1. 姓名: 朱柏綸 系級: 資工110 學號: F84064014

姓名: 高造擎 系級: 工資111 學號: H34076021

1. 競賽敘述與目標

透過機器學習，實作出可以準確預測某顧客未來是否會繼續使用該銀行的模型，是一個binary classification的問題。各組會有排行榜的public排名作為依據，實際排名則是由private決定。

1. 資料前處理

高造擎:

刪除RowNumber和CustomerId明顯對預測沒有幫助的欄位後，試圖使用SelectKBest和ExtraTreeClassifier尋找最有助於分類的欄位，但放入各式模型結果皆不甚理想。因此嘗試其他方法，於使用RandomForest時，把所有欄位取組合放入模型，看是否能找到達到最高數值的欄位組合，並應用於其他模型，結果也是差強人意，所以最後決定維持原本欄位，只刪除RowNumber和CustomerId。

朱柏綸:

* 連續型資料: 嘗試做normalize
* 類別型資料: label encoding / target encoding
* 特徵篩選: Drop customerID, Surname, RowNumber
* Outlier處理: 使用IForest決定outlier並移除
* 資料分割: 使用stratified的方式，訓練測試集皆維持原本的比例(8:2)
* 資料取樣: 對標記少的那類(佔20%)做oversampling，在分割訓練 測試資料集後，隨機複製訓練集中佔少數標記的資料點，增加訓練資料的平衡性

1. 特徵處理與分析

* Oversampling:   
  未做Oversampling  
  一張含有 文字 的圖片

  自動產生的描述  
  Oversampling 2000筆後  
  一張含有 文字 的圖片

  自動產生的描述  
  Oversampling增加資料平衡度可以讓f1\_score提升，但同時precision會因此下降
* Outlier Clipping:  
  沒有去除Outlier:  
  一張含有 文字 的圖片

  自動產生的描述  
  有去除Outlier(4% of data removed):一張含有 文字 的圖片

  自動產生的描述  
  去除outlier效果可能不顯著
* Normalization  
  沒有做normalization  
  一張含有 文字 的圖片

  自動產生的描述  
  有做normalization  
  一張含有 文字 的圖片

  自動產生的描述  
  Normalization對非tree based模型較有幫助，如LogisticRegression和SVM的效果都提升了不少，LGBM和RandomForest則幾乎沒有影響

1. 預測訓練模型

高造擎:

預測訓練模型:因缺乏經驗，為尋找最適合此問題的演算法，我將上課學過的套件逐一測試，包括:MLP, GaussianNB, KNeighbor, SVC, DecisionTree, RandomForest, Linear Regression, XGB, EasyEnsemble, BalancedRandomForest，觀察accuracy, precision, f1-score的數值是否夠佳，而最後只有ensemble類型的RandomForest, XGB, EasyEnsemble, BalancedRandomForest足夠好，所以針對他們調整參數。調參主要以gridsearch設定max-depth為10, n\_estimators=250, class\_weight=’balanced subsample’，以解決Exited比例不均的問題。

朱柏綸:

* 最終模型  
  一張含有 文字 的圖片

  自動產生的描述  
  主要使用Xgboost、LGBM和RandomForest組合效果最好。LGBM可以使用class\_weight，在不平衡資料中給予class\_weight可以讓判斷準確度提升，使用較多的estimators和較大的max\_bin會提升模型準確度，小的learning\_rate在足夠的iteration也應該可以得到更好的結果。最後使用voting classifier與權重做組合分類器，使用機率總和的soft voting。

1. 預測結果分析
   * + 不同模型原始效能  
       LBGM和RandomForest表現明顯大幅超越其他模型
     + LGBM的class\_weight調整:一張含有 文字 的圖片

       自動產生的描述  
       越大的class\_weight可以提升precision，但相反的f1 score會跟著下降，較接近分群比例的class\_weight
     + LGBM learning rate調整 [0.2, 0.15, 0.1, 0.05, 0.03, 0.02]  
       一張含有 文字 的圖片

       自動產生的描述  
       越小的learning rate也會有越高的precision與越低的f1 score
     + LBGM max\_bin調整  
       一張含有 文字 的圖片

       自動產生的描述  
       結果差異不大
     + LBGM min\_gain\_to\_split調整  
       一張含有 文字 的圖片

       自動產生的描述  
       透過控制min\_gain\_to\_split可以控制得到更高的precision
     + LGBM n\_estimator調整  
       一張含有 文字 的圖片

       自動產生的描述  
       estimator的數量多precision會下降f1 score會上升，f1 score的上升趨緩的很快速，但precision則是繼續大幅下降，太多estimators對預測未必是好事
     + 調整模型  
       一張含有 文字 的圖片

       自動產生的描述  
       調整參數能小幅增進效果
     + Ensemble多個classifier  
       一張含有 文字 的圖片

       自動產生的描述  
       Ensemble後的模型可以在precision和f1score之間做平衡並達到整體上更好的效果，也可以透過調整各個模型的權重來控制precision與f1score
     + 透過oversampling調整訓練結果  
       一張含有 文字 的圖片

       自動產生的描述  
       也可以透過在訓練集oversampling犧牲一點precision拿到更高的f1 score
2. 感想與心得

高造擎:

我從競賽學習到，上課內容與實際解決問題的差距真的很大，上課藉由二維圖形，可以很容易知道甚麼模型適用於問題，但實際問題的欄位通常都會多到無法馬上知道，所以只能透過觀察和反覆嘗試才能有所進展。我一開始就花費太多時間，盲目的測試我知道的所有模型，而沒有發現Exited欄位的比例差很多。另外，我花費最多時間在研究演算法的參數，很多參數都不知道其意義，即使知道也不清楚設定多少可以達到最佳效果，所以只能一直排列組合，看能不能有所進步。總結來說，我覺得機器學習真的好玄，一直不斷的嘗試，但結果都不好，即使有變化，也不太清楚原因。我個人覺得，如何從資料處理到最後模型的產出，應該要有系統化的處理方式，這也是我未來需要學習的方向。

朱柏綸:

從競賽中自學到在面對不平衡資料時會遇到的一些問題與處理方法，還有自己寫一些class和function來讓各個不同模型參數間的效果關係能更快更簡潔的看出來。花最多時間的部分在於模型參數的調整，最後當結果都差不多的時候靠著些微的參數調整調出擠牙膏般的績效差異時在花了不少時間，最困難的地方可能來自資料的不平衡，還有X跟y的相關性不夠高可能也是讓問題有一些困難。我覺得競賽的網站可以不要每次載入畫面的時候都自動下滑到自己排名所在的那個位置，因為排名太下面每次都要看他在那邊下滑有一點久，可以就停在最上面讓我自己滑下去看我的排名有多低就可以了。

Hw5 git hub

F84064014 朱柏綸

Repo: <https://github.com/F84064014/DataScienceHw5>

Page: https://f84064014.gitrfgttt hub.io/DataScienceHw5/hw3/main.html

H34076021 高造擎

Repo: <https://github.com/Rob12312368/H34076021_DataScience>

Page: <https://rob12312368.github.io/H34076021_DataScience/HW3/%E7%B6%B2%E9%A0%81/Tsao-Ching.html>