**資料科學導論**

**Competition 1**

組員：施其均、蕭捷晨、黃書璿

1. 資料分析

(一)初始化變量

dataset\_names儲存每個數據集，X\_trains, y\_trains ,X\_tests分別儲存訓練集的Feature、訓練集的標籤、測試集的標籤。

(二)遍歷Competition\_data

folder\_path指向包含數據集的主目錄，再使用os.listdir(folder\_path)獲取路徑中所有資料夾，爾後運用for迴圈遍歷數據集，透過panda的read\_csv方法讀取X\_train ,y\_train ,X\_test，並添加到相對應列表。

(三)辨別數據型和類別型資料

定義auto\_identify\_features方法，numeric\_features儲存數據型資料，categorical\_features儲存類別型資料，利用pd.api.types.is\_numeric\_dtype判斷是否為數據型資料。定義X\_train\_processed, X\_test\_processed, y\_train\_processed儲存處理後的數據。

(四)不同特徵處理

ColumnTransformer對數據集的不同特徵做相異的處理，StandardScaler進行標準化，接著挑 出類別特徵進行OneHot編碼，把類別型資料轉換成二進制向量。X\_train\_scaled和X\_test\_scaled是標準化後的數據。

(五)SMOTE應用

處理類別的不平衡，X\_resampled和y\_resampled是經過SMOTE平衡的數據。

貳、方法嘗試

(一)前處理

嘗試做資料的前處理，線性回歸等資料調節，但表現不盡理想。

(二)model

將model換成lightgbm，會在部分dataset表現不佳，得到的AUC結果是0.82。

(三)remote

在前處理階段加入remote，使得不平衡的資料，在之後的AUC表現進一步提升。

(四)soft voting

加入soft voting，最後的集成模型根據每個基礎模型的類別機率進行加權平均，接著VotingClassifier將多個模型集合成最終模型。結果上升至0.845，表示能夠彌補模型不同帶來的缺點。為何不直接voting，而使用soft voting? 因為soft voting根據每個模型的概率計算，可以針對不同模型設定不同權重，照模型的表現配予影響力。

(五)手動分析

試著不使用soft voting，分析每個dataset能取得最好成績的方法，但結果似乎overfitting，且AUC不如soft voting的結果高。再者，手動分析的running time相當的久，而且最佳結果只能在0.7左右。

(六)OonHotEncoder

最後在soft voting的基礎上增加OneHotEncoder幫助處理類別型資料，將其轉化為二進制向量，得到最高的0.86。

參、結果觀察

將X\_test\_processed, y\_train\_processed分割成訓練集(80%)和測試集(20%)。實作四個基模型，分別是lightgbm-基於梯度提升的高效樹模型，適合大數據集、rf-有多個決策樹的集合模型，適用於非線性整理、svc-使用線性基礎和機率估計，適用線性可分類的數據、lr-簡單基礎線性模型，作為基準。

prob\_lgb、prob\_rf、prob\_svc、prob\_lr分別存放對測試集進行機率預測的結果，爾後計算出各個基模型在測試集的AUC表現，以便後續分配權重。透過soft voting訓練出最終模型後，對測試集實作機率預測，最後輸出AUC分數，觀察其表現。

經過觀察，大部分模型的AUC超過0.8甚至0.9，表明這些模型在相應測試集上表現良好，但也有部分模型產生0.65等接近0.5(隨機猜測)的AUC，體現出諸如此類的模型表現相對較差，猜測原因是資料集特徵不足或者基模型並沒有充分訓練。

肆、方法選定

(一)資料前處理方式

將資料拆分成數據型特徵和類別型特徵，數據型特徵使用StandardScaler進行標準化處理，而類別型特徵使用OneHotEncoder轉換為數值向量。爾後，套用SMOTE解決類別不平衡的問題，透過生成合成樣本，以增加少數類別的樣本量。

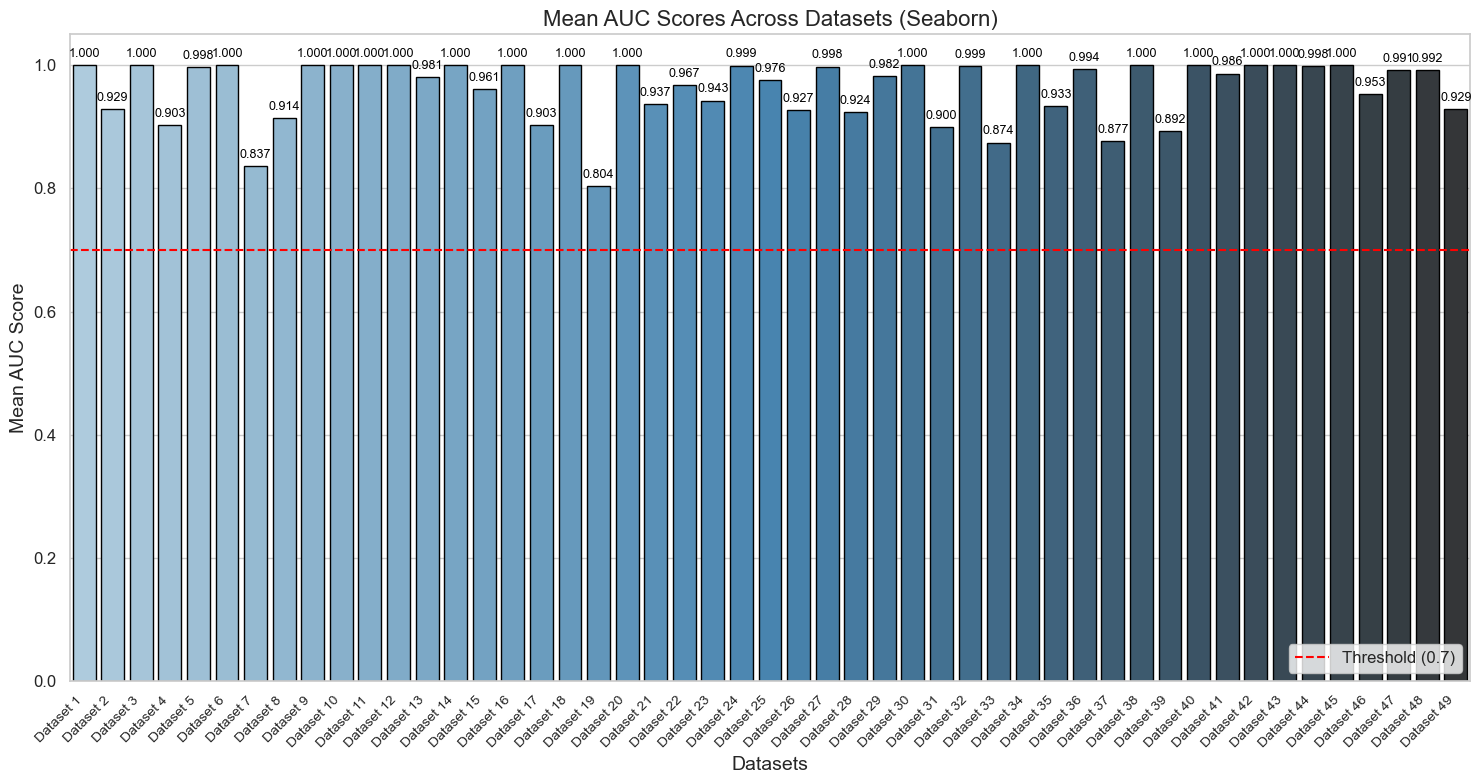
(二)基模型建立

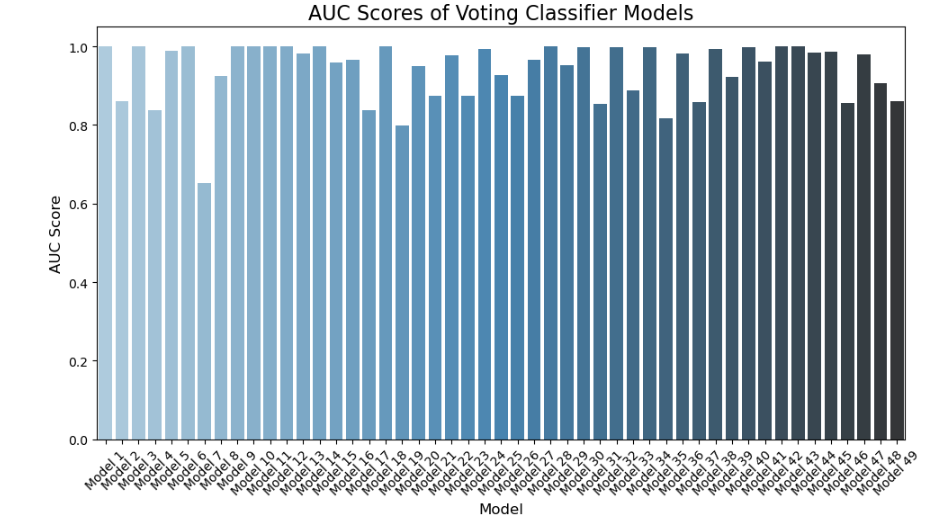
選擇LightGBM、RandomForest、SVC、邏輯回歸四種常用的機器學習模型作為基模型，這些基模型涵蓋梯度提升樹、集成樹、非線性支持向量機和線性模型，比起只使用單一模型更具模型多樣性。

(三)評估方法

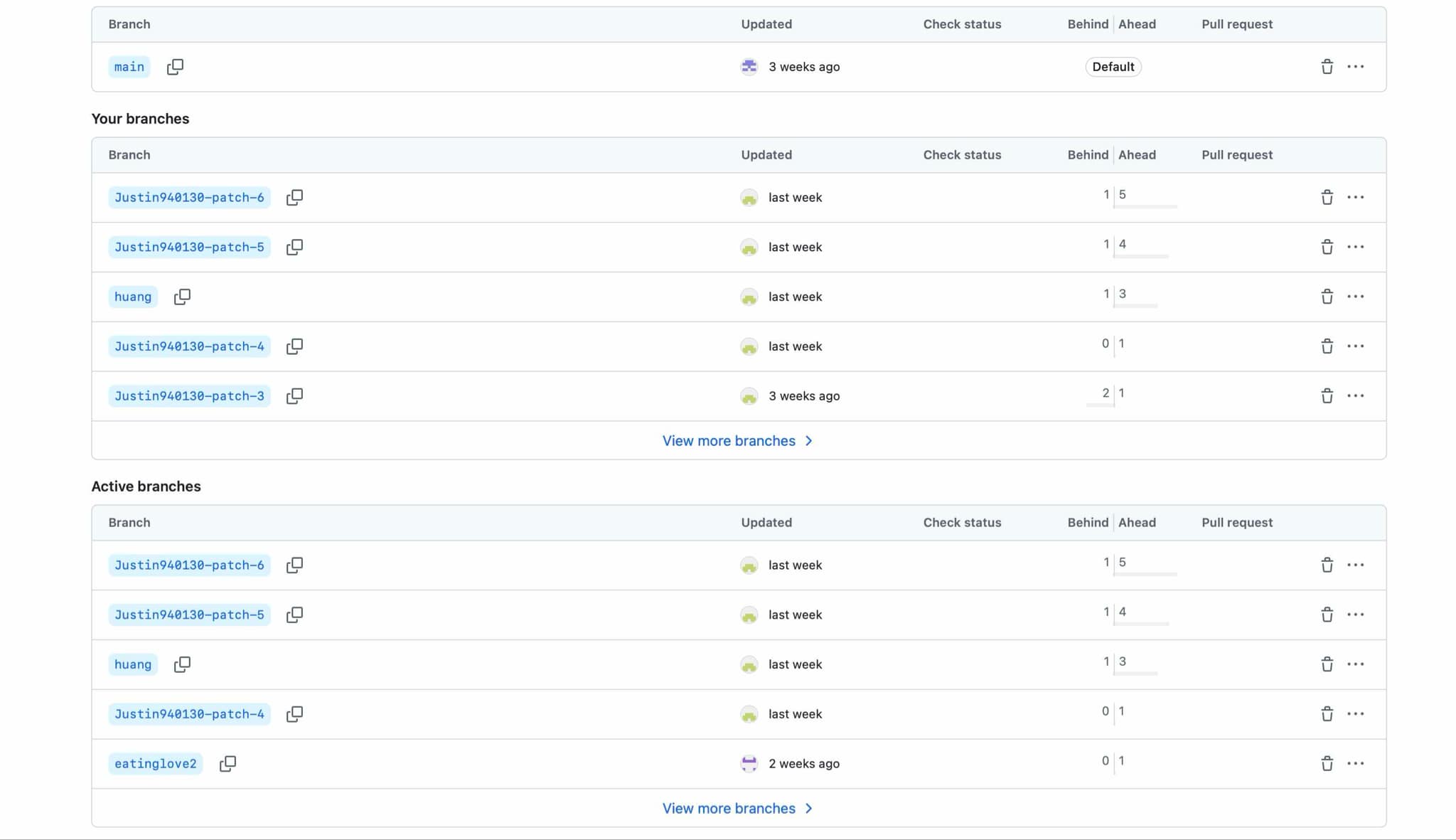
針對每個基模型，我們去計算其在測試集上的AUC表現，評斷其在二元分類問題上的區別能力。

接著根據模型的AUC設置權重，採用soft voting將四個模型進行加權集成。最後在每個dataset中，對soft voting模型進行訓練，並計算最終的AUC。

\*沒有OneHotEncoder的表現

\*有OneHotEncoder的表現

雖然在原本的圖auc表現較佳 但是在上傳之後測試隱藏資料 發現表現並沒有比較好 可能是因為過擬合的關係。

Github