資料科學導論期末

專案報告

1. 比賽介紹

我們這組這次所報名的比賽是 《AI CUP 2023 玉山人工智慧公開挑戰賽 - 信用卡冒用偵測》，正如其名其提供的訓練資料是由多筆有關交易資訊的訓練集，我們的預測目標就是判斷出這筆資料是否為盜刷，是一個較為單純的二元分類問題，並且會使用 f1 score 來作為分數計算準則。

2. 資料清理

a. 非數值資料

我們對於非數值資訊並沒有做什麼過於複雜的編碼，因此使用了 scikit-learn 中的 label encoder，將類別資訊 mapping 到一個數值化資訊，以便將來進行訓練。

b. 資料移除

在這個資料集中也可以發現一個對訓練沒有幫助的 feature，就是交易序號，因為每筆的交易序號不會重複，因此將這一筆資料從訓練資料中移除。

c. 缺失值處理

我們觀察其中有一 feature 為狀態碼，他的數值非 0 即 1，並且也可以觀察到，他造成的缺失量為最大宗，因此我們做的處理就是，在這一項 feature 中有缺失值都填入 0，作為缺失填補，而其他 feature 的缺失值，我們使用 scikit-learn 的 IterativeImputer 對缺失值進行預測，以填補缺失值。

d. 處理後資料變化

圖1 是 feature etymd 在經過數據清理前後的變化，從此圖可觀察到，他在資料上的處理前後的分布不會相差太遠，而我們觀察大部分的 feature 都有這樣的特性，因此我們相信在資料清理上，有達到一定程度的準確性。

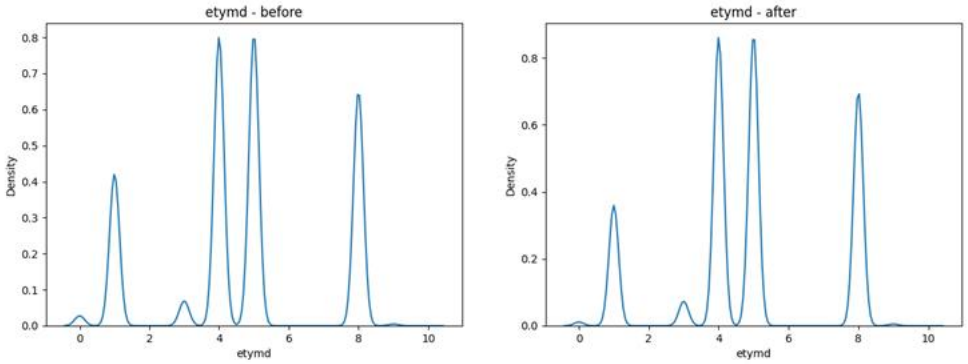

圖 1

3. 資料分布分析

a. 資料讀取優化

由於這次由主辦方所提供的訓練資料的維度不小，總共有約 870 萬筆左右的資料量，遠超我們過去所處理過的數據量，所以我們決定要先使用這資料集的部分資料來去分析資料的分佈狀態，以決定是否有需要將所有資料用來作為訓練資料。

為了表達資料分布的大方面趨勢，所以我們用 kernel density 來表示資料的分布，可以在呈現上形成較為平滑的線型。

b. Positive 和 Negative 的平衡性問題

我們發現在訓練資料的上 Positive(為盜刷的資料) 和 Negative (非盜刷的資料) 的數量相差甚遠，數量差距在 260 倍左右，這項特徵在往後我們作超參數調整時有相當影響。

由圖2 可以看出，在 label (使否盜刷) 的欄位中，屬於 0 的數量占了大部分的占比，這就是前述的資料不平衡的問題，

並且透過圖2 也可以看出在資料量不同的狀況下，依然有相似的數據分布，這也可以作為在此 feature 中，資料在訓練集中的分布位置也較為平均，可以做為不用將所有資料作為訓練資料的根據之一。

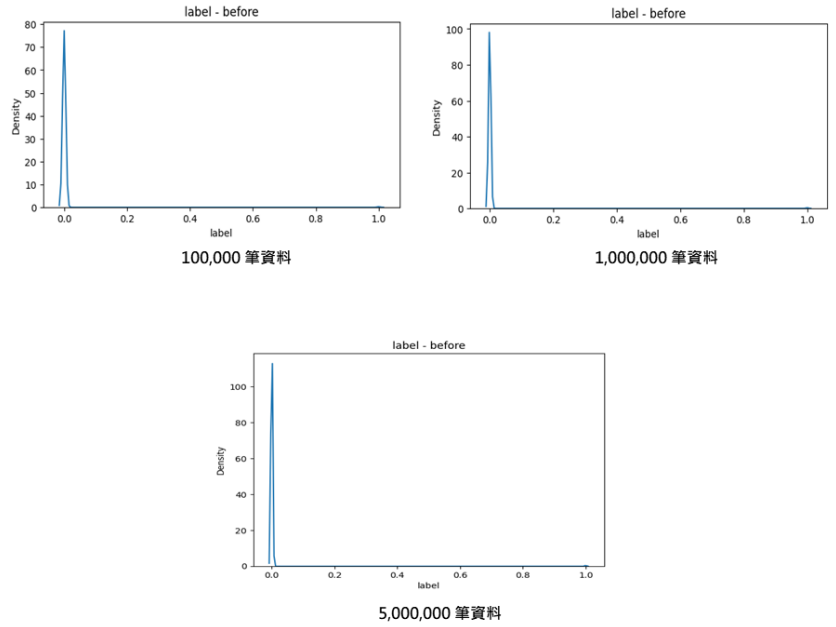

圖 2

c. 數據大小對數據分布影響

我們在觀察大多數 feature 的分布概況在各種資料量大小下的分布基本上不大方向上都有相似的趨勢，但其中可以觀察到在特定 feature 上較大的資料量會呈現較為陡峭的資料分布。

以下圖而言，圖 3表示授權時間，而圖4 表示授權日期，可以看出陡峭程度隨著資料量的增加而增強。

在這邊可以推斷的是在數據量較大的情況下，在某些 feature 上會更能顯現其可能擁有的週期性變化，進而導致某些時間點的數據量分布較少。

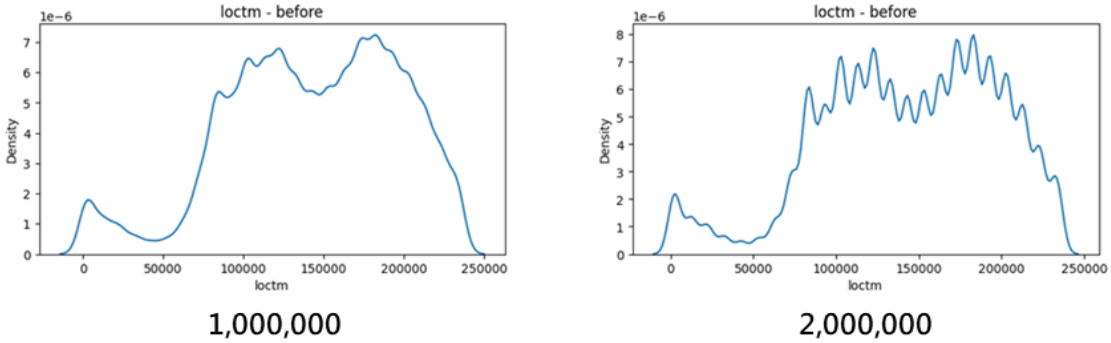

圖 3

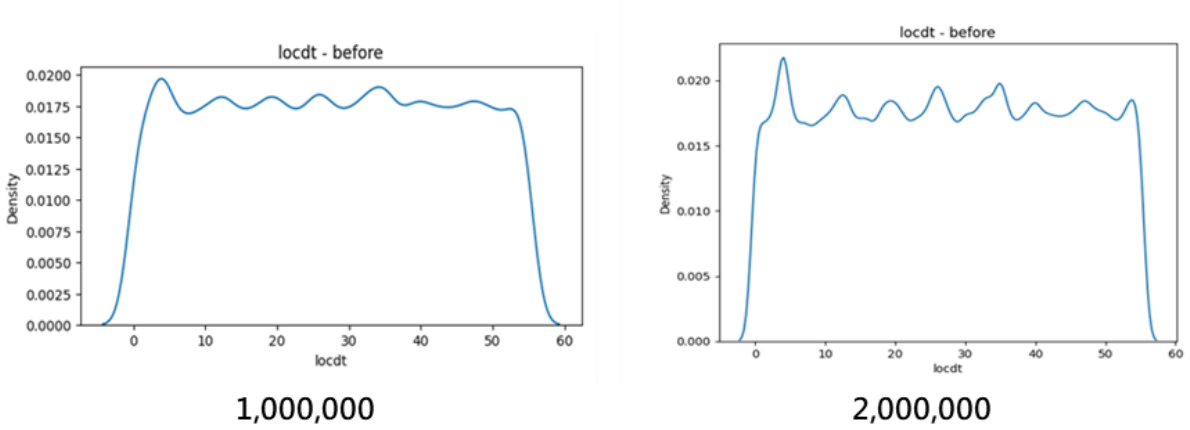

圖 4

4. 資料相關性分析

為了展現 feature 間的相關性，我們這邊使用了 Heatmap，在圖5 中，除了對角線外值得注意的是有兩 feature 的相關性也為 1，這兩個 feature 分別是的交易金額(台幣)和實付金額。

我們在想，既然兩者相關性如此強烈，那選擇其一作為訓練資料就好，然而在我們後面的分析，做這樣的 feature 篩選後，使用 f1 score 計算結果成效，會發現成效有所衰退，此特性會在後序有關 feature 篩選的節數呈述。

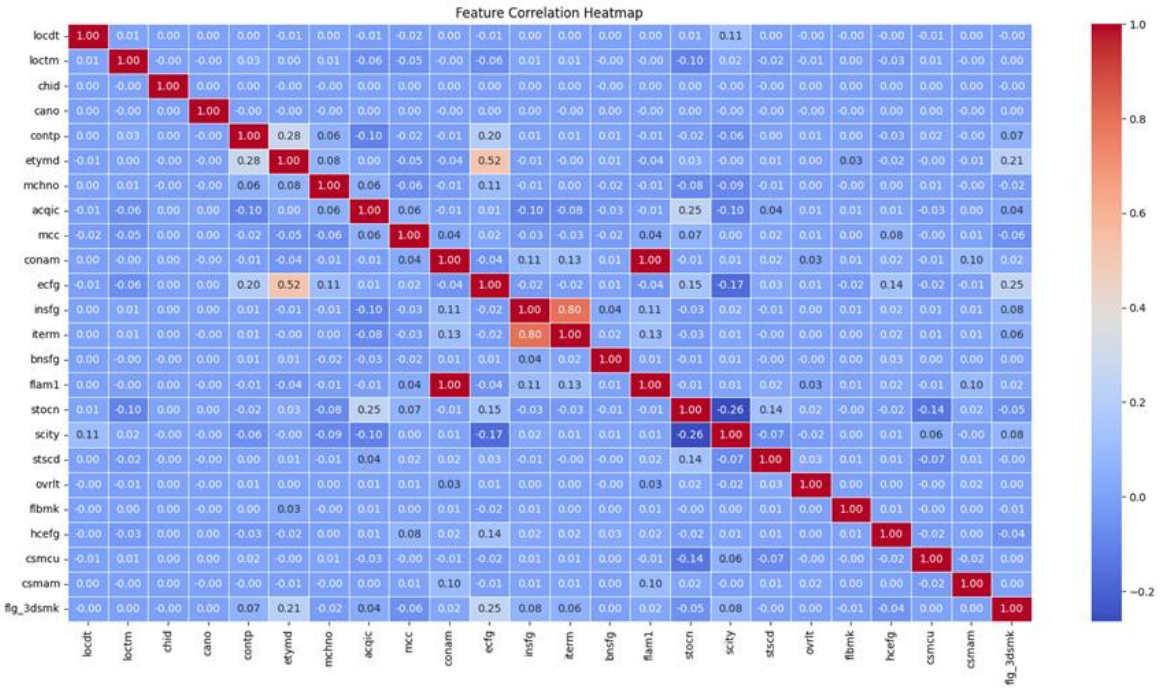

圖 5

5. 訓練

a. 模型選用

在此次 AICUP 的比賽中我們使用的是 xgboost 作為我們訓練的基底，xgboost 就基本面而言，是一個決策樹模型，而 xgboost 提供了在 loss function 或平衡數據的相關參數作為可調項，所以我們在訓練過程要做的就是提供一組足夠好的超參數讓模型去訓練。

b. 參數調整

i. RamdomizedSearchCV

為了讓參數調整更有自動性，我們使用了 scikit-learn 的RandomizedSearchCV 進行條整參數，因為在以往經驗下，若提供的參數選項列表使用太多的 random 值反而會容易找不到較好的參數組合，所以如圖6 顯示，在這邊提供的參數組合都是有固定值的選項。

首先，我們針對較小數據量(100萬筆)的資料進行參數調整，圖6 我們先給予 xgboost 調整參數的基本起始條整參數的搜索範圍。

而根據於此所跑出來的最佳參數可以從圖7 所觀測，首先可以確定的是在 learning rate 上都給了不小的數值，或許可以推斷的是訓練起始點到最佳解的過程上， loss 變化所呈現的趨勢可能較為平滑，導致需要更大的 learning rate 來加速訓練效率。

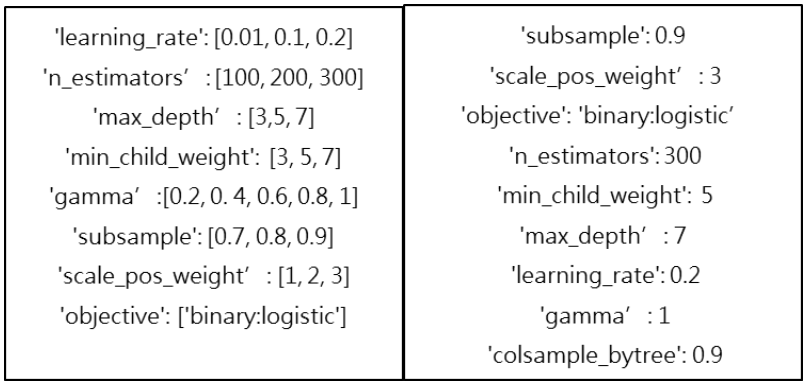

圖 6

圖7

ii. 超參數調整方向

圖8 中，我們挑選了幾個較為重點的超參數來觀察，分別是 max_depth，這邊用三種不同顏色的線來表示，並且還有 learning _rate，在這邊分為了三張表，接著是 n_estimator ，這邊作為 x 軸，最後的 y 軸由 f1 score 的均值所表示。

由此可以觀察到，的確在 learning rate 較高的情況下，模型能獲得較高分，並且隨著深度以及 n_estimator 的增加，對模型的提升有所幫助。

看到圖9 ，這裡將 y 軸改為在做 cross validation 的過程中所獲分數的標準差，這裡可以看到越往右側的圖標準差也較低。

綜上所述，當 n_estimator 的數值增加和給予其適當大的 learning_rate 和 depth 都對模型訓練有所幫助，大致上是此模型的調整參數的大方向。

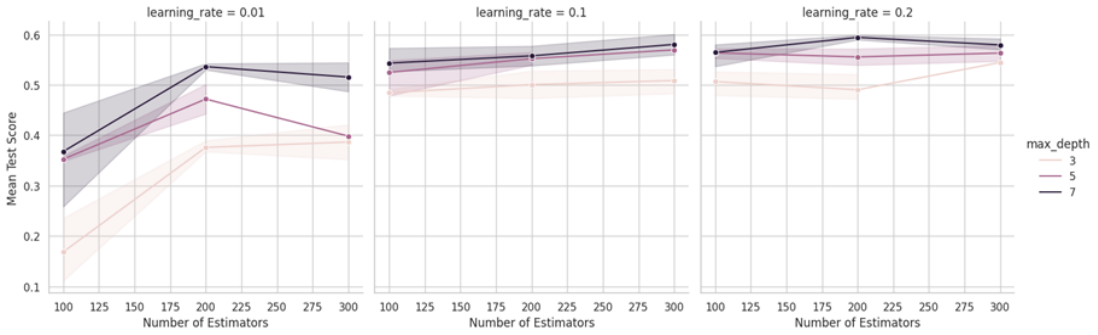

圖 8

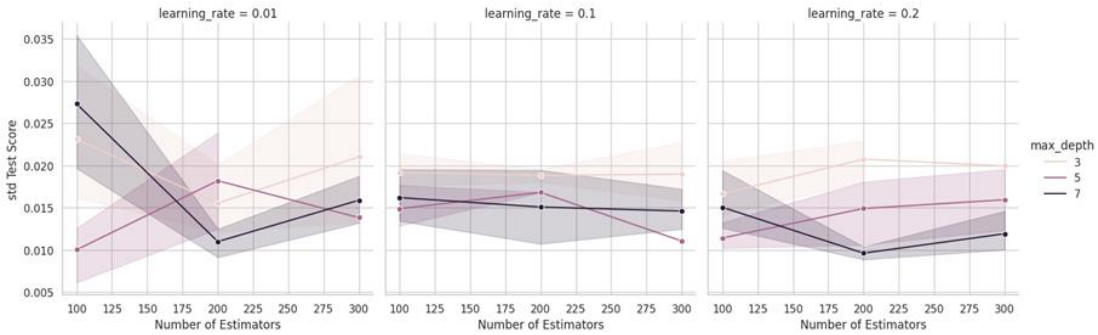

圖9

iii. 資料 Imbalance 的影響

在最一開始所提到 label 有數量不平衡的問題，而在最初步的調整參數結果出爐後，也展現出了其問題，我們一開始搜索出scale_pos_weight 的值是 3，但之後發現此參數需要隨著不平衡的狀況而倍數調整，因為此參數就是在為此訓練資料較為不平的 label 給予權重上的倍數，以達到平衡效果。

由圖10，此 Confustion Matrix 中可以觀察到，的確在此訓練資料的影響上，右上角有大量的分佈，因為此資料集會使 label 為零的狀況更容易去猜測，但看到在原始 label 為 1 的情況下，所猜測的準確度只有大概一半，明顯效果不彰。

由圖11 可以觀察，當我們直接將 scale_pos_weight 改為符合資料差距倍數左右的數值 260，便可看到在 recall 的數值有所提升，證明在預測 1 的狀況有所改善。

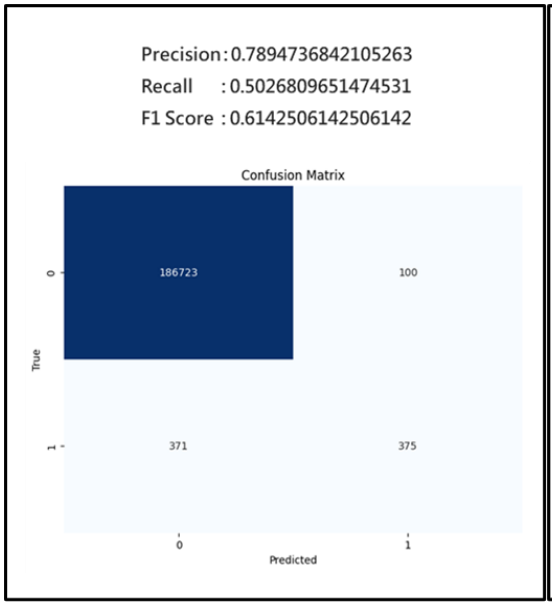

圖10

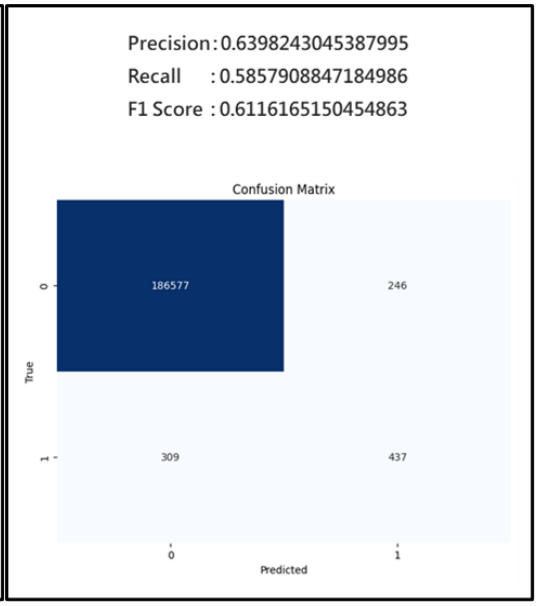

圖 11

c. 資料量大小選擇

在一開始的描述中，表示由於資料量的大小對於資料的分布狀況影響不大，因此判斷可以不用將所有資料讀入進行訓練，但如圖12 (訓練數量為 100 萬) 和圖13 (訓練數量為 200萬)，可看到在 Validation Set 的進步，在資料量大的部分獲得比較好的訓練水準。

因此可推斷，在一定的資料量級下，訓練資料的增加還是對模型在訓練的過程有所幫助。

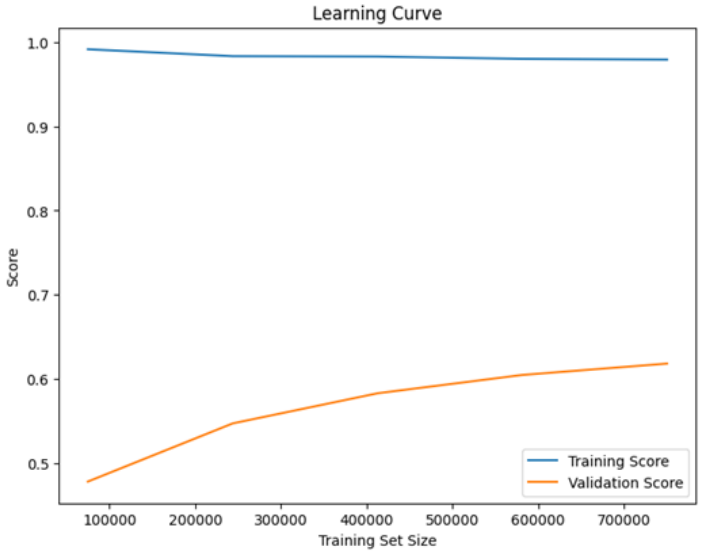

圖 12

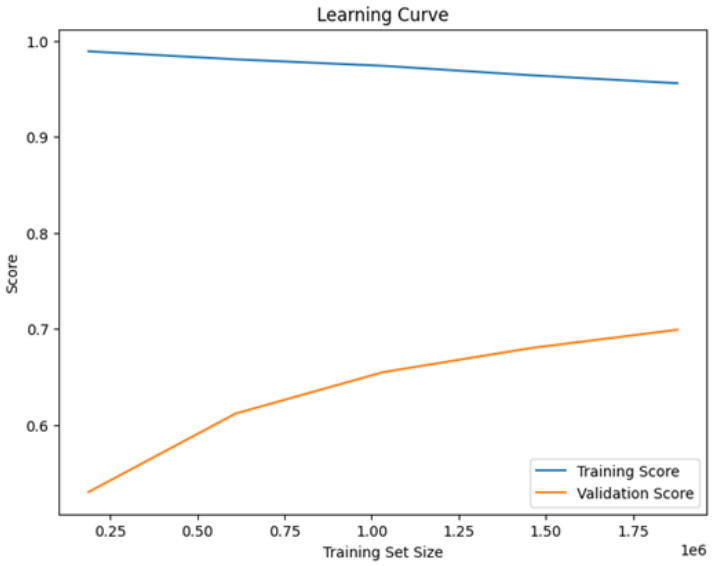

圖 13

d. Feature 的選擇

首先，針對不同量級的資料量，我們選擇了全部的 Feature 來做訓練，如圖14 (訓練數量 100萬) 和圖15 (訓練數量 200萬)，可以看出在 Feature 對模型的重要性，兩者幾乎是沒甚麼差距的，再次驗證資料分布相當勻稱。

因此，接下來對於 feature 篩選結果都透過數量級為100 萬的作為主要分析對象。

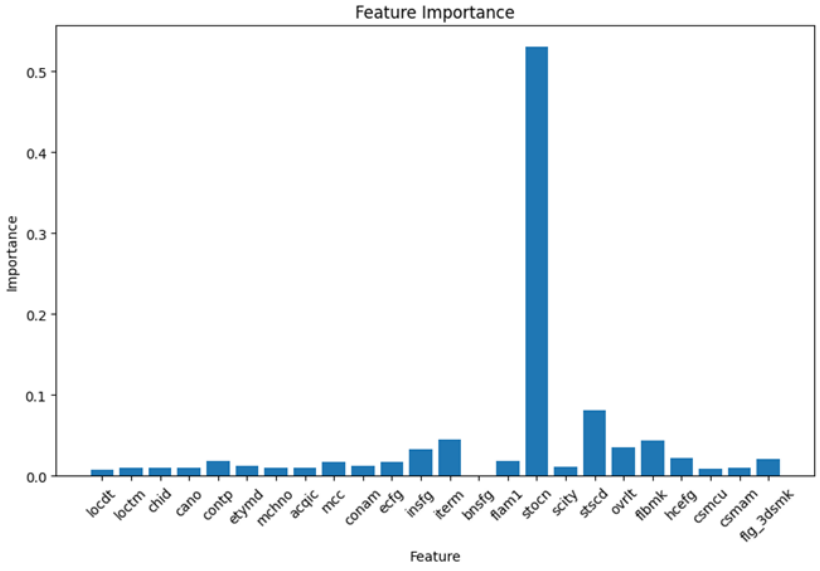

圖 14

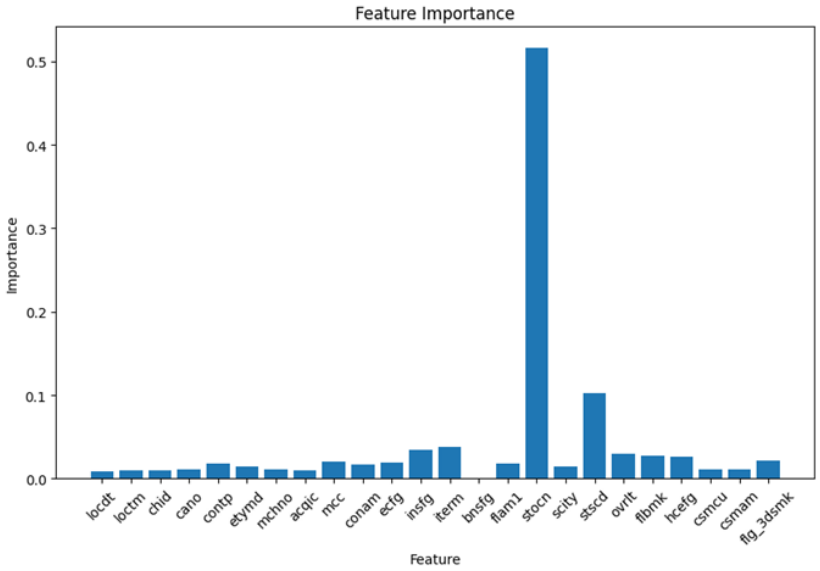

圖 15

由上述的圖15 可以觀察出，除了重要性高的很突出的 stocn 這項 feature 外，大部分的重要性都不甚相去太遠，因此透過去試驗小範圍改變的 threshold 作為判斷基準是比較適合的。

圖16 和圖17 分別展現了全部 feature 以及 threshold = 0.011 作為篩選機制的訓練節果，feature 量從 24 降到 16，圖 17 雖展現出來的 True Positive 比較高，但是可以看到在 False Negative 的部分大幅增加，造成 Recall 連帶下降，而使 f1 score 下跌。

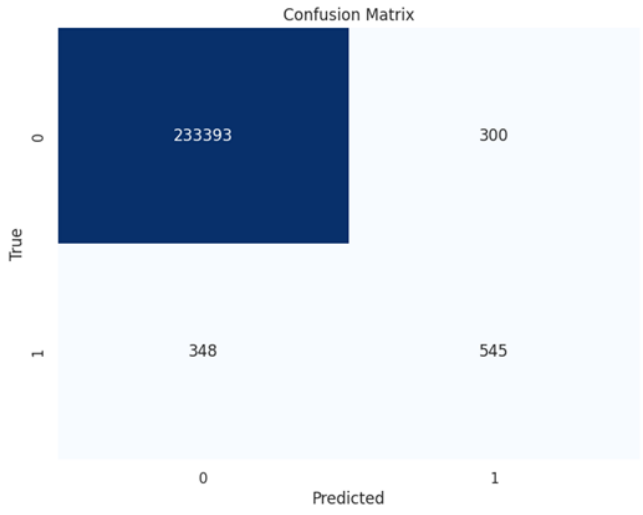

圖 16

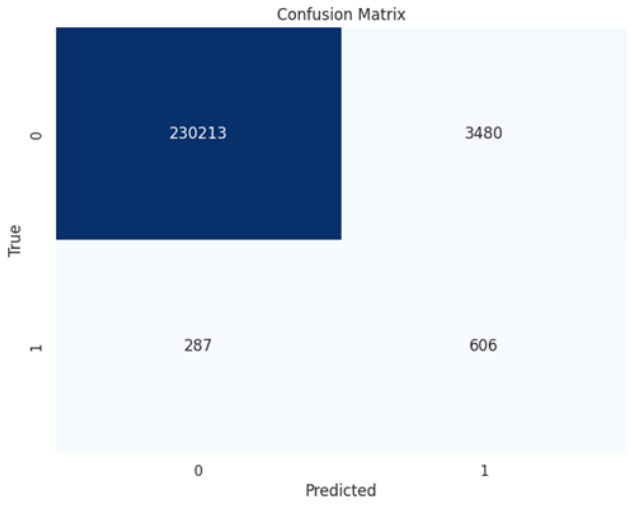

圖 17

接著嘗試把 threshold 更改為 0.01， feature 量從 24 降為 20，圖 18 可以看出 Recall 比 threshold 0.011 的時候回復了不少，但 Precision 的部分就落回原來水準，但比原來不更動 feature 數量是還要差一點，多次的調整 feature 也無法獲得更好的進步，甚至有可能導致退步。

總歸而言，我們認為 feature 的縮減同時也代表資料量的縮減，而我們認為此訓練集對數據數量的敏感度大於對於 feature 改變的敏感度。

因此，我們最終決定不去篩選 feature ，以全部 feture 都選擇的此態下去訓練。

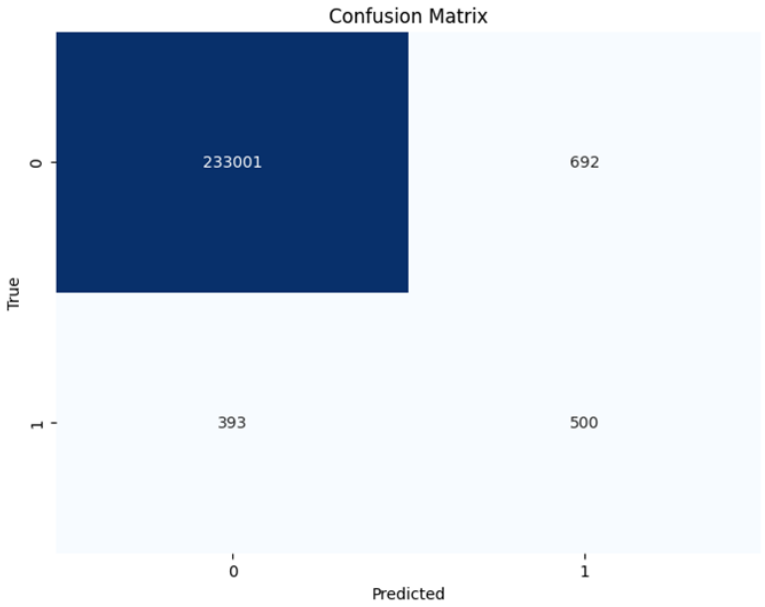

圖18

e. 模型調整總結

我們最終的模型調整是將資料量設定為 200 萬後，將 Feature 全都選為訓練資料，然後將 scale_pos_weight 設定為 260 的固定值，以這樣的狀態去讓 RamdomizeSearchCV 去找最佳參數，並且找到的這組參數最為模型的超參數固定值。

6. 競賽結果與心得

我們這組最終在 Public 的分數獲得 0.539269 的分數，在 394 組中排名 72，我們認為在最終的結果或許還有蠻大的進步空間，因為在賽後分析的過程中也發現了不少可以再運用的特徵，或許能將分數再往前帶，但最終是無法驗證，著實可惜。

總歸而言，透過這學期的上課以及作業製作，讓我們能在這次的比賽中得以運用並且驗證，而由比賽結果來看，還有需多我們要學習的，在資料分析這們學問上，還有許多需要深究的。