飆車族的械鬥，他環視週遭發現總共有三群飆車族，身上的衣服分別是黃、灰、紅。俊傑已處於車陣中心，處境非常的危險，現在俊傑有兩條路可以選：

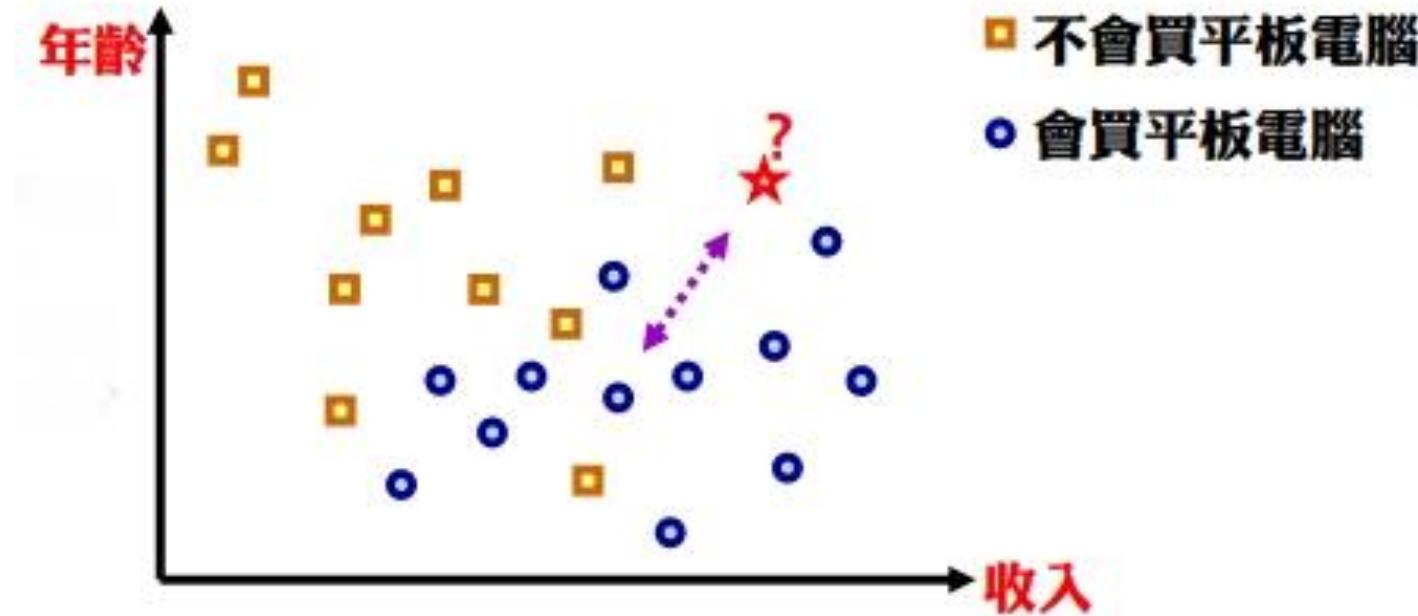
- 1.不加入任何陣營，來一個打一個，以一擋百。
- 2.選擇加入某個陣營

基本上選了1就沒什麼好討論了，所以我們先假設他選2好了。俊傑本身就是個識時務的人，所以他看了一下離他最近的7個人( $k=7$ )，發現有黃色衣服的有4個，灰色有2個，紅色有一個，所以他當下立刻從包包來出一件黃色外套，加入黃色陣營，這就是所謂的「西瓜偎大邊」。從這個故事我們了解，俊傑本身應該是有學過KNN演算法的。

# 最近鄰居法 k-Nearest Neighbors

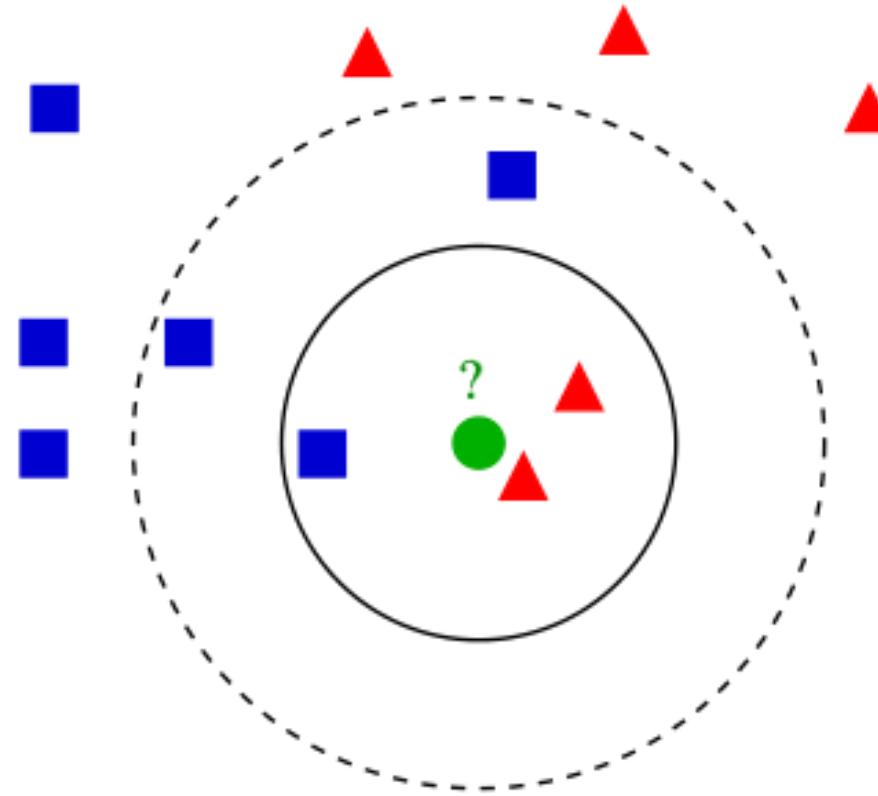
- 1967年，由 Cover 和 Hart 所提出
- 根據基礎為同物以類聚  
*近朱者赤，近墨者黑*  
*西瓜偎大邊*  
*孟母三遷*
- 目的：分類  
屬於機器學習中的監督式學習(Supervised learning)  
根據現有已分類好的資料集合，找出與**待分類資料最為鄰近資料**，根據此**最鄰近資料的所屬類別**，對待分類資料進行類別判定或預測

# 最近鄰居法 k-Nearest Neighbors



- 找尋待分類資料與資料集合內各資料點之間的距離。
- 與待分類資料距離最短之現存資料的類別，即為該待分類資料之所屬類別

# 最近鄰居法 k-Nearest Neighbors



圖中的綠色圓點，是屬於哪一類呢

# 最近鄰居法 k-Nearest Neighbors

## KNN演算法步驟

步驟1：確定參數K = 最近鄰居的數量

步驟2：計算查詢實例和所有訓練樣例之間的距離。

步驟3：根據第k個最短距離進行排序並確定最近的鄰居。

步驟4：收集最近鄰居的類別Y。

步驟5：使用最近鄰居多數類別作為查詢實例的預測值。

*k*-Nearest Neighbor

Classify ( $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $x$ ) //  $\mathbf{X}$ : training data,  $\mathbf{Y}$ : class labels of  $\mathbf{X}$ ,  $x$ : unknown sample

**for**  $i = 1$  **to**  $m$  **do**

    Compute distance  $d(\mathbf{X}_i, x)$

**end for**

    Compute set  $I$  containing indices for the  $k$  smallest distances  $d(\mathbf{X}_i, x)$ .

**return** majority label for  $\{\mathbf{Y}_i \text{ where } i \in I\}$

# 最近鄰居法 k-Nearest Neighbors

## 距離函數

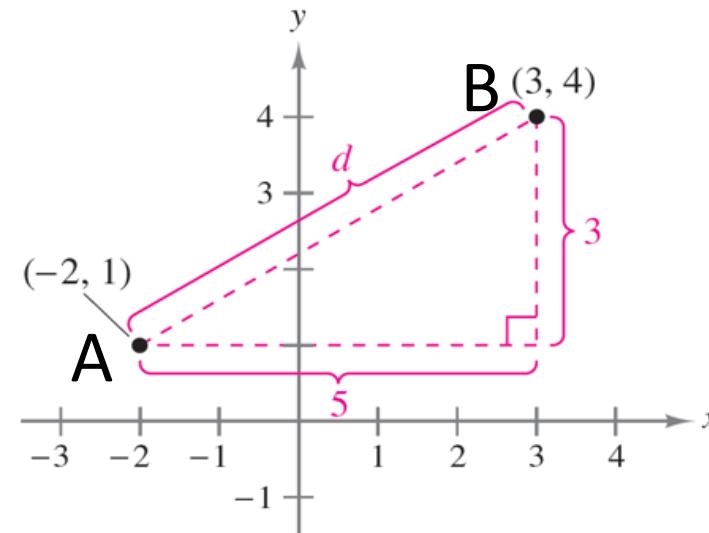
- 負責判斷兩筆資料差異到底有多大
- 距離愈小：表示差異愈小
- 通常採用歐幾里德距(Euclidean Distance)或曼哈頓距離  
假設兩物的位置：

- $X = (x_1, x_2, x_3, \dots, x_n)$  和  $Y = (y_1, y_2, y_3, \dots, y_n)$

- 則公式為  $D(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$

- 例子  $A(-2, 1), B(3, 4)$

$$\begin{aligned}\overline{AB} &= \sqrt{(3 - (-2))^2 + (4 - 1)^2} \\ &= \sqrt{34}\end{aligned}$$



# 最近鄰居法 k-Nearest Neighbors

## KNN 優缺點

### 優點

易於理解，可以用來做分類也可以用於處理回歸問題

### 缺點

計算量大，空間複雜度高，需要大量記憶體

# 最近鄰居法 k-Nearest Neighbors

- 測試當  $X_1=3$ ,  $X_2=7$  為哪個屬性  
當  $K=3$

X1(value)	X2(value)	Y( Classification)
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

# 最近鄰居法 k-Nearest Neighbors

X1(value)	X2(value)	Square Distance to query instance (3,7).
7	7	$(7-3)^2 + (7-7)^2 = 16$
7	4	$(7-3)^2 + (7-4)^2 = 25$
3	4	$(3-3)^2 + (7-4)^2 = 9$
1	4	$(3-1)^2 + (7-4)^2 = 13$

# 最近鄰居法 k-Nearest Neighbors

X1(value)	X2(value)	Square Distance to query instance (3,7)	Rank Minimum Distance	Is it included in 3-Nearest neighbors?
7	7	$(7-3)^2 + (7-7)^2 = 16$	3	Yes
7	4	$(7-3)^2 + (7-4)^2 = 25$	4	No
3	4	$(3-3)^2 + (7-4)^2 = 9$	1	Yes
1	4	$(3-1)^2 + (7-4)^2 = 13$	2	Yes

# 最近鄰居法 k-Nearest Neighbors

X1(value)	X2(value)	Square Distance to query instance (3,7)	Rank Minimum Distance	Is it included in 3-Nearest Neighbors?	Y= Category of nearest Neighbors
7	7	$(7-3)^2 + (7-7)^2 = 16$	3	Yes	Bad
7	4	$(7-3)^2 + (7-4)^2 = 25$	4	No	-
3	4	$(3-3)^2 + (7-4)^2 = 9$	1	Yes	Good
1	4	$(3-1)^2 + (7-4)^2 = 13$	2	Yes	Good

經由上表，點X1=3，X2=7為 Good 屬性



有一天小明口渴想喝飲料，看 Goolge Map 發現學校附近有很多飲料店，但不知道要喝哪一間，後來小明就想起KNN的選擇方法，來幫助他選擇買哪間店的飲料當

**K = 1 時**  
小明會選擇哪間飲料店呢？

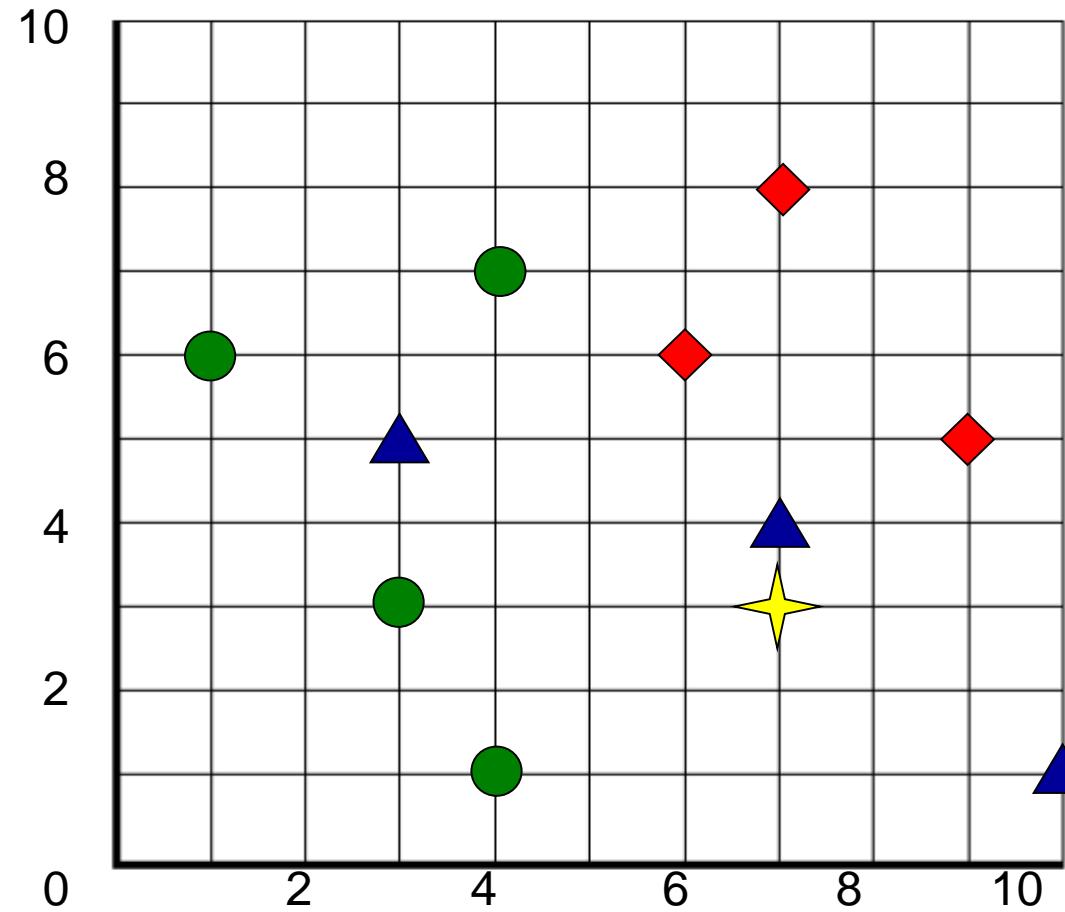
**K = 3 時**  
是哪間呢？

**K = 5 時**  
是哪間呢？

**K = 7 時**  
是哪間呢？

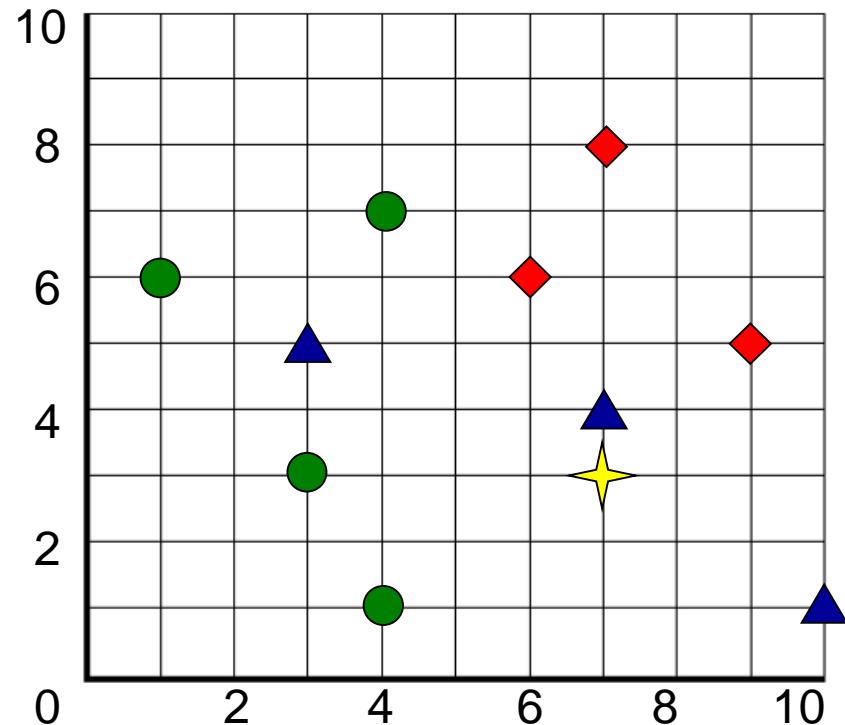
# 最近鄰居法-習題 1

圖中的 $\star$ 和哪個類別最接近呢？



# 最近鄰居法-習題 1

圖中的 $\star$ 和哪個類別最接近呢？



$\star (7, 3)$

順序	位置	距離	分類	
1	7,4	1	▲	1NN ▲
2	9,5	2.82	◆	
3	6,6	3.16	◆	3NN ◆
4	10,1	3.61	▲	
5	4,1	3.61	●	5NN ◆▲
6	3,3	4	●	
7	3,5	4.47	▲	7NN ▲
8	7,8	5	◆	
9	4,7	5.66	●	9NN ◆▲●
10	1,6	6.71	●	

投票機制



**E**

## Exercise 0: Environment Setting

# Exercise 0: Environment Setting

- Create a new project with “pure python” in PyCharm.
  - Using venv as the virtual environment.
- Install necessary packages
  - **pip install -U scikit-learn**
  - pip install pandas
  - pip install seaborn



## Exercise 1: Data exploratory and visualization

# Exercise 1: Data exploratory and visualization

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 from sklearn.model_selection import train_test_split
4 from sklearn.tree import DecisionTreeClassifier, plot_tree
5 from sklearn import metrics
6 from sklearn.naive_bayes import GaussianNB
7 from sklearn.neighbors import KNeighborsClassifier
8 from sklearn.svm import SVC
9
10 data = pd.read_csv('data.csv')
11 print(data.head(5))
12 print(data.describe())
13 print(data.groupby('species').size())
```

## Head(5)

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

## describe()

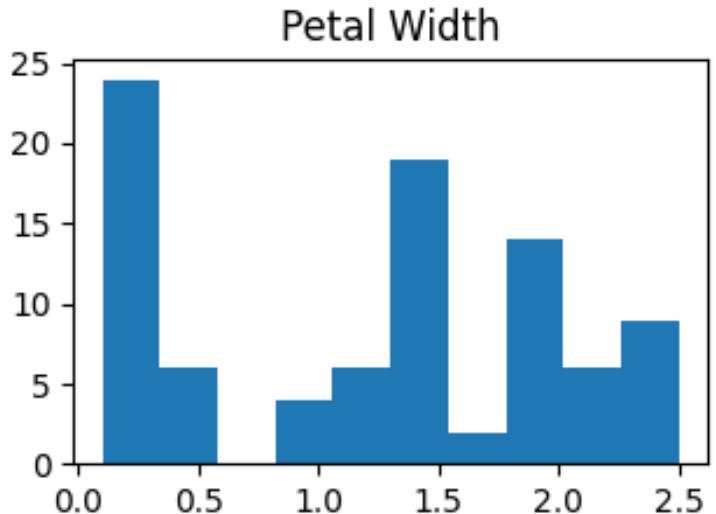
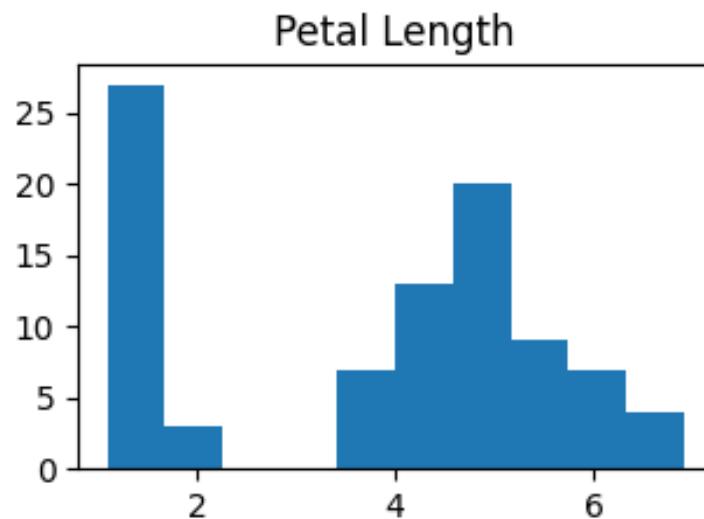
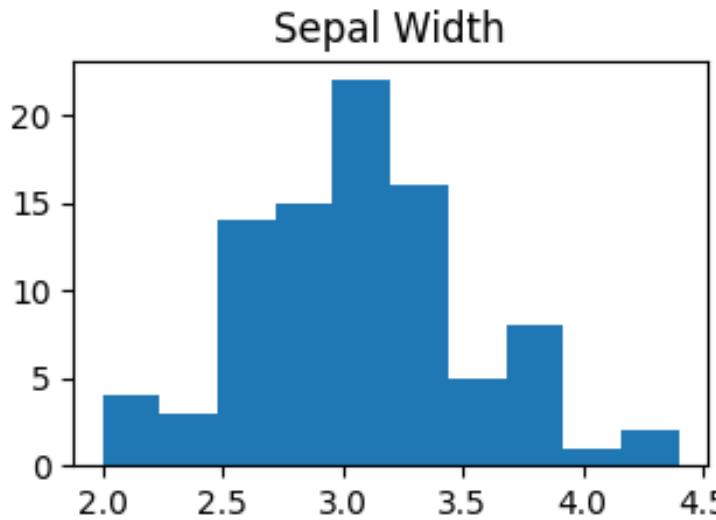
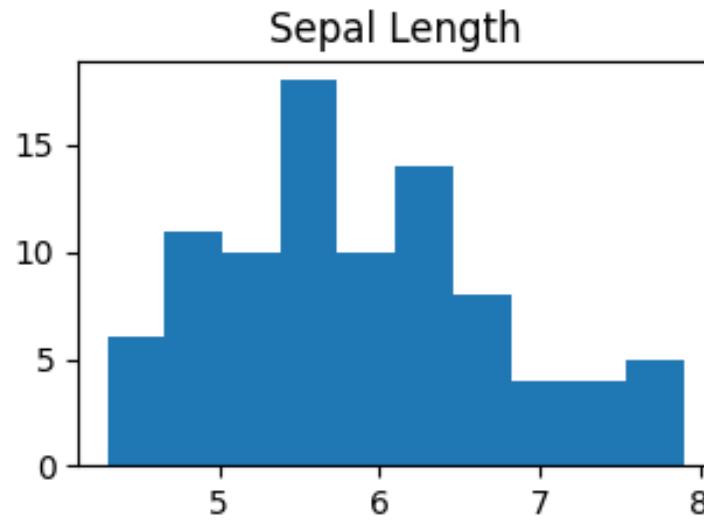
	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

## groupby()

```
species  
setosa      50  
versicolor  50  
virginica   50  
dtype: int64
```

```
L5      # Holdout split  
L6      train, test = train_test_split(data, test_size=0.4, stratify=data['species'], random_state=42)  
L7  
L8      n_bins = 10  
L9      fig, axs = plt.subplots(2, 2)  
L10     axs[0, 0].hist(train['sepal_length'], bins=n_bins);  
L11     axs[0, 0].set_title('Sepal Length');  
L12     axs[0, 1].hist(train['sepal_width'], bins=n_bins);  
L13     axs[0, 1].set_title('Sepal Width');  
L14     axs[1, 0].hist(train['petal_length'], bins=n_bins);  
L15     axs[1, 0].set_title('Petal Length');  
L16     axs[1, 1].hist(train['petal_width'], bins=n_bins);  
L17     axs[1, 1].set_title('Petal Width');  
L18     # add some spacing between subplots  
L19     fig.tight_layout(pad=1.0);  
L20     fig.show()
```

`fig.show()`



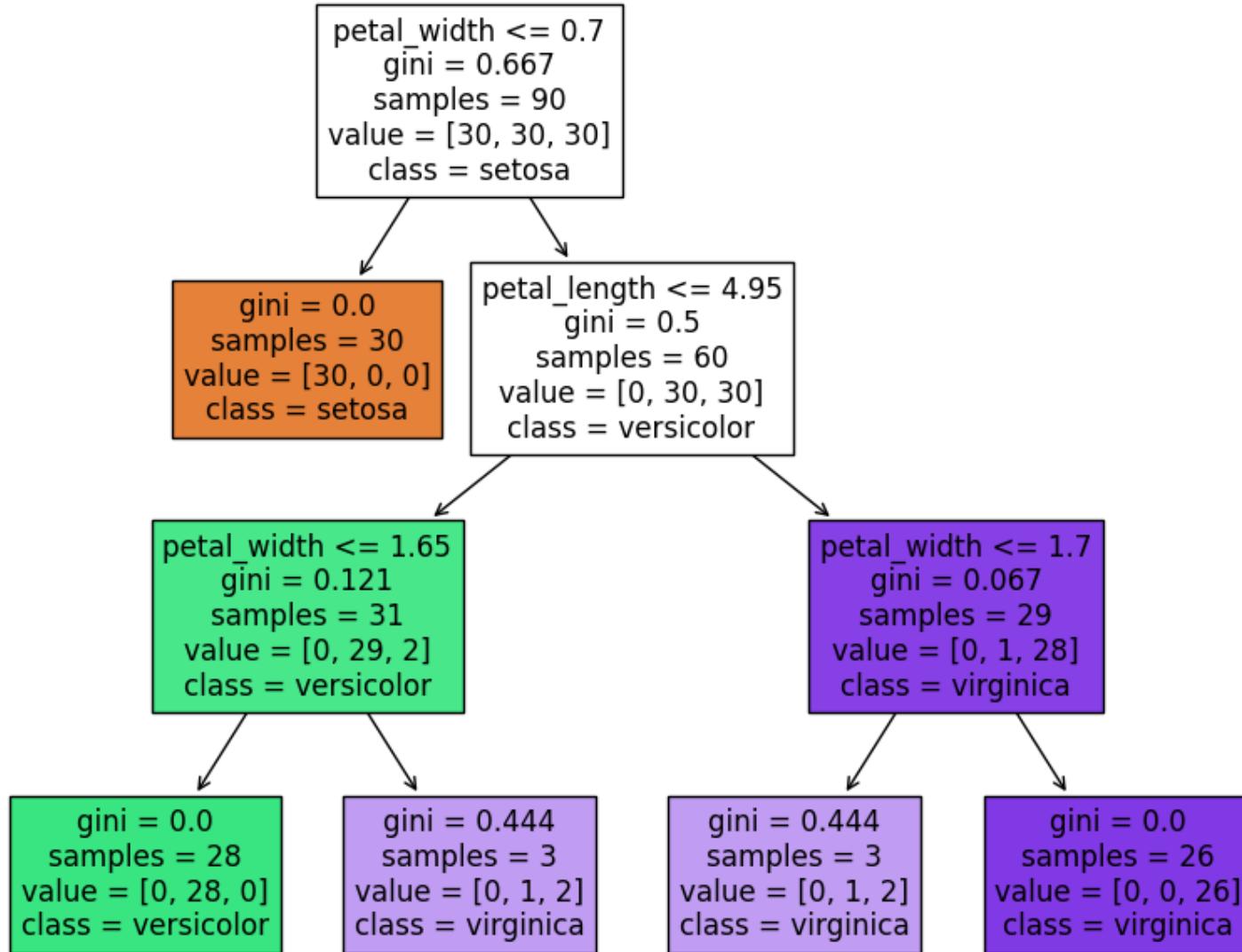


**E**

## Exercise 2: Decision Tree

# Decision Tree

```
33 ## split the X, Y
34 X_train = train[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']]
35 y_train = train.species
36 X_test = test[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']]
37 y_test = test.species
38
39 ## Classification Tree
40 mod_dt = DecisionTreeClassifier(max_depth=3, random_state=1)
41 mod_dt.fit(X_train, y_train)
42 prediction = mod_dt.predict(X_test)
43 print('The accuracy of the Decision Tree is', "{:.3f}".format(metrics.accuracy_score(prediction, y_test)))
44
45 fn = ["sepal_length", "sepal_width", "petal_length", "petal_width"]
46 cn = ['setosa', 'versicolor', 'virginica']
47
48 plt.figure(figsize=(10, 8))
49 plot_tree(mod_dt, feature_names=fn, class_names=cn, filled=True)
50 plt.show()
```





**E**

## Exercise 3: KNN

# KNN

```
52 ## KNN
53 mod_dt = KNeighborsClassifier()
54 mod_dt.fit(X_train, y_train)
55 prediction = mod_dt.predict(X_test)
56 print('The accuracy of the KNN is', "{:.3f}".format(metrics.accuracy_score(prediction, y_test)))
```



**E**

## Exercise 4: SVC

# Decision Tree

```
58     ## SVC  
59     mod_dt = SVC()  
60     mod_dt.fit(X_train, y_train)  
61     prediction = mod_dt.predict(X_test)  
62     print('The accuracy of the SVC is', "{:.3f}".format(metrics.accuracy_score(prediction, y_test)))
```



**E**

## Exercise 4: GaussianNB

# GaussianNB

```
64 ## GaussianNB  
65 mod_dt = GaussianNB()  
66 mod_dt.fit(X_train, y_train)  
67 prediction = mod_dt.predict(X_test)  
68 print('The accuracy of the GaussianNB is', "{:.3f}".format(metrics.accuracy_score(prediction, y_test)))
```

The accuracy of the Decision Tree is 0.983  
The accuracy of the KNN is 0.933  
The accuracy of the SVC is 0.967  
The accuracy of the GaussianNB is 0.933