機器學習工作坊

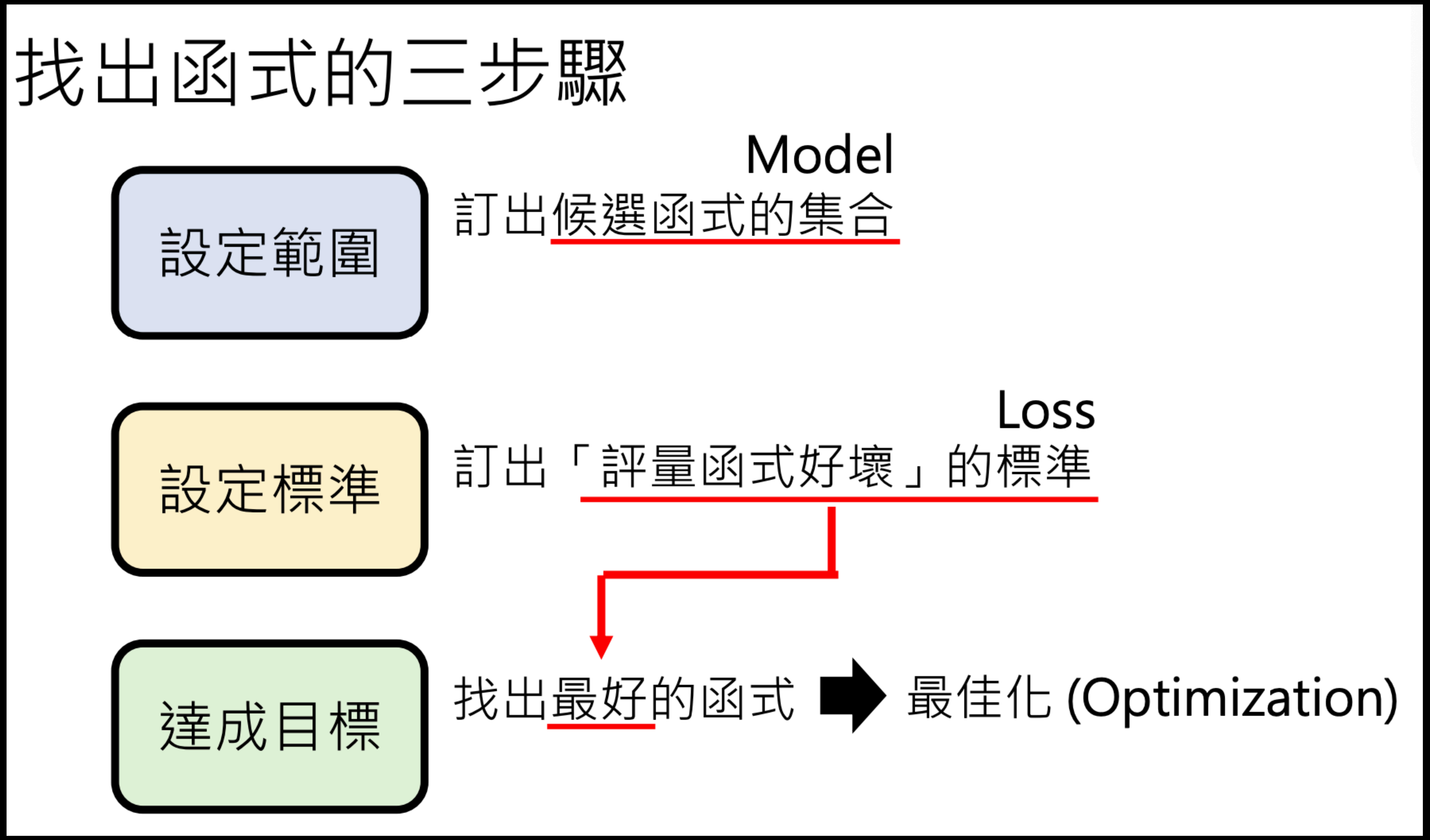
Part1: 監督學習 Supervised Machine Learning

Central Maintenance, IT Department (Internal use)

簡介

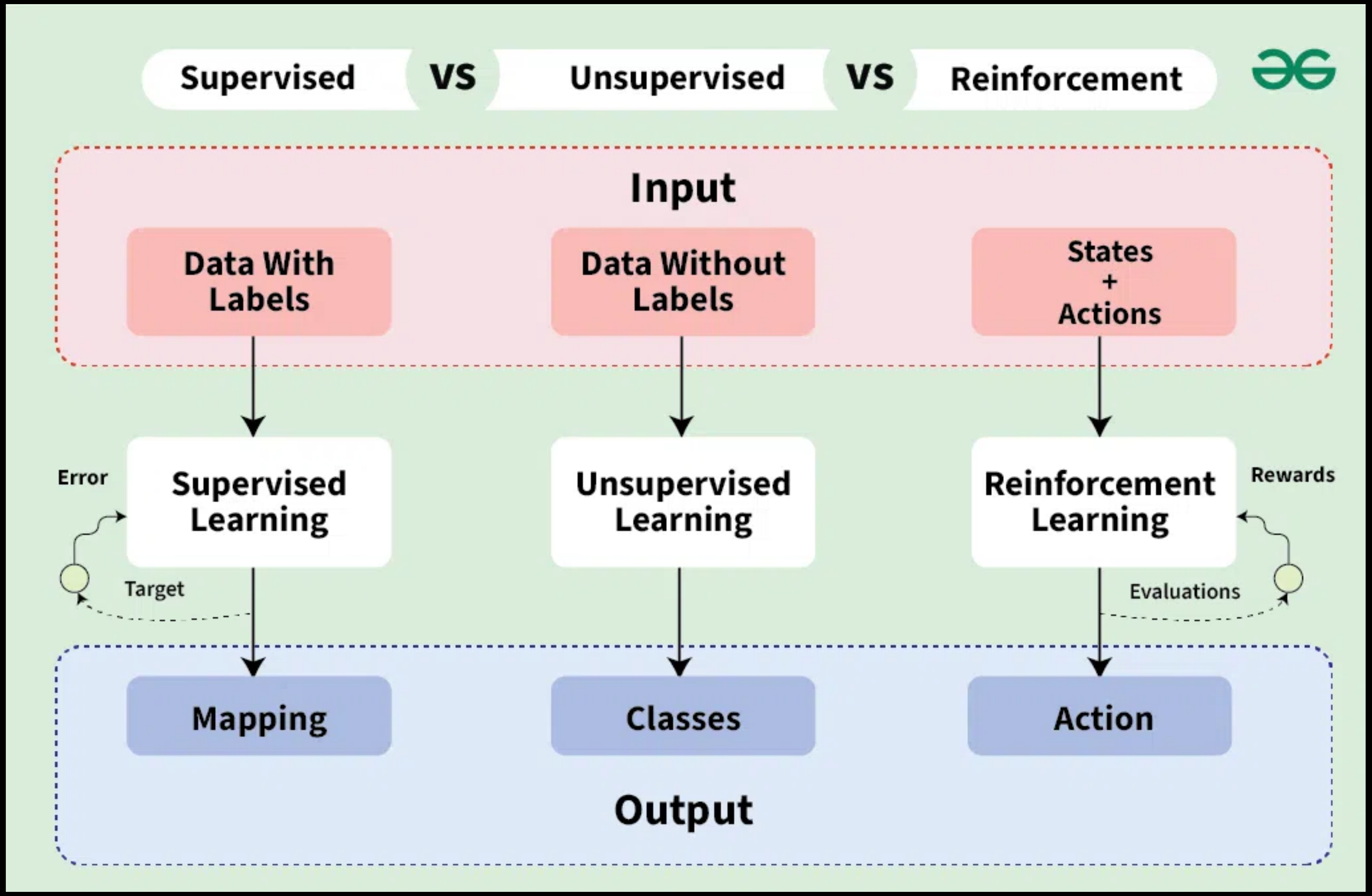
定義和類型

- 什麼是機器學習？
  - 機器自動從數據中學習“規律”(函式) - $y = f(x) + e$
    - 為什麼要找f()? : Predict or Inference



Step 1: 設定範圍：訂出候選函式的集合

機器學習的三大類型

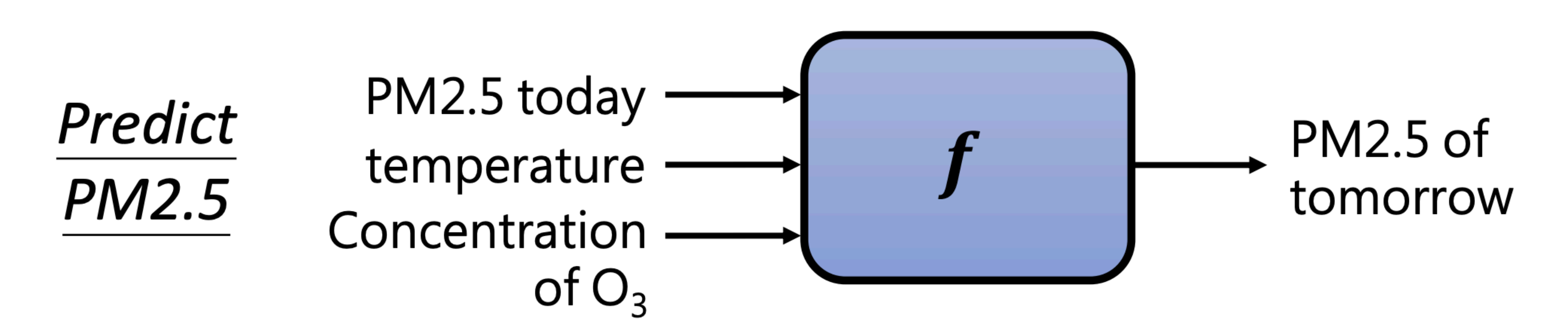


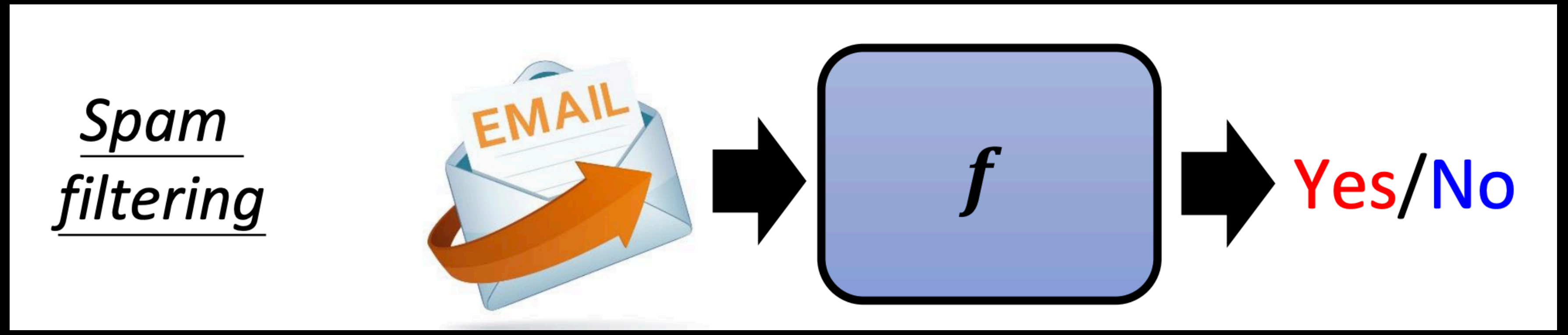
- 機器學習的類型
  - 監督學習（Supervised ML)
    - 有 Target (y)
  - 非監督學習（Unsupervised ML)
    - 沒有 Target (y)
  - 強化學習（Reinforced ML)
    - 通過回饋（Rewards) 學習

監督學習（Supervised ML)

基本概念

- 定義
  - 訓練數據中需要有目標標籤Target（Y）：模型學習如何從輸入特徵（Features)預測目標輸出（Target)
  - 目標：最小化預測值與真實值之間的誤差
- 方法
  - 回歸（Regression): 函數的輸出是一個數值
  - 分類 （Classification): 函數的輸出是一個類別





監督學習常用算法

回歸算法（用於預測連續值）

1. Ordinary Least Squares (OLS) Regression
   普通最小二乘回歸
2. Logistic Regression
   邏輯迴歸

分類算法（用於預測離散類別）

1. Logistic Regression
   邏輯迴歸
2. Decision Tree
   決策樹
3. k-Nearest Neighbors (k-NN)
   k-最近鄰算法
4. Ensemble Methods
   集成方法（或集成學習）
5. Naive Bayes
   樸素貝葉斯
6. Support Vector Machine (SVM)
   支持向量機
7. Artificial Neural Networks (ANN)
   人工神經網絡

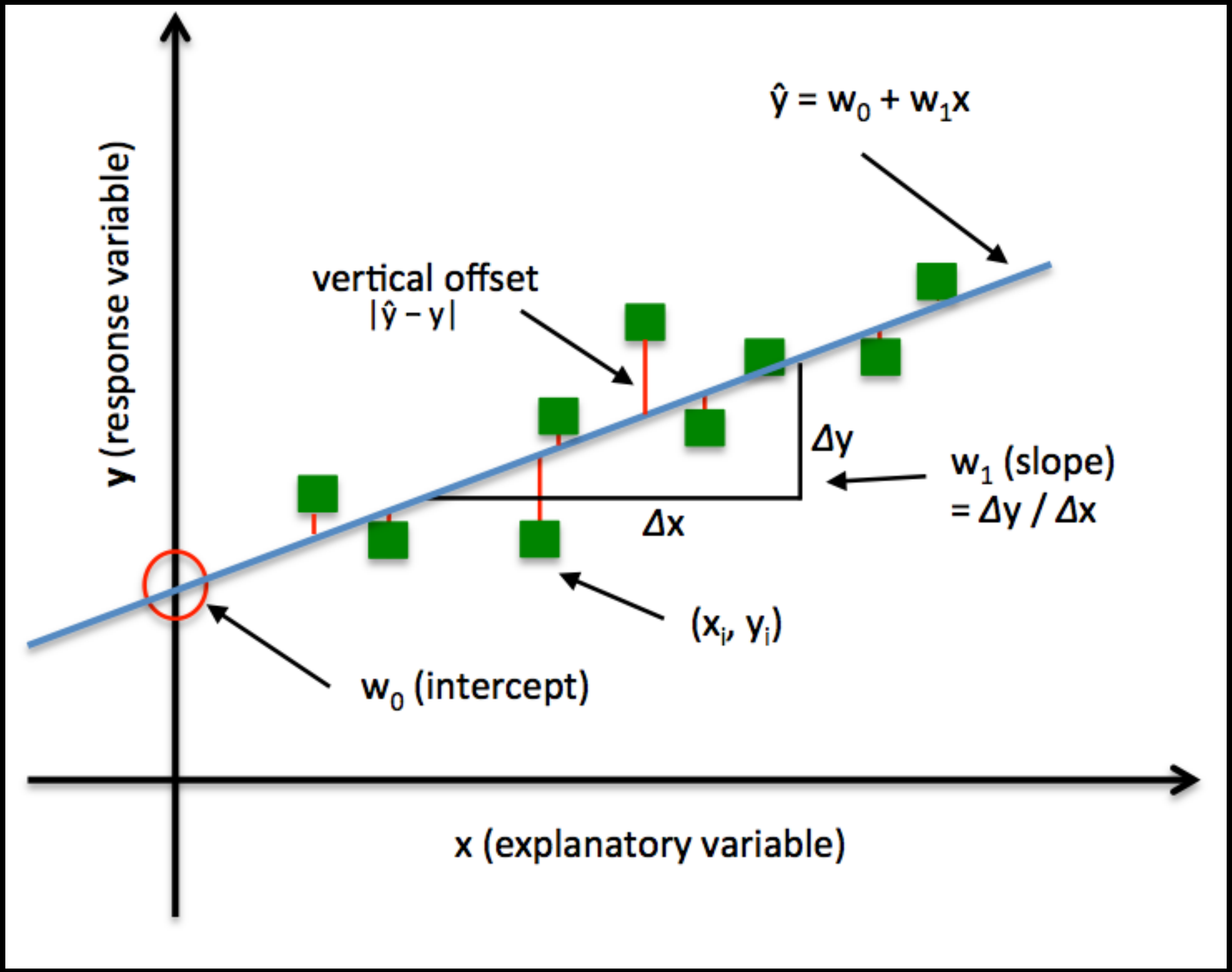
回歸算法 1. Ordinary Least Square (OLS) Regression
普通最小二乘回歸

$$Y \approx \beta_0 + \beta_1 X + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

$$RSS = SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Goal: Minimize the Residual Sum of Square (RSS)

- RSS: 評定函式好壞的一個基本標準
- 其他標準：MSE, RSE, R Square, Information Criteria, etc



Demo (in Python)

Linear regression

- Step 1: import packages: pandas, numpy, sklearn
- Step 2: import dataset
- Step 3: Data Cleaning
- Step 4: EDA
- Step 5: Split the dataset in to training and testing
- Step 6: fit the model and assess performance

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import cross_val_score, validation_curve, train_test_split
from sklearn.linear_model import LinearRegression, LogisticRegression, Ridge, Lasso
from sklearn.metrics import mean_squared_error, confusion_matrix
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.preprocessing import StandardScaler
```
Python

```python
# import dataset
college = pd.read_csv('../data/college.csv')
college.head(2)
```
Python

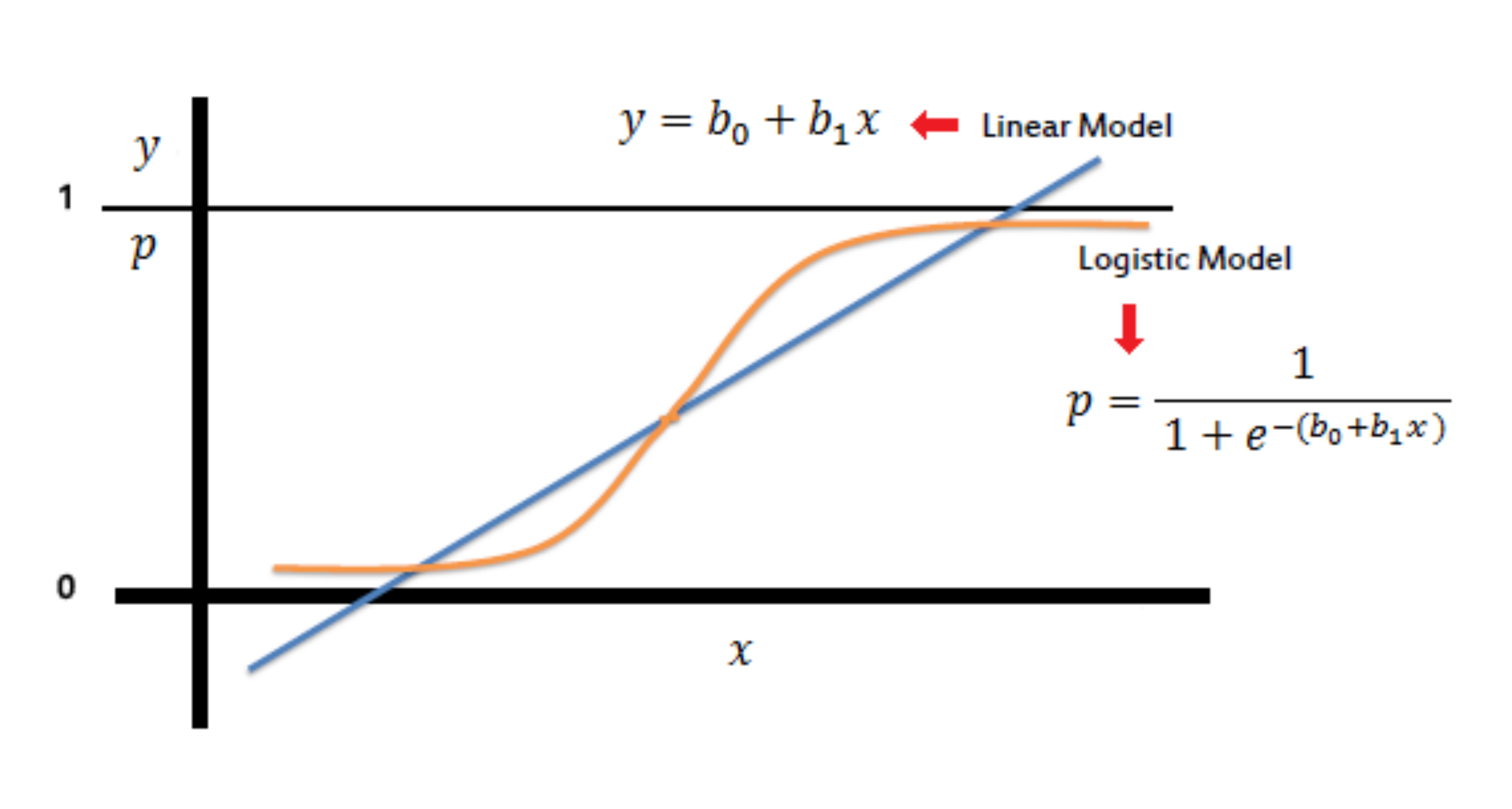
```python
# data cleaning: convert categorical features to numeric
college = pd.get_dummies(college, columns=['Private'], drop_first=True)
college.head(2)
```
Python

```python
# EDA
# check for missing values
# college.isnull().sum()
```
Python

```python
# split into training and test sets
y = college['Apps']
X = college.drop(['Name', 'Apps'], axis=1)
Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, test_size=0.3, random_state=10)
```
Python

```python
# Fit the model with training data and assess performance with test data
lm = LinearRegression()
lm.fit(Xtrain, ytrain)
lm.score(Xtest, ytest)
```
Python

分類算法 1. Logistic Regression
邏輯回歸*



$$log\left(\frac{Pr(Y = c|X)}{1 - Pr(Y = c|X)}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \epsilon$$

```python
# model fitting
logit = LogisticRegression()
logit.fit(Xtrain, ytrain)
```
Python

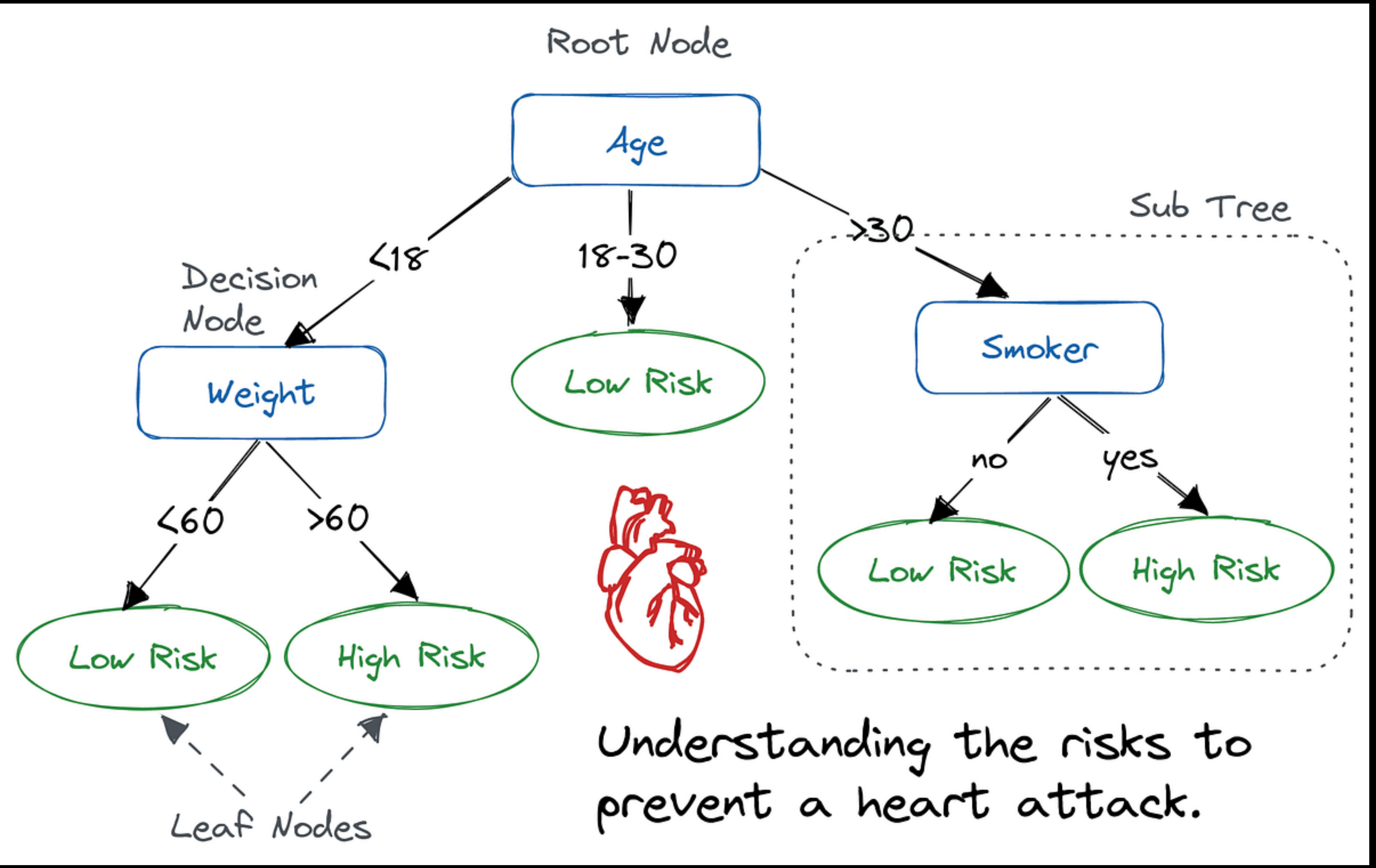
```python
# confusion matrix for the binary case

#              Predicted
#        ----------------------
# True       0           1
# ----------------------------
#   0    True Neg     False Pos
#   1    False Neg    True Pos
# ----------------------------
```
Python

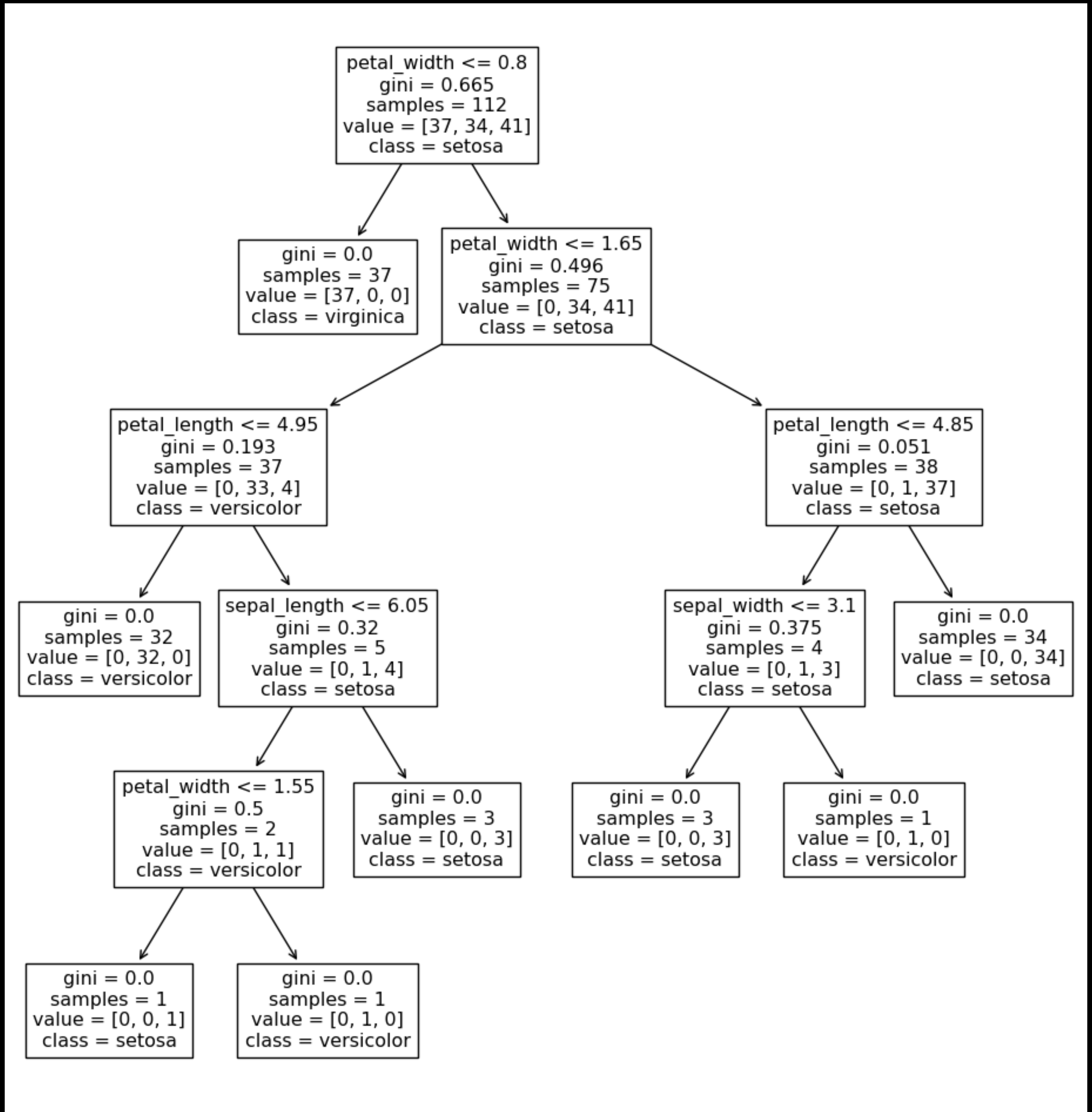
```python
# examine classification errors
print('Accuracy:', logit.score(Xtest, ytest))
confusion_matrix(ytest, logit.predict(Xtest))
```
Python

*： 也可以用於回歸，但多用於二元分類

分類算法 2. Decision Tree
決策樹



- 決策樹分類器——使用一組規則（if-then 條件語句）來創建同質化的群組



Step 2: 訂出標準：訂出評量函式好壞的標準

$$\text{accuracy} = \frac{\#\text{ of correct decisions}}{\#\text{ of decisions}}$$

$$\text{precision} = \frac{\#\text{ of cases where } \hat{y} = y}{\#\text{ of cases predicted as } \hat{y}}$$

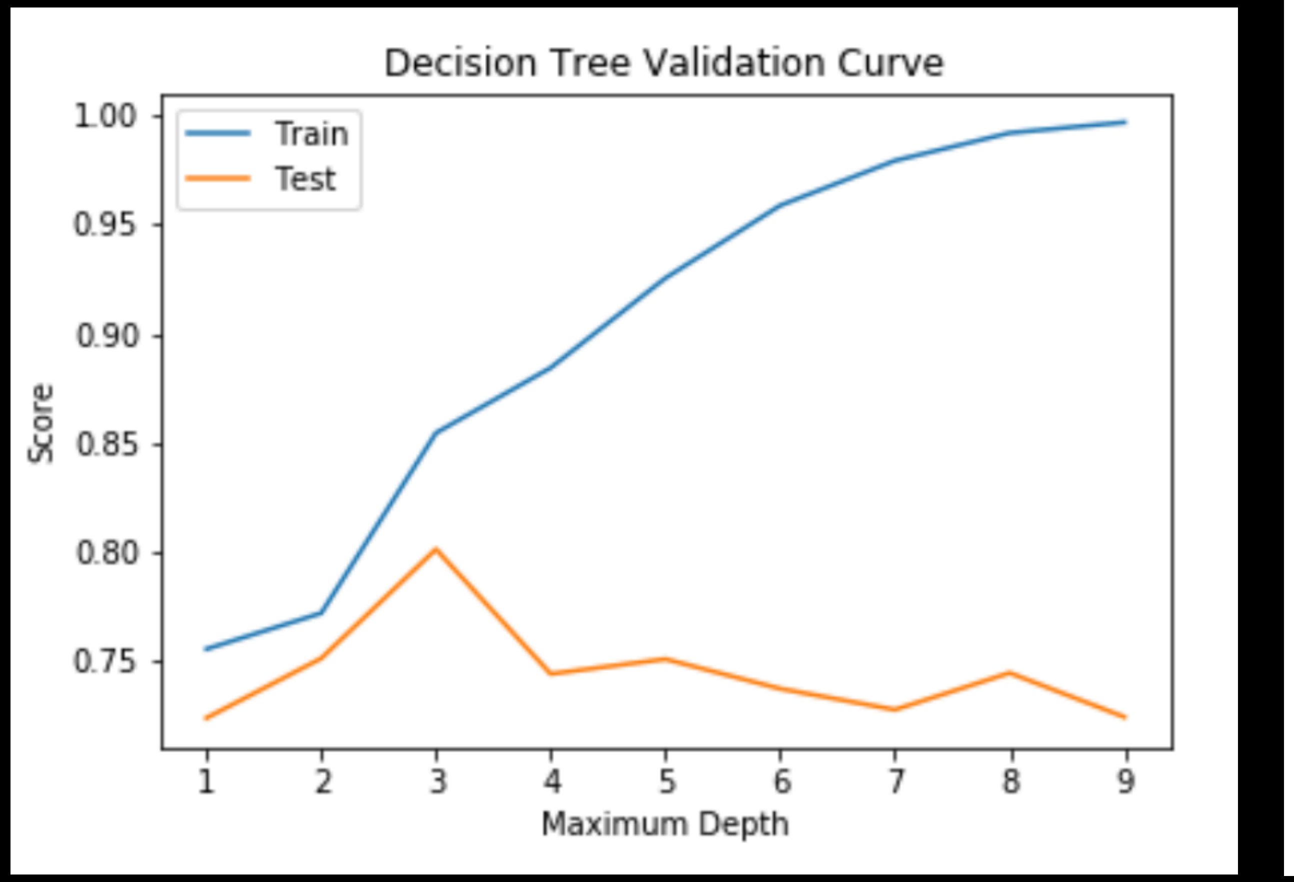
$$\text{recall} = \frac{\#\text{ of cases where } \hat{y} = y}{\#\text{ of cases of } y}$$

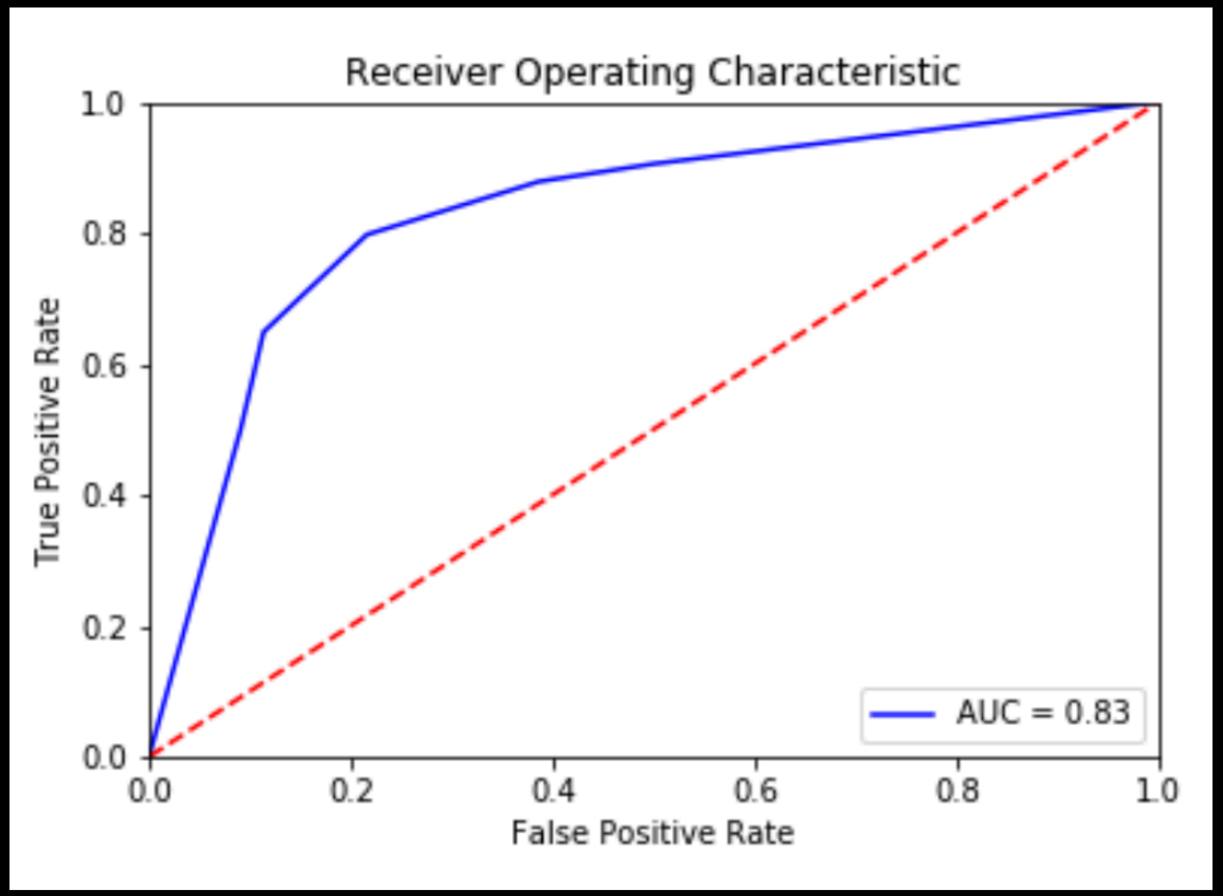
Diagnostics
- Scores: Accuracy, Precision, Recall
- Confusion matrix
- Validation (fitting) curve
- Learning curve
- ROC curve
- Area under the curve (AUC)

Cross-validation
- Generalizable performance measures
- Choosing a model family
- Choosing hyper-parameters

$$\text{F-measure} = \frac{2(\text{precision})(\text{recall})}{\text{precision} + \text{recall}}$$

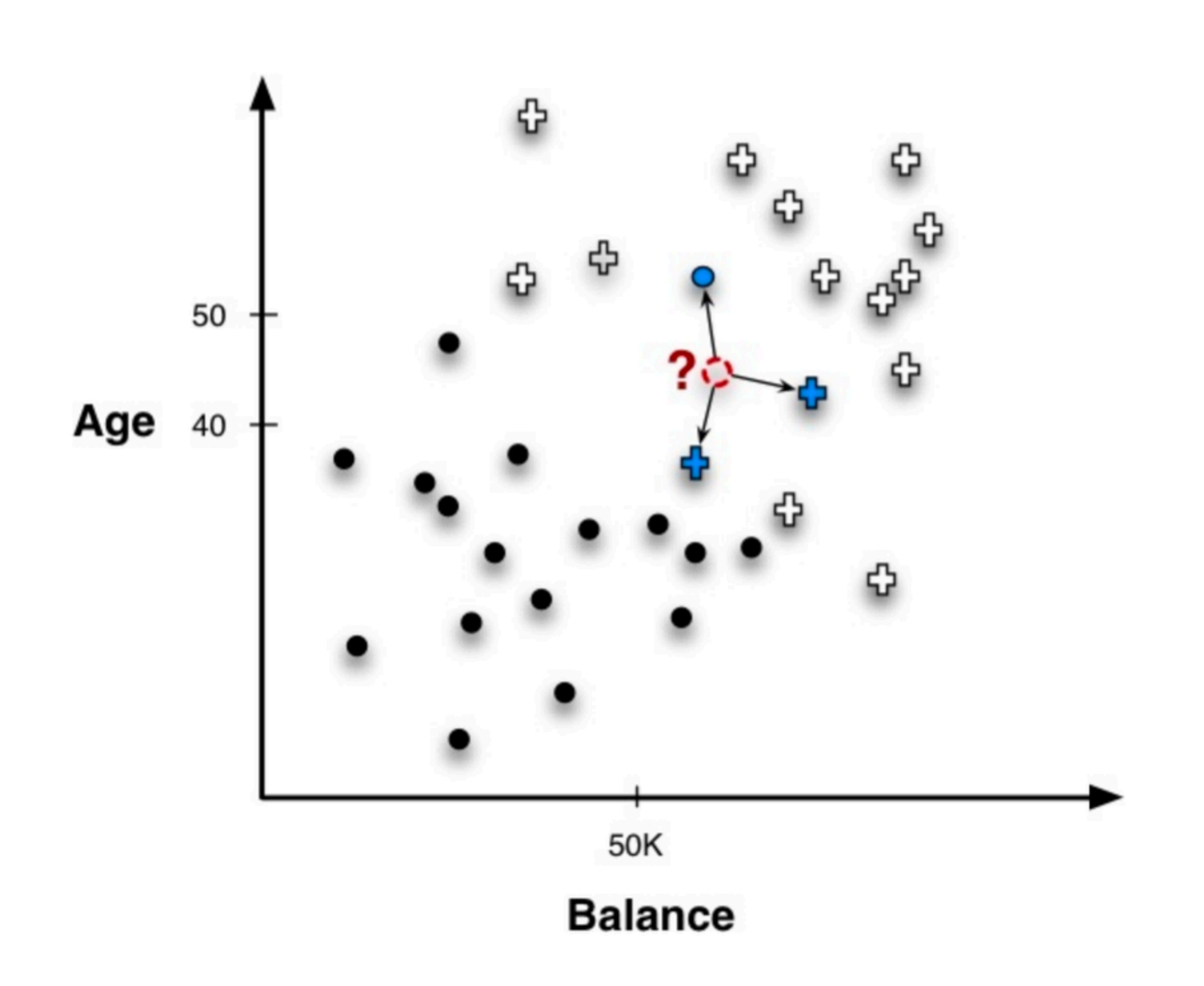
		Actual values 1	Actual values 0
Predicted values	1	TP	FP
	0	FN	TN



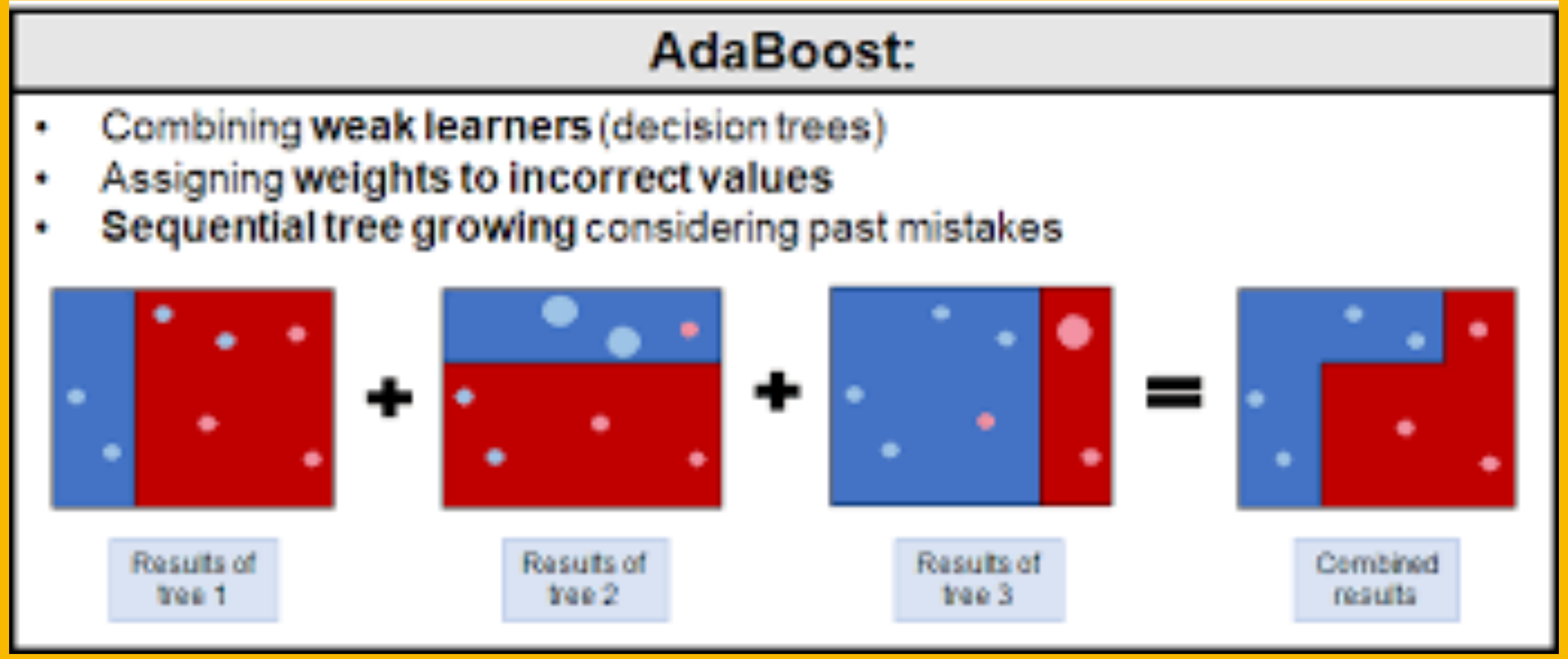


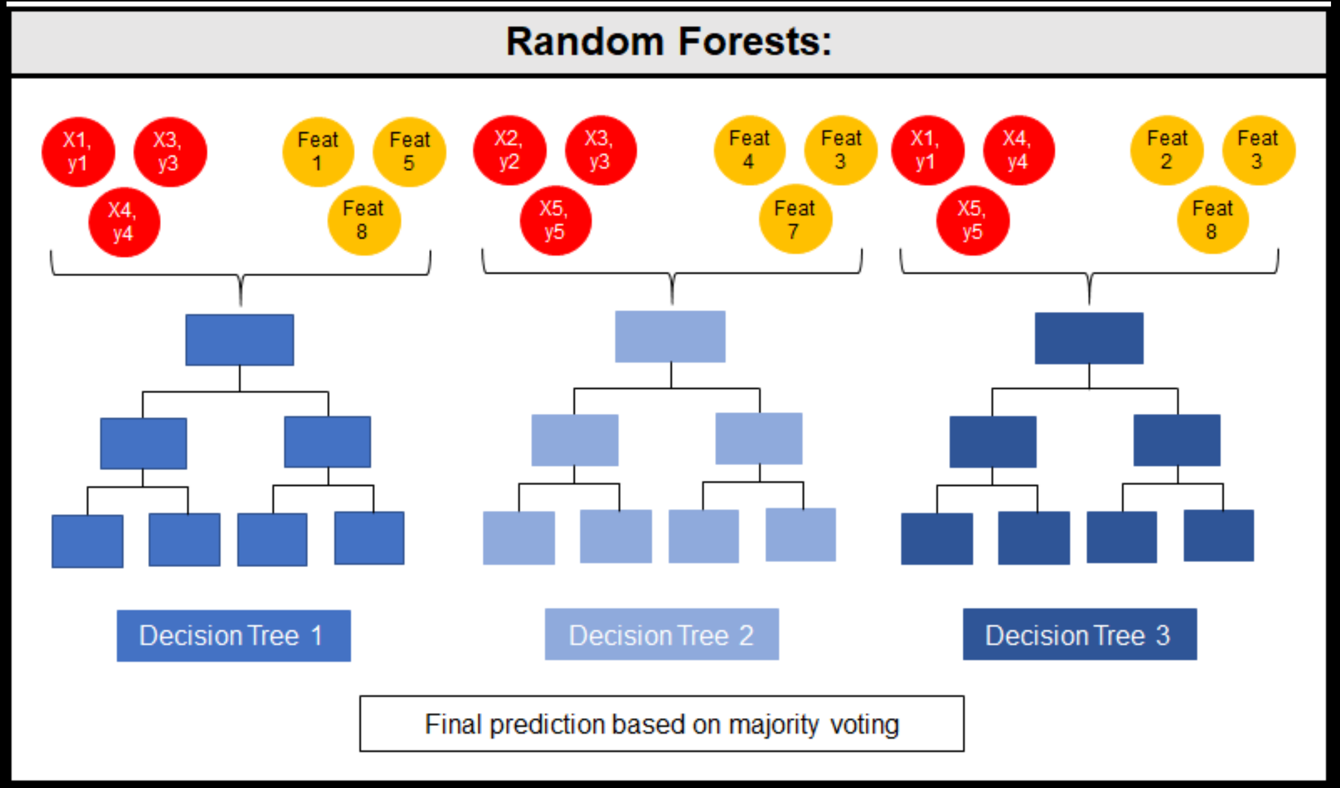
3. 分類算法 k-Nearest Neighbors (k-NN)
k-最近鄰算法

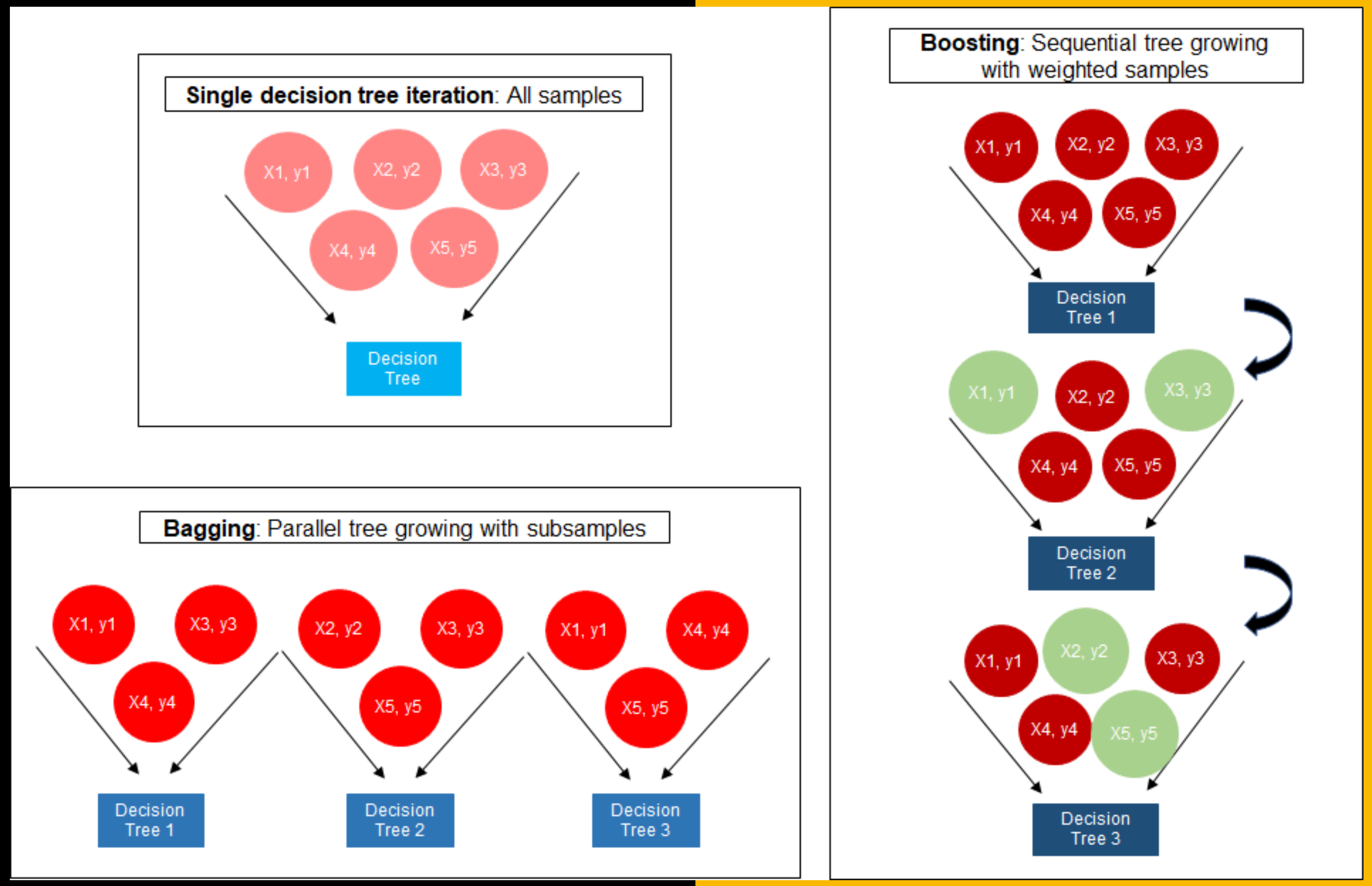
- KNN 算法
  1. 設定k 為最近鄰數量，並準備好訓練數據
  2. 對於每個新的數據點
     - 計算z與所有訓練樣本的距離
     - 選出離z最近的k個樣本
     - 根據這k個樣本的多數類別決定z的類別

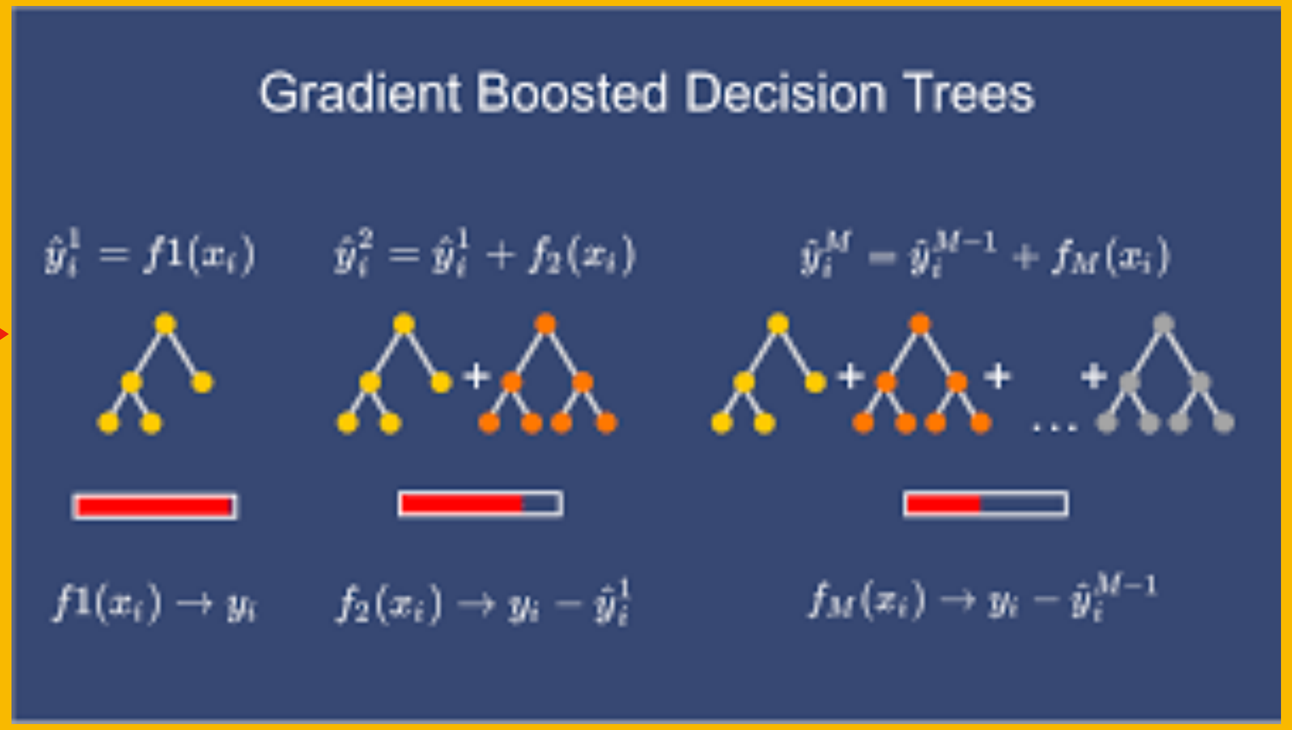


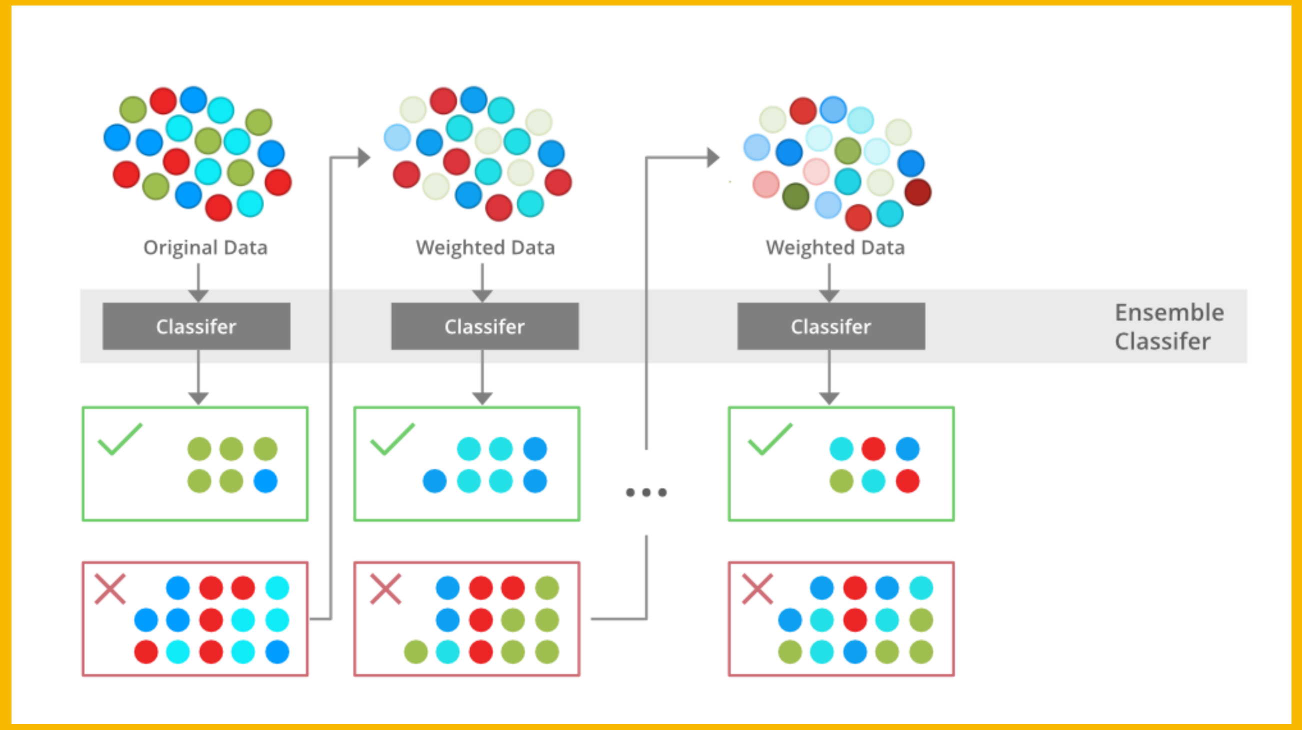
分類算法 4. Ensemble Method
集成算法

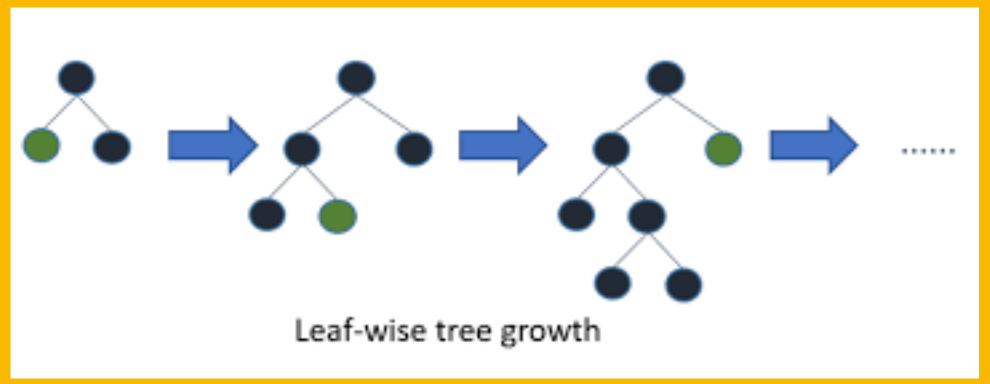












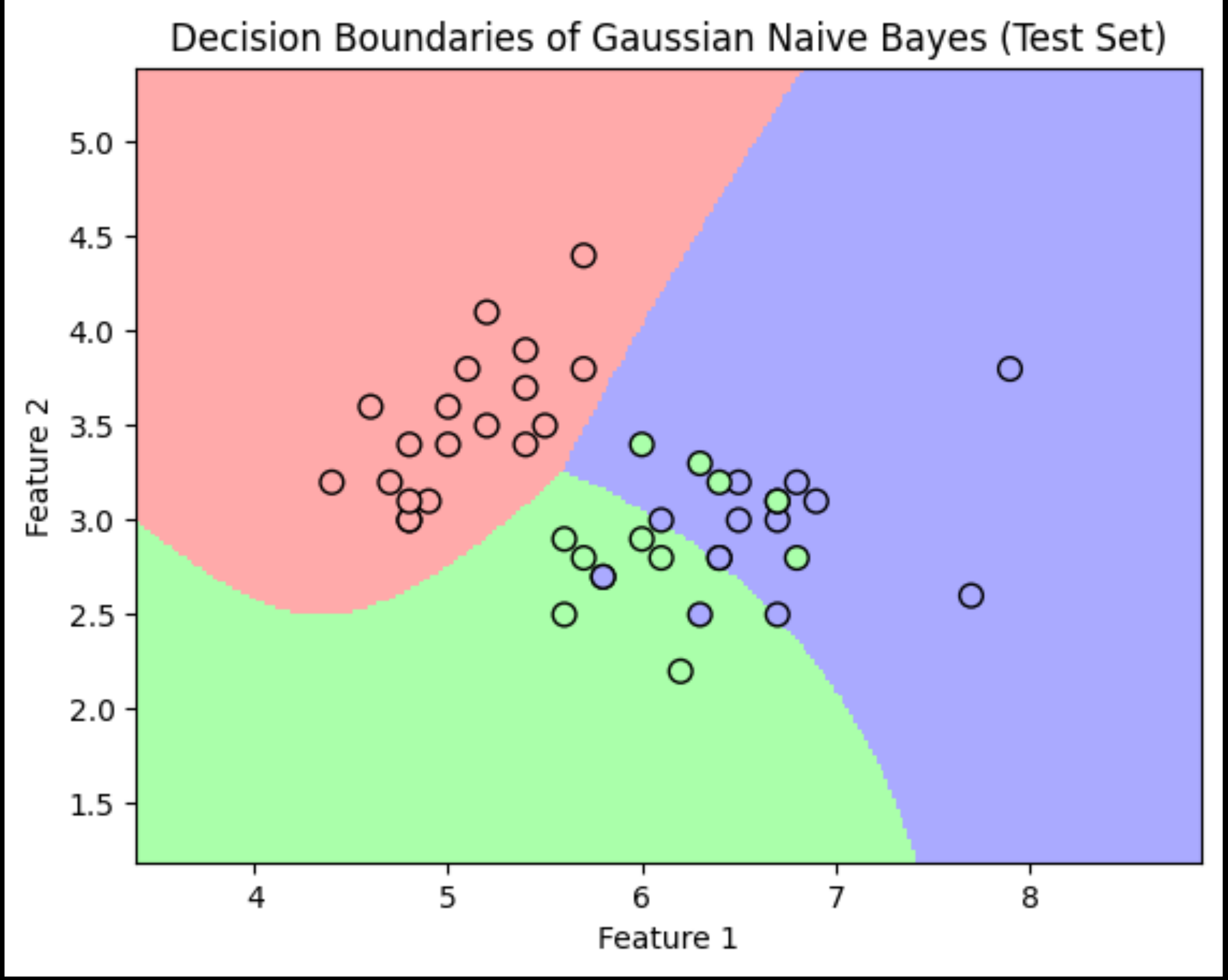
Bagging

Boosting

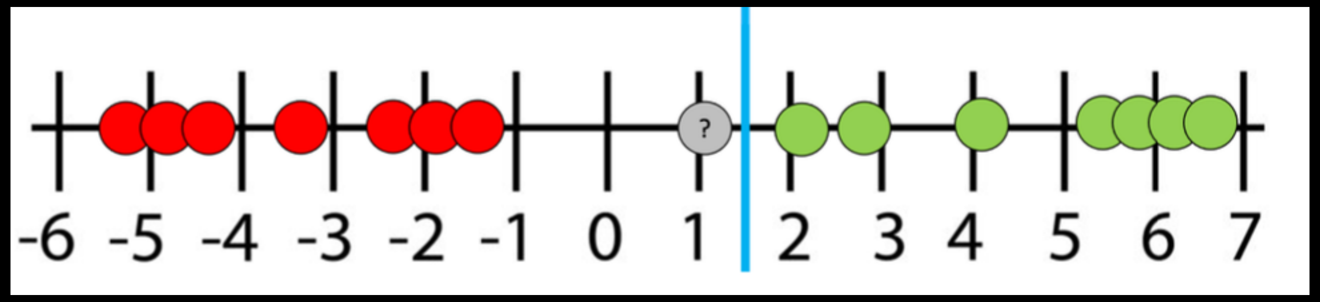
分類算法5：樸素貝葉斯
Naive Bayes

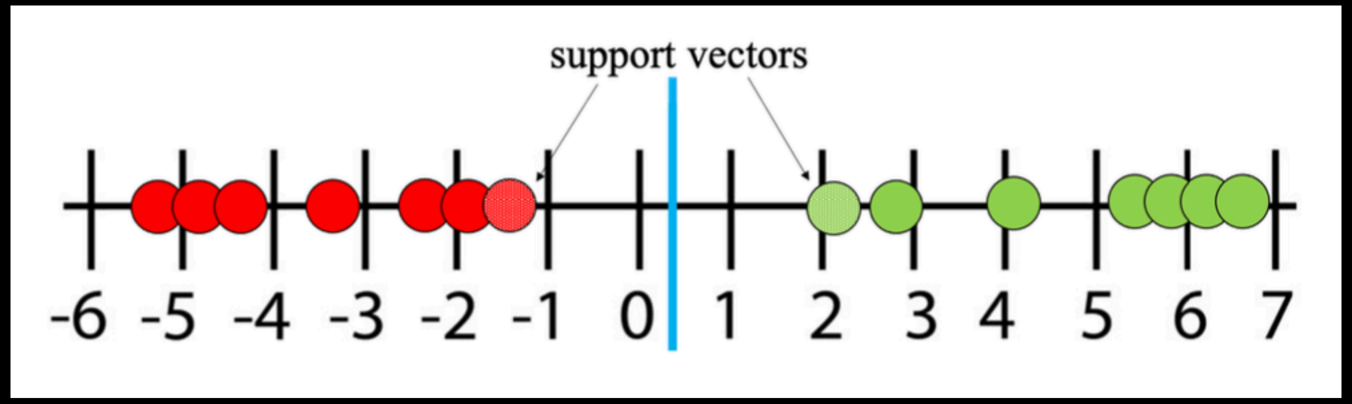
Naive Bayes 算法簡介：

- **基於概率的分類算法**：
  Naive Bayes 根據條件概率公式（Bayes 定理）計算每個類別的可能性，並選擇概率最高的類別。
- **假設特徵條件獨立**：
  「Naive（簡單）」的原因是它假設所有特徵是相互獨立的，這在實際情況下可能不成立，但效果仍然不錯。
- **計算公式**：$$Pr(Y = y|X) = \frac{Pr(Y = y)Pr(X|Y = y)}{Pr(X)}$$
- **適用場景**：
  適合用於文本分類（如垃圾郵件檢測）、情感分析和醫學診斷等問題。



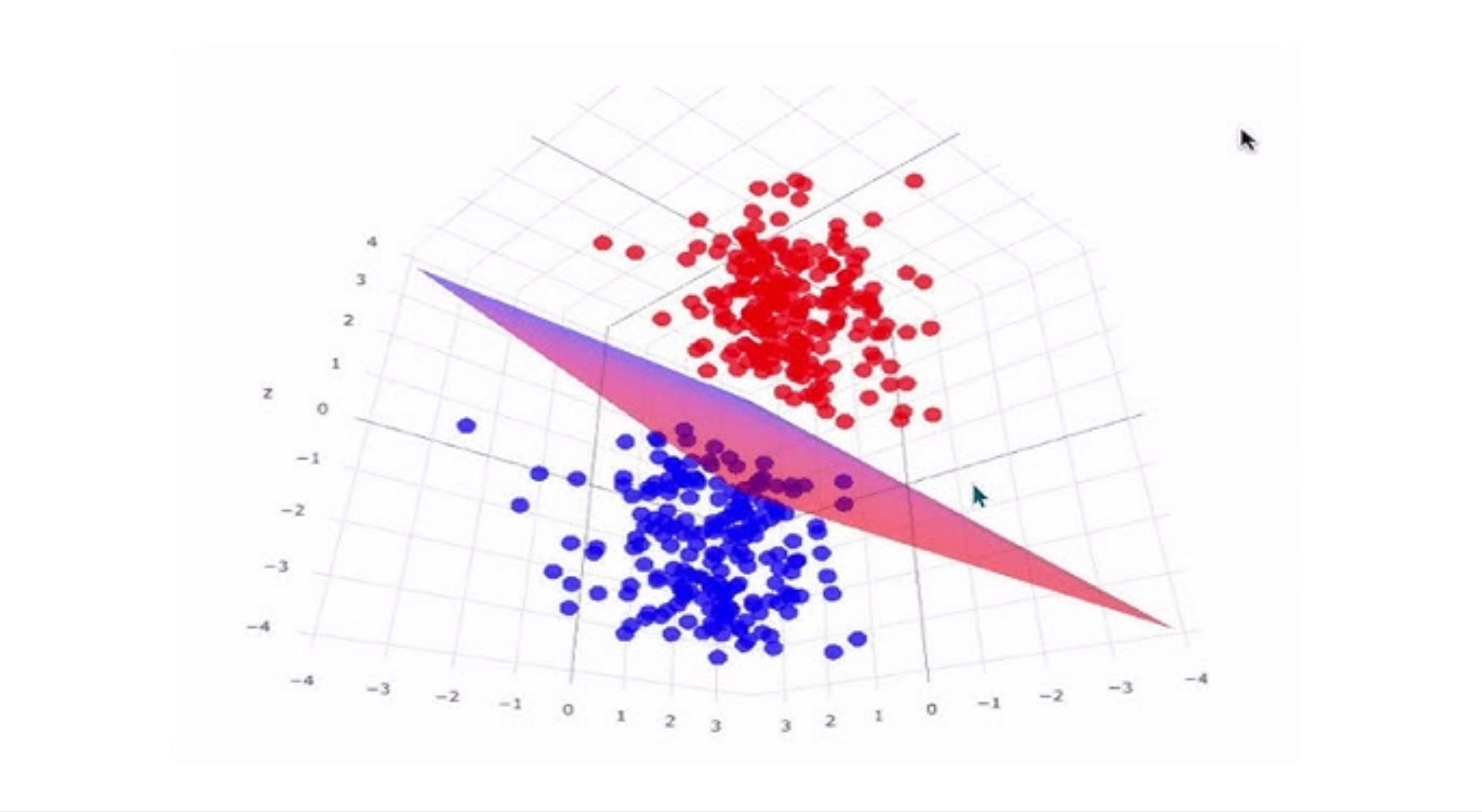
分類算法6: 支持向量機
Support Vector Machine (SVM)





SVC（支持向量分類器）算法

- **目標**：找到能最大化兩類數據間距的分類邊界。
- **允許誤差**：引入鬆弛變量，允許部分樣本落在錯誤的一側。
- **控制誤差**：通過超參數 C 限制誤差的數量，平衡模型的準確性與容錯能力。
- **調整模型**：使用交叉驗證選擇最合適的 C



樸素貝葉斯： 文本分析

垃圾郵件檢測（Spam Detection）

1.

"She ate an apple today"

$\downarrow$

[an, apple, ate, she, today]

$$d_1 = [w_1, w_3, w_4]$$
$$d_2 = [w_1, w_2, w_3]$$
$$d_3 = [w_3, w_6]$$

2.

	w_1	w_2	w_3	w_4	w_5	w_6
d_1	1	0	1	1	0	0
d_2	1	1	1	0	0	0
d_3	0	0	1	0	0	1

	dear
d_1	1
d_2	1
d_3	0
d_4	1

	label
d_1	spam
d_2	spam
d_3	not spam
d_4	?

3.

Step 1: Compute the posterior $Pr(spam|dear)$

$$Pr(spam|dear) = \frac{Pr(spam) * Pr(dear|spam)}{Pr(dear)}$$

Step 2: Compute the posterior $Pr(notspam|dear)$

$$Pr(notspam|dear) = \frac{Pr(notspam) * Pr(dear|notspam)}{Pr(dear)}$$

	dear	friend	...	money	send
d_1	1	0	...	1	0
d_2	1	1	...	1	1
d_3	0	0	...	0	0
d_4	1	1	...	1	0

	label
d_1	spam
d_2	spam
d_3	not spam
d_4	?

4.

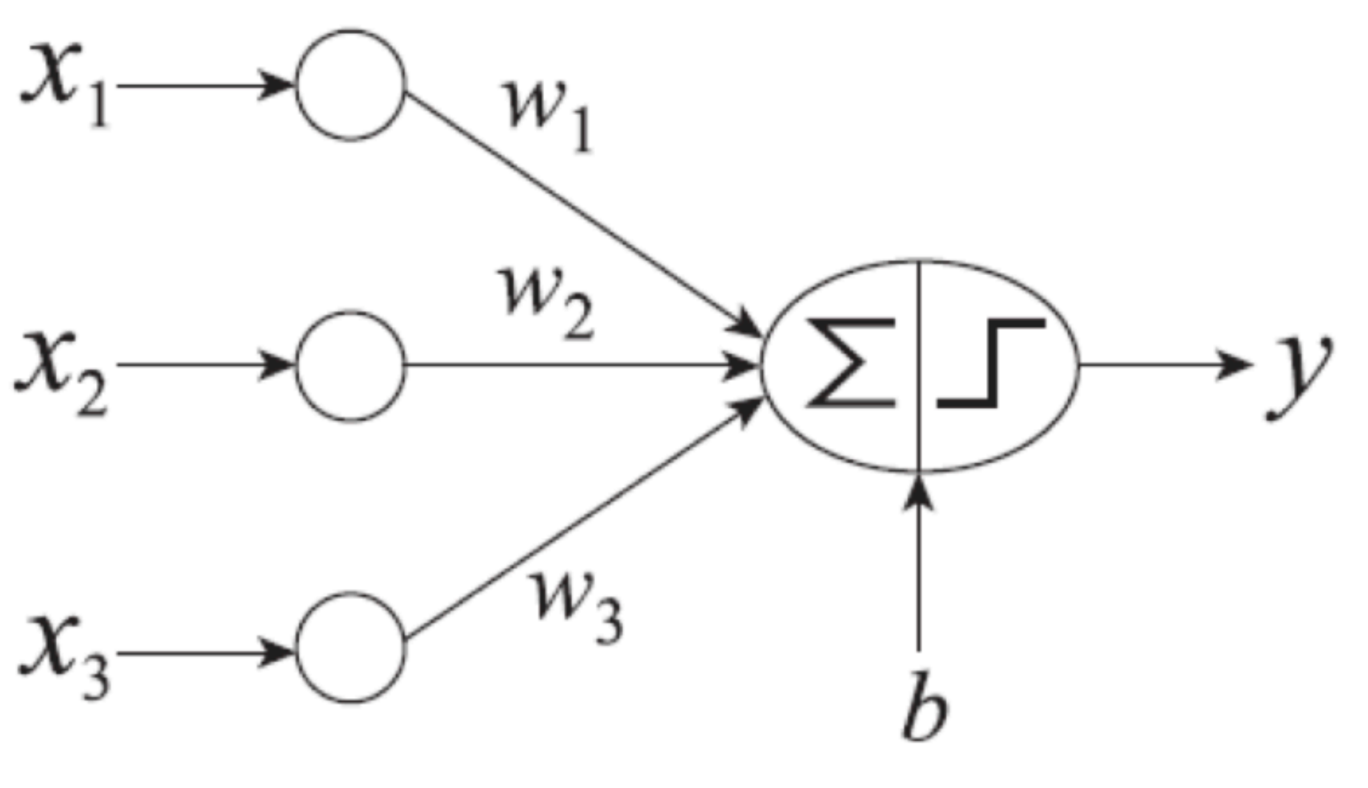
$$Pr(X_i = c|y) = \frac{n_c + 1}{n + v}$$

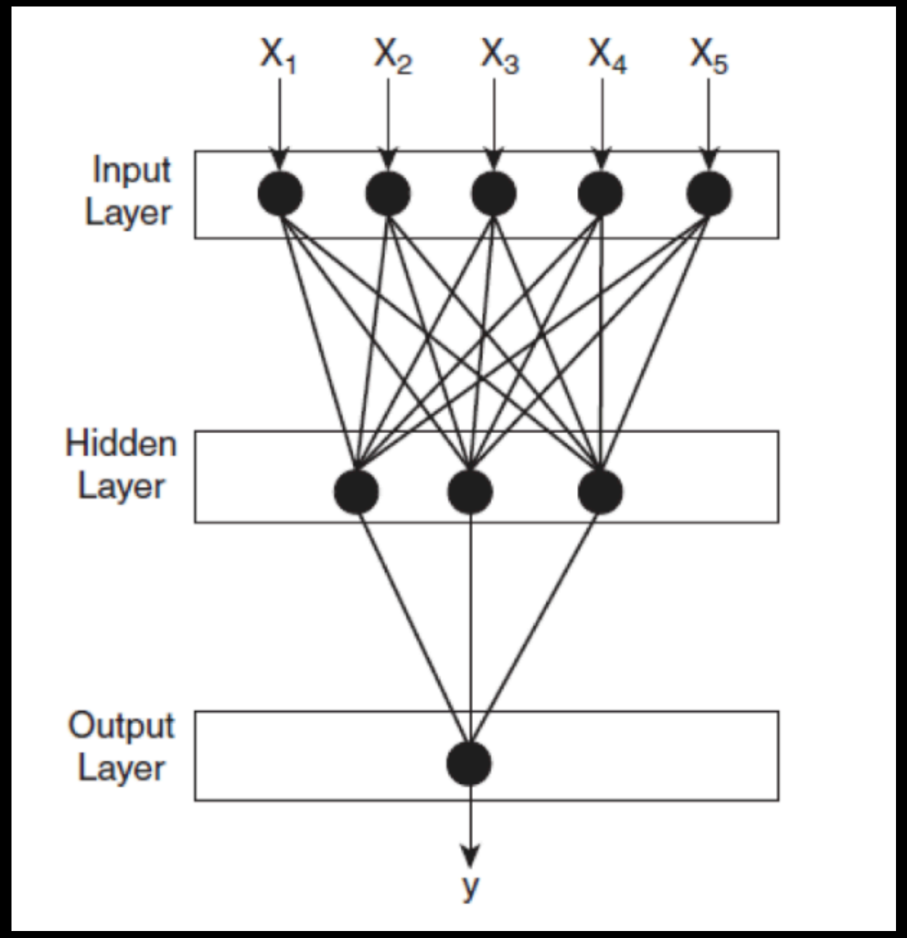
分類算法6: 人工神經網絡

Artificial Neural Networks (ANN)

$$y = \begin{cases} 0 \text{ if } x_1w_1 + x_2w_2 + x_3w_3 < 10 \\ 1 \text{ if } x_1w_1 + x_2w_2 + x_3w_3 \geq 10 \end{cases}$$

$$y = \begin{cases} 0 \text{ if } \mathbf{xw} + \mathbf{b} < 0 \\ 1 \text{ if } \mathbf{xw} + \mathbf{b} \geq 0 \end{cases}$$





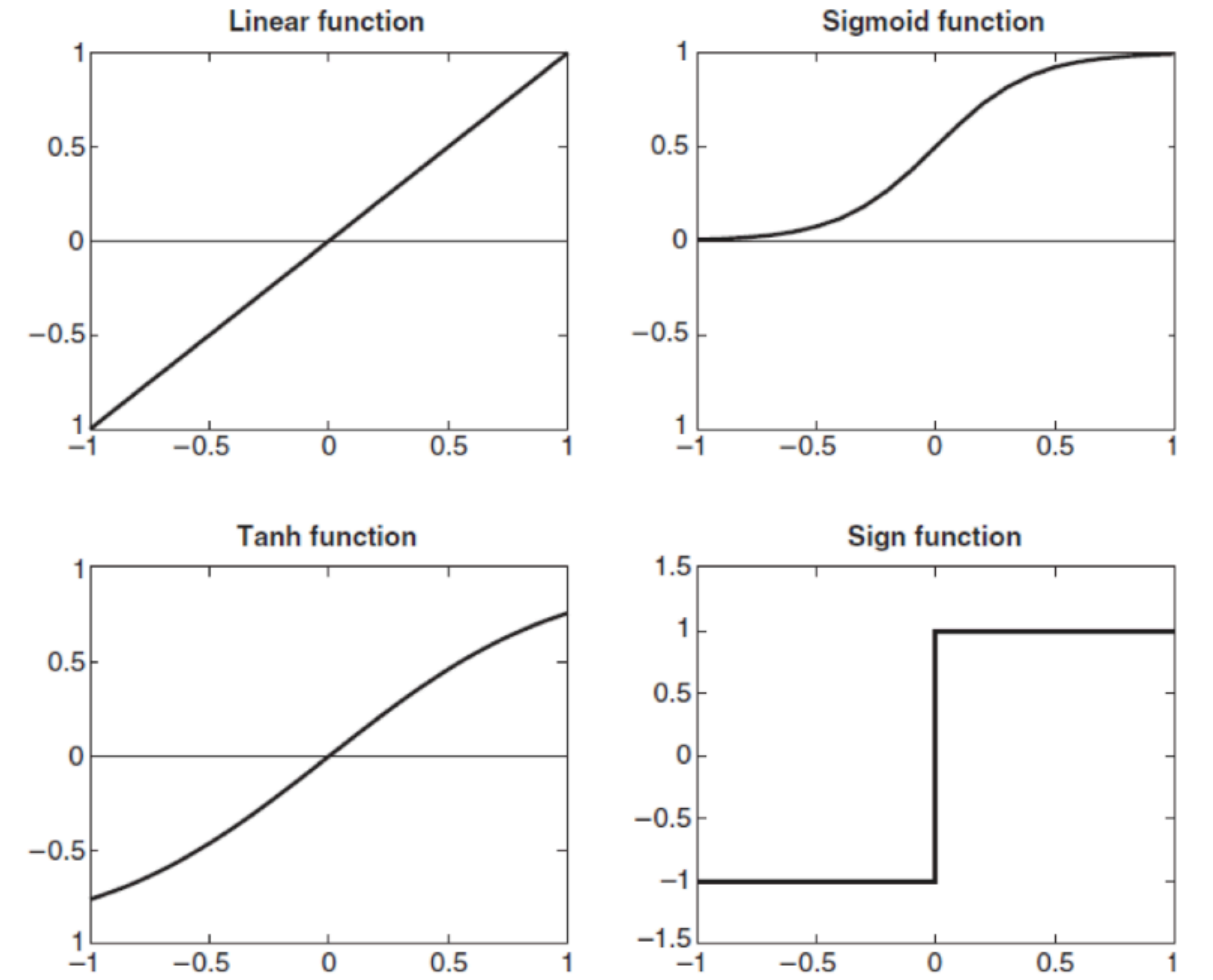
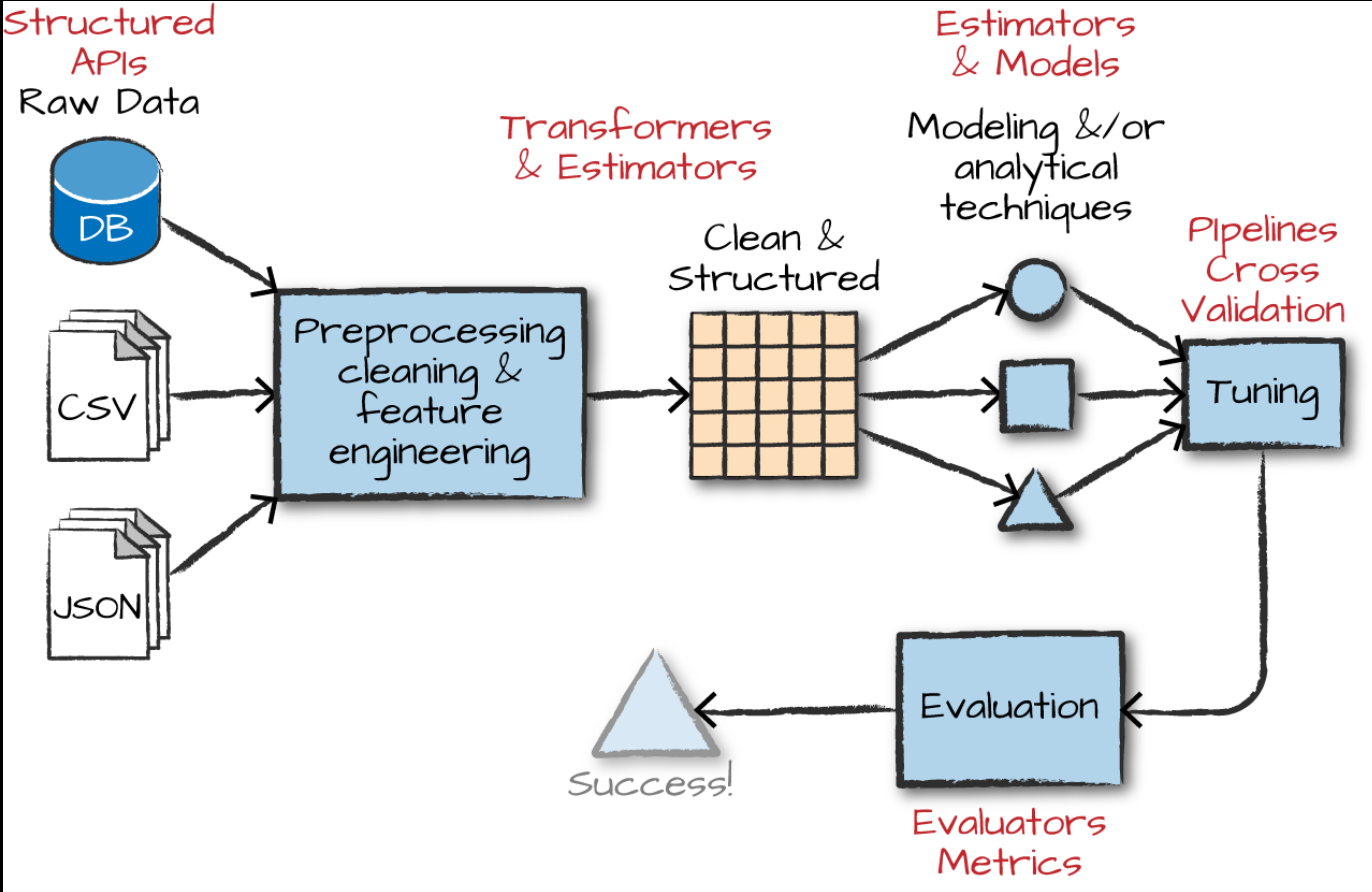
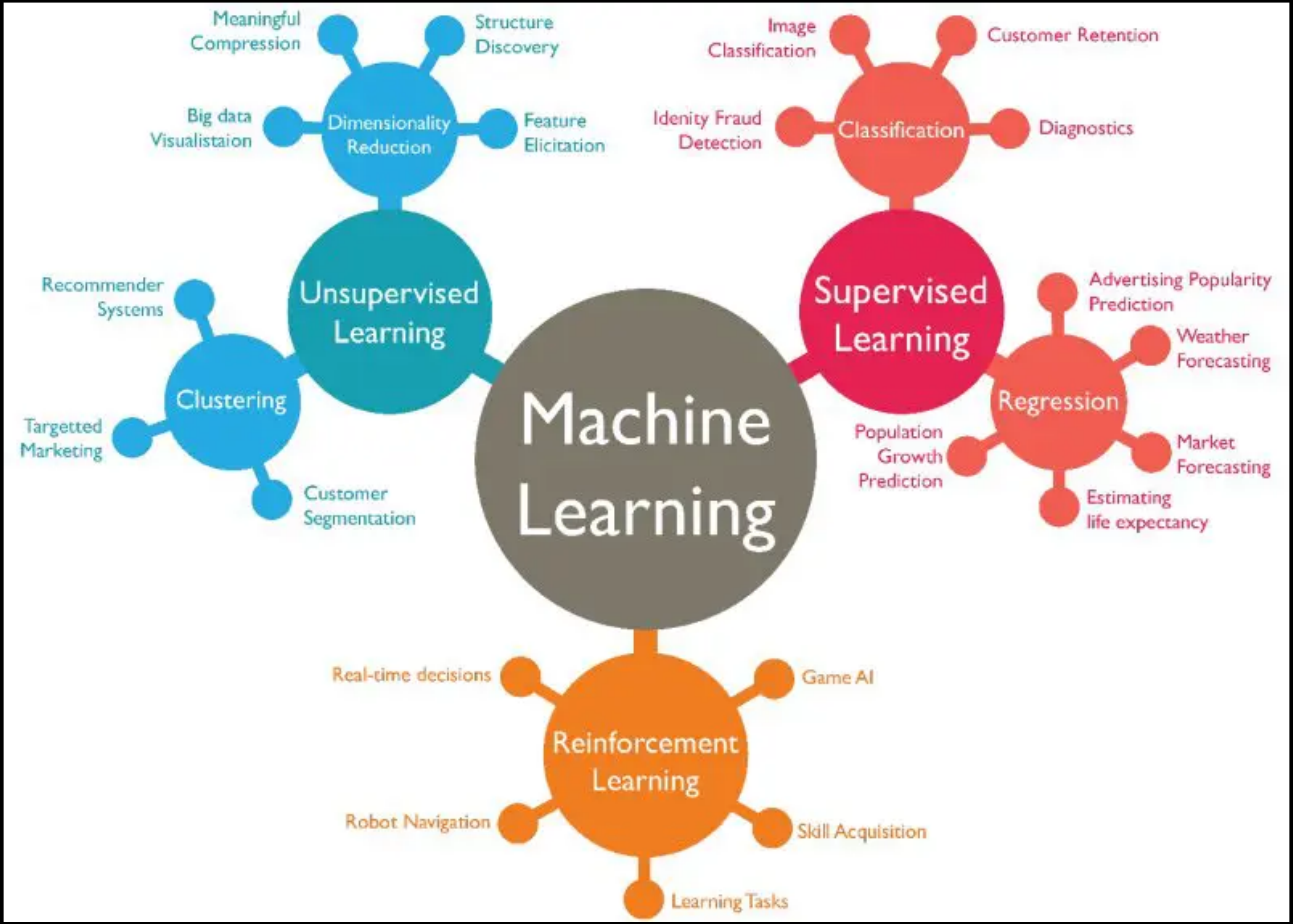


Figure 4.26. Types of activation functions used in multi-layer neural networks.


Structured
APIs
Raw Data
DB
CSV
JSON
Transformers
& Estimators
Preprocessing
cleaning &
feature
engineering
Clean &
Structured
Estimators
& Models
Modeling &/or
analytical
techniques
Pipelines
Cross
Validation
Tuning
Evaluation
Evaluators
Metrics
Success!


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks