大數據資料分析實作

評 估

博雅(科技)課程

模型評估

📌 當完成分類模型的建立後，**模型評估**是非常關鍵的步驟，可幫助我們了解模型的**預測效能**與**實際表現**。

✅ 不同類型任務的評估指標差異

類型	常用指標
分類問題	準確率、混淆矩陣、Precision、Recall、F1-score、ROC、PR
迴歸問題	MSE、RMSE、MAE、R^2（決定係數）

模型評估

✅ 分類問題 - 三個常用的模型評估指標：

- **準確率**（Accuracy）
- **混淆矩陣**（Confusion Matrix）
- 分類報告（Classification Report）
  - **Precision**（精確率）
  - **Recall**（召回率）
  - **F1-score**

準確率（Accuracy）

✅ 定義：**準確率**是最簡單也最直觀的評估指標，表示模型**預測正確的比例**。

📐 計算公式：$$\text{Accuracy} = \frac{\text{正確預測的樣本數}}{\text{總樣本數}} = \frac{TP + TN}{TP + TN + FP + FN}$$

- TP（True Positive）：實際是正，預測也為正
- TN（True Negative）：實際是負，預測也為負
- FP（False Positive）：實際是負，卻預測為正（**誤報**）
- FN（False Negative）：實際是正，卻預測為負（**漏報**）

準確率（Accuracy）

📘 **解釋：高準確率代表大部分預測正確，但不一定代表模型好（在不平衡資料集上可能誤導）。**

📌 **適用情況：**

- 當各分類樣本數量大致相同，準確率是可靠的指標。
- 在資料「不平衡」（例如：90% 是 A 類別，10% 是 B）時不適用。

混淆矩陣（Confusion Matrix）

✅ 定義：混淆矩陣是一種用來衡量模型預測結果的表格工具，呈現模型預測與實際分類的對照關係，即能夠具體了解模型在各類別上的預測表現。

📊 二元分類範例如下：

	實際 Positive	實際 Negative
預測 Positive	TP	FP
預測 Negative	FN	TN

混淆矩陣（Confusion Matrix）

	實際為正類（1）	實際為負類（0）
預測為正類（1）	TP（真正）	FP（假正）
預測為負類（0）	FN（假負）	TN（真負）

- TP（True Positive）：實際是正類，模型也預測為正 → 預測正確
- TN（True Negative）：實際是負類，模型也預測為負 → 預測正確
- FP（False Positive）：實際是負類，但模型預測為正 → **誤報**
- FN（False Negative）：實際是正類，但模型預測為負 → **漏報**

分類報告（Classification Report）

✅ **定義：分類報告提供每個類別的四個指標。**

指標名稱	說明
Precision（精確率）	預測為正類中，有多少是真的正類
Recall（召回率）	真實正類中，有多少被正確預測
F1-score	Precision 與 Recall 的加權平均
Support	該類別的真實樣本數

分類報告（Classification Report）

🔎 精確率 Precision：

$$\text{Precision} = \frac{TP}{TP + FP}$$

📌 說明：

- 意思是「**你預測為正類的資料中，有幾個是真的**」
- 若 FP 很多，Precision 就會低
- **高精確率**：表示模型預測為正的樣本中，絕大多數都是對的。
- 適用於「**不想誤判**」的場景，如：癌症篩檢（別隨便說人有病）

分類報告（Classification Report）

🔎 召回率 Recall：

$$\text{Recall} = \frac{TP}{TP + FN}$$

📌 說明：

- 意思是「**所有真的正類中，有多少被模型正確預測**」
- 若 FN 很多，Recall 就會低
- **高召回率**：表示模型能抓到幾乎所有真正的正類。
- 適用於「**不能漏判**」的場景，如：垃圾郵件過濾、毒品快篩

分類報告（Classification Report）

🔎 F1 分數（F1-score）：

$$\text{F1-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

📌 說明：

- Precision 與 Recall 的加權調和平均值，綜合考量 **Precision 和 Recall 的平衡性**
- 非常適合處理不平衡資料
- 若 Precision 或 Recall 有一個非常低，F1 也會低
- 適用於「**需要精確與完整兼顧**」的情境（**不想誤報也不想漏報**）

實務評估指引

建議指標	重點用途	應用時機	解釋
Accuracy	整體準確率	**類別分布均衡**時可用	直觀簡單
Recall（召回率）	實際正類被找出的比例	不想漏報時。 應用於垃圾郵件偵測（**防漏判**）	寧可誤報 不要漏報
Precision（精確率）	預測為正時， 對的機率有多高	不想誤報時。 應用於醫療診斷（**防誤報**）	寧可漏報 不要誤判
F1-score	精確率與召回率的平衡	Precision/Recall 落差大時。 應用於**整體考量**	平衡 Precision 與 Recall

ROC 曲線（Receiver Operating Characteristic）

👉 **主要應用於：二元分類問題**

📌 **定義：**ROC 曲線是以：

- X 軸：False Positive Rate（FPR）

$$FPR = \frac{FP}{FP + TN}$$

- Y 軸：True Positive Rate（Recall / TPR）

$$TPR = \frac{TP}{TP + FN}$$

- 畫出一條隨著閾值變化的曲線，衡量模型在不同預測閾值下的分類能力。

ROC 曲線（Receiver Operating Characteristic）

📈 AUC（Area Under Curve）

- 曲線下的面積，數值在 **0.5～1.0** 之間
- 越接近 **1** 表示**分類性能越好**
- 0.5 表示亂猜，沒有預測能力

💡 適用情境：

- 類別不平衡時更能反映模型整體分類能力
- 適合想了解模型「**整體預測分數表現**」的情境

PR 曲線（Precision-Recall Curve）

👉 **主要應用於：資料不平衡的二元分類問題**

📌 **定義：**

- X 軸：Recall（召回率）
- Y 軸：Precision（精確率）

PR 曲線（Precision-Recall Curve）

💡 特點：

- 比 ROC 更適合 **正類極少數**、**不平衡資料** 的問題
- 可以觀察當你希望模型多找出正類時，會犧牲多少精確率

📈 PR-AUC：

- 與 ROC-AUC 相似，指**曲線下的面積**
- 曲線**越靠近右上角**，表示**模型越好**

R^2 分數（決定係數 / R-squared）

👉 **主要應用於：迴歸問題**

📌 **定義：**R^2 衡量的是模型「**對資料變異量的解釋能力**」，即模型對資料解釋能力的比例，取值範圍：0,1（理論可為負）。

$$R^2 = 1 - \frac{RSS}{TSS}$$

- RSS（Residual Sum of Squares）：殘差平方和，即預測誤差平方和
- TSS（Total Sum of Squares）：總平方和，即實際資料的總變異

R^2 分數（決定係數）

💡 解讀方式：

R^2 → 模型對資料解釋能力的比例

- $R^2 = 1$：完美預測
- $R^2 = 0$：模型毫無解釋能力
- $R^2 < 0$：模型甚至比隨機亂猜還差

MSE（Mean Squared Error）

💡 定義：平均平方誤差，**越小越好**

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- 用途：對離群值敏感，懲罰大誤差

RMSE（Root Mean Squared Error）

💡 **定義：MSE 的平方根，單位與原始目標變數相同**

$$\text{RMSE} = \sqrt{MSE}$$

- 用途：直觀衡量預測值與實際值的偏差程度

MAE（Mean Absolute Error）

💡 **定義：平均絕對誤差，不平方，因此對離群值不敏感**

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- 用途：比 RMSE 更穩健（對 outliers 的影響較小）

何時選哪一種指標

問題特性	適合使用的指標
類別平衡，模型整體準確	Accuracy、F1-score
類別不平衡（如詐騙檢測）	Precision-Recall、PR-AUC
想了解整體預測能力	ROC-AUC
預測數值（迴歸問題）	MSE、RMSE、MAE、R^2

課堂實作範例操作流程

實作練習：二元分類 - 乳癌資料集

📌 從資料讀取、模型訓練、預測、評估到混淆矩陣視覺化，展示了完整的二元分類流程

實作練習：二元分類 - 乳癌資料集

安裝與版本控制

```
!pip install japanize-matplotlib | tail -n 1
!pip install xgboost==0.90 | tail -n 1
```

- !pip install japanize-matplotlib：這是在 Jupyter Notebook 中安裝 Python 套件的指令（前面的 ! 是執行 shell 指令）。
- japanize-matplotlib：這個套件能讓 matplotlib 圖表支援顯示日文（例如標籤、標題等），適用於需要顯示日文資料的視覺化。
- 指定安裝 xgboost 的 0.90 版本，因為新舊版本可能在演算法實作或預設參數上略有差異。此做法常見於應用於教材或比賽結果時，避免版本造成模型表現差異。

實作練習：二元分類 - 乳癌資料集

⚙ 改變顯示參數設定

```
from sklearn import set_config

set_config(print_changed_only=False)
```

- set_config(print_changed_only=False)：告訴 Scikit-learn 在印出模型（例如 print(model)）時，要列出所有參數，而不是只顯示使用者改變的部分。這對學習與理解模型的內部參數非常有幫助。

實作練習：二元分類 - 乳癌資料集

⚠ 警告管理

```
import warnings
warnings.filterwarnings('ignore')
```

- warnings.filterwarnings('ignore')：忽略 Python 中的警告訊息，讓 Notebook 介面更乾淨。但要注意這也可能忽略重要警告，不建議在正式部署中使用。

實作練習：二元分類 - 乳癌資料集

套件載入

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

- pandas（pd）：資料處理與表格分析的主要工具。
- numpy（np）：處理數值與矩陣運算的基礎工具。
- matplotlib.pyplot（plt）：用來畫圖與視覺化數據。

實作練習：二元分類 - 乳癌資料集

🔍 顯示控制

```
import japanize_matplotlib
```

- 這行確保使用 matplotlib 畫的圖能正確顯示日文。對於需要處理亞洲語系資料集很有用（雖然如果是繁體中文，會需要額外中文字型設定）。

```
from IPython.display import display
```

- 匯入 display 函數：在 Jupyter Notebook 中更漂亮、完整地顯示資料表格（比 print(df) 更清楚且支援樣式）

實作練習：二元分類 - 乳癌資料集

🔍 顯示控制

```
# numpy 浮點顯示精確度
np.set_printoptions(suppress=True, precision=4)
```

- suppress=True：不使用科學記號（例如 1e-05 變成 0.0001）
- precision=4：顯示 4 位小數

實作練習：二元分類 - 乳癌資料集

🔍 顯示控制

```
# 在 Pandas 中顯示浮點數的精確度
pd.options.display.float_format = '{:.4f}'.format
```

- 設定 Pandas 表格中的數字也使用 4 位小數顯示，整齊一致。

```
# 顯示資料框中的所有項目
pd.set_option("display.max_columns", None)
```

- 預設 Pandas 顯示資料時，若欄位太多會省略。這行可以確保 所有欄位都會顯示出來，很實用於大型資料表。

實作練習：二元分類 - 乳癌資料集

隨機種子設定

```
# 指定圖表的預設字體大小
plt.rcParams["font.size"] = 14
```

- 設定 matplotlib 繪圖時所有文字（如標籤、標題）的預設字體大小為 14，使圖表更清晰易讀。

```
random_seed = 123
```

- 設定一個固定的「隨機數種子」可以讓程式中的隨機動作（如資料切分、模型隨機初始化等）在每次執行時保持一致，有利於 重現性。

- 常見的做法是在多種資料前處理或模型訓練中加入 random_state=random_seed。

實作練習：二元分類 - 乳癌資料集

🧪 混淆矩陣流程：從資料載入到資料劃分

```
from sklearn.datasets import load_breast_cancer
```

- 從 scikit-learn 載入內建的乳癌資料集（Breast Cancer Wisconsin dataset），這是常見的二元分類練習資料集。

```
cancer = load_breast_cancer()
```

- 將資料集儲存到變數 cancer。這是一個 字典風格的物件，包含：
  - cancer.data：輸入特徵（每筆樣本有 30 個數值特徵）
  - cancer.target：0 表示良性，1 表示惡性

實作練習：二元分類 - 乳癌資料集

```
x = cancer.data
```

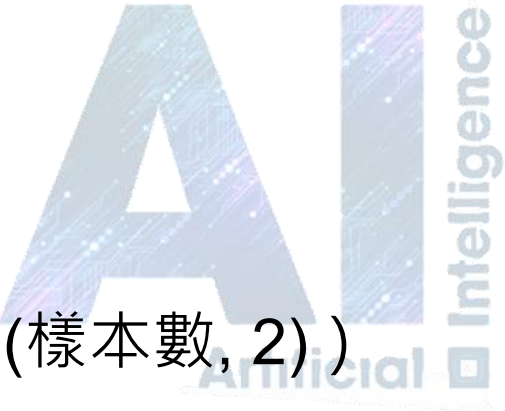
- 將輸入特徵存為變數 x（維度為 (樣本數, 特徵數)）

```
y = 1 - cancer.target
```

- 資料集中，target=0 是良性，target=1 是惡性
- 為了讓模型更直觀判斷「是否為惡性」，這裡反轉標籤：使 惡性 = 1，良性 = 0

```
x2 = x[:, :2]
```

- 為了視覺化簡單，我們只取前兩個特徵來建模。
- x[:, :2] 表示取所有樣本的前 2 個欄位（簡化後的特徵維度變成 (樣本數, 2)）



實作練習：二元分類 - 乳癌資料集

📊 分割資料（訓練 / 測試集）

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(
    x2, y, train_size=0.7, test_size=0.3, random_state=random_seed)
```

- train_test_split：將資料切分為「訓練集」與「測試集」。
- train_size=0.7：70% 用來訓練模型。
- test_size=0.3：30% 用來測試模型效果。
- random_state=random_seed：保持隨機性一致，讓結果可重複。

實作練習：二元分類 - 乳癌資料集

🧠 建立與訓練模型（邏輯迴歸）

```
from sklearn.linear_model import LogisticRegression
algorithm = LogisticRegression(random_state=random_seed)
```

- 導入並初始化邏輯迴歸模型（Logistic Regression），常用於二元分類。
- random_state 確保隨機初始化可重現。

```
algorithm.fit(x_train, y_train)
```

- 將訓練資料送進模型進行訓練（學習如何分類）

實作練習：二元分類 - 乳癌資料集

```
y_pred = algorithm.predict(x_test)
```

- 使用訓練好的模型來預測測試資料，將結果存在 y_pred。

```
score = algorithm.score(x_test, y_test)
```

- 用 score() 方法來計算 模型正確率（accuracy）。
- 這裡會將模型預測與實際標籤比較，輸出正確預測比例。

```
print(f'score: {score:.4f}')
```

- 使用 f-string 印出正確率，保留小數點後 4 位。

實作練習：二元分類 - 乳癌資料集

📉 混淆矩陣分析

```
from sklearn.metrics import confusion_matrix
```

- 混淆矩陣（Confusion Matrix）是分類問題中重要的評估工具，能顯示「正確分類」與「誤判」的細節。

實作練習：二元分類 - 乳癌資料集

📉 混淆矩陣分析

```
matrix = confusion_matrix(y_test, y_pred)
```

- confusion_matrix 根據實際標籤 y_test 和預測結果 y_pred 計算矩陣，回傳如下格式：

	預測=0（良性）	預測=1（惡性）
真實=0（良性）	TN（真正）	FP（假陽）
真實=1（惡性）	FN（假陰）	TP（真陽）

```
print(matrix)
```

- 印出混淆矩陣原始的 numpy 陣列。

實作練習：二元分類 - 乳癌資料集

混淆矩陣美化顯示

- 這是一個自訂函數，將原本純數字的混淆矩陣加上「行與列的標籤」，幫助人更直觀閱讀：
- 列：實際值（正確答案）
- 欄：預測值（模型輸出）

```
def make_cm(matrix, columns):
    # 矩陣 numpy 數組

    # 列項目名稱列表
    n = len(columns)

    # 建立列索引與欄索引標籤
    act = ['正確答案數據'] * n
    pred = ['預測結果'] * n

    # 建立 DataFrame，讓混淆矩陣加上標籤
    cm = pd.DataFrame(matrix,
         columns=[pred, columns],
         index=[act, columns])
    return cm
```

實作練習：二元分類 - 乳癌資料集

混淆矩陣美化顯示

```
# 使用 make_cm 進行混淆矩陣標記
cm = make_cm(matrix, ['良性', '惡性'])
display(cm)
```

- 將剛剛產生的混淆矩陣 matrix 套用我們自訂的標籤美化函數 make_cm。
- display() 是 Jupyter 專用的函數，比 print() 更能美觀顯示 Pandas 表格。

實作練習：二元分類 - 乳癌資料集

📌 **「精準率（Precision）、召回率（Recall）、F1 分數」是分類任務中非常重要的評估指標，特別是在類別不平衡的情況下，這些指標比單純的準確率更能反映模型的品質。**

◆ 導入評估指標所需函數

```
from sklearn.metrics import precision_recall_fscore_support
```

- 這是從 sklearn.metrics 模組中引入一個多功能的函數。
- 它可以一次計算出 精準率（precision）、召回率（recall）、F1 分數（f-score）和 support（樣本數）。
- 適用於二元與多分類問題。

實作練習：二元分類 - 乳癌資料集

◆ 計算三個指標

```
precision, recall, fscore, _ = precision_recall_fscore_support(
    y_test, y_pred, average='binary')
```

- y_test：實際的正確標籤（Ground Truth）
- y_pred：模型預測的標籤
- average='binary'：適用於「二元分類問題」，系統會以 1 為正類（positive），0 為負類（negative）來計算。
- 若是多類別任務，可使用 average='macro'、'micro'、'weighted' 等選項，含義如下：
  - 'macro'：對所有類別計算指標的「簡單平均」
  - 'weighted'：根據每個類別樣本數加權平均
  - 'micro'：全體混合計算 TP/FP/FN

實作練習：二元分類 - 乳癌資料集

◆ 計算三個指標

```
precision, recall, fscore, _ = precision_recall_fscore_support(
    y_test, y_pred, average='binary')
```

- 回傳值：
  - precision（精準率）= TP / (TP + FP)：模型預測為正的樣本中，實際為正的比例。
  - recall（召回率）= TP / (TP + FN)：實際為正的樣本中，成功預測為正的比例。
  - fscore：精準率與召回率的調和平均。F1 = 2 × (precision × recall) / (precision + recall)
  - _：support（每個類別在 y_test 中的樣本數），此處用 _ 忽略。

實作練習：二元分類 - 乳癌資料集

◆ 顯示計算結果

```
print(f'準確率: {precision:.4f}')
print(f'召回率: {recall:.4f}')
print(f'F1分數:   {fscore:.4f}')
```

- 使用 Python f-string 格式化輸出，.4f 表示小數點後保留四位數。
- 列印出三個常見分類模型的性能指標。

實作練習：二元分類 - 乳癌資料集

🧠 使用情境簡介（為什麼這些指標很重要？）

指標	適合關注的情境
精準率	當「錯把負類當成正類」的成本很高時， 例如垃圾信分類、醫療診斷中的誤診。
召回率	當「漏判正類」的成本很高時， 例如癌症檢測、金融詐欺預測。
F1 分數	當你需要在精準率與召回率間做出綜合考量時。

實作練習：二元分類 - 乳癌資料集

📘 PR 曲線（Precision-Recall Curve）

PR 曲線是在**不均衡分類問題（例如詐欺檢測）中非常實用的指標，它顯示模型在不同閾值下的準確率（precision）與召回率（recall）**之間的關係。

🔷 建立 PR 曲線資料

```
# 導入 precision_recall_curve 函數
from sklearn.metrics import precision_recall_curve
```

- 匯入 sklearn 中的 precision_recall_curve 函數，用來根據機率值與實際值，計算出 PR 曲線所需的數據。

實作練習：二元分類 - 乳癌資料集

```
# 獲取 precision、recall、以及對應的 threshold 閾值
precision, recall, thresholds = precision_recall_curve(y_test, y_proba1)
```

- y_test：實際標籤（0 或 1）
- y_proba1：預測為 1 的機率（即 predict_proba 中的第 1 欄）
- 回傳三個陣列：
  - precision：各閾值下的準確率
  - recall：各閾值下的召回率
  - thresholds：不同的閾值

實作練習：二元分類 - 乳癌資料集

```
# 將結果放入 DataFrame 中方便檢視
df_pr = pd.DataFrame([thresholds, precision, recall]).T
df_pr.columns = ['臨界點', '準確率', '召回率']
```

- 使用 pandas.DataFrame() 將三個列表組成一個表格（每列代表一個閾值的評估值），再重新命名欄位。

```
# 顯示閾值 0.5 附近的區段（間隔為10）
display(df_pr[52:122:10])
```

- 只顯示部分資料，方便檢查不同閾值下的 precision/recall 變化情況。

實作練習：二元分類 - 乳癌資料集

◆ 繪製 PR 曲線

```
plt.figure(figsize=(6,6))                    # 設定圖形大小為 6x6 吋
plt.fill_between(recall, precision, 0) # 使用填滿的區域來表示曲線下的面積
plt.xlim([0.0, 1.0])                         # X 軸範圍限制為 0~1
plt.ylim([0.0, 1.0])                         # Y 軸範圍限制為 0~1
plt.xlabel('召回率')                          # 設定 X 軸標籤
plt.ylabel('準確率')                          # 設定 Y 軸標籤
plt.title('PR曲線')                          # 設定圖形標題
plt.show()                                   # 顯示圖形
```

實作練習：二元分類 - 乳癌資料集

◆ PR 曲線的 AUC（面積）

```
from sklearn.metrics import auc          # 匯入 AUC 計算工具
pr_auc = auc(recall, precision)          # 計算 PR 曲線下面積
print(f'PR曲線下面積: {pr_auc:.4f}')    # 顯示結果（保留四位小數）
```

- auc(x, y)：計算以 x 為橫軸、y 為縱軸下的曲線面積。
- PR 曲線的面積越大，代表模型在不平衡資料上表現越穩定。

實作練習：二元分類 - 乳癌資料集

📘 ROC 曲線（Receiver Operating Characteristic）

ROC 曲線用來描述模型在不同閾值下的 真陽性率（TPR） 與 假陽性率（FPR），是最常見的二元分類評估工具。

🔷 產生 ROC 曲線資料

```
from sklearn.metrics import roc_curve
```

- 匯入 roc_curve，可用來計算不同閾值下的 FPR、TPR。

實作練習：二元分類 - 乳癌資料集

```
fpr, tpr, thresholds = roc_curve(y_test, y_proba1, drop_intermediate=False)
```

- fpr：假陽性率（False Positive Rate）
- tpr：真陽性率（True Positive Rate），也叫敏感度
- thresholds：對應這些比率的閾值

```
df_roc = pd.DataFrame([thresholds, fpr, tpr]).T
df_roc.columns = ['臨界點', '假陽性率', '敏感度']
```

- 將資料組成 DataFrame，方便觀察不同閾值下的 ROC 指標

```
display(df_roc[21:91:10])  # 顯示第21~91筆（每10筆取一），方便查看臨界點區段
```

實作練習：二元分類 - 乳癌資料集

◆ 使用更準確的模型重新繪圖

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(
  x, y, train_size=0.7, test_size=0.3, random_state=random_seed)
```

- 將資料切分為訓練集與測試集，70% 訓練，30% 測試，設定 random_seed 確保可重現。

實作練習：二元分類 - 乳癌資料集

```
algorithm = LogisticRegression()          # 建立邏輯迴歸模型
algorithm.fit(x_train, y_train)           # 用訓練資料訓練模型
```

```
y_pred = algorithm.predict(x_test)     # 預測分類結果（0/1）
y_proba1 = algorithm.predict_proba(x_test)[:,1]  # 取出屬於類別 1 的機率
```

```
fpr, tpr, thresholds = roc_curve(y_test, y_proba1)  # 取得 ROC 曲線資料
```

實作練習：二元分類 - 乳癌資料集

◆ 繪製 ROC 曲線並計算 AUC

```
plt.figure(figsize=(6,6))
plt.plot([0, 1], [0, 1], 'k--')
plt.fill_between(fpr, tpr, 0)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.0])
plt.xlabel('假陽性率')
plt.ylabel('敏感度')
plt.title('ROC曲線')
plt.show()
```

```
roc_auc = auc(fpr, tpr)
print(f'ROC曲線下面積:{roc_auc:.4f}')
```

- auc(fpr, tpr)：計算 ROC 曲線下面積（AUC）
- AUC 越接近 1，模型表現越好；0.5 表示與隨機猜測差不多。

實作練習：二元分類 - 乳癌資料集

✅ 總結：PR vs. ROC

項目	PR 曲線	ROC 曲線
橫軸	召回率（Recall）	假陽性率（FPR）
縱軸	精準率（Precision）	真陽性率（TPR）
適用場景	標籤不平衡（少量陽性樣本）	標籤較均衡時使用
衡量指標	AUC（PR-AUC）	AUC（ROC-AUC）
越高越好	✅	✅

實作練習：房價預測

🧭 單元流程架構

1. 資料讀取與預覽
2. 資料前處理
3. 切分訓練與測試資料
4. 建立回歸模型（以線性回歸為例）
5. 模型訓練
6. 模型預測
7. 模型評估（MSE、RMSE、MAE、R^2）
8. 結果視覺化與解釋

實作案例：使用加州房價資料集

1 載入資料與預覽python複製程式碼

🔍 說明：

- fetch_california_housing() 是 sklearn 提供的房價資料集
- X 是特徵資料（例如：人口、房間數、收入等）
- y 是目標變數（房價，以千美元為單位）

```
from sklearn.datasets import fetch_california_housing
import pandas as pd

# 載入加州房價資料集
data = fetch_california_housing()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = pd.Series(data.target, name='HouseValue')

# 顯示前五筆資料
display(X.head())
display(y.head())
```

實作案例：使用加州房價資料集

2 資料切分

```
from sklearn.model_selection import train_test_split


X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)
```

📌 將資料分成訓練集 80% 和測試集 20%。

實作案例：使用加州房價資料集

3 建立與訓練模型

```
from sklearn.linear_model import LinearRegression


# 建立線性回歸模型
model = LinearRegression()
model.fit(X_train, y_train)
```

📌 LinearRegression() 是最基礎的回歸模型，適合觀察初始表現。

實作案例：使用加州房價資料集

4 預測與模型評估

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np


# 模型預測
y_pred = model.predict(X_test)
```

實作案例：使用加州房價資料集

4 預測與模型評估

```
# 評估指標計算
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'MSE: {mse:.3f}')
print(f'RMSE: {rmse:.3f}')
print(f'MAE: {mae:.3f}')
print(f'R²: {r2:.3f}')
```

實作案例：使用加州房價資料集

📊 模型結果視覺化

```
import matplotlib.pyplot as plt


plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.plot([y_test.min(), y_test.max()],
         [y_test.min(), y_test.max()],
         'r--', lw=2)
```

```
plt.xlabel('實際房價')
plt.ylabel('預測房價')
plt.title('預測 vs 實際')
plt.grid(True)
plt.show()
```


📌 若點多數集中在紅色虛線（理想預測）附近，代表預測效果佳。

迴歸問題 - 指標應用說明

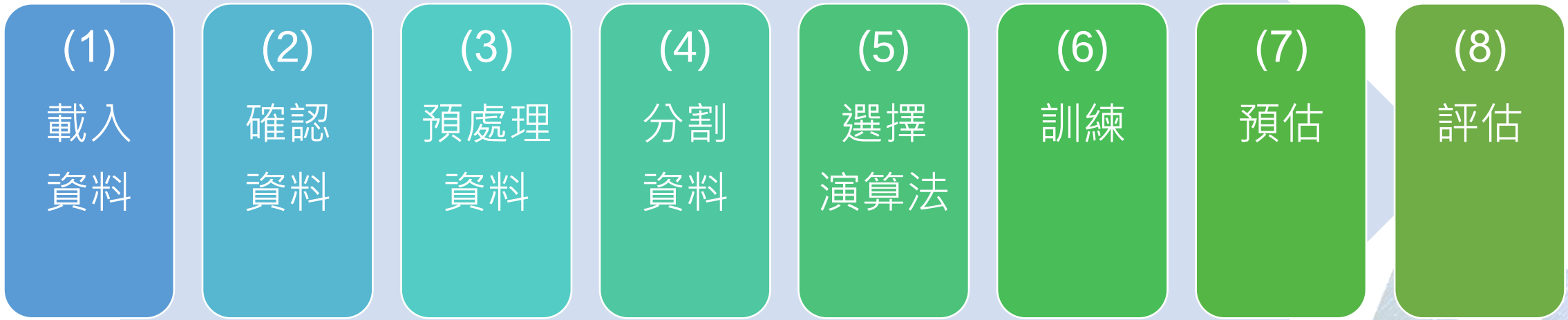
指標	效果	適用情境
MSE	對誤差懲罰大	偏好準確率但容忍離群值
RMSE	與原始單位相符	對於直觀呈現效果佳
MAE	對離群值不敏感	預測較穩健
R^2	模型解釋能力	判斷整體模型品質

大數據資料分析實作

總　結

博雅(科技)課程

「大數據資料分析」完整流程八大步驟



「大數據資料分析」完整流程八大步驟

(1) 載入資料（Load Data）

- 目標：將資料來源（CSV、資料庫、API、JSON、網頁等）匯入程式中。
- 常用工具：pandas.read_csv()、SQL、爬蟲、資料庫連線等。

📌 重點：

- 確保資料來源可靠、格式正確。
- 檢查編碼問題、遺漏值或異常值。

「大數據資料分析」完整流程八大步驟

(2) 確認資料（Understand Data）

- 目標：初步理解資料結構、欄位意義與資料型態。
- 常用工具：**df.head()**、**df.info()**、**df.describe()**、**df.plot()** 等。

📌 重點：

- 了解各欄資料分布與單位。
- 判斷目標變數（Label）與特徵（Features）。
- 判別問題類型：分類、迴歸、群聚等。

「大數據資料分析」完整流程八大步驟

(3) 預處理資料（Preprocess Data）

- 目標：將資料清理並轉換為機器學習可接受的格式。
- 處理方式：
  - 缺失值處理（補值、刪除）
  - 類別資料轉換（One-hot Encoding、Label Encoding）
  - 數值正規化（Normalization/Standardization）
  - 特徵選擇與轉換（PCA、特徵工程）

📌 重點：

- 資料乾淨是模型學習的基礎。
- 對類別型、時間型、文字型資料特別注意。

「大數據資料分析」完整流程八大步驟

(4) 分割資料（Split Data）

- 目標：將資料分為訓練集與測試集，以便訓練與驗證模型效能。
- 常用方式：train_test_split()（例如 80% 訓練 / 20% 測試）

📌 重點：

- 保證測試資料在訓練時不被洩漏。
- 可以加入驗證集（Validation Set）或使用交叉驗證（Cross-Validation）。

「大數據資料分析」完整流程八大步驟

(5) 選擇演算法（Select Algorithm）

- 根據問題類型選擇對應的 AI 演算法：

問題類型	常用演算法
分類	Logistic Regression, SVM, Random Forest, XGBoost, CNN
迴歸	Linear Regression, XGBoost Regressor, SVR
時間序列預測	ARIMA, LSTM
關聯分析	Apriori, FP-Growth
分群	K-Means, DBSCAN

📌 重點：

- 問題決定演算法。
- 可比較多種演算法挑選效果最佳者。

「大數據資料分析」完整流程八大步驟

(6) 訓練模型（Train Model）

- 目標：使用訓練集讓演算法學習輸入資料與目標資料的關係。
- 範例語法：**model.fit(X_train, y_train)**

📌 重點：

- 訓練過程會調整模型參數。
- 可以使用「超參數調整」（Hyperparameter Tuning）提升效果（如 GridSearchCV）。

「大數據資料分析」完整流程八大步驟

(7) 預估（Predict）

- 目標：讓訓練好的模型針對新資料做出預測。
- 範例語法：**model.predict(X_test)**

📌 重點：

- 確保輸入的資料格式與訓練時一致。
- 可以套用在即時預測、推薦系統、商業應用中。

「大數據資料分析」完整流程八大步驟

(8) 評估模型（Evaluate Model）

- 目標：使用測試集評估模型的預測效果。
- 常用指標：

任務類型	評估指標
分類	Accuracy, Precision, Recall, F1, AUC
迴歸	MSE, RMSE, MAE, R^2 Score
分群	Silhouette Score, Calinski-Harabasz

📌 重點：

- 選擇正確的評估指標依據任務。
- 可視覺化混淆矩陣、ROC 曲線、預測 vs 實際圖。

機器學習在大數據分析的應用與意義

流程步驟	大數據應用重點	實際意義
1. 載入資料	整合多源資料（IoT、社群媒體、感測器、ERP、網頁點擊）	支援數百 GB 至 TB 級資料存取，奠定資料治理基礎
2. 確認資料	快速掃描數據分布與異常、資料類型統計	幫助了解資料的維度與品質、確認目標與特徵欄位
3. 預處理資料	分散式清洗（如 Spark）、特徵工程自動化（如 Feature Store）	提升資料一致性、準確性，是 AI 成敗的關鍵
4. 分割資料	進行大規模交叉驗證或即時切分（Real-time Splitting）	確保模型不過擬合，同時可處理線上學習或滾動預測
5. 選擇演算法	根據任務與資料類型自動推薦最佳模型（AutoML, HPO）	提升模型開發效率與可擴展性
6. 模型訓練	使用 GPU 或分散式架構訓練大型模型（如深度學習、XGBoost）	快速處理大量特徵與樣本，支持企業級應用
7. 預估/推論	即時預測（Real-time Inference）、批次推論（Batch Scoring）	將模型成果轉為實際決策支援工具
8. 模型評估	結合視覺化儀表板、持續監控模型表現（MLOps）	持續追蹤模型品質、實現可解釋性與信任度

大數據與機器學習整合的核心意義

重點面向	說明
規模與速度 （Scale & Speed）	可處理海量資料，支援即時分析與預測
自動化決策 （AI-driven Decisions）	將模型嵌入業務流程中，提供預測性洞察
模型可解釋性 （Explainability）	尤其在醫療、金融等領域，確保模型透明可信
商業價值最大化	從資料中找出模式，改善流程、預測趨勢、降低風險

各領域的大數據 + 機器學習應用實例

領域	應用場景	解決問題	使用演算法
零售	顧客購物行為預測、動態定價	增加銷售、減少庫存成本	分群（K-Means） 分類（RF、XGBoost）
金融	信用風險評估、詐欺偵測	降低壞帳與損失風險	分類（SVM、XGBoost） 異常檢測（Isolation Forest）
醫療	疾病預測、醫療影像辨識	提高診斷準確度與效率	CNN、分類、強化學習
製造	預測性維護、良率分析	減少停機時間、提升品質	時間序列分析、分類、分群
智慧城市	交通流量預測、能源管理	提升城市效率與永續發展	LSTM、回歸、即時預測
物流供應鏈	需求預測、最佳路徑規劃	優化配送與存貨管理	時間序列、強化學習、回歸模型
教育科技	學習成效預測、個人化推薦	提升學習效果與資源配置	分類、回歸、推薦系統
社群媒體	情緒分析、熱門貼文預測	掌握趨勢與行銷策略	NLP 模型（BERT）、分群、分類

應用 AI 解決不同問題的方式

應用領域	問題類型	使用方法與說明
醫療診斷	分類	預測是否患病（癌症判別、病症分型）
房價預測	迴歸	根據面積、地段等預測房價
銷售預測	時間序列	根據歷史銷售資料預測未來銷售量
市場分析	分群	將顧客群分成不同類型（顧客細分）
電商推薦	關聯分析	根據消費者購物行為推薦其他商品
客服問答系統	自然語言	使用 NLP 模型進行語意理解與回覆
智慧工廠品質檢測	分類	影像辨識缺陷產品
智慧交通預測	時間序列	根據天氣與流量預測車流量

AI × 大數據 = 資料價值的實現

✅ 「機器學習 + 大數據技術」能夠：

- 將資料轉為 可行動的決策依據
- 將 AI 從「理論研究」推向「商業實戰」
- 協助企業與組織在競爭中贏得先機

「大數據資料分析實作」課程