金融科技 第三組 Final Project 成果報告

組員：林柏辰 110061530、章家綾 110061610、王靖淳 111061585、
李家妤 111061613、李弈萱 111061641

I. Abstract

我們參加了玉山銀行底下 T-Brain 平台所舉辦「你說可疑不可疑？－疑似洗錢交易預測」比賽，我們將資料集做預處理，並且依據洗錢定義設計適合的 feature，使用 SVM 與 XGB，並用 voting 的方式來相互結合兩個模型的優點，以達到預測洗錢交易的目的。

II. Introduction

a. 動機

在寫前幾次作業的過程中，利用 yahoo 股票紀錄實做了許多在股票市場中常見的指標，在了解這些指標的同時，也讓我們更理解股市交易的專有名詞和金融科技與人工智慧之間的關係，然而金融界很廣，我們也想了解其他面向的金融數據該如何操作，因此參加玉山銀行底下 T-Brain 平台所舉辦「你說可疑不可疑？－疑似洗錢交易預測」比賽，並且仔細研究後也發現這個比賽的資料庫整合麻煩、在前處理與特徵提取的部分非常的艱鉅，加上洗錢對我們來說是比較陌生的，我們需要了解洗錢的定義，以及數據上有關金錢的流動的所有相關紀錄代表的意義，同時，因為各家銀行交易的資料通常都是機密，除非有接到產學合作，或者是未來進入金融領域工作，否則皆沒有機會拿到這些資料庫並進行進一步的使用，所以這個比賽機會對我們來說很難得、有挑戰性且可以在過程中學到許多額外的知識，使我們的鬥志被熊熊燃起，期待可以迎面克服這比賽的挑戰。

b. 背景

洗錢指的是以非法手段獲取金錢後變成合法資金的過程，而反洗錢則是一連串防範洗錢的措施。在國際上的經驗洗錢和反洗錢的主要活動集中在金融領域，因此幾乎所有國家與跨國合作都針對金融機構進行反洗錢規範。

因為對反洗錢的重視，所以台灣也持續在反洗錢上訂定政策。除了政府訂定政策之外，目前最常見的技術提升是將 AI 應用在洗錢防制，以自動化機器人協助偵測可疑交易，主要在洗錢防制上的應用有以下幾點。

1. 自動交易監控：AI 自動化流程可以從大量交易資料中偵測跟辨識可疑特徵，快速、大量、準確的分析資料。洗錢交易風險越大數值顯示也越大，可以讓銀行有優先確認的名單。
2. 預警偵測 : AI 同時可以定期分析新聞、社交媒體、網路等文字，在發生相關事件時，可掃描跟分析敏感文字，甚至可因此了解商品樣態，進一步調查是否涉及融資洗錢。
3. 自動生成初步報告 : 在政府的法規中要求金融機構偵測到可疑交易時，要提交可疑交易報告，否則會被裁罰。若能將 AI 結合其他技術在偵測時自動生成報告，可以提高效率。
4. 客戶分群與設定門檻 : 不同客戶有不同的交易行為，例如個人帳戶跟法人帳戶模式即是不同，若以相同門檻監控交易行為，容易產生大量誤判警示。因此應該將相似客戶分群，再設定合適門檻，此舉也符合風險基礎。

玉山銀行底下 T-Brain 平台所舉辦「你說可疑不可疑？－疑似洗錢交易預測」比賽正是反洗錢挑戰的一部份，其提供了相關交易資料，又可分為 training、public、private 三部分，等同於切分訓練集與測試集，training 是訓練集，public 則是測試集，我們在過程中是以 public 做為測試集來看結果並與他人比較排名，最後在比賽截止的前一天主辦方會公布 private 的資料以及 public 的解答，最後比賽成績是依據 private 的分數與排名來看，並且比賽結束後也仍未公布 private 的解答，所以在這份報告中我們會呈現 public 的成績並且能對照已公布的 public 資料中真正洗錢的幾筆資料找出其排名以及網站公布的 private 成績。

c. 目標

這個比賽評分依據，是從所有有洗錢嫌疑的交易中，將最有可能是真正洗錢的交易以機率由大到小做排序，假設答案是 N 筆真正洗錢的交易，那麼會以 N-1 作為分子，第 N-1 筆交易的排序名次 x 為分母，使用$\frac{N-1}{x}$*作為分數*。經過觀察數據以及其他組別的成績後，我們將目標訂在將第 N-1 筆真正洗錢的交易排在所有數據的一半以上。

d. Related Work

現在大型機構中的檢測洗錢制度是基於一定規則的系統，這種方法效率低，目前較普遍基於數據科學的反洗錢模型致力於客戶關係管理與交易行為的時間特徵，多是分為兩部分研究，一為研究模型，包括 Decision Tree、Random Forest、SVM、Naive Bayes、 Logistic Regression、Neural Network，另一部分為研究特徵工程，這裡關注的面向也很多樣，特徵本身就非常多，篩選手段各有不同，此外，可以更換抽樣技術以增加事件發生率，因為洗錢事件的發生率很低，也可以利用時頻分析，觀察可疑資料與非可疑資料的時頻特徵是否有顯著不同。

III. Preliminary

a. 辨識模型

i. SVM

SVM 用統計風險最小化的原則來估計一個分類的超平面(hyperplane)，也就是說找到一個決策邊界(decision boundary)讓各類之間的邊界(margins)最大化，使其可以完美區隔開來。這邊 SVM 是假設有一個 hyperplane($w^T x + b = 0$)可以完美分割兩組資料，所以 SVM 就是在找參數 w、b 讓兩組之間的距離最大化。

$$\max_w \{2/\|w\|\} \rightarrow \min_w \frac{1}{2} w^T w$$

$$subject\ to \quad y_i(w^T \boldsymbol{x}_i + b) \geq 1, \qquad \forall i = 1, \ldots, n$$

ii. XGboost

XGboost 全名為 eXtreme Gradient Boosting，它是以 Gradient Boosting 為基礎，添加一些新的技巧，可以說是結合 Bagging 和 Boosting 的優點。XGboost 保有 Gradient Boosting 的做法，每一棵樹是互相關聯的，目標是希望後面生成的樹能夠修正前面一棵樹犯錯的地方。此外 XGboost 是採用特徵隨機採樣的技巧，和隨機森林一樣在生成每一棵樹的時候隨機抽取特徵，因此在每棵樹的生成中並不會每一次都拿全部的特徵參與決策。在訓練時為了擬合訓練資料，會產生很多高次項的函數，但反而容易被雜訊干擾導致過度擬合，因此對目標函數做標準化並加上 L1/L2 Regularization 讓損失函數更佳平滑，且抗雜訊干擾能力更大；最後 XGboost 還用到了一階導數和二階導數來生成下一棵樹。

iii. 常見模型的差異、優勢

1. DNN
   a. 優點：生成較複雜模型、概括能力較好、可計算隱藏特徵
   b. 缺點：容易過擬合、訓練時間長、注重模型架構設計
2. Random Forest
   a. 優點：速度快且泛用性高、適合大量或高維度的資料、降低離群值影響
   b. 缺點：特徵本身須為有效預測因子、可調整參數少
3. SVM
   a. 優點：類可分離時的最佳算法、適用極端狀況下的分類任務
   b. 缺點：訓練時間長、參數調整難度較高
4. XGBoost
   a. 優點：速度快、降低離群值影響、可計算特徵重要性、不易過擬合
   b. 缺點：參數多難調整

iv. Majority vote

由於我們嘗試了多個模型，而每個模型進行預測的方式又各不相同，如果只用單一個可能產生盲點，於是我們採用多數決加權投票來結合各模型的優點，產生最終預測結果。

b. Dataset (逐筆交易資料標註為紫色)

資料集_檔案名稱	訓練資料欄位名稱(欄位說明)	備註
public_train_x_ccba_full_hashed (ccba)	cust_id (顧客編號) lupay (上月繳款總額) byymm (帳務年月) cycam (信用額度) usgam (已使用額度) clamt (本月分期預借現金金額) csamt (本月預借現金金額) inamt (本月分期消費金額) cucsm (本月消費金額) cucah (本月借現金額)	
public_train_x_cdtx0001_full_hashed (cdtx)	cust_id (顧客編號) date (消費日期) country (消費地國別) cur_type (消費地幣別) amt (交易金額-台幣)	
public_train_x_custinfo_full_hashed (custinfo)	cust_id (顧客編號) alert_key (alert 主鍵) risk_rank (風險等級) occupation_code (職業) total_asset (行內總資產) AGE (年齡)	* 同顧客可能有數筆 alert_key * 無職業者 (occupation_code) 歸類在 21

public_train_x_dp_full_hashed (dp)	cust_id (顧客編號) debit_credit (借貸別) tx_date (交易日期) tx_time (交易時間) tx_type (交易類別) tx_amt (交易金額) exchg_rate (匯率) info_asset_code (資訊資產代號) fiscTxId (交易代碼) txbranch (分行代碼) cross_bank (是否為跨行交易) ATM (是否為實體 ATM 交易)	* debit_credit (借貸別) 原始資料為 CR、DB、Nan，CR 轉換為 0，DB 轉換為 1，Nan 轉換為 2 * tx_amt 資料中有"負值"，代表為沖正交易，包括匯款轉帳、取現等交易失敗，但是銀行端仍扣錢時，銀行系統會採取的一種補正程序。 * tx_amt 單位為原始幣值，需乘上 exchg_rate 後才是台幣 * txbranch 空值得部分代表非跨行交易，有值代表交易對手銀行代碼
public_train_x_remit1_full_hashed (remit)	cust_id (顧客編號) trans_date (外匯交易日(帳務日)) trans_no (交易編號) trade_amount_usd(交易金額(折合美金))	
public_x_alert_date	alert_key (alert 主鍵) date (alert 主鍵發生日期)	
train_x_alert_date	alert_key (alert 主鍵) date (alert 主鍵發生日期)	
train_y_answer	alert_key (alert 主鍵) sar_flag (alert 主鍵報 SAR 與否)	

c. 評分標準

依照官方的評分標準是以 Recall@N-1 的 Precision 進行評分，意即在抓到 N-1 個真正報 SAR 案件名單下的 Precision（將四捨五入計算至小數點後 7 位），公式如下：

$$Recall@N-1\ 的Precision = \frac{N-1}{第N-1\ 筆真正報SAR的預測機率排名}$$

$$，N為時間內所有真正報SAR的案件總數$$

範例：

N=11

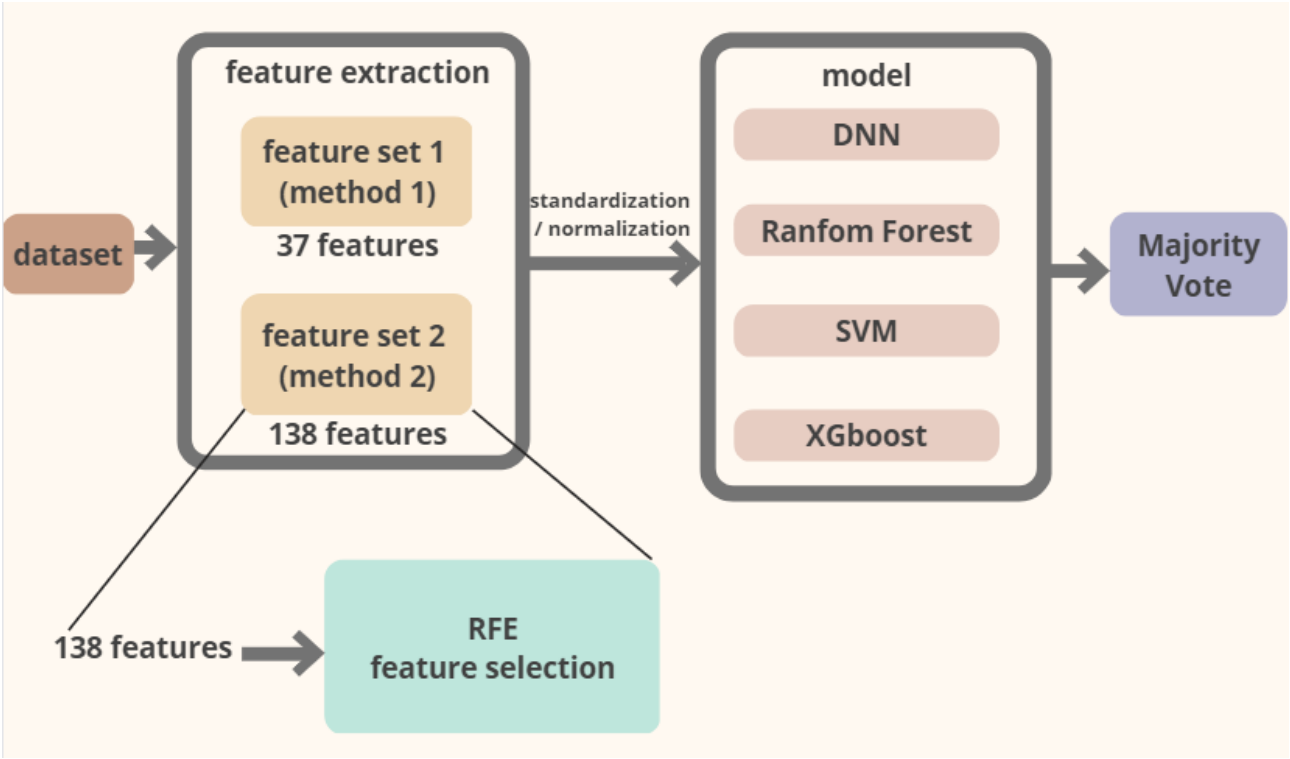
可疑案件總數 3000 筆

預測機率由高排到低後，若在第 1000 筆時抓到 10 筆真正報 SAR 案件

$$Recall@N-1\ 的Precision = \frac{11-1}{1000} = 0.01$$

IV. Method

Work Flow



Feature Engineering

Feature Set1 (第一組特徵設計)

cdtx、dp、remit 皆有逐筆的交易資料，交易日期分別記錄在 date、tx_date 、trans_date 欄位。我們以天為尺度合併這些資料，並結合 public_x_alert_date 與 train_x_alert_date 的 alert_date 劃分同一顧客 alert_key 之間的時間區間來將資料分群標註並存為 combined_data.csv。每位客戶都或多或少會有幾次的 "洗錢預警"(也就是被 alert_key 標記)，我們將"上一個 alert_key 結束到這一個 alert_key 預警日期" 區間設為這一個 alert_key 資料範圍，而之後將以 alert_key 作為單位進行下一步的特徵提取。此時 combined_data.csv 包含的資料有：cust_id、date、country、cur_type、amt、debit_credit、tx_time、tx_type、tx_amt、exchg_rate、info_asset_code、fiscTxId、txbranch、cross_bank、ATM、trans_n、trade_amount_usd、alert_key。我們將用這些資料來提取 feature set1 的特徵。

我們先將資料分成與時間有關(上表中的紫色)、與時間無關。根據以下所提到的洗錢定義，我們針對這些定義提取可能有效的預測因子，再合併與時間無關的資料，將特徵的亂度降低、僅提取有效的資料，以增加訓練的效率與精準度。

i. 洗錢定義：

- 現金存、提，累積達特定金額以上，以“略低於須申報金額(50 萬)”，頻繁的現金存、提。
- 存、提金額相當，相距時間不久、達特定金額以上
- 帳戶累積大量餘額，經常匯至國外帳戶達特定金額以上
- 開戶後立即有達特定金額以上款項存、匯入，又迅速轉移
- 匯至國外達特定金額以上
- 匯至國外的次數過於頻繁
- 自國外收到達特定金額以上款項後，立刻再將該筆款項匯回同一國家/地區的另一個人
- 突然存款達特定金額以上(同時有多張支票存入)
- 不活躍帳戶突然有大量資金存入、迅速轉出

ii. 以下是我們提取的特徵，我們基於這個 combined_data.csv 進行特徵提取，在同一個 alert_key 區間內提取出下列 37 種 feature：

- trade_amount_usd:取'trade_amount_usd'(交易金額)的最大值
- amt_sum:’amt’(交易金額-台幣)加總
- amt_std: ’amt’(交易金額-台幣)的變異數
- d_cur_type: 'cur_type'(消費地幣別)非台幣的交易次數
- day:同一個 alert_key 區間的交易資料有多少筆
- trans_no_final: 'trans_no'(交易編號)出現最多次的編號
- d_cur_type_ratio:取得跨國交易占總交易次數的比例('d_cur_type'除以'day')
- sum_0 / sum_1： 存/貸款總金額
- var_0 / var_1 :存/貸款變異數
- transnation_rate： 跨國交易占總交易次數的比例
- total_amount_money_in / total_amount_money_out： 跨國交易的匯入 / 匯出總金額
- var_money_in / var_money_out :跨國交易的匯入 / 匯出變異數
- cross_bank_rate :跨行交易占總交易次數的比例
- risk_rank : custinfo 裡顧客的風險等級(銀行處理的資料)
- occupation_code : custinfo 裡顧客的職業分類(銀行處理的資料)
- total_asset : custinfo 裡顧客的行內總資產(銀行處理的資料)
- AGE : custinfo 裡顧客的年齡(銀行處理的資料)
- lupay_max / lupay_std : ccba 裡 'lupay'(上月繳款總額)最大值/變異數
- clamt_max / clamt_std : ccba 裡 'clamt'(本月分期預借現金金額)最大值/變異數
- csamt_max /csamt_std : ccba 裡 'csamt'(本月預借現金金額)最大值/變異數
- inamt_max / inamt_std : ccba 裡 'inamt'(本月分期消費金額)最大值/變異數
- cucsm_max / cucsm_std： ccba 裡 'cucsm'(本月消費金額)最大值/變異數
- cucah_max / cucah_std : ccba 裡 'cucah'(本月借現金額)最大值/變異數
- cytocam_times： ccba 裡 'cytocam'(信用額度)調整次數
- cytocam_first / cytocam_last : ccba 裡最初的 / 最後的 'cytocam'(信用額度)
- cytocam_variance : ccba 裡 'cytocam'(信用額度)調整變異數

Feature Set2 (第二組特徵設計)

i. 我們先將資料做了預處理以及定義需要的特徵：

1. likely_trans 定義：若一連續交易為[100,101,102,103,104,105]，而目標閾值為 3%，則以每筆交易為基準算出相差在 3%以內的交易數(扣除自己)為 [3,4,5,5,4,3]，回傳最大值 5
2. dp 的 trans type 分為 4 種，以 cross_bank, txbranch, ATM 欄位的值區分。先將 txbranch 的空值補 0，有值則訂為 1，則以三個欄位的 0/1 值劃分
    a. trans0: [0,0,1]
    b. trans1: [0,0,0]
    c. trans2: [0,1,1]
    d. trans3: [1,1,1]
3. dp 的 tx type 分為兩種：
    a. 臨櫃現金交易 tx_type = 1 且 info_asset_code = 12)：tx1
    b. 其他：tx0
4. dp 的交易金額乘上匯率轉為台幣表示
5. 有關金額的特徵皆取 log scale

另外我們統計 alert_key 的間隔後發現中位數大約是 5 天左右，因此設計了不同時間區間來提取特徵，分別是：兩相鄰 alert_key 的時間區間(period_)、五天內(day5_)、十天內(day10_)。我們分別在以上三種區間計算下列 43 種特徵：

- cdtx_n_country: 消費涵蓋的國家數
- cdtx_n_country_switch: 連續消費時變換國家的次數
- cdtx_max_country: 消費最多的國家的消費次數
- cdtx_n_foreign: 外國消費次數
- cdtx_n_foreign_switch: 連續消費時在本國/外國間交換的次數
- cdtx_n_cur: 消費涵蓋的幣別數
- cdtx_n_cur_switch: 連續消費時變換使用幣別的次數
- cdtx_n_forcur: 使用外幣消費的次數
- cdtx_n_forcur_switch: 連續消費時幣別在本國/外國間交換的次數
- cdtx_tx_sum: 總消費金額
- cdtx_likely_trans_[05/10]: 金額相差在[5% / 10%]內的 likely_trans
- cdtx_tx_num: 總消費金額
- dp_trans[0/1/2/3]: trans[0/1/2/3] 的交易數
- dp_tx[0/1]: tx[0/1]的交易數
- dp_CR_sum: credit 交易總額
- dp_CR_likely_trans_[05/10]: credit 交易金額相差在[5% / 10%]內的 likely_trans
- dp_DB_sum: debit 交易總額
- dp_DB_likely_trans_[05/10]: debit 交易金額相差在[5% / 10%]內的 likely_trans
- dp_tx_sum: tx 交易總額，tx 定義為不分 credit/debit，數字為正的交易
- dp_tx_likely_trans[05/10]: tx 金額相差在[5% / 10%]內的 likely_trans
- dp_neg_sum: neg 交易總額，neg 定義為不分 credit/debit，數字為負的交易，先取絕對值再取-log
- dp_n_[CR/DP/tx/neg]: [credit/debit/tx/neg]交易總數

- remit_n_trans_type: trans_no 的種類數
- remit_trans[0/1/2/3]: trans_no 為[0/1/2/3]的外匯交易總數
- remit_tx_sum: 外匯交易總額
- remit_likely_trans[05/10]: 外匯交易金額相差在[5% / 10%]內的 likely_trans
- remit_tx_num: 外匯交易總數

ii. 若時間區間內無交易資料則補 0

iii. 提取 ccba 中 lupay, cycam, usgam, clamt, csamt, inamt, cucsm, cucah 欄位，負數先取絕對值再取-log

iv. 提取 custinfo 中 total_asset 欄位

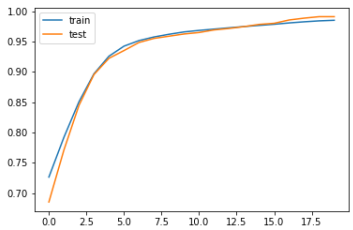
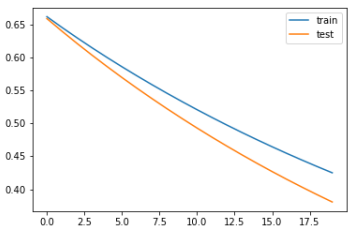
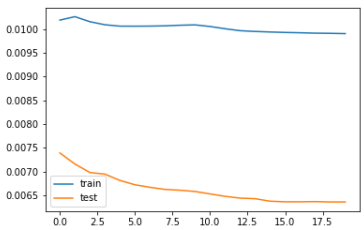
v. 合併以上特徵得到維度為 138(43×3+8+1)的特徵組

Model

我們嘗試了 Random Forest、SVM、XGB、DNN，各個 model 分別取上面幾種較適合的 feature 來做訓練。DNN 是我們首要嘗試的模型，其效果不如我們預期，可能的原因是 loss 計算的方式與評分方式不同以及模型不好訓練，因此我們決定不往深度學習的方向，改用 bagging、boosting 的概念，選擇了 Random Forest 與 XGB，這裡的效果是能夠將洗錢的資料照預測的機率排名後確實分類在總數的一半以上，但兩者排名分布較相似，較後面的資料比較重疊，這裡我們選擇結果較好的 XGB，為了能夠精進模型的預測能力，我們列出所有洗錢資料的排名後仔細研究明顯落後的兩筆資料的特徵，藉由 SVM 能改善這兩筆資料的預測排名，顯示 SVM 學習到的特徵是與 XGB 學習的方向不同，所以為補全兩個面向，我們使用 XGB 與 SVM 做 voting。

V. Result

Baseline - DNN

accuarcy	loss	score
		

可以看出訓練時 accuracy 有隨著 epoch 越訓練越好，Loss 也有逐步的收斂，但其實洗錢標籤 0/1 分布為 100:1，資料本身很不平衡，所以高準確率可能只是偏重預測標籤多的那一方，並且從 score 上來看結果也沒有隨著 loss 的下降有任何正比與反比關係，驗證損失函數計算的更新方向本來就跟我們計算成績的方式無關，這份資料庫用這個模型並不好訓練也無法產生足夠好的預測。

競賽結果

model		Random Forest (RF)	SVM	XGB	Voter(RF +XGB)	Voter(SVM +XGB)
score		0.0109	0.0082	0.0161	0.0108	**0.0163**
alert key / SAR prob rank	354939	19	1420	70	22	71
	355091	60	429	96	40	97
	355152	916	578	1430	1470	1398
	355724	10	199	64	25	65
	359668	377	823	434	543	431
	356602	41	102	33	42	33
	363320	15	1217	29	27	30
	358453	248	894	619	412	612
	363896	619	1162	324	511	324
	361617	1533	957	584	925	574
	363033	918	1176	399	837	398

上表為 random forest、SVM、XGB 以及結合 Random Forest 與 XGB 做 voting 和結合 SVM 與 XGB 做 voting 五個模型的分數以及洗錢資料依據機率的排名，可以看到 Random Forest 與 XGB 排名分布較接近，排名較後面的重疊率較高，推測兩者學習的方向較為接近，尤其可以看到 alert key 355152 的排名在 Random Forest 偏後，在 XGB 甚至是最後一名，讓排名後的始終無法透過其中一個模型學到其特徵並改善，因此使用 voting 可能無法達到好的效果，而 SVM 的排名分布看起來是顯著與 XGB 不同的，所以這裡使用 SVM 與 XGB 做 voting 能夠結合兩模型的優點達到更好的成效。

這裡的分數是 public data 測試的結果，總數為 1845 筆，排名的一半為 922 以上，其中含了 11 筆真正洗錢的資料，若我們要達到第 N-1 筆真正洗錢的資料能夠排在總數的一半以上，分數必須達到$\frac{10}{922} = 0.0108$，我們使用的 SVM 與 XGB 做 voting 確實能夠做到並且效果更好。

Public Leaderboard | Private Leaderboard

#	隊伍名稱	成員	提交次數	分數	上傳時間
17	十六減九	5	23	0.014104	12/26/2022 12:39:33 PM

檔案說明	上傳時間	Public分數	Private分數	訊息
private_xgbsvmvoting_np.csv xgbsvmvotingnp 上傳成員 王 靖淳	2022-12-26 12:39:33	1.0	0.0141044	Scoring success.

在競賽中我們繳交 SVM 與 XGB 做 voting 的模型預測結果，在主辦方提供的 private data 上的分數是 0.014104，排名位於 685 組參賽隊伍中的 17 名。至於所有真正洗錢的資料是否排名位於總資料量的一半以上我們無法確認，因為主辦方並無提供 private data 的解答，我們只能確認已公布解答的 public data 在我們的設計下確實能達到真正洗錢的資料排名位於總資料量的一半以上。

後續成果

競賽後期，主辦單位提供了 public data 的解答，雖然 private data 在競賽結束後已無法再上傳，但我們得以用 public 的解答繼續嘗試不同的做法。在學到更多洗錢知識與對資料更深入的了解後，我們設計了另一個系列的特徵：Feature Set2。這些設計出的特徵有 138 維，我們接著使用 ExtraTreeClassifier 為基礎的 Recursive Feature Elimination(RFE)做特徵篩選，挑出重要性前 100 的特徵來訓練，結果如下：

透過 RFE 挑選的前 10 重要特徵。可發現第一名與交易失敗有關，其餘則大多跟外幣、國外交易有關，確實與洗錢的定義相符。

	period_dp_neg_sum	day5_cdtx_n_forcur	period_remit_trans1	ccba_csamt	day10_cdtx_n_forcur	period_remit_trans3	day5_cdtx_n_cur_switch	day5_cdtx_n_cur	day10_cdtx_n_country_switch	day10_cdtx_max_country
0	-46.051702	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
1	-46.051702	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
2	-46.051702	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
3	-46.051702	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
4	-46.051702	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0

模型訓練結果：

model		Random Forest (RF)	SVM	XGB	Voter(XGB+ SVM+RF)
score		0.0100	0.0101	**0.0197**	0.0160
alert key / SAR prob rank	354939	126	664	6	10
	355091	144	36	211	159
	355152	325	1611	191	232
	355724	274	993	86	112
	359668	1380	288	506	462
	356602	25	38	16	14
	363320	76	16	53	46
	358453	367	574	205	246
	363896	826	632	624	784
	361617	1000	627	454	625
	363033	314	93	336	229

最後發現這組用 XGB 能得到比之前更好的效果，並且和 Feature Set1 的預測結果分布也不相同，若能將使用不同輸入特徵的模型進一步整合做多數決投票，或許能達到更好的互補效果，這也是我們未來的努力方向之一。

VI. Discussion

這份資料庫原始資料是許多逐筆交易資訊，所以我們最初的想法是將這些資料以洗錢警示(alert key)為依據做切分，切成許多時間序列後利用 Transformer 學習序列內天與天之間的關係，或許能從中學到一些關於洗錢的交易線索，但實際上，這樣的效果並無法達到我們的預期，我們認為最大的原因是未經過資料處理，我們的資料是將許多不同種類的交易資訊單純以時間作排列並合併，可想而知，資料是非常髒亂的，有許多空值，並且會遇到資料的時間單位對不上的問題，這些原因都是造成 Transformer 效果不好的可能，因此設計特徵是一項重要工程，我們參考了洗錢定義，根據這些定義來猜想什麼是我們這裡可以歸納出的相關因子，可以看到以上的結果都是使用經過設計後的特徵，使用設計後的特徵確實有效提升我們的成效，不過我們在金融方面的知識可能也不夠充足，所以設計的特徵不夠有效，這是一個未來可以學習、可做改善的地方。

在模型設計上，我們嘗試了多種模型並做多數決投票，不過輸入特徵必須相同是內建函示庫的一大限制。實際上我們也可以整合多種輸入訓練各種模型，例如時序列模型、特徵工程產生的模型、以背景資料做預測的模型等等，或許能產生更好的互補效果，以多模態的輸入訓練模型也是我們未來想嘗試的做法之一。

除此之外，想探討的是本次競賽的評分，雖然我們有達成目標，但以這種方式評分成果要好的話就需要將第 N-1 筆的真正洗錢的資料盡可能地往前排，然而在實作過程中我們可以發現，某幾筆資料特別不同，比賽使用這種評分方式或許是希望我們能找出方法使這幾筆異常值能被抓到，我們可以從上面的 DNN-score 圖看到我們目前計算 loss 的方法與評分方式是毫無關係的，因為其更新通常會往能準確預測更多洗錢資料的方向，因此容易拋棄一些異常值，造成這些異常值排名變成很後面，所以我們並不能確保以如此方式做參數更新也能夠使評分往好的方向成長，但目前我們的技術還無法改善這部分，我們認為損失函數會是這個題目一個未來可研究的方向。

VII. Reference

洗錢手法:

http://www.fullgoal.com.cn/contents/2016/6/22-3bbe1d7f2c28489abfa1aaf2a1b919e7.html

反洗錢技術:

https://www.tedu.tw/blog/artificial-intelligence-preventing-money-laundering.html

https://ai.iias.sinica.edu.tw/ai-on-aml-and-cft-in-practice/

https://www.bnext.com.tw/article/63799/fc-award1

反洗錢相關研究:

https://arxiv.org/pdf/2011.08492.pdf

https://xtglxb.sjtu.edu.cn/CN/abstract/abstract1208.shtml

https://www.x-mol.com/paper/1340384585059057664/t?recommendPaper=1340448598824501248

模型:

https://scikit-learn.org/stable/modules/svm.html

https://xgboost.readthedocs.io/en/stable/python/python_intro.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html