機器學習技術 Lab3

0116F137 陳廣能

一、實驗目的

本實驗旨在利用樹狀分類器（Decision Tree、Random Forest、XGBoost）對心臟病資料集進行預測，並比較各模型的分類效能。

二、實驗方法

1. 資料集來源：使用 Kaggle heart-disease-uci 資料集。

2. 資料前處理：
   - 補齊缺失值（數值型以中位數，類別型以眾數填補）。
   - 類別型欄位進行編碼。
   - 移除不必要欄位。
   - 主要程式碼片段如下：

```python
for feature in ['trestbps', 'chol', 'thalch', 'oldpeak', 'ca']:
    if data[feature].isnull().any():
        data[feature].fillna(data[feature].median(), inplace=True)
for feature in ['fbs', 'restecg', 'exang', 'slope', 'thal']:
    if data[feature].isnull().any():
        data[feature].fillna(data[feature].mode()[0], inplace=True)
for col in data.select_dtypes(include='object').columns:
    data[col] = data[col].astype('category').cat.codes
```

3. 資料切分：將資料分為訓練集與測試集（8:2）。

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

4. 模型訓練與評估:

- 決策樹（Decision Tree）

  ```python
  dt_model = DecisionTreeClassifier(
      random_state=42,
      max_depth=6,
      min_samples_split=10
  )
  dt_model.fit(X_train, y_train)
  dt_pred = dt_model.predict(X_test)
  print("Decision Tree Accuracy:", accuracy_score(y_test, dt_pred))
  ```

  - 決策樹的優點是模型結構直觀、易於解釋，可以直接觀察每個特徵對預測的貢獻。

- 隨機森林（Random Forest）

  ```python
  rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
  rf_model.fit(X_train, y_train)
  rf_pred = rf_model.predict(X_test)
  print("Random Forest Accuracy:", accuracy_score(y_test, rf_pred))
  ```

  - 隨機森林透過多棵樹的集成，能有效降低過擬合，提升泛化能力。

- XGBoost

  ```python
  from xgboost import XGBClassifier
  xgb_model = XGBClassifier(
      n_estimators=100,
      max_depth=6,
      learning_rate=0.1,
      random_state=42,
      use_label_encoder=False,
      eval_metric='mlogloss'
  )
  xgb_model.fit(X_train, y_train)
  xgb_pred = xgb_model.predict(X_test)
  print("XGBoost Accuracy:", accuracy_score(y_test, xgb_pred))
  ```

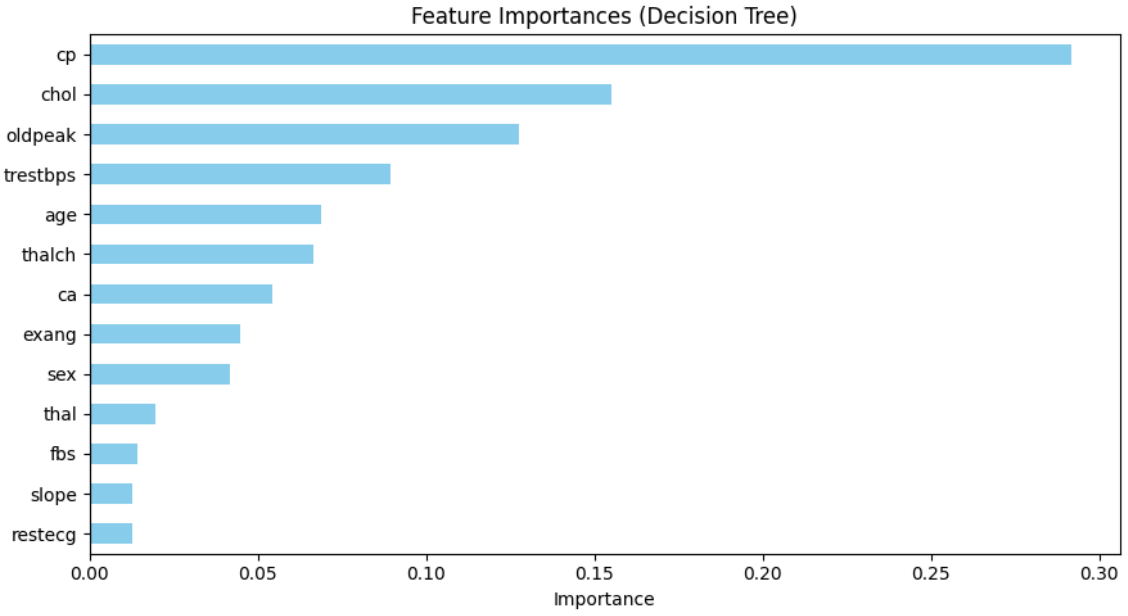
  - XGBoost 屬於梯度提升樹，能逐步修正前一輪模型的錯誤，對於複雜資料有極佳的擬合能力。

- 評估指標：Accuracy、Confusion Matrix、Classification Report

5. 特徵重要性分析：繪製決策樹特徵重要性圖。

```python
importances = pd.Series(
    dt_model.feature_importances_,
    index=X.columns
)
importances.sort_values().plot(kind='barh', color='skyblue')
plt.title("Feature Importances (Decision Tree)")
plt.xlabel("Importance")
plt.tight_layout()
plt.show()
```

- 透過特徵重要性圖，可以直觀了解哪些變數對模型預測影響最大。



三、實驗結果

1. 決策樹（Decision Tree）
   - Accuracy: 約 0.8033
   - 混淆矩陣與分類報告顯示模型對兩類皆有不錯的預測能力。
   - 重要特徵：如 cp、thalach、oldpeak 等。

2. 隨機森林（Random Forest）
   - Accuracy: 約 0.8689
   - 整體表現優於單一決策樹，泛化能力較佳。

3. XGBoost
   - Accuracy: 約 0.8852

- 三種模型中表現最佳，對於複雜資料有更強的擬合能力。

4. 模型比較

Model	Accuracy
Decision Tree	0.8033
Random Forest	0.8689
XGBoost	0.8852

四、討論與心得

在這次實作過程中，我深刻體會到決策樹模型的結構非常清晰，讓我能夠直接觀察每個特徵對預測的影響，這對於模型的解釋性非常有幫助。不過，實際操作時也發現決策樹容易過擬合，對測試資料的泛化能力有限。進一步嘗試隨機森林後，我發現透過集成多棵樹，模型的穩定性與準確率都明顯提升，也比較不容易受到單一特徵或雜訊的影響。XGBoost 的梯度提升策略則讓我印象深刻，它能針對難以分類的樣本進行修正，因此在本次實驗中表現最佳。透過特徵重要性分析，我發現像 cp（胸痛型態）、thalach（最大心跳）、oldpeak（運動誘發ST段壓力）等特徵對預測心臟病有顯著影響，這讓我更理解資料中各變數的實際意義。此外，這次也讓我體會到資料前處理的重要性，像是缺失值填補、類別編碼等步驟，對模型效能有很大的影響，良好的前處理是提升模型表現的基礎。

五、結論

本實驗比較三種樹狀分類器於心臟病預測之效能，XGBoost 準確率最高，建議在實務應用中優先考慮集成方法。

樹狀分類器的精隨在於：

- 能處理非線性特徵與異質資料
- 支援特徵重要性分析
- 集成方法（如隨機森林、XGBoost）能大幅提升預測效能與穩定性