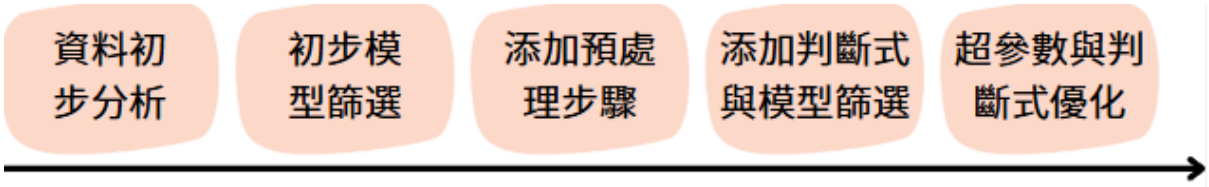
資料科學導論 CP1 報告

組別：Sorry my bad (DS105)

組員：何寬羿(C34104032)、林業誠(E24105038)、曾柏誠(E24126717)

一、 專案思考與製作歷程



二、資料初步分析

1. 本次比賽包含 49 個資料集，每個資料集結構相同，因此在進行移除極端值等預處理時，能設計統一的預處理流水線(仍需考慮不同資料集的資料差異)。

2. 數值型的數值成呈正負分布，並且每個特徵的分布尺度都不同。因此，可能需要進一步標準化或其他預處理。

3. 特徵的標籤不明，無法直覺性地假設某一特徵與預測值的關聯性，因此需要花較多時間去進行資料處理，以釐清不同特徵與預測值的關係。

以上為單就初步分析資料集來看，我們發現的特別之處。在預處理、模型訓練、結果分析時，本組有採用更多技巧以提高預測的準確性，其中也不乏許多以往我們進行預測時不曾使用過的預處理方式、模型。

三、初步模型篩選與過濾過程

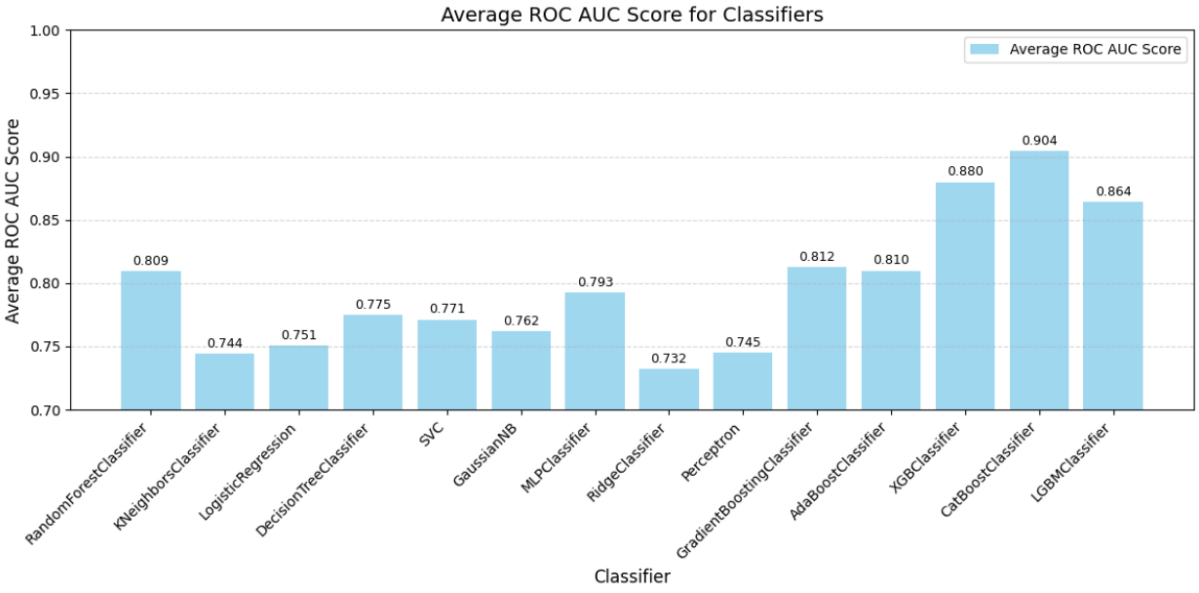
```
## Train & Test & Split & Build Model
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression, RidgeClassifier, Perceptron
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neural_network import MLPClassifier
from xgboost import XGBClassifier
import lightgbm as lgb
```

1. **篩選方法**：

- **篩選對象**：
  - 如上圖所示，包括多種常見的分類模型，涵蓋樹模型、線性模型和集成模型等。
- **選擇依據**：
  - 這些模型被廣泛應用於各類數據集，具有良好的適應性和穩定性。
- **參數調整**：
  - 通過簡單的手動參數調整（如樹的深度、學習率等），提升每個模型的平均 ROC AUC 分數。
  - 此階段的目標並非追求最優結果，而是快速測試各模型的適配性，找出能在此資料集上取得不錯表現的模型。
- **篩選基準**：
  - 根據調整後的平均 ROC AUC 分數設定最低基準分數。
  - 僅保留達到基準的模型，刪除那些表現較差且可行性較低的模型。

2. **結果**：

- 初步篩選後，僅保留了表現良好且具有潛力的模型作為候選模型。
- 篩選出來的模型：`AdaBoostClassifier`，`GradientBoostClassifer`，`RandomForestClassifier`，`CatBoostClassfier`，`XGBClassfier`，`LGBMClassifier`
  (也有使用 MLP 等進階模型，不過因為準確率問題，後續不再嘗試使用)
- 大幅減少了後續需要優化的模型數量，集中資源於潛力最高的模型進行深入分析。
- 這樣的篩選策略不僅提高了效率，也避免了在後續階段花費不必要的精力在效果不佳的模型上。



四、 添加預處理步驟

1. 最終選擇的預處理步驟：

(1) 處理數值型特徵

- **移除異常值 (Remove Outlier)**：根據四分位距（`IQR`）將數值型特徵中超出上下界的值調整到合理範圍內，減少極端值對模型的影響。
- **移除低方差特徵 (Remove Low Variance)**：刪除變異性極低的特徵（閾值為 0.0001），確保模型關注有用的特徵。
- **調整偏態 (Adjust Skewness)**：對偏態過高的數值型欄位進行處理：
  - 全為正數時使用 `Box-Cox` 轉換。
  - 包含非正數時採用對數轉換（`log1p`）。
  - 目的是使數據分布更平衡，提升模型穩定性。

(2) 處理分類特徵（One-hot Encoding）

- **獨熱編碼**：將分類特徵進行 `One-hot Encoding`，轉換為數值格式：
  - 使用 `OneHotEncoder`，自動忽略未知類別，保證兼容性。
  - 如果分類特徵為空，則返回空數組以保證數據結構一致性。

(3) 結合數值型與分類特徵

- 使用 `np.hstack` 合併數值特徵與獨熱編碼後的分類特徵，形成完整的數據集。
- 同時生成所有特徵名稱，確保列數與數據列對應，避免列名過多或過少的問題。

(4) 選擇重要特徵 (Select Important Features)

- 使用特徵重要性篩選方法，保留對模型貢獻較大的特徵：
  - 評估特徵的重要性分數，並保留重要性超過閾值的特徵。
  - 確保至少保留累計重要性占前 80% 的特徵。
- 過濾掉多餘特徵後，輸出篩選出的重要特徵名稱。

(5) 處理類別不平衡問題 (Handle Class Imbalance)

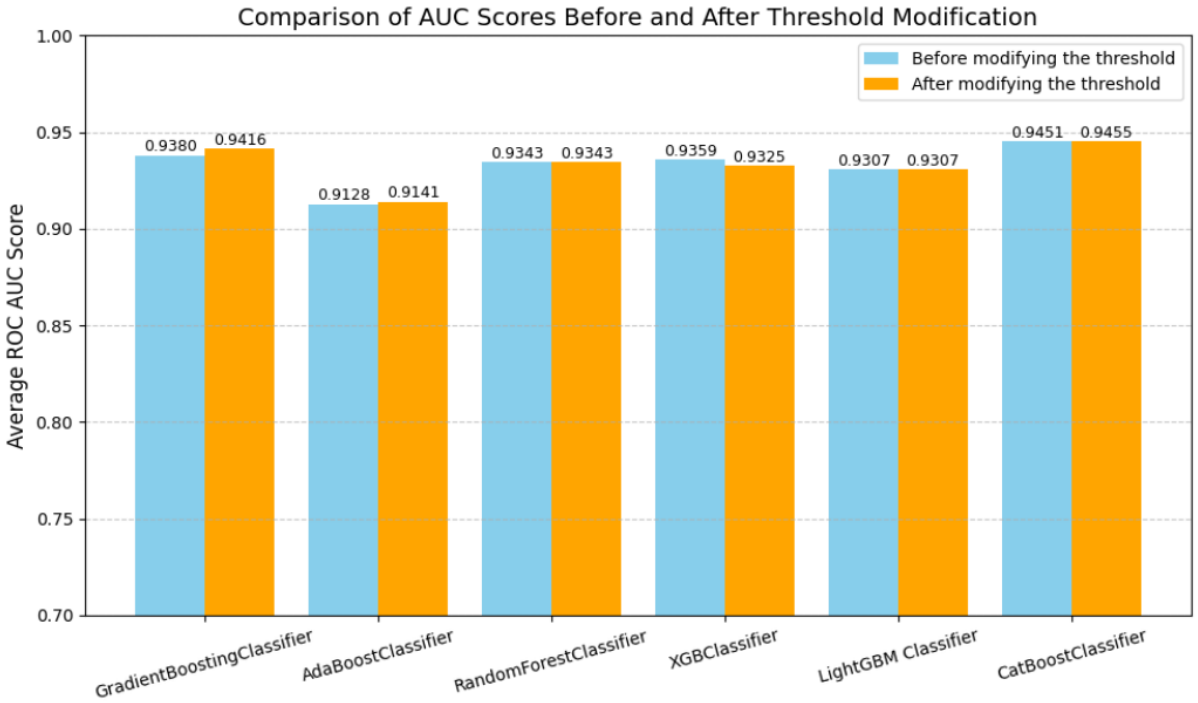
- 利用 `SMOTE` 或 `ADASYN` 技術生成少數類別的合成樣本，平衡數據中的類別比例。
- 默認採用 `SMOTE` ，提升模型對少數類別的預測能力。

(6) 確保數據格式一致

- 將最終處理後的數據轉換為 `DataFrame` 格式，並將篩選出的特徵名稱對應到數據列中。
- 確保數據集的結構與模型兼容，便於後續建模。

※附註

當然，以上的預處理步驟所使用的閾值（`threshold value`）或其他參數均是經過多次嘗試後得出的相對最佳設定，旨在提升數據質量與模型性能。



2. 曾嘗試但未使用的預處理方法

(1) 數據標準化 (Standardization)

- 將數值型特徵按均值和標準差進行標準化，轉換為均值為 0、標準差為 1 的分佈。
- **結果**：對樹模型效果不佳，因為這類模型不依賴特徵的尺度，反而可能破壞數據的結構。

(2) 數據歸一化 (Normalization)

- 將數值型特徵縮放到固定範圍（如 0 到 1）。
- **結果**：對於樹模型幫助有限，甚至壓縮了數據範圍，難以捕捉特徵的非線性關係。

(3) 移除高度相關性特徵 (Remove High Variance)

- 刪除相關性高的特徵以避免冗餘。
- **結果**：導致部分重要信息被移除，對樹模型的效果反而變差，因為冗餘特徵可能有助於模型學習。

(4) 生成交互特徵 (Generate Interaction Features)

- 創造特徵間的交互項（如相乘或非線性組合）。
- **結果**：生成大量特徵後，數據維度增加，但信息增益有限，反而導致過擬合。

3. **Optuna 在閾值優化中的嘗試**

- 嘗試使用 Optuna 調整預處理的閾值（如 IQR、方差閾值等），但試驗數量有限，難以涵蓋所有可能的參數組合。
- **最終方案**：採用手動調整，針對不同情境進行測試，獲得更穩定的效果。

4. **預處理整合的挑戰與解決方案**

(1) **效能限制**：
- 預處理大多在 CPU 上執行，影響了 GPU 訓練效率，限制了 Optuna 試驗數量。

(2) **工具限制**：
- 嘗試使用 Dask 提高效率，但部分 scikit-learn 方法與 Dask 不兼容，導致流程受限。

(3) **解決方案**：
- 回歸使用 NumPy 和 scikit-learn，確保穩定且高效的處理效果。

五、添加判斷式

Conditional Preprocessing 的說明：

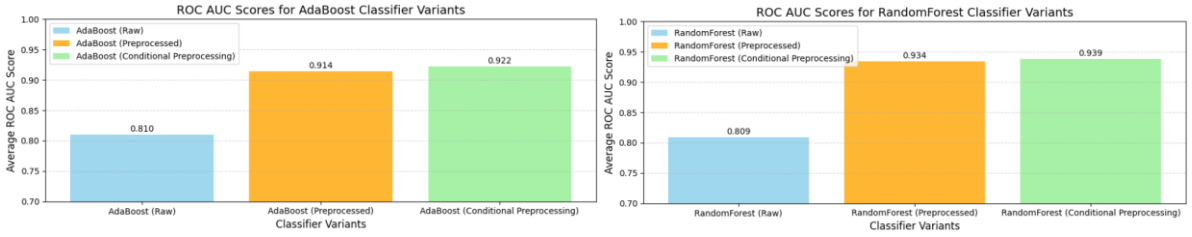
在主程式中，我們設置了一個判斷式，用於動態決定是否進行預處理：

1. **預處理前的 AUC 已達到 1**：
   直接使用原始數據進行建模，無需進行預處理，從而減少不必要的計算時間。
2. **預處理前的 AUC 不為 1**：
   進行預處理後，對比預處理前與後的 AUC，選擇較高的 AUC 作為最終結果。

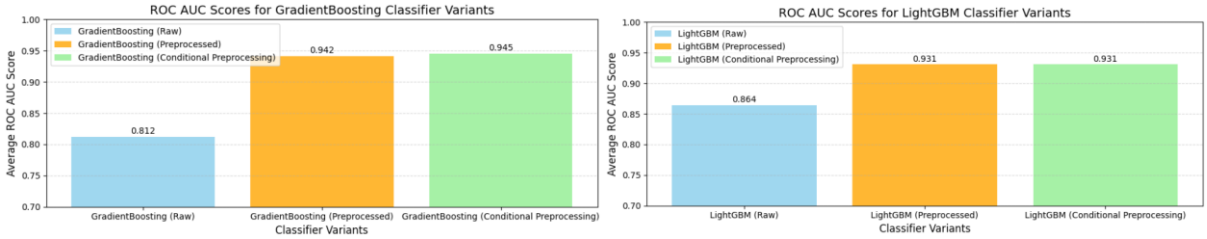
這種方法的優勢在於：

- **減少執行時間**：避免對已經表現良好的資料集進行冗餘處理。
- **提升結果質量**：動態選擇最佳結果，最大化模型性能。
- **降低過擬合風險**：對某些資料集而言，直接使用原始數據有助於保留更多特徵信息，避免過度清理引發的過擬合問題。

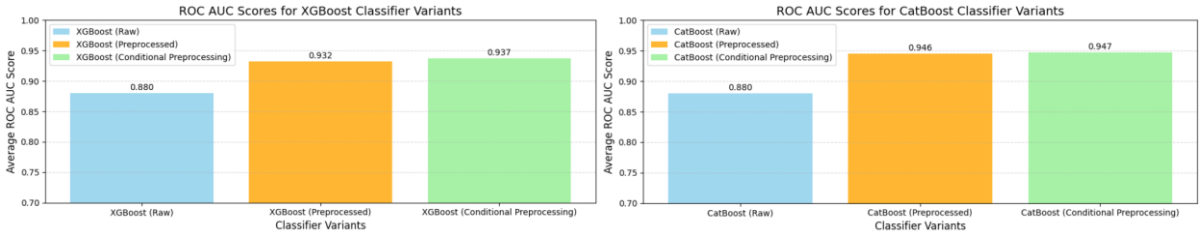
`AdaBoost / RandomForest Classfier:`



`GradientBoosting / LightGBM Classifier:`



`XGBoost / CatBoost Classifier:`



經過上傳各個模型經過 `Conditional Preprocessing` 的結果，根據 `Public Leaderboard` 的 `AUC` 表現，最終確定 `GradientBoostingClassifier`、`CatBoostClassifier` 和 `XGBoostClassifier` 的結果最佳，因此後續僅針對這三個模型進行更深入的優化與調整。

在模型調整階段，曾嘗試整合 `CatBoost`、`XGBoost` 和 `LightGBM`，讓 `Optuna` 能同時在多個模型間進行選擇，期望通過增加靈活性來提升優化效果。然而，在實驗過程中發現以下問題：

- **版本衝突**：不同套件之間可能存在不兼容情況，導致部分試驗執行時間異常延長。
- **效率低下**：單次試驗執行時間超過 10 分鐘，試驗總數僅 20 次的情況下，運

行超過 10 小時仍未完成所有試驗。

基於上述限制，為了提高效率，最終選擇分別針對各模型進行單獨調整，以避免資源浪費並確保試驗結果的穩定性。

六、超參數的更深入調整以及優化/測試判斷式

在進行超參數的更深入調整時，主要採用 Grid Search 和 Optuna 進行實驗與優化。然而，實驗過程中發現：

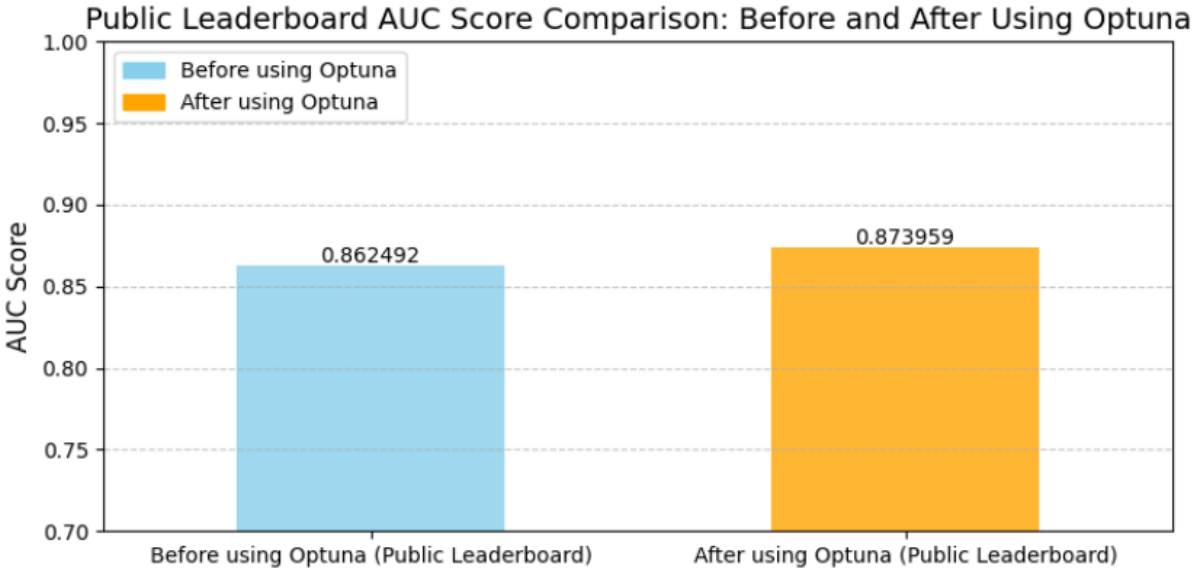
1. Grid Search 的限制：

(1) **超參數設置方式**：Grid Search 需要明確設置每個超參數的固定數值組合，而不能設置參數範圍進行自動搜索。這導致在需要嘗試多種參數組合時，Grid Search 的靈活性較差。
(2) **執行速度**：Grid Search 的運算效率較低，對於高維參數空間的搜索非常耗時。即使是較少的參數組合，執行速度也遠不及 Optuna。

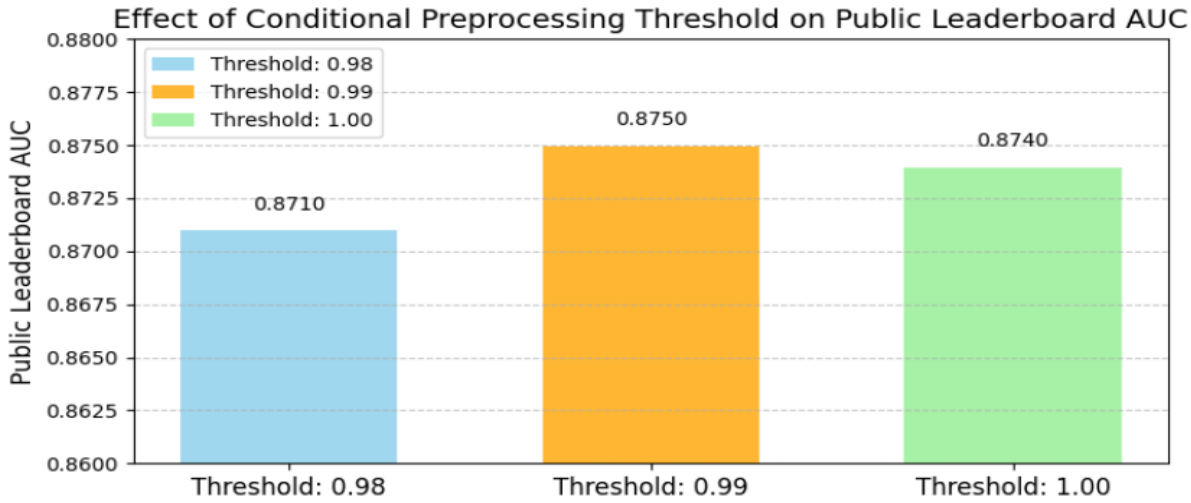
2.Optuna 的優勢：

(1) **靈活性**：Optuna 支援範圍搜索，並能根據試驗結果動態調整搜索方向，顯著提升了搜索效率。
(2) **執行效率**：在多次對比中，Optuna 的執行速度明顯優於 Grid Search，能在短時間內完成更多參數組合的測試。

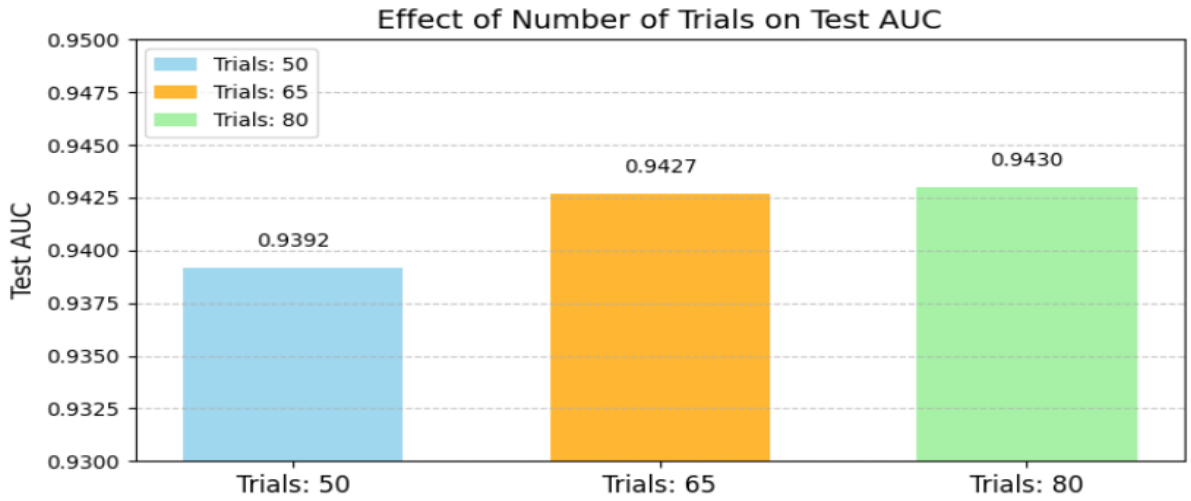
基於上述原因，最終選擇完全使用 Optuna 作為超參數調整工具，以提升優化效率和結果的質量。



- 為了在降低過擬合風險的同時進一步提升 Public Leaderboard 的 AUC，我們決定嘗試調整 Conditional Preprocessing 的閾值，通過實驗不同的閾值設定來優化結果。



- 結果顯示，當 Conditional Preprocessing 的閾值設定為 0.99 時，在 Public Leaderboard 上的 AUC 表現最佳。此外，與 1.00 相比，0.99 更能有效降低過擬合風險，可謂一舉兩得。因此，我們最終決定在剩餘時間內以 0.99 作為主要測試的閾值進行實驗
- 此外，我們也測試了 number of trials 對模型性能的影響。如理論所示，number of trials 越高，超參數調整應該越精準，預期會促使 Test AUC 和 Public Leaderboard AUC 一同提升。然而，實驗結果卻出現了反直覺的情況——Test AUC 雖有所升高，但 Public Leaderboard AUC 卻呈下降趨勢，這可能是因為超參數調整過度導致模型出現了 overfitting 的徵兆。
- 需要特別說明的是，以下圖片僅顯示了 Test AUC 的結果，由於當時測試時未記錄對應的 Public Leaderboard AUC，目前無法回溯這些數據。當時的測試目的僅為初步驗證，並未充分考量記錄的重要性，因此未能完整保存這部分資料。



最後衝刺階段策略

目標：

- 提升 `Public Leaderboard` 的 `AUC`。
- 確保不增加模型過度擬合（overfitting）的風險。

策略：

1. **專注於最終結果使用預處理前的資料集**：

   原因：

   - 每個資料集都獨立比較過預處理前與預處理後的結果，最終選擇預處理前結果的資料集，表明這些情況下預處理後的效果較差。
   - 專注於預處理前的結果進行人工超參數調整，可以在不增加過擬合風險的前提下進一步提升 `AUC`。

2. **避免處理最終結果選擇預處理後結果的資料集**：

   原因：

   - 這些資料集在現階段的表現已經最佳，進一步調整可能會增加過度擬合的風險。
   - 保持現有的狀態，可以確保結果的穩定性與可靠性。

3. **設定 `number of trials` 為 80，並執行程式碼**：

   原因：

   - **提升超參數搜索效率**：將 `number of trials` 設置為 80，有助於增加搜索空間的覆蓋範圍，從而提高找到更優超參數組合的可能性。
   - **合理利用時間**：每次執行程式需耗時約 40 分鐘至 1 小時，考慮剩餘時間有限，選擇較高的 `number of trials`（80）能更有效

率地提升結果的性價比。

操作：

每次執行的結果上傳至 Public Leaderboard 並記錄 AUC，以此進行人工比較，操作步驟如下：

標記資料集類型：

- **紅色標記**：最終結果選擇使用 **預處理前結果** 的資料集。
- **綠色標記**：人工選擇 AUC 表現較高的超參數（超參數事先已保存）。

選擇策略：

- **優先選擇 Public Leaderboard AUC 較高的執行結果**，並進一步比較內部資料集的 AUC 表現。
- 並非單純選擇 AUC 最高的結果，還需考慮該執行整體的 Public Leaderboard AUC 表現，確保最終選擇更穩定的結果。

處理其餘資料集：

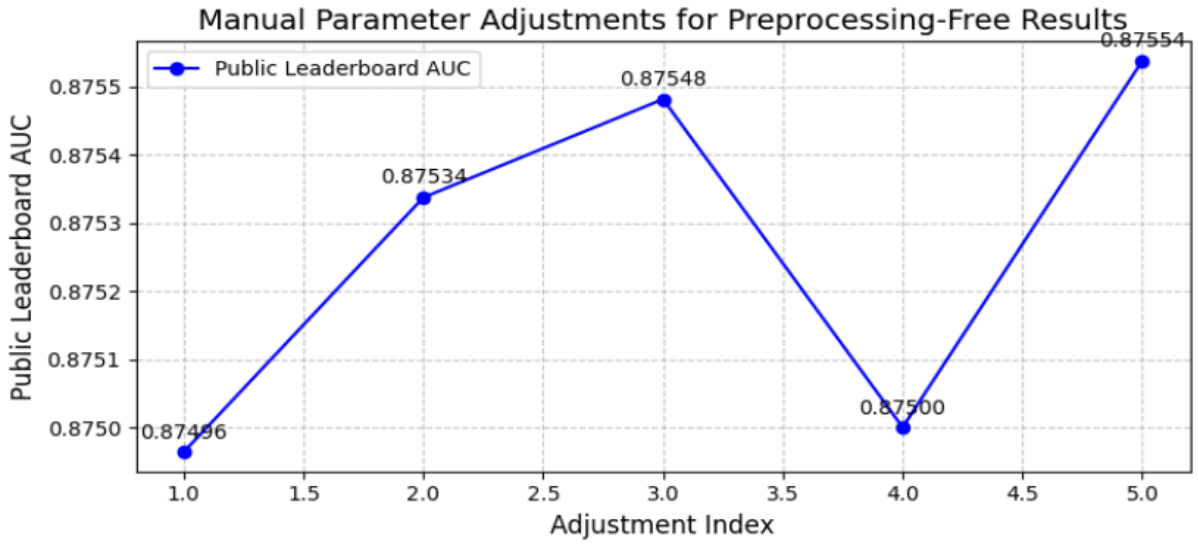
- 未被紅色標記的資料集，統一採用 0.874964 的 Optuna 結果作為基礎超參數。

最終調整：

- 根據選擇結果，對每個資料集進行客製化的超參數調整。
- 雖然是客製化，但實際上僅基於 0.874964 的超參數組合進行小幅度更動：
    - 這樣的微調確保改動幅度不會過大，以降低增加過擬合風險的可能性。
    - 在提升性能的同時，保持模型的穩定性。

AUC		0.868684	0.874964	0.869401	0.873279	0.871689	0.868996
	1	0.98	0.99	0.99	0.99	0.99	0.99
	1	2	3	4	5	6	7
1	9.0689E+14	9.1879E+15	9.2677E+15	9.2079E+15	9.2471E+15	9.13E+14	9.2497E+15
2	7.903E+15	8.0474E+15	7.96812E+15	7.9514E+15	7.9856E+15	7.9665E+15	8.0416E+15
3	8.8643E+15	9.4183E+15	9.33518E+15	9.3075E+15	9.5291E+15	8.6427E+15	9.1136E+15
4	9.9107E+15	9.9702E+15	9.97024E+15	9.9256E+15	9.9702E+15	9.9554E+15	9.9702E+15
5	9.6932E+15	9.6742E+15	9.75758E+15	9.7424E+15	9.7197E+15	9.7083E+15	9.6553E+15
6	1	1	1	1	1	1	1
7	7.0983E+15	7.147E+15	7.16087E+15	7.1102E+15	7.1688E+15	7.1748E+15	7.1668E+15
8	1	1	9.99877E+15	1	1	1	1
9	9.8236E+15	9.8236E+15	9.82363E+15	9.8236E+15	9.8236E+15	9.8236E+15	9.8236E+15
10	1	1	1	1	1	1	1
11	9.9482E+15	9.9362E+15	9.93616E+15	9.9293E+15	9.9384E+15	9.9362E+15	9.9339E+15
12	9.9963E+15	1	1	1	1	9.9975E+15	1
13	9.7121E+15	9.6439E+15	9.75758E+15	9.7879E+15	9.7576E+15	9.7197E+15	9.7197E+15
14	9.8836E+15	9.8979E+15	9.89175E+15	9.9224E+15	9.9081E+15	9.8897E+15	9.9122E+15
15	9.2181E+15	9.305E+15	9.32432E+15	9.2361E+15	9.2882E+15	9.1988E+15	9.2708E+15
16	9.6783E+15	9.6505E+15	9.6465E+14	9.6465E+15	9.5763E+15	9.673E+15	9.5829E+15
17	7.4415E+15	7.3973E+15	7.37616E+15	7.3698E+15	7.4847E+15	7.4137E+15	7.3851E+15
18	1	1	1	1	1	1	1
19	9.6604E+15	9.5839E+15	9.61741E+14	9.6891E+15	9.6126E+15	9.6174E+14	9.5887E+15
20	1	1	1	1	1	1	1
21	8.8576E+15	8.8541E+15	8.93962E+15	8.9542E+15	8.8604E+14	8.9883E+15	8.8437E+15
22	9.7236E+15	9.7475E+15	9.7395E+15	9.7554E+15	9.7554E+15	9.7315E+15	9.7342E+15
23	8.9286E+15	9.1327E+15	8.97959E+15	9.0816E+15	9.0816E+15	9.0306E+14	8.9796E+15
24	7.9689E+15	8.6695E+14	8.18802E+15	8.1576E+15	8.6332E+15	8.6033E+15	8.2418E+15
25	9.3007E+15	9.0385E+15	9.03846E+15	9.528E+15	9.3182E+15	9.528E+15	9.528E+15

以下是人工調整/客制後的 Public Leaderboard AUC 的變化圖



七、上傳結果分析

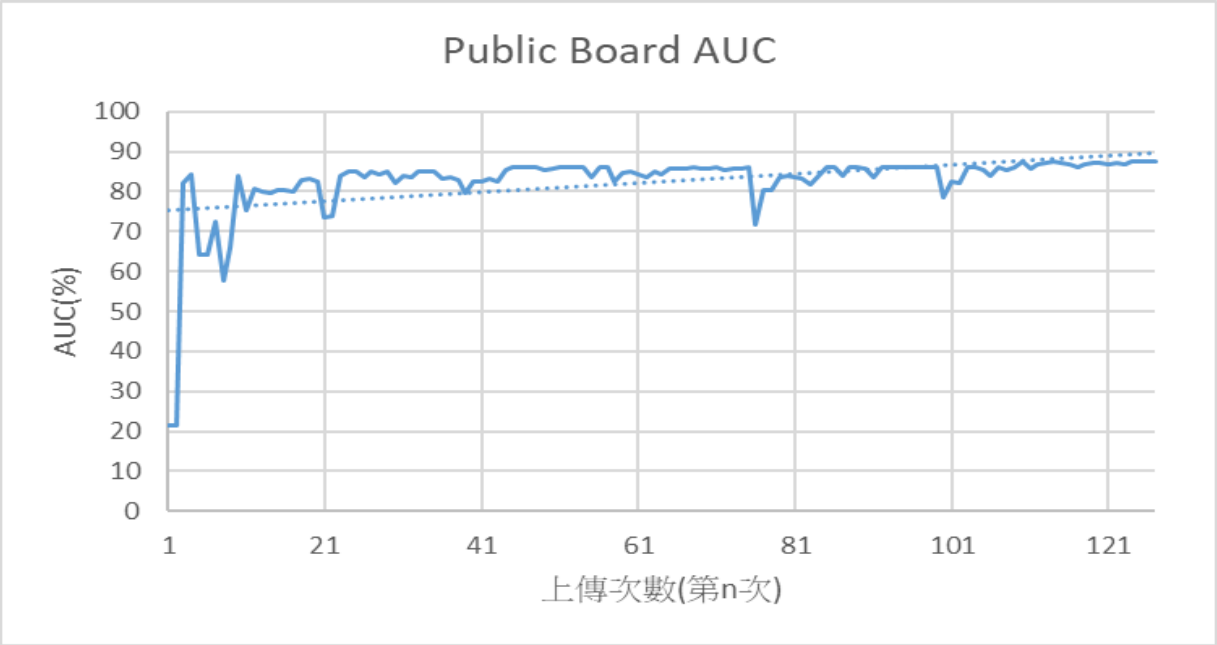
1. 最後上傳的模型

模型： **GradientBoostingClassifier**

超參數：根據不同資料集，找到對應的最佳超參數。(下圖僅列出預測其中一部分的資料集所採用的超參數)

```
# 資料集對應的最佳參數
dataset_best_params = [
    {'learning_rate': 0.010077337485660235, 'n_estimators': 593, 'subsample': 0.6211723814759181, 'max_depth': 4, 'min_samples_leaf': 1, 'max_features': None},
    {'learning_rate': 0.02142631575342677, 'n_estimators': 519, 'subsample': 0.7883256609731156, 'max_depth': 9, 'min_samples_leaf': 1, 'max_features': None},
    {'learning_rate': 0.04348841200617906, 'n_estimators': 520, 'subsample': 0.6465482025888715, 'max_depth': 5, 'min_samples_leaf': 2, 'max_features': 'sqrt'},
    {'learning_rate': 0.01138098873554563, 'n_estimators': 431, 'subsample': 0.7470197815550073, 'max_depth': 8, 'min_samples_leaf': 2, 'max_features': None},
    {'learning_rate': 0.058105178960300145, 'n_estimators': 600, 'subsample': 0.6624082184492188, 'max_depth': 8, 'min_samples_leaf': 9, 'max_features': 'log2'},
    {'learning_rate': 0.03790156439639546, 'n_estimators': 492, 'subsample': 0.7528481892490324, 'max_depth': 2, 'min_samples_leaf': 8, 'max_features': 'sqrt'},
    {'learning_rate': 0.09330345811448283, 'n_estimators': 435, 'subsample': 0.6761660507326365, 'max_depth': 1, 'min_samples_leaf': 10, 'max_features': 'sqrt'},
    {'learning_rate': 0.02354299305473999, 'n_estimators': 508, 'subsample': 0.6163254576756O1, 'max_depth': 3, 'min_samples_leaf': 7, 'max_features': None},
    {'learning_rate': 0.010156512732095198, 'n_estimators': 456, 'subsample': 0.6377256504554993, 'max_depth': 6, 'min_samples_leaf': 9, 'max_features': 'sqrt'},
    {'learning_rate': 0.03167039640943675, 'n_estimators': 498, 'subsample': 0.7856305137888209, 'max_depth': 7, 'min_samples_leaf': 5, 'max_features': None},
    {'learning_rate': 0.0230510568137622A, 'n_estimators': 542, 'subsample': 0.6286213882002543, 'max_depth': 1, 'min_samples_leaf': 9, 'max_features': 'sqrt'},
```

2. 歷史上傳 AUC 記錄



上圖為我們隨著時間，一次一次將預測結果上傳至競賽平台的分數記錄。

隨著比賽的進行，本組多次上傳預測結果以優化模型性能。從上傳的歷史記錄趨勢線可以看出，通過不斷調整資料預處理方式、優化模型與訓練方式，Public Board 上的 AUC 分數呈現穩步上升趨勢。值得注意的是，本組最後一次上傳的分數也是歷史最高分。以上反映了我們對資料處理、模型訓練調整的有效性。

3. Public & Private Board 情形

CP1 Public & Private Board 上傳比較		
	Public Board	Private Board
本組最後上傳名次	10 (前 15%)	18 (前 26%)
① 本組最後上傳 AUC	0.875537	0.854843
② 排行榜最高 AUC	0.890892	0.864254

本組與第一名的差距 (即②- ①)	0.015355	0.009411

CP1 全班整體上傳結果(共 68 組)		
AUC	Public Board	Private Board
平均值	0.824949	0.803903
標準差	0.149803	0.144396

儘管 Public Board 的預測準確率較為理想，在將近 70 組中獲得第 10 名，但在 Private Board 上本組的排名下降了近 10 名，說明模型在未公開測試數據上的泛化能力仍有提升空間。不過，相較於 Public Board，Private Board 上本組與第一名的 AUC 差距縮小，顯示本組模型集及預測方式在未知數據上的預測精確度，以全班的分數分布上看來，仍具競爭力。

八、 預測改善方法與建議

在本次競賽結束後，整理出以下幾個可改進的方向：

1. 模型與參數記錄

在每次模型訓練與超參數調整後，需明確記錄當前觀察到的準確率變化(包含 Local 和 Public Board)。否則，即使有 Optuna 作為輔助，仍可能每次都從零開始嘗試，未能基於先前調整結果進一步縮小搜尋範圍。

2. 超參數調整策略

當前未能將 Optuna 與 GridSearch 有效整合，導致在面對大量超參數時，缺乏適合的方法來系統性尋找最佳值。需探索更全面的超參數優化策略，提升調參效率。

3. 工具使用與模型整合

目前僅使用 Optuna 和 GridSearch 進行調參，但仍有其他工具（如 Ray Tune、MLflow、PyTorch Lightning 等）值得嘗試。未來可以引入這些工具，並探索不同模型之間的整合，嘗試組合模型以提高效果。

4. 資料切割方法

當前資料切割多採用 train_test_split 的預設方式，其他方法（如 K-Fold 或 Stratified K-Fold）尚未深入嘗試。未來需針對不同資料集與切割方式進行測試，以評估其對模型表現的影響。

九、 學習收穫

本次課內比賽，對於都沒有參與過 AI 相關競賽的我們三個人來說，是一次寶貴的學習經歷，讓我們在以下方面都獲益良多：

1. 資料處理能力

在預處理階段，我們學會了如何應用技術處理異常值、調整偏態、進行數據標準化等，並結合工具（如 SMOTE）解決類別不平衡問題。同時，我們也熟悉了特徵選擇的重要性。

2. 模型調參與性能優化

比賽過程中，我們深入了解了多種模型（如 CatBoost、XGBoost）的特性，以及學會從眾多模型中，選擇出最適合預測這個題目的模型。訓練過程中，加深了我們對模型性能優化的練習，也讓我們體會到如何平衡模型的準確率與泛化能力。

3. 團隊合作與工具應用

本次團隊分工中，我們使用之前助教上課時有交教的 Git 及 GitHub 進行版本管理和任務協作，有效提升了工作效率。同時，我們學會了如何從多次上傳結果中分析問題並優化模型。

4. 理解評估指標

通過分析 Public 與 Private Board 的分數差異，我們深刻理解了模型泛化能力的重要性，這對未來參與其他競賽或真實預測問題時有很大的幫助。

十、給老師及助教的建議

1. 預測任務的細節設計

本次數據集的特徵沒有具體名稱，且預測任務缺乏額外的上下文描述（具體目的、背景等），這使得我們難以直觀分析(或初步假設)特徵的重要性。這樣的預測任務確實能加強我們對於資料集處理、分析的能力，不過若是往後課程的競賽中能增加一些預測的背景資訊，相信幫助同學更有針對性地進行資料預處理與建模，也能提高同學們對於預測任務的興趣。

2. 歷史上傳的每一次結果在 Private 上的表現，都讓同學們得知

由於競賽因素，以最後一次上傳的結果作為 Private Board 的排名，這點我們贊

同。但是希望在比賽結束後，也能提供每一次上傳結果在 Private 的表現，以利我們了解過往結果的模型泛化能力。

3. **上傳預測結果時能讓我們打上備註**

由於上傳次數非常多，若能在上傳時簡短備註使用哪個模型、相關處理，應能讓同學更方便地去分析每次上傳的結果好壞，以利用訓練方向的調整。

十一、檔案上傳說明

`analyze.ipynb`

- 這檔案裡面主要是**資料初步分析**以及**進行初步模型篩選與過濾**

`AdaBoostClassifier.ipynb / CatBoostClassifier.ipynb / GradientBoostingClassifier.ipynb / LightGBM.ipynb / RandomForestClassifier.ipynb / XGBoost.ipynb`

- 這些都是各個被初步篩選的模型，然後裡面主要是在**添加了預處理方法以及判斷式**後進行測試各個模型的準確率。

`GradientBoostingClassifier_FinalAdjustments.ipynb`

- 這檔案裡面主要是**超參數的更深入調整以及優化判斷式**的測試以及**最後衝刺階段**

`FinalModel.ipynb`

- 這檔案裡面主要是存取完整的所有步驟以及**最終使用模型**。