題目：銀行客戶去留之預測

指導教授：鄭又仁

組長：工工所 楊韻玄

組員：工工所 劉維仁

工工所 蔡雨築

工工所 潘縉緯

工工所 謝瑞璟

目錄

一、 簡介 3
二、 資料概述分析 4
2.1 變數介紹 4
2.2 視覺化探索資料 5
2.3 資料預處理 9
三、 研究方法 10
3.1 Random Forest 10
3.2 XGboost (Extreme Gradient Boosting) 11
3.3 CatBoost 12
四、 資料分析 14
4.1 模型衡量指標 14
4.2 單一模型 16
4.2.1 CatBoost 16
4.2.2 Random Forest 17
4.2.3 XGBoost 18
4.2.4 單一模型比較 19
4.3 重抽樣法 20
4.3.1 CatBoost 20
4.3.2 Random Forest 21
4.3.3 XGBoost 22
4.3.4 SMOTE 比較 23
4.4 投票法 24
4.5 產生新特徵 24
4.5.1 方法介紹 24
4.5.2 Random Forest + CatBoost 26
4.5.3 XGBoost + CatBoost 27
4.6 模型 overfitting 28
五、 結論 29
六、 工作分配 29
七、 資料來源 29

一、 簡介

隨著資料科學以及機器學習的蓬勃發展，目前已經廣泛應用到諸多的領域上，透過模型的建立將大量資料萃取出有價值知識，提供專業領域或決策單位具有前瞻性的指標做為參考。

客戶對於企業來說是最重要的資產，透過完善的客戶服務和深入的客戶分析來滿足客戶的個性化需求，提高客戶的滿意度和忠誠度，才能保證銀行利潤增長的實現。客戶流失的定義為該客戶不再參與原業務、不再重複購買產品或者終止原先使用的服務，客戶流失量會對公司的業務產生影響，本研究收集過去客戶六個月在銀行的資料進行分析，根據不同的客戶屬性（例如年齡，性別，地理位置等）來預測客戶是否會流失。

由於客戶在銀行的資料組合多樣，無明確的指標可以看出客戶是否會流失的關鍵性依據，造成資料處理以及預測困難性，因此本研究目的是建立機器學習模型去預測客戶是否會在六個月後離開銀行，並使用不同的方法進行比較驗證每個模型的效度。

二、 資料概述分析

此章節為原始資料的基本介紹與探索，原始資料有 10000 筆的客戶資料、變數共有 14 個、無缺值資料，前 13 個變數為自變數，最後一列為 1 或 0 的二進制反應變數表示客戶是否會離開銀行，為本研究想要預測的目標。接下來會進行變數的介紹、用視覺化統計圖表去解釋變數間存在的潛在關係以及資料預處理三個部分。

2.1 變數介紹

原始變數資訊如 2.1 所示，包括變數名稱、型態以及說明。

表 2.1 變數介紹

	變數名稱	變數型態	說明
1	RowNumber	類別型	紀錄該資料為第幾筆。
2	CustomerId	類別型	客戶在該銀行的代號。
3	Surname	類別型	客戶的姓氏。
4	CreditScore	數值型	客戶在該銀行的信用分數，由該銀行所衡量。
5	Geography	類別型	客戶所在之國家，包含法國、西班牙、德國三個國家。
6	Gender	類別型	客戶的生理性別。
7	Age	數值型	客戶的年齡。
8	Tenure	數值型	客戶成為該銀行客戶的年數。
9	Balance	數值型	客戶在該銀行的帳戶餘額。
10	NumOfProducts	類別型	客戶在六個月內曾向銀行購買的產品種類個數。

11	HasCrCard	二進制類別型	客戶是否在該銀行有信用卡，表示方式：有信用卡為 1，否為 0。
12	IsActiveMember	二進制類別型	客戶是否被該銀行判定為活躍客戶，表示方式：活躍客戶為 1，否為 0。
13	EstimatedSalary	數值型	客戶被該銀行所評估的薪資。
14	Exited	二進制類別型	客戶是否在六個月後會離開該銀行，表示方式：離開該銀行為 1，否為 0。

2.2 視覺化探索資料

透過原始數據的變數關聯性以及不同的視覺化統計圖表去呈現並解釋可能存在的潛在關係進行初步的分析，訂定後續分析方向並與模型分析後的結果做驗證檢查是否具有一致性。

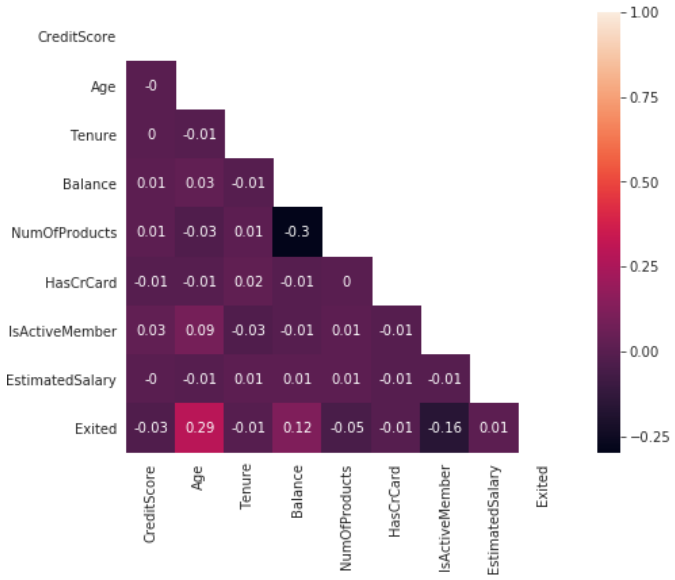

圖 2.1 原始數值型變數間相關性

由上圖可發現數值型解釋變數間並無高度相關性，僅在變數 Exited 與 Age 之間有 0.29 的低相關性，變數 NumOfProducts 與 Balance 之間具有-0.3 的低

附相關性，因此可排除資料帶有高度共線性的疑慮。當資料帶有高度共線性時，將會使預測受到特定的變數或是變數組合之影響程度增加。

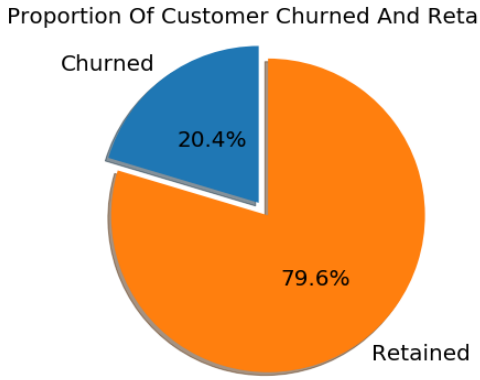

圖 2.2 客戶流失與否比例

由上圖可發現原始資料流失與否的比例懸殊，離開該銀行的客戶佔所有客戶的 20%，因此若將整份原始資料集投入模型訓練，則可能對於「佔少數」的類別在預測方面有較低的敏感度（此處佔少數的為「流失」）。意即：若將整份訓練資料集投入，則可造成模型在預測方面會有「比起預測為 1(流失)，有更大的機率將其預測為 0(未流失)」。接下來針對類別型與數值型變數分別進行探討。

(1)類別型資料

1. 客戶性別

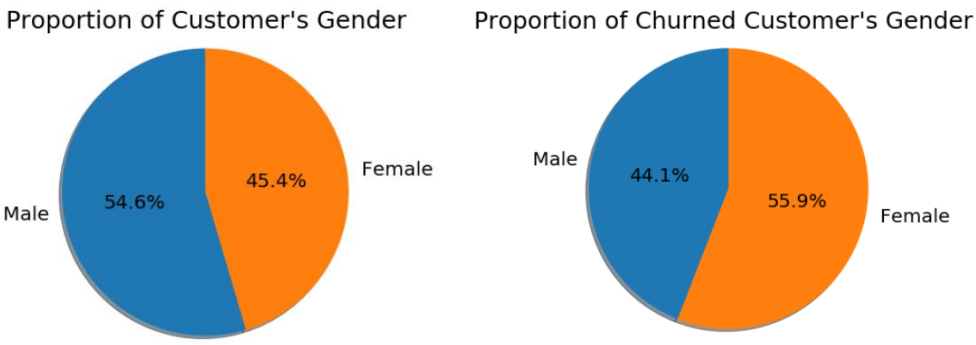

圖 2.3 客戶的性別比例以及流失客戶的性別比例

左圖為客戶中男女(Gender)的比例，右圖為流失客戶中男女的比例，從上圖之差異可以看出在銀行總客戶中男性比例較女性高，但在流失客戶中女性比

例超出男性許多，相對於男性來說女性比較容易成為銀行的流失客戶。

2. 客戶所在國家

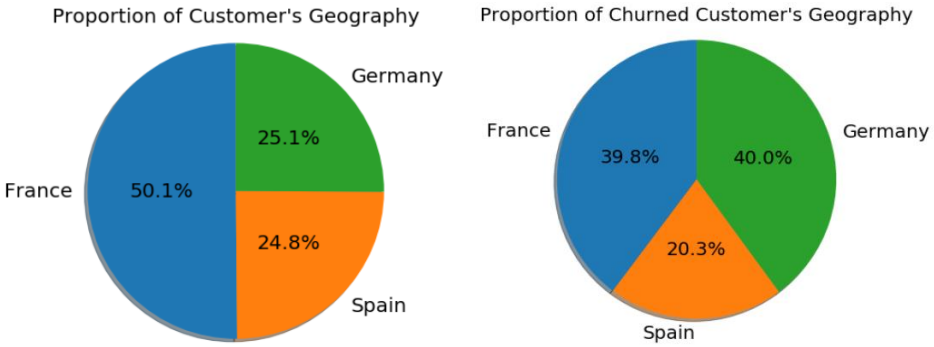

圖 2.4 客戶的國家比例以及流失客戶的國家比例

左圖為銀行的客戶中居住地區(Geography)的類別分佈情形，而右圖則為流失的客戶。發現德國流失人數相對於法國與西班牙在比例上來說高出許多，表示在德國的客戶有較大的機會沒有銀行的忠誠度，會選擇離開該銀行，而法國與西班牙在人數比例之間僅有些許變化。

3. 客戶購買產品數量

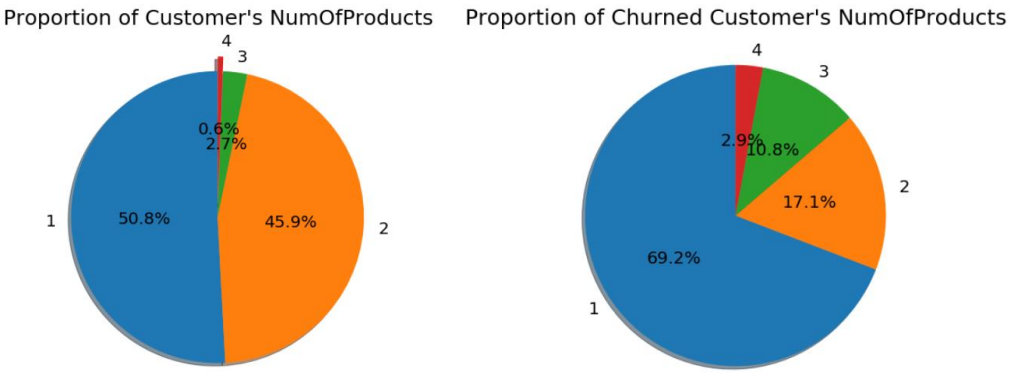

圖 2.5 客戶購買產品數量的比例以及流失客戶購買產品數量的比例

左圖為銀行的客戶購買產品(NumOfProducts)的類別分佈情形，而右圖則為流失的客戶。發現大約一半的客戶購買該銀行一個產品或是兩個產品，僅有極少部分客戶會購買三個產品或是四個產品；然而在流失客戶中購買三、四個產品的客戶流失率相對提高許多，購買兩個產品的客戶僅有少部分會選擇離開該銀行，購買一個產品的客戶有稍微提高一些。

4. 客戶是否擁有信用卡

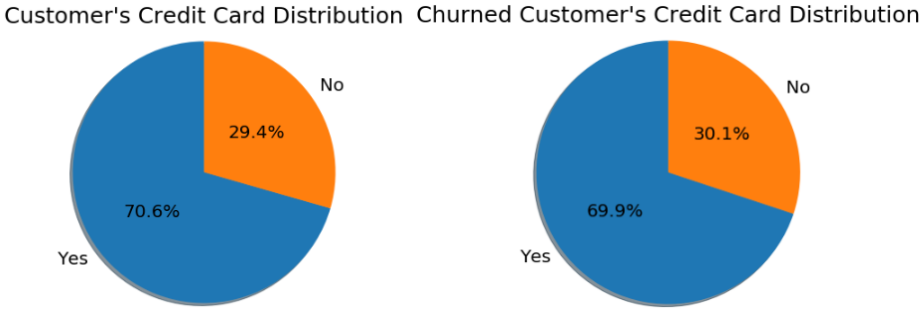

圖 2.6 客戶擁有信用卡的比例以及流失客戶擁有信用卡的比例

左圖為銀行的客戶擁有信用卡(HacCrCard)的類別分佈情形，而右圖則為流失的客戶。從上方二圖可以發現其比例並無顯著差異，代表客戶是否擁有信用卡不會影響客戶是否會流失。

5. 客戶是否為活躍客戶

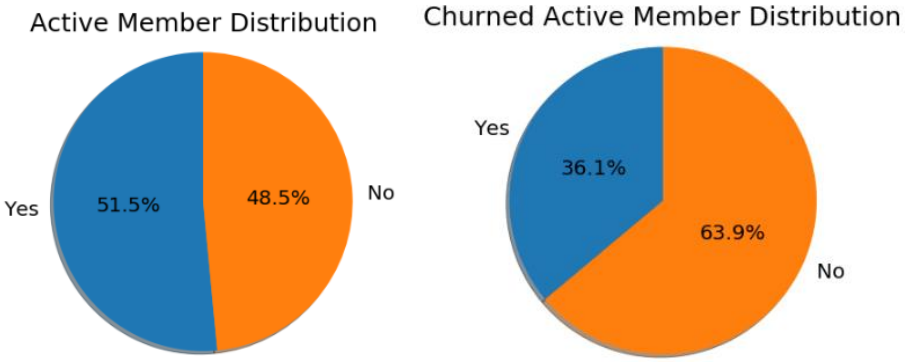

圖 2.7 客戶為活躍客戶的比例以及流失客戶為活躍客戶的比例

左圖為銀行的客戶是否為活躍客戶 (IsActiveMember)的類別分佈情形，而右圖則為流失的客戶。從上方二圖可以發現活躍與非活躍客戶大約各占銀行客戶的一半，然而在流失客戶中較大的比例為非活躍客戶，符合現實中正常的情況。

(2)數值型資料

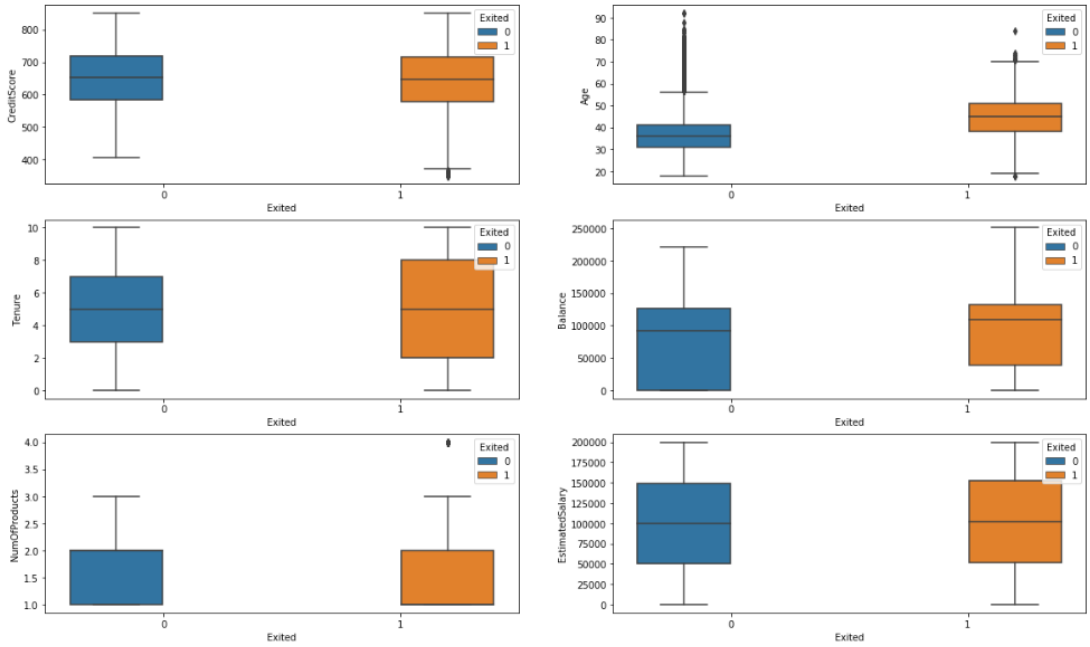

圖 2.8 數值型變數與反應變數箱型圖

將數值型變數透過反應變數的分類繪製成箱型圖去探討，從上圖可以看到在 CreditScore、NumOfProduct、EstimatedSalary 的箱型圖中在客戶是否會離開該銀行的分布無明顯差異。Age 箱型圖則可以看出在流失客戶的分類其分部較為年長的客戶，代表年長客戶有較大的可能會離開該銀行。Tenure 的箱型圖在 Exited 為 1 的分類其分布較廣，表示當客戶成為該銀行客戶年數較為極端時(較大或是較小)有較大的可能性會離開該銀行。在 Exited 為 1 的 Balance 箱型圖表示該銀行正在流失擁有銀行結餘的客戶，可能造成該銀行可用資金受限。

2.3 資料預處理

此筆銀行客戶資料為乾淨的資料，在 10000 筆資料當中並無缺失值，前三個變數 RowNumber, CustomerId, Surname 為不影響銀行客戶是否會流失並將其刪除，保留剩餘 10 個變數納入後續的模型建立，分別為 CreditScore, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary。

三、 研究方法

3.1 Random Forest

Random Forest 是基於決策樹分類器的集成學習演算法，由 Leo Breiman 於 2001 年提出。原始隨機森林演算法中分類器為 CART (Classification and Regression Tree) 樹，影響模型性能的主因為強度(Strength)與相關性(Correlation)，每棵樹的分類強度越大、相關性越小，模型性能越好。

✓ Bagging (Bootstrap aggregating)

Random Forest 透過 bagging 進行集成學習，在 bagging 中，從原始樣本重複抽取(bootstrap)多組訓練集(x_i^{*b}, y_i^{*b})，其中，$b=1,...B$，$i=1,...n$，共有 B 組相互獨立的訓練集，並針對每組訓練集建立未剪枝的決策樹模型。以分類資料為例，使用多數投票法建構模型函數，其式如下：

$$\hat{f}^{bag}(x) = arg \max_{k=1,...K} \sum_{b=1}^{B} 1\{\hat{f}^{tree,b}(x) = k\} \quad (3.1)$$

✓ Random Forest 演算法

1. 使用 bagging 演算法產生訓練資料：從原始 N 個樣本中，使用重複抽樣(bootstrap)的方式，產生 n 組訓練集。

2. 針對每組訓練集，建立未剪枝的分類或回歸決策樹，在節點找特徵進行分裂時，並非找最適的特徵以產生最佳分枝，而是在 M 個特徵中隨機抽取 m 個特徵，並在抽到的特徵中找到最佳解並進行分枝。

3. 透過 n 棵樹之結果進行集成，若為分類資料，則使用多數投票法；若為回歸資料，則使用平均法。

✓ Out-Of-Bag (OOB)估計

OOB 資料為在使用 bootstrap 方式生成訓練集資料時，未被取樣之資料，其可以用來估計樹的泛化誤差(Generalization error)，也可以用來計算單

個特徵的重要性。Random Forest 不需要再進行交叉驗證以獲取測試集誤差的不偏估計，相較交叉驗證，OOB 估計能透過少量資料的計算量達到近似於交叉驗證的結果。

3.2 XGboost (Extreme Gradient Boosting)

XGboost 最初由 Tianqi Chen 開發，此演算法成功的重要因素為在所有情況下的可擴充套件性。XGboost 的可擴充套件性主要來自幾個重要的系統與演算法優化，這些創新包括：可處理稀疏資料的創新樹學習演算法、加權分位數略圖程式（Weighted Quantile Sketch）以尋找近似樹學習中之候選分割點、加快學習速度的平行與分散式運算、以及用於核外樹形學習的快取感知塊結構。

假設 XGboost 模型中有 M 棵樹，模型函數如式(3.2)：

$$\hat{f}(x_i) = \sum_{m=1}^{M} h_m(x_i)\,, h_m \epsilon \mathcal{F} \tag{3.2}$$

目標函數如式(3.3)：

$$Obj^{(t)} = \sum_{i=1}^{n} L_n(y_i, \hat{f}(x_i)) + \sum_{m=1}^{M} \Omega(h_m) \tag{3.3}$$

其中，前項為訓練誤差，用來衡量模型在訓練資料上的預測能力，典型的訓練誤差包括均方誤差(Square loss) $L\big(y, f(x)\big) = (y - f(x))^2$ 以及羅吉斯誤差(Logistic loss) $L\big(y, f(x)\big) = y \ln\big(1 + e^{-f(x)}\big) + (1 - y)\ln(1 + e^{f(x)})$。目標函數後項為正則項，用來控制模型的複雜度，防止過擬合(overfitting)的發生，其定義為 $\Omega(h) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} {w_j}^2$，其中，$T$ 為樹節點數量。前項訓練誤差使用泰勒展開式逼近，重新定義目標函數如式(3.4)：

$$Obj^{(t)} = \sum_{j=1}^{T}\left[\left(\sum_{x_i \epsilon R_{jt}} g_i\right) w_j + \frac{1}{2}\Big(\sum_{x_i \epsilon R_{jt}} s_i + \lambda\Big) w_j^2\right] + \gamma T \tag{3.4}$$

其中，若訓練誤差為均方誤差，$g_i = 2(\hat{f}^{(t-1)}(x_i) - y_i)$、$s_i = 2$。透過目標函數，我們可以得知 XGboost 透過式(3.5)衡量候選分枝並建立分枝：

$$L_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \quad (3.5)$$

XGboost 使用增量訓練(Additive training)透過多次疊代逐步使目標函數下降，其尋找分割點的演算法包括 Exact Greedy Algorithm 與 Approximate Algorithm，前者為暴力尋找所有可能的分割點，而當資料量過大，XGboost 則提出後者使用加權分位數略圖程式之近似演算法以加快學習速度。此外，在分割樹時，XGboost 將每棵樹的節點增加一個預設方向，適合處理缺失值多的稀疏資料。

3.3 CatBoost

Gradient boosting 的學習模型存在類別型特徵前處理與預測偏差(prediction shift)的問題，因此俄羅斯搜尋引擎 Yandex 開發 CatBoost 演算法，導入 Ordered boosting 一種改良的 Gradient boosting 演算法以及能處理類別型特徵資料的演算法，以避免 Target leakage 的發生。

✓ 類別型特徵

類別型特徵的前處理方式包括 One-hot-encoding、Target Encoding 等，前者適合用於類別數較少的類別型特徵，而後者則是使用樣本標籤之統計值代替各類別特徵的值，樣本 i 的特徵 k 值之替代值如式(3.6)：

$$\hat{x}_k^i = \frac{\sum_{j=1}^{n} \mathbb{I}_{\{x_j^i = x_k^i\}} \cdot y_j + ap}{\sum_{j=1}^{n} \mathbb{I}_{\{x_j^i = x_k^i\}} + a} \quad where \;\; a > 0 \quad (3.6)$$

然而，若某類別只有一個或少量的樣本，容易造成過擬合的問題。因此 CatBoost 使用另一種有效策略，針對訓練集隨機排列，計算樣本的標籤統計值時，只將此樣本以前的樣本標籤值納入計算，此方法可以有效使用全部的訓練集資料訓練模型並避免過擬合的發生，如式(3.7)，此外，CatBoost 演算法同時也將多個類別型特徵組合而成的新特徵納入考量。

$$\hat{x}_k^i = \frac{\sum_{X_{j\in D_k}} \mathbb{I}_{\{x_j^i = x_k^i\}} \cdot y_j + ap}{\sum_{X_{j\in D_k}} \mathbb{I}_{\{x_j^i = x_k^i\}} + a} \quad where \;\; a > 0, D_k = \{X_j : \sigma(j) < \sigma(k)\} \quad (3.7)$$

✓ Ordered boosting

CatBoost 提出一種演算法解決傳統 gradient boosting 存在的預測偏差(Prediction shift)問題，其第一階段採用梯度步長的無偏估計，第二階段使用傳統的 GBDT 方案執行，其演算法如下。

Algorithm 2: Building a tree in CatBoost

input : $M, \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}, \alpha, L, \{\sigma_i\}_{i=1}^{s}, Mode$

$grad \leftarrow CalcGradient(L, M, y)$;
$r \leftarrow random(1, s)$;
if $Mode = Plain$ **then**
  $G \leftarrow (grad_r(i)$ for $i = 1..n)$;
if $Mode = Ordered$ **then**
  $G \leftarrow (grad_{r,\sigma_r(i)-1}(i)$ for $i = 1..n)$;
$T \leftarrow$ empty tree;
foreach *step of top-down procedure* **do**
  foreach *candidate split* c **do**
    $T_c \leftarrow$ add split c to T;
    if $Mode = Plain$ **then**
      $\Delta(i) \leftarrow \text{avg}(grad_r(p)$ for $p : leaf_r(p) = leaf_r(i))$ for $i = 1..n$;
    if $Mode = Ordered$ **then**
      $\Delta(i) \leftarrow \text{avg}(grad_{r,\sigma_r(i)-1}(p)$ for $p : leaf_r(p) = leaf_r(i), \sigma_r(p) < \sigma_r(i))$ for $i = 1..n$;
    $loss(T_c) \leftarrow \cos(\Delta, G)$
  $T \leftarrow \arg\min_{T_c}(loss(T_c))$
if $Mode = Plain$ **then**
  $M_{r'}(i) \leftarrow M_{r'}(i) - \alpha\,\text{avg}(grad_{r'}(p)$ for $p : leaf_{r'}(p) = leaf_{r'}(i))$ for $r' = 1..s, i = 1..n$;
if $Mode = Ordered$ **then**
  $M_{r',j}(i) \leftarrow M_{r',j}(i) - \alpha\,\text{avg}(grad_{r',j}(p)$ for $p : leaf_{r'}(p) = leaf_{r'}(i), \sigma_{r'}(p) \le j)$ for $r' = 1..s$, $i = 1..n, j \ge \sigma_{r'}(i) - 1$;
return T, M

四、 資料分析

前面章節已做過資料前處理，共 10000 筆資料做分析。資料將切割為 80%訓練集資料（1632 筆流失、6368 筆留下，共 8000 筆），20%測試集資料（405 筆流失、1595 筆留下，共 2000 筆）。接著使用訓練集資料建立模型，並用測試集資料做驗證，以 F1 Score 做為模型衡量指標。本章節分為六個小節，首先介紹 4.1 模型衡量指標，接著為 4.2 單一模型建立，4.3 重抽樣法，4.4 投票法，4.5 產生新特徵，最後 4.6 模型過度擬合。本章架構如下圖所示。

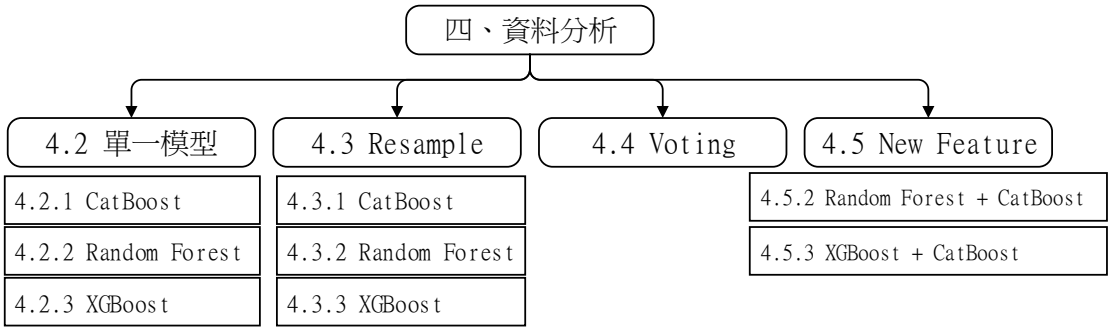

圖 4.1 分析架構

4.1 模型衡量指標

分類問題的指標計算通常以混淆矩陣做為基礎，混淆矩陣如下表格所示，每格的意義如下：

TP : True Positive 真陽性，預測流失且實際流失

FP : False Positive 偽陽性，預測流失但實際留下

FN : False Negative 偽陰性，預測留下但實際流失

TN : True Negative 真陰性，預測留下且實際留下

表 4.1 混淆矩陣

	真實		
預測		陽性 Positive (流失 1)	陰性 Negative (留下 0)
	陽性 Positive (流失 1)	TP	FP
	陰性 Negative (留下 0)	FN	TN

根據混淆矩陣，可以常用的指標如下表格所整理：

表 4.2 常用指標

指標	公式	意義
Accuracy 準確率	$\frac{TP+TN}{TP+FP+FN+TN}$	正確被預測為陽性與陰性的資料占所有資料的比例
Precision 精確率	$\frac{TP}{TP+FP}$	正確被預測為陽性的資料占所有被預測為陽性的比例
Recall 召回率	$\frac{TP}{TP+FN}$	正確被預測為陽性的資料占所有真實為陽性的比例
F1 Score	$2*\frac{Precision*Recall}{Precision+Recall}$	Precision 與 Recall 的調和均值

由於如 2.2 節所述，本研究的資料為不平衡資料，可能預測時會有較大機率預測為全是 0（留下），但依然有高度的 Accuracy。而本研究關心的是客戶是否會流失，所以使用關注 True Positive 的指標（Precision、Recall、F1 Score）會較適合。F1 score 是同時考量 Precision 和 Recall 的指標，為 F Measure 的特例，其公式如下：

$$F_\beta = (1+\beta^2)*\frac{Precision*Recall}{(\beta^2*Precision)+Recall}$$

本研究認為兩種指標同等重要，因此$\beta = 1$，公式如表格所示。若認為 Precision 較重要，β數值可以設定較小；若認為 Recall 較重要，則相反。

4.2 單一模型

4.2.1 CatBoost

表 4.3 混淆矩陣

訓練集	真實	
		流失	留下
預測	流失	849	180
	留下	783	6188

測試集	真實	
		流失	留下
預測	流失	213	72
	留下	192	1523

表 4.4 訓練集和測試集之 Accuracy、F1 score 比較

	訓練集	測試集
Accuracy	0.879625	0.868
F1 Score	0.638106	0.617391

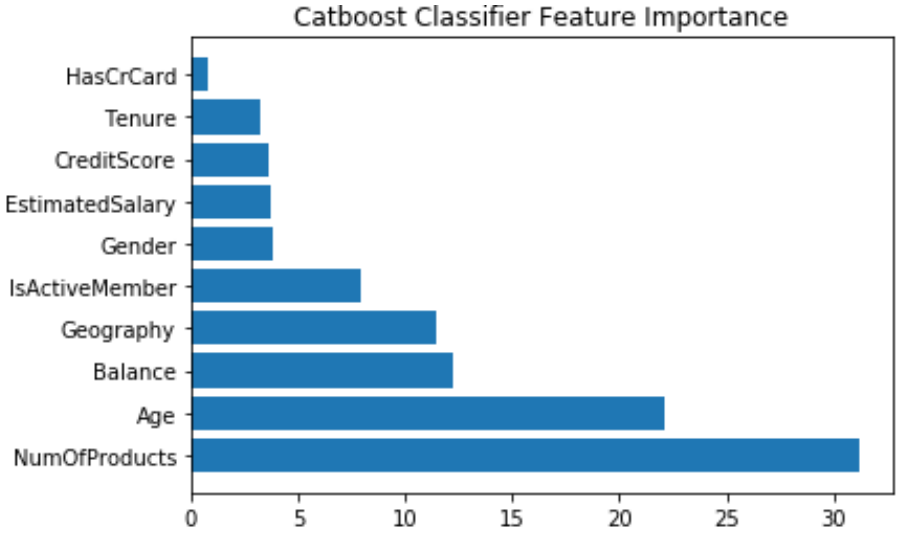

圖 4.2 CatBoost’s Feature importance

4.2.2 Random Forest

表 4.5 混淆矩陣

訓練集	真實		
預測		流失	留下
	流失	1632	0
	留下	0	6368

測試集	真實		
預測		流失	留下
	流失	208	68
	留下	197	1527

表 4.6 訓練集和測試集之 Accuracy、F1 score 比較

	訓練集	測試集
Accuracy	1	0.8675
F1 Score	1	0.610866

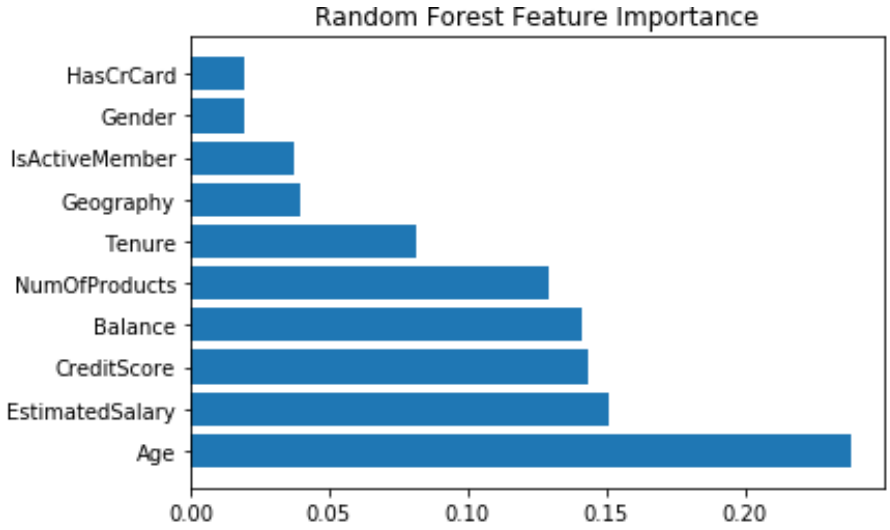

圖 4.3 Random Forest’s Feature importance

4.2.3 XGBoost

表 4.7 混淆矩陣

訓練集	真實		
預測		流失	留下
	流失	742	166
	留下	890	6202

測試集	真實		
預測		流失	留下
	流失	197	61
	留下	208	1534

表 4.8 訓練集和測試集之 Accuracy、F1 score 比較

	訓練集	測試集
Accuracy	0.868	0.8655
F1 Score	0.584252	0.594268

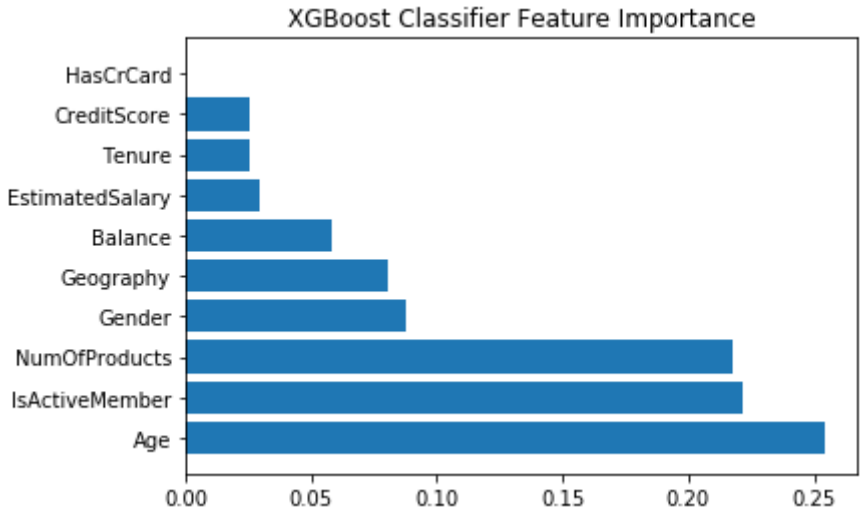

圖 4.4 XGBoost’s Feature importance

4.2.4 單一模型比較

表 4.9 比較

Accuracy			
		訓練集	測試集
Label encoding	CatBoost	0.879625	0.868
	Random Forest	1	0.8675
	XGBoost	0.868	0.8655

F1 Score			
		訓練集	測試集
Label encoding	CatBoost	0.638106	0.617391
	Random Forest	1	0.610866
	XGBoost	0.584252	0.594268

將上述章節彙整成表 4.9，由於 CatBoost 可以處理類別型資料，因此不需做 Label encoding 的類別型轉換。Accuracy 在三種模型的表現都約 0.8，F1 Score 約 0.6 左右。Random Forest 在訓練集的指標都為 1，因此推測有可能 overfitting。而根據 F1 Score，XGBoost 的預測能力最差。

CatBoost 模型中的重要變數排名如圖 4.1，前三名為 NumOfProducts、Age、Balance。Random Forest 如圖 4.2 所示，前三名為 Age、EstimatedSalary、CreditScore，但後兩者的重要程度和 Balance、NumOfProducts 差不多。XGBoost 如圖 4.3 所示，前三名為 Age、IsActiveMember、NumOfProducts，和其他變數的重要程度有明顯落差。根據第二章的數值型分析，Age、Tenure、Balance 和客戶是否流失有關連性。分析結果顯示，三種模型中，Age 都是變數重要性的前三名，Balance 變數在前兩種模型中的重要程度算高，驗證了第二章的分析。但 Tenure 則沒有明顯的關聯性。

4.3 重抽樣法

從第二章可以知道客戶流失與否的比例大約為 80%：20%相差懸殊，針對不平衡資料的情況下，本研究採用 2002 年的論文「SMOTE: Synthetic Minority Over-sampling Technique」提出 SMOTE (Synthetic Minority Oversampling Technique) 是來進行重抽樣(Resampling)，使兩類資料量一致，其演算法如下：

1. 找出與陽性個體 $\mathbf{x}_i$ 的最近的 k 個陽性鄰點 (k-nearest neighbors)
2. 在 k 個鄰點中隨機選擇一個，稱作 $\mathbf{x}_j$，我們會利用該鄰點用來生成新樣本
3. 計算 $\mathbf{x}_i$ 與 $\mathbf{x}_j$ 的差異 $\Delta = \mathbf{x}_j - \mathbf{x}_i$
4. 產生一個 0 到 1 之間的隨機亂數 η
5. 生成新的樣本點 $\mathbf{x}_i^{(new)} = \mathbf{x}_i + \eta\Delta$

將訓練集資料中使得反應變數客戶流失與否(Exited)為 0 及為 1 的資料各 6368 筆，總共 12736 筆，並分別使用 Catboost、Random Forest 及 XGBoost 建立模型後使用 Accuracy 及 F1 Score 檢驗各模型效度。

4.3.1 CatBoost

表 4.10 混淆矩陣

訓練集	真實		
預測		流失	留下
	流失	1264	566
	留下	368	5802

測試集	真實		
預測		流失	留下
	流失	277	224
	留下	128	1371

表 4.11 訓練集和測試集之 Accuracy、F1 score 比較

	訓練集	測試集
Accuracy	0.88325	0.824
F1 Score	0.73021	0.611479

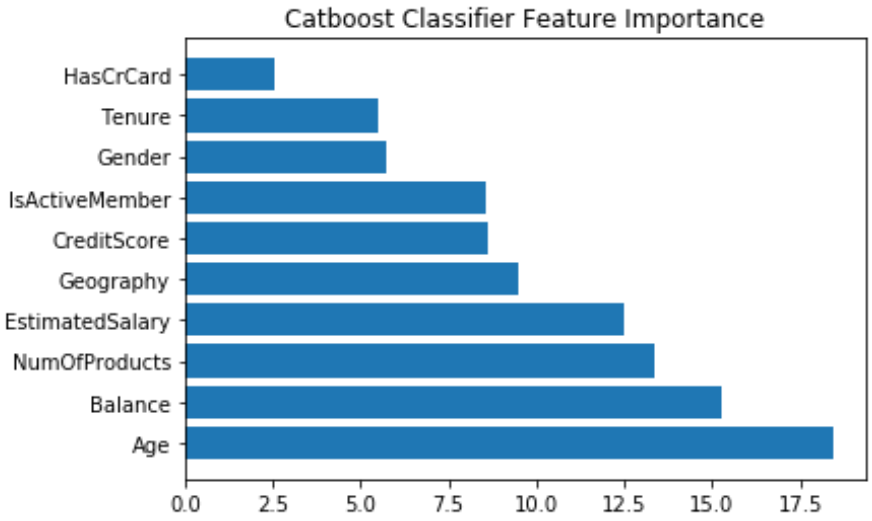

圖 4.5 CatBoost’s with SMOTE Feature importance

4.3.2 Random Forest

表 4.12 混淆矩陣

訓練集	真實		
預測		流失	留下
	流失	1632	1
	留下	0	6367

測試集	真實		
預測		流失	留下
	流失	264	243
	留下	141	1352

表 4.13 訓練集和測試集之 Accuracy、F1 score 比較

	訓練集	測試集
Accuracy	0.999875	0.808
F1 Score	0.999694	0.578947

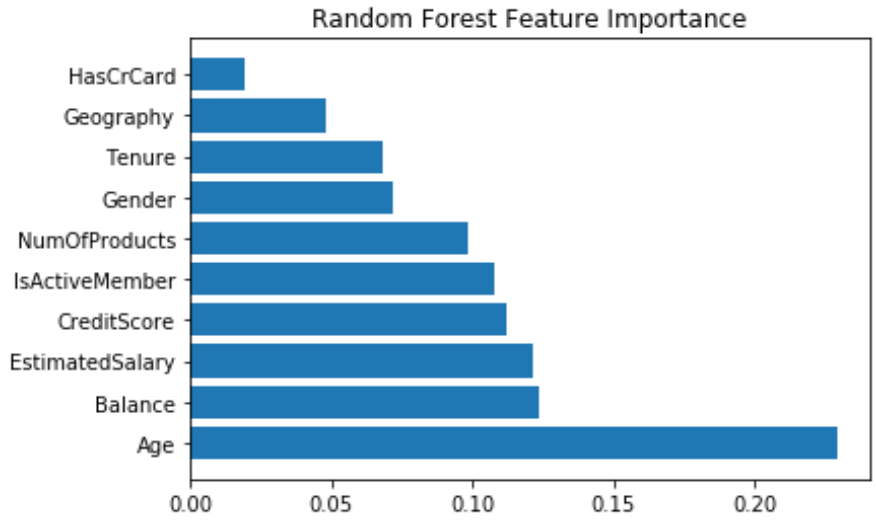

圖 4.6 Random Forest with SMOTE Feature importance

4.3.3 XGBoost

表 4.14 混淆矩陣

訓練集	真實		
預測		流失	留下
	流失	1141	927
	留下	491	5441

測試集	真實		
預測		流失	留下
	流失	288	284
	留下	117	1311

表 4.15 訓練集和測試集之 Accuracy、F1 score 比較

	訓練集	測試集
Accuracy	0.82275	0.7995
F1 Score	0.61676	0.58956

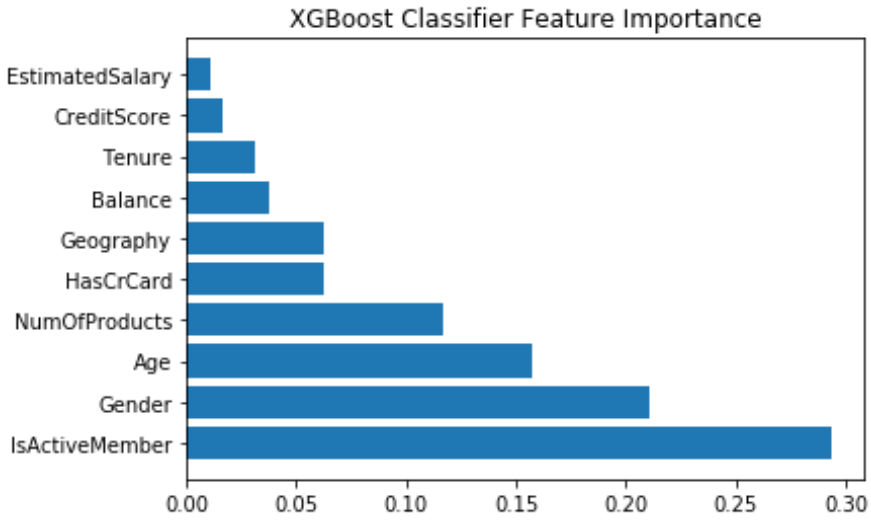

圖 4.7 XGBoost’s with SMOTE Feature importance

4.3.4 SMOTE 比較

表 4.16 比較

Accuracy			
		訓練集	測試集
Label encoding	CatBoost	0.88325	0.824
	Random Forest	0.999875	0.808
	XGBoost	0.82275	0.7995

F1 Score			
		訓練集	測試集
Label encoding	CatBoost	0.73021	0.611479
	Random Forest	0.999694	0.578947
	XGBoost	0.61676	0.58956

將上述章節彙整成表 4.16，與表 4.9 做比較可以發現進行 SMOTE 的結果比原先單一模型還要差，不論在 Accuracy 及 F1 Score 的都比單一模型還要低。而在三個模型的比較，CatBoost 的預測能力最佳與單一模型的結果一樣，Random

Forest 與 XGBoost 的預測能力差不多。

4.4 投票法

投票法(Voting)是一種常用多個分類模型，於以下是基於 4.2 單一模型的模型結果進行投票，投票的方法為三個模型中只要有兩個以上的模型預測為流失就認定為流失，並計算混淆矩陣如表 4.17，及訓練集和測試集的 Accuracy、F1 score 如表 4.18。

表 4.17 混淆矩陣

訓練集	真實		
預測		流失	留下
	流失	864	123
	留下	768	6245

測試集	真實		
預測		流失	留下
	流失	202	61
	留下	203	1534

表 4.18 訓練集和測試集之 Accuracy、F1 score 比較

	訓練集	測試集
Accuracy	0.888625	0.868
F1 Score	0.659794	0.60479

投票的方法其結果在 Accuracy 與 Catboost 一樣高，但在 f1 score 方面並沒有提高。

4.5 產生新特徵

4.5.1 方法介紹

基於 Facebook 在 2014 年的論文「ractical Lessons from Predicting Clicks on Ads at Facebook」介紹了通過 GBDT+LR 的方案，在這篇論文中他們提到了一種將 Xgboost 作為 feature transform 的方法，不過在這裡，我們採用

Random forest 後面接上 CatBoost 以及 Xgboost 後面接上 CatBoost 兩種方法。

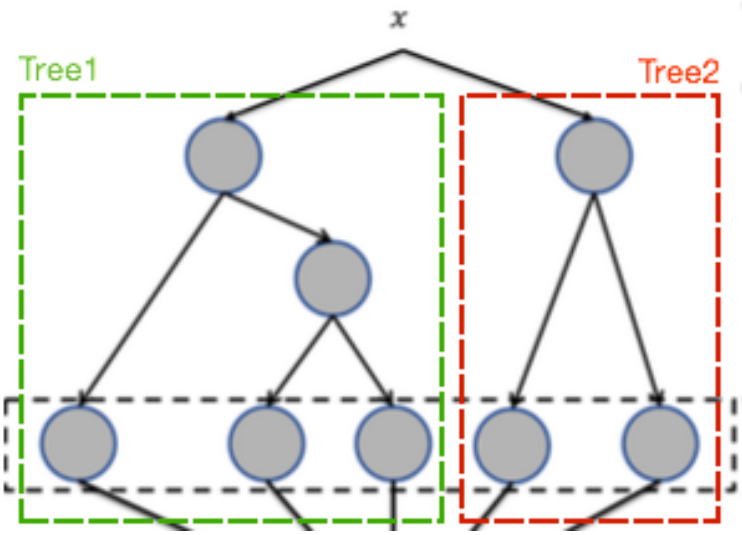

圖 4.8 產生新特徵介紹圖

舉例來說，上圖中第一棵樹有 3 個葉子節點，第二棵樹有 2 個節點，當我們輸入一個樣本點 x，若 x 在第一棵樹落在第三個葉子節點，而在第二棵樹落在第一個葉子節點，那麼通過此方法獲得的新特徵向量為[3,1]，其中向量的 3 對應第一棵樹的 3 個葉子節點，1 則對應第二棵樹的 2 個葉子節點。

利用上述的方法，我們將 Random forest 產出的新特徵與原有的特徵合併後，套入 CatBoost 的模型，看看產生的新特徵有沒有使模型的準確率提升。XGBoost 也同樣的重複以上步驟，流程圖如下圖。使用這種方式，我們將 Random forest 以及 XGBoost 的樹固定為 10 棵，一方面使資料維度不巨量增加，一方面又要有足夠的資訊，但這個數字是可以變動的，也許 100 棵的效益更高，但在本研究，我們就只是探討這個方法對於我們的分類結果有沒有增加效益，所以對於這個數字我們並沒有更深入的探討。

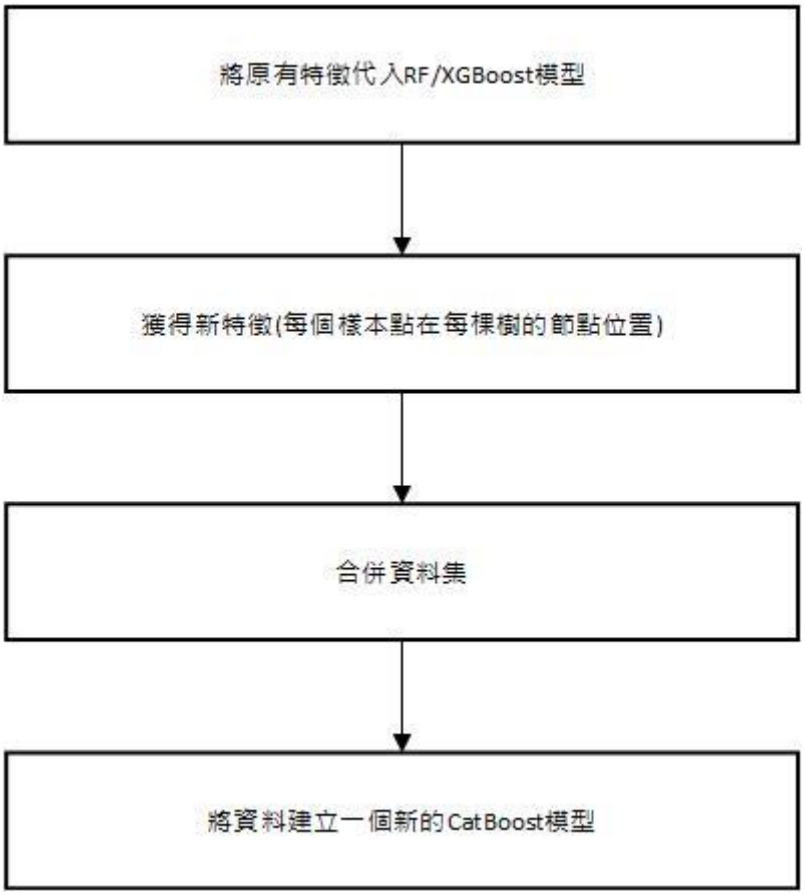

圖 4.9 產生新特徵流程圖

4.5.2 Random Forest + CatBoost

表 4.19 訓練集和測試集之 Accuracy、F1 score 比較

	訓練集	測試集
Accuracy	0.951875	0.8605
F1 Score	0.871452	0.603129

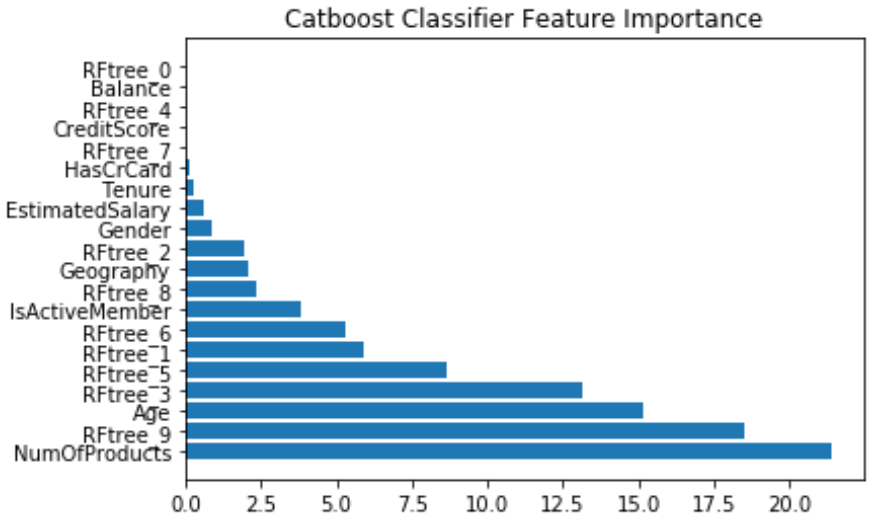

圖 4.10 Random Forest + CatBoost Feature importance

從 Feature importance 可以看到第 9 棵樹對於 CatBoost 的分類的重要性很高，可惜的是加入 Random forest 的節點資訊，對於模型的分類準確度沒有提升。

4.5.3 XGBoost + CatBoost

表 4.20 訓練集和測試集之 Accuracy、F1 score 比較

	訓練集	測試集
Accuracy	0.876875	0.8705
F1 Score	0.628722	0.630528

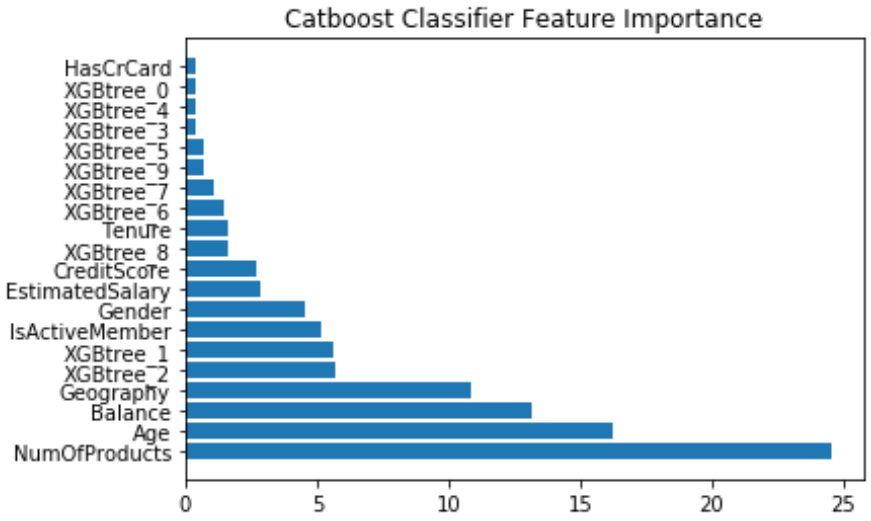

圖 4.11 XGBoost + CatBoost Feature importance

從 Feature importance 中也能看到 XGBoost 的樹對於 CatBoost 也有所貢獻，也能從 Accuracy 和 F1 score 看到加入 XGBoost 產生的新特徵，能對於我們的模型有些許的改善，也是本研究目前為止最好的結果。

4.6 模型 overfitting

雖然在本研究當中，所有的模型都使用預設的參數，但我們還是畫出下圖，來解釋模型 overfitting 的問題，在下圖當中，使用的是 CatBoost 的模型。

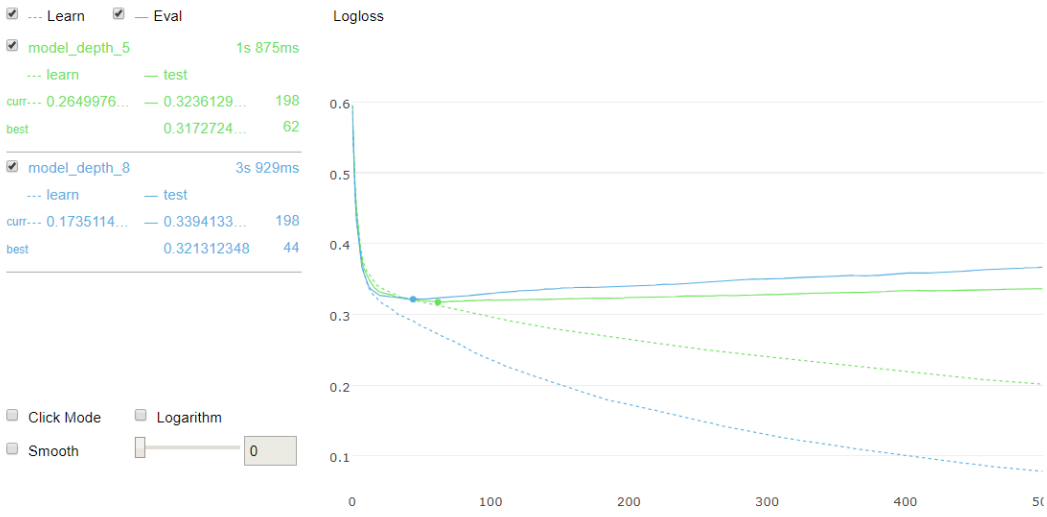

圖 4.12 CatBoost 之 Logloss

綠色為樹的深度為 5，藍色則為 8。虛線代表的是訓練集，實線則為測試集。可以看到在迭代 500 次時，depth = 8 的 loss function 在 train 雖然很小，但在 test 中卻比 depth = 5 的模型還來的大，這張圖充分的表現模型過度擬合的問題。

五、 結論

在本研究中，我們拿銀行客戶流失的資料，運用了上課所學的 Random forest 以及 XGBoost 還有我們另外查詢到的 CatBoost 三種模型去建模，對於這種不平衡的資料，這些以樹為基底的模型其實已經幫我們處理了這種不平衡的資料還能有一定的準確度，雖然我們在這次的研究中，對於 feature selection 的方法都沒有使用，但我們嘗試了在處理不平衡資料當中 Resampling 的方法以及利用 RF 和 XGBoost 加入新特徵的方法，在本研究當中，我們學習到了很多我們沒有嘗試過的方法。接下來，我們可以嘗試用調整模型參數、做其他 Resampling 的方法或是做一些 feature selection 的方法來嘗試提高我們模型的準確度。

六、 工作分配

簡介	謝瑞璟
資料概述分析	謝瑞璟
研究方法	蔡雨築
資料分析	潘縉緯、劉維仁、楊韻玄
結論	潘縉緯

七、 資料來源

https://www.kaggle.com/kmalit/bank-customer-churn-prediction/data