



國立臺灣科技大學  
資訊工程系

---

## 碩士學位論文

基於機器學習模型於新進玩家流失預測

New Player Churn Prediction Based On Machine Learning  
Models

研 究 生：柯名鴻

學 號：M10915Q05

指導教授：戴文凱博士

中華民國一一年八月十五日

# 中 文 摘 要

近年來，手機遊戲商大多以免費遊戲為主。但是，免費遊戲在定義玩家是否流失上極為困難，遊戲商容易失去挽回玩家的機會。此外，新進玩家的流失率較高。因此，如果能準確地預測出可能流失的新進玩家，就能即時的採取挽留策略，希望能因此提高新進玩家的留存率，進而有效地提高營收。

本論文對此議題提出一巨量資料探勘框架，此框架將由五大階段組成：(1) 資料前處理階段、(2) 資料分析階段、(3) 機器學習階段、(4) 預測結果分析階段及 (5) 產業應用分析階段。將需先將資料進行前處理以及預測前之資料分析，隨後進行機器學習的訓練，再將預測之結果導入資料特徵重要性分析之中，最後用代理人模型找出玩家流失的規則，完成整體預測與分析之工作。

關鍵字：新進玩家、流失預測、免費遊玩遊戲、機器學習

# ABSTRACT

In recent years, most mobile game developers focus on free-to-play (F2P) games. However, it is extremely difficult for F2P games to define whether players are churning, which makes game operators hard to retain players. In addition, new players have a higher churn rate. Therefore, if operators can accurately predict the new players that may churn, they can immediately adopt a retention strategy to lost players, looking forward that this will increase the retention rate of new players and thus effectively increase revenue.

This thesis proposes a machine learning framework for this topic, which is composed of five major stages: (1) pre-processing stage, (2) analysis stage, (3) machine learning stage, (4) feature extraction analysis stage and (5) business application analysis stage. To be specific, the framework will need to process the data and data analysis before prediction, and train the model via machine learning algorithm, then analyze important features with the prediction results. Finally, by using surrogate model, the framework will be able to find out the patterns of player churn.

Keywords: New Player, Churn Prediction, Free-to-Play, Machine Learning

# 誌謝

非常感謝指導教授戴文凱博士，在我的碩士生涯中給予我許多的指導，在討論研究時，老師都會適時地指出不足之處；在撰寫論文時，老師也都會幫忙審查，並時常會叮嚀我注意小細節，讓我能順利的完成本論文，再次感謝老師的教導與協助。

另外，特別感謝張國清博士，時常在我研究時給予我建議，並指正我錯誤的部分，讓我的實驗結果有不錯的表現；在我於公司實習時，也都會額外花時間教學與照顧，讓我額外學到許多與產業應用相關的部份，再次感謝張博士的鼎力相助。

最後感謝一路上陪伴我求學的家人們，在我感到壓力時，能給我許多溫暖與依靠；在我遇到困難時，能給我極大的鼓勵與支持，讓我能夠持續堅持下去，且不顧一切地繼續往前走，真的很感謝我的家人們。

# 目 錄

中文摘要 . . . . .	I
ABSTRACT . . . . .	II
誌謝 . . . . .	III
目錄 . . . . .	IV
圖目錄 . . . . .	VII
表目錄 . . . . .	IX
符號說明 . . . . .	X
1 緒論 . . . . .	1
1.1 研究背景與動機 . . . . .	1
1.2 研究目標 . . . . .	1
1.3 研究方法概述 . . . . .	2
1.4 研究貢獻 . . . . .	3
1.5 本論文之章節結構 . . . . .	3
2 文獻探討 . . . . .	4
2.1 免費遊戲興起 . . . . .	4
2.2 資料前處理 . . . . .	4
2.3 學習模型選擇 . . . . .	5
2.4 資料不平衡處理及其評估方式 . . . . .	6
3 研究方法 . . . . .	8
3.1 資料前處理階段 . . . . .	9
3.1.1 整合資料 . . . . .	9

3.1.2	資料過濾 . . . . .	11
3.1.2.1	刪除空缺值 . . . . .	11
3.1.2.2	無價值玩家資料處理 . . . . .	12
3.1.3	目標值準備 . . . . .	13
3.1.4	建立資料特徵 . . . . .	14
3.2	資料分析階段 . . . . .	15
3.2.1	探索性資料分析 . . . . .	15
3.2.1.1	高資訊量之資料特徵 . . . . .	15
3.3	機器學習階段 . . . . .	17
3.3.1	分割訓練與測試資料集 . . . . .	17
3.3.2	學習模型選擇 . . . . .	17
3.3.3	不平衡資料權重調整 . . . . .	18
3.3.4	搜尋最佳參數解 . . . . .	18
3.3.4.1	模型參數 . . . . .	19
3.3.4.2	網格搜索與交叉驗證 . . . . .	19
3.3.5	評估驗證最佳模型 . . . . .	20
3.4	預測結果分析階段 . . . . .	23
3.4.1	資料特徵重要性分析 . . . . .	24
3.5	產業應用分析階段 . . . . .	25
3.5.1	代理人模型 . . . . .	25
4	實驗結果與分析 . . . . .	26
4.1	實驗系統架構 . . . . .	26

4.2	資料前處理評估 . . . . .	27
4.2.1	預測受眾評估 . . . . .	27
4.2.2	預測特徵評估 . . . . .	28
4.3	資料分析評估 . . . . .	31
4.3.1	探索性資料分析評估 . . . . .	31
4.4	機器學習評估 . . . . .	34
4.4.1	分割訓練與測試資料集評估 . . . . .	34
4.4.2	資料不平衡處理評估 . . . . .	34
4.4.3	最佳模型評估 . . . . .	37
4.4.4	時間框架評估 . . . . .	38
4.5	預測結果分析評估 . . . . .	40
4.5.1	資料特徵重要性評估 . . . . .	40
4.6	產業應用分析評估 . . . . .	42
4.6.1	代理人模型評估 . . . . .	42
5	結論與未來研究 . . . . .	45
5.1	結論 . . . . .	45
5.2	未來研究 . . . . .	45
	參考文獻 . . . . .	46

# 圖 目 錄

圖 2.1	裝袋算法方式建樹示意圖 . . . . .	6
圖 2.2	提升方法方式建樹示意圖 . . . . .	6
圖 3.1	本論文之巨量資料探勘框架示意圖 . . . . .	9
圖 3.2	資料庫群內資料之示意圖 . . . . .	10
圖 3.3	依各項遊戲為整合目標之示意圖 . . . . .	10
圖 3.4	原始資料集示意圖 . . . . .	10
圖 3.5	刪除空缺值示意圖 . . . . .	11
圖 3.6	判別有價值與無價值玩家之示意圖 . . . . .	12
圖 3.7	非流失玩家與流失玩家之示意圖 . . . . .	13
圖 3.8	流失玩家與非流失玩家范氏圖 . . . . .	14
圖 3.9	建立資料特徵示意圖 . . . . .	14
圖 3.10	探索性資料分析之使用圖表類型 . . . . .	15
圖 3.11	高資訊量之資料特徵示意圖 . . . . .	16
圖 3.12	低資訊量之資料特徵示意圖 . . . . .	16
圖 3.13	分割訓練與測試資料集示意圖 . . . . .	17
圖 3.14	RepeatedStratifiedKFold 示意圖 . . . . .	20
圖 3.15	ROC Curve 示意圖 . . . . .	21
圖 3.16	PR Curve 示意圖 . . . . .	22
圖 3.17	不平衡資料中 ROC Curve 失準示意圖 . . . . .	22
圖 3.18	代理人模型示意圖 . . . . .	25



圖 4.1	新進玩家之流失長條圖	28
圖 4.2	作業系統 One Hot Encoding	29
圖 4.3	第一層特徵變數	30
圖 4.4	第二層特徵變數	31
圖 4.5	觀察期間玩家登入天數之人數統計圖	32
圖 4.6	觀察期間玩家完成任務之次數統計圖	32
圖 4.7	玩家創帳號後第三天之老虎機有贏分平均次數統計圖	33
圖 4.8	作業系統各玩家統計圖	33
圖 4.9	不平衡資料處理前後比較之 ROC Curve 圖	35
圖 4.10	不平衡資料處理前後比較之 PR Curve 圖	36
圖 4.11	三種學習模型之曲線比較圖	37
圖 4.12	決策樹資料特徵重要性比較圖	40
圖 4.13	隨機森林資料特徵重要性比較圖	41
圖 4.14	極限梯度提升資料特徵重要性比較圖	41
圖 4.15	代理人模型圖	44

# 表 目 錄

表 3.1	學習模型參數調校表 . . . . .	19
表 3.2	學習模型參數說明表 . . . . .	19
表 3.3	混淆矩陣 . . . . .	21
表 3.4	$\beta$ 數值意義表 . . . . .	23
表 4.1	實驗系統架構之研究環境表 . . . . .	26
表 4.2	有價值玩家觀察表 . . . . .	27
表 4.3	非流失玩家及流失玩家定義表 . . . . .	28
表 4.4	資料特徵總數表 . . . . .	29
表 4.5	訓練與測試資料集玩家數表 . . . . .	34
表 4.6	最佳模型評估表 . . . . .	37
表 4.7	最佳模型參數解表 . . . . .	38
表 4.8	參數搜尋範圍表 . . . . .	38
表 4.9	觀察期評估表 . . . . .	39
表 4.10	表現期評估表 . . . . .	39
表 4.11	特徵重要性前十名 . . . . .	42

## 符 號 說 明

$O$	觀察期天數
$R$	挽留期天數
$P$	表現期天數
$class\ 1$	付費玩家
$class\ 0$	非付費玩家
$X$	訓練資料集占比
$Y$	測試資料集占比
$N_1$	流失玩家樣本數
$N_0$	非流失玩家樣本數
$G()$	<i>Gini Impurity</i>
$GI()$	<i>Gini Importance</i>
$fi()$	資料特徵重要性 ( <i>Feature Importance</i> )
$D$	某節點
$D_p$	父節點
$D_{left}$	左子節點
$D_{right}$	右子節點
$N_p$	父節點樣本數
$N_{left}$	左子節點樣本數
$N_{right}$	右子節點樣本數
$x$	欲求其重要性之資料特徵
$k$	節點分割時所用資料特徵為 $x$ 之所有節點

$l$	樹中所有節點
$t$	學習模型中的所有樹

# 第 1 章 緒論

## 1.1 研究背景與動機

對於許多遊戲商而言，準確預測玩家流失對於長期成功至關重要。近年來，手機遊戲商大多以免費遊戲為主，不再是以往的買斷制或月費制。在免費遊戲的模式之下，遊戲商之營收有非常顯著的成長，例如：「絕地要塞 2」(Team Fortress 2) 原本為買斷制型式遊戲，新收入的來源僅限於沒購買過遊戲的人，這樣的商業模式與他們的發展策略並不完全吻合。於是，在 2011 年改為免費遊戲，除了擴大玩家受眾，還透過遊戲不斷地更新保持玩家的興趣。最終，遊戲營收提高達 12 倍之多 [1]。但是，免費遊戲在定義是否為流失玩家上極為困難，一旦玩家想離開了隨時都能停止遊玩，沒有必要告訴遊戲商其決定，遊戲商也容易因此失去挽回玩家的機會。

根據哈佛商業評論 [2] 中指出，企業爭取一個新顧客的成本是保留老顧客成本的 5 倍。一個公司如果能將其顧客流失率降低 5%，其利潤就能增加 25% - 85%。也就是說，獲得一個新玩家的成本比留住一個玩家要昂貴許多。因此，留住玩家成為重要議題，精準地預測玩家流失，即使是微小地提升，也可能導致營收有顯著的提升。

另外，根據網站 Swrve [3] 於 2014 年提供的報告指出，數十款的遊戲中，有 19.3% 的新玩家只玩一次特定遊戲，新玩家的次日留存率為 33.9%，而第 30 天的留存率只剩下 5.5%。而 Mustač 等人 [4] 則是針對歐洲一款休閒遊戲進行研究，研究中可以看到，此遊戲有大約 60% 的玩家在玩了一天後就離開了，3 天後，玩家們只剩下大約 20%。上述的 2 份報告都清楚表明了新進玩家的流失率是很嚴重的問題。因此，本文將針對新進玩家做流失預測分析，希望能因此提高新進玩家的留存率，進而有效地增加營收。

## 1.2 研究目標

本論文的研究目標是預測免費遊戲的新進玩家是否會流失。透過新進玩家在遊戲初期的遊玩歷程和儲值紀錄等特徵訓練出準確率最高的模型，並藉由模型的

結果分析各個特徵的突出性，來進行玩家流失原因的解釋說明，甚至能從中了解玩家們的喜好、趨勢，也可以讓市場操作人員有挽留玩家的操作依據，有了明確的挽留策略就能夠對症下藥，藉此來強化留存，並進一步增加營收。

## 1.3 研究方法概述

本論文將新進玩家創帳號後的天數切分為三個時期：(1) 觀察期：玩家創帳號後前幾天，會將此時期的玩家行為軌跡作為資料特徵來訓練模型；(2) 挽留期：於觀察期之後，作為給市場操作人員實施挽留策略的時間；(3) 表現期：於挽留期之後，決定玩家是否流失，如果玩家在此時期有任一登入紀錄則視為非流失玩家，反之將視為流失玩家。

此外，本文提出一巨量資料探勘框架：此框架將由五大階段組成，(1) 資料前處理階段：首先從資料庫群中整合所有需要的資料，並過濾掉無價值玩家，再著手目標值準備並建立資料特徵，以利後續分析及機器學習使用；(2) 資料分析階段：使用前階段產出之資料，透過統計圖表來觀察資料特性，進行探索性資料分析 ( Exploratory Data Analysis ) [5]，藉由流失玩家與非流失玩家的資料分佈，來觀察資料特徵是否可以提供給學習模型較多之資訊；(3) 機器學習階段：首先將處理後的資料集分割為訓練集及測試集，隨後針對訓練集進行多數群樣本權重值縮小，並透過網格搜索 ( Grid Search ) [6] 與交叉驗證 ( Cross Validation ) [7] 找出機器學習模型的最佳超參數，以獲得最佳模型，其中學習模型選用決策樹 ( Decision Tree ) [8]、隨機森林 ( Random Forest ) [9] 及極限梯度提升 ( Extreme Gradient Boosting ) [10]，最後藉由測試集來驗證評估最佳模型，產出預測結果；(4) 預測結果分析階段：使用前階段產出之預測結果進行資料特徵重要性分析，透過計算各資料特徵於各學習模型中之基尼重要性 ( Gini Importance ) [11]，以利更加了解及解釋資料特徵與遊戲所提供之體驗綜合評估；(5) 產業應用分析階段：參考代理人模型 ( Surrogate Model ) [12]，用較簡單的模型來模擬較複雜的模型，能協助了解流失玩家的行為規則，作為市場操作人員的操作依據。

在方法驗證上，本論文將藉由混淆矩陣 ( Confusion Matrix ) 所延伸之 ROC Curve ( Receiver Operating Characteristic Curve ) [13] 與 PR Curve ( Precision-Recall Curve ) [14] 來協助驗證學習模型之優劣，並同時利用 Weighted  $F_\beta$  - Score [15] 來選出最佳模型與最佳參數解，隨後計算特徵重要性 ( Feature Importance ) 於各資料特徵中，以了解到何者於學習模型中貢獻了最多的資訊量，以利學習模型進行訓

練與分類。

## 1.4 研究貢獻

本論文之研究貢獻為：

1. 訓練一新進玩家流失預測模型並運用於遊戲領域，利用其預測結果從中了解玩家流失的原因與動機，並作為市場操作人員後續挽留玩家的操作依據，以強化留存並進一步增加營收。
2. 提出一建立資料特徵的方法：對資料集以統計手法建立資料特徵，並用多個時間框架做拆分，以獲得第一層特徵變數，再對第一層特徵變數做計算，進一步來得到第二層特徵變數，如變化量特徵等。
3. 於資料集中進行資料特徵之探勘，藉由不同種類與面向之方式，挑選出適合用來呈現流失玩家的資料。
4. 整理出適合於不平衡資料集中的評估值方式，將對於學習模型之預測結果提供合理的評估，進而進行比較。
5. 整理出資料特徵重要性之計算，以利分析資料特徵的突出性與其貢獻的資訊量。
6. 透過代理人模型了解流失玩家的行為規則，能做為市場操作人員的操作依據，方便產業應用。

## 1.5 本論文之章節結構

本論文第 2 章為文獻探討，探討關於資料前處理、學習模型選擇、資料不平衡處理以及其評估方式的相關文獻。第 3 章為研究方法，詳述講述各階段之研究流程，並分為五節來介紹資料前處理、資料分析、機器學習、預測結果分析及產業應用分析。第 4 章為實驗結果與分析，分為六節：第 4.1 節實驗系統架構；第 4.2 節資料前處理評估；第 4.3 節資料分析評估；第 4.4 節機器學習評估；第 4.5 節預測結果分析評估；第 4.6 節產業應用分析評估。第 5 章為結論與未來研究，總結本論文提出的方法與實驗結果，並討論未來的研究方向。

## 第 2 章 文獻探討

本章節針對免費遊戲興起介紹，並探討關於資料前處理、學習模型選擇、資料不平衡處理以及其評估方式的相關文獻。

### 2.1 免費遊戲興起

免費遊戲，是一種玩家無需支付任何費用，即可遊玩該遊戲之大部分內容，與付費型遊戲（買斷制或月費制）形成對比。在免費遊戲中，遊戲商可以藉由遊戲內購買或遊戲內置入廣告等方式來賺取營收 [16]。近幾年內，遊戲商皆轉以開發免費遊戲為主，因其類型所帶來之營收，已遠大於付費型遊戲 [17]。另外，免費遊戲還能夠有效的讓玩家流失量降低，透過其無需支付任何費用就能遊玩遊戲的特性，使玩家進入遊戲的門檻大為降低 [18]。

雖然免費遊戲能讓玩家流失量降低，但相對的，對於玩家是否流失變得極難定義，因為玩家可以在沒有任何通知的情況下停止遊戲，沒有明確的流失事件，例如玩家取消訂閱。儘管玩家流失模型在商業領域已經存在了幾十年，但隨著機器學習方法近年來的進步，它們的複雜性和準確性也都在提升，因此，即使在高維數據上，也可以應用極限梯度提升等機器學習來創建非常準確的模型 [19]。

### 2.2 資料前處理

在將巨量資料應用於機器學習前，資料的前處理也是極為重要，相較於在學習模型上進行深入研究與改進，透過資料特徵之轉化及選擇顯得更為重要且有效 [4] [17] [19]。

首先將對資料進行清理，只收集有價值之資料，例如：Tamassia 等人只收集遊玩時間超過給定門檻之玩家 [20]、Periáñez 等人只收集消費金額遠高於一般玩家者 [21] 或 Runge 等人只取付費玩家中前 10 % 者 [22]，上述之清理方式皆只著重於具有高資訊量的資料，而不將無價值的資料放入機器學習中。

而針對資料特徵之探勘，Sifa 等人 [23] 探勘玩家基本資料及玩家行為，再將



其進行轉化，例如：取平均值與偏差值於玩家遊玩時間、將玩家國籍分類等。Xie 等人 [24] 使用遊戲內的事件發生頻率來預測玩家對遊戲的參與度。Lee 等人 [25] 探勘玩家行為、玩家購買商品數量、玩家遊戲內交易及玩家遊戲內社交，主要針對玩家每日於遊戲內的行為軌跡。Gregory [19] 透過相對時間及絕對時間建立新的特徵，以提升資料集的特徵數量。Hadiji 等人 [26] 探勘玩家消費商品數量、玩家遊玩天數等。

## 2.3 學習模型選擇

在機器學習中，於分類預測的應用上，樹狀結構之學習模型最為主流且有效，因其建樹之方式，可以清楚的解釋該筆樣本之預測路徑，進而針對各資料特徵進行重要性的計算與分析 [17]。另外，在樹狀結構之學習模型中，主流以裝袋算法 ( Bagging ) [27] 及提升方法 ( Boosting ) [28] 兩種建樹想法為準：

- 裝袋算法：從訓練資料集中，隨機取樣並訓練成多份分類器，而每次訓練時會將資料取出後放回，並再次抽取，最後的預測結果將由多個分類器投票選出，採多數決，且各分類器間的權重關係皆為相等，如圖 2.1。例如：隨機森林即為裝袋算法 + 決策樹。
- 提升方法：從訓練資料集中，每次訓練使用相同資料，而第  $n$  個分類器於訓練時，將針對第  $n-1$  個分類器分類錯誤的資料增大其權重值，以修正分錯的資訊，希望將分錯的資料減少，預測結果將由多個分類器投票選出，各分類器間的權重關係不同，錯誤率越低的分類器，擁有越高的權重，如圖 2.2。例如：極限梯度提升即為提升方法 + 決策樹。

從上述得知，選用樹狀結構之學習模型將有助於預測分類問題。因此，為求本論文之預測新進玩家流失能夠達到預期，將採用決策樹、隨機森林與極限梯度提升來驗證樹狀結構之優勢。

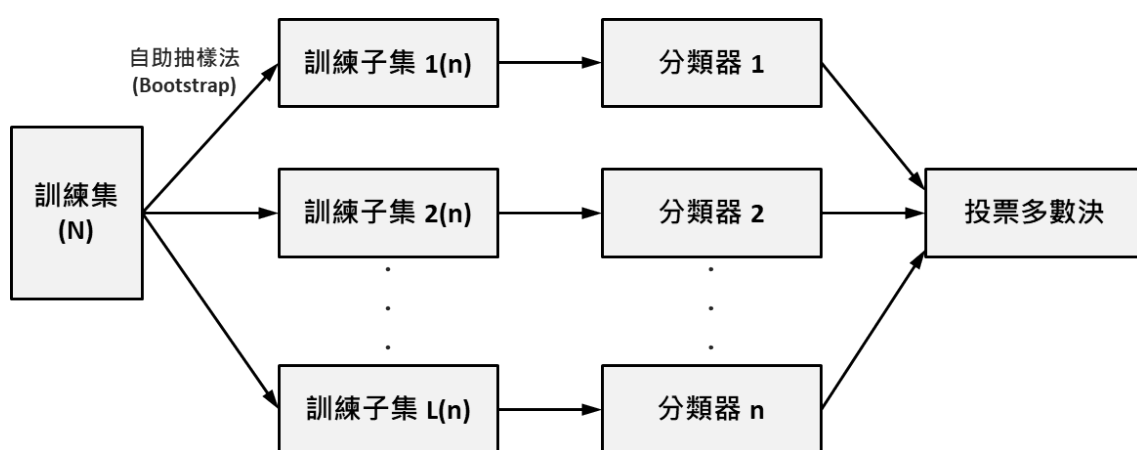


圖 2.1: 裝袋算法方式建樹示意圖

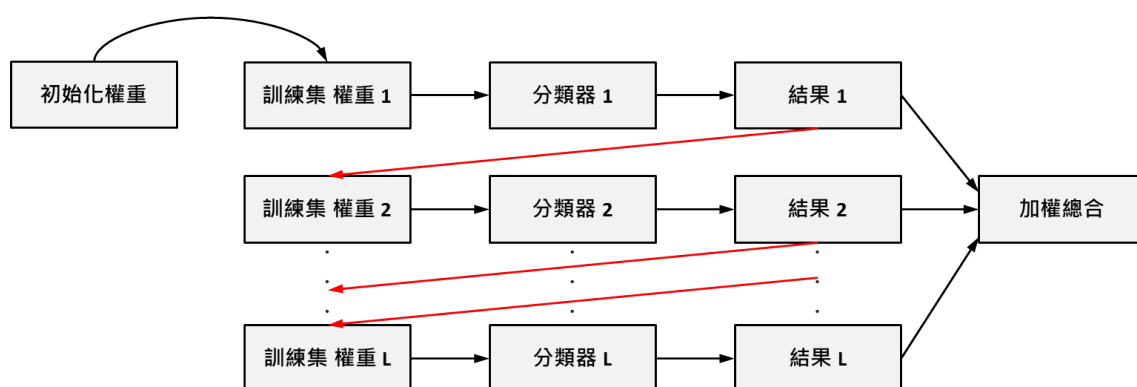


圖 2.2: 提升方法方式建樹示意圖

## 2.4 資料不平衡處理及其評估方式

在遊戲領域進行機器學習訓練時，往往會遭受到資料不平衡的影響；例如：於預測是否付費上，非付費玩家會遠多於付費玩家，導致付費玩家資料過少 [23]。於預測是否流失上，流失玩家會遠多於非流失玩家，導致非流失玩家資料過少 [25]，前述研究都採以針對資料集進行處理的方式解決資料不平衡，例如：SMOTE (Synthetic Minority Over-sampling Technique)，於少數群添加模擬資料，使得少數群之樣本數與多數群相等 [29]。而本論文不只預測玩家是否會流失，還需分析其原因，如在資料集中填入模擬資料，將會使得分析失準，無法得到有效的資訊，所以我們將採用在機器學習訓練時，放大少數群之樣本權重值，使得學習模型更加著重於少數群的資訊，如同提升方法建樹時，藉由權重值的不同，修正分類錯誤的資訊 [28]。

在評估資料不平衡資料集時，如果單純計算學習模型之精確率 ( Precision )、召回率 ( Recall ) 或 F - Score [30]，將導致多數群之評估結果壓過少數群之評估結果，使得最終評估失真，無法有效驗證學習模型之成效。因此，在評估不平衡資料時，Sifa 等人額外運用幾何平均數 ( Geometric Mean ) [31] 來評估學習模型之成效 [23]。藉由上述的概念，本論文將採用 Weighted  $F_{\beta}$  - Score 來評估不平衡資料，使得少數群之評估不被多數群所壓過，使用樣本間的數量權重差來計算多數群與少數群的  $F_{\beta}$  - Score，希望能夠合理的評估學習模型間的表現。

### 第 3 章 研究方法

針對遊戲領域巨量資料進行新進玩家流失預測，先將資料進行前處理以及預測前之資料分析，隨後訓練機器學習與其最佳化處理，最後再依預測之結果導入資料特徵重要性分析之中，完成整體預測與分析之工作。

本論文提出一巨量資料探勘框架，圖 3.1 為巨量資料探勘框架示意圖，此框架將由五大階段組成：

- 資料前處理階段：首先將從資料庫群中整合所有所需資料，並過濾出有價值之原始資料，再著手目標值準備並建立資料特徵，以利後續分析及機器學習使用。
- 資料分析階段：使用前階段產出之有價值原始資料進行探索性資料分析，透過統計圖表來觀察資料特性，藉由流失玩家與非流失玩家的資料分佈，來觀察資料特徵是否可以提供給學習模型較多之資訊。
- 機器學習階段：首先將有價值原始資料集進行分割為訓練及測試集，隨後針對訓練集進行網格搜索與交叉驗證，並搭配參數表，以獲得模型的最佳參數解，最後藉由測試集來驗證評估最佳模型，產出預測結果。
- 預測結果分析階段：使用前階段產出之預測結果進行資料特徵重要性分析，以利更加了解及解釋資料特徵與遊戲所提供之體驗綜合評估。
- 產業應用分析階段：參考代理人模型，用較簡單的模型來模擬較複雜的模型，能協助了解流失玩家的行為規則，作為市場操作人員的操作依據。

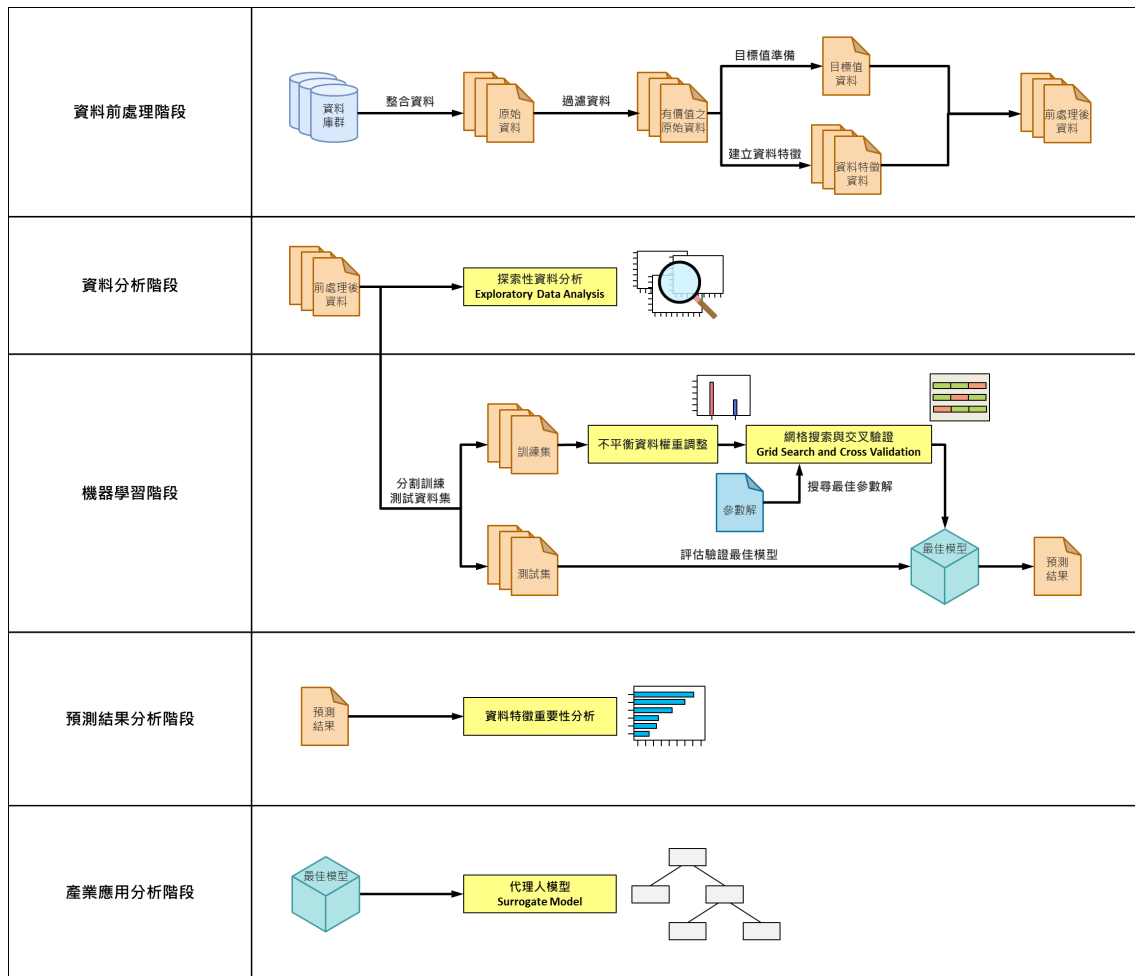


圖 3.1: 本論文之巨量資料探勘框架示意圖

## 3.1 資料前處理階段

此階段將著重於資料之整合與過濾，為求能收集到有價值之原始資料，以提高後續分析研究之價值，同時進行目標值的準備，並建立資料特徵，協助機器學習之訓練，目標產出有價值之玩家遊戲行為軌跡資料集。

### 3.1.1 整合資料

首先資料庫群中之資料皆以天為單位，記錄了各項遊戲之玩家行為軌跡，如圖 3.2。此步驟將依各項遊戲為整合目標，重整為多個原始資料集，每個原始資料集中，只會記錄該類遊戲之每位玩家行為軌跡，如圖 3.3，將可提升後續目標值準備及資料特徵探勘速度。

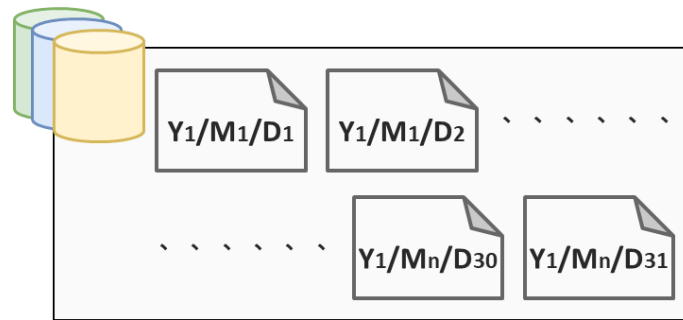


圖 3.2: 資料庫群內資料之示意圖

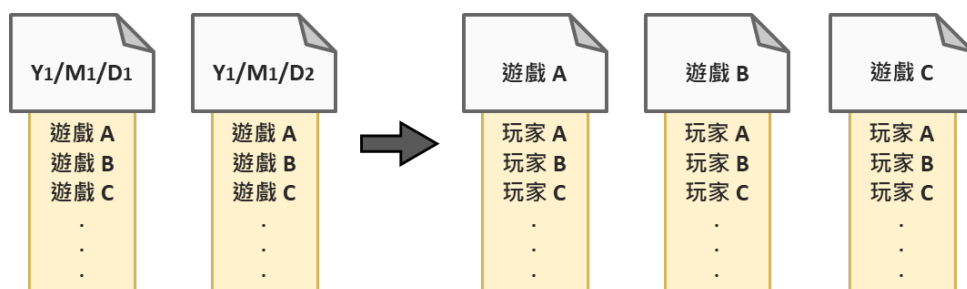


圖 3.3: 依各項遊戲為整合目標之示意圖

除了上述之玩家遊戲行為軌跡原始資料集外，還另外收集了玩家輪廓資料 (含國家、玩家等級等) 與玩家平台操作紀錄 (含消費紀錄、客訴紀錄等)，最終此步驟將產出三大類原始資料集，如圖 3.4：

- 玩家輪廓資料 (含國家、玩家等級等)
- 玩家平台操作紀錄 (含消費紀錄、客訴紀錄等)
- 玩家遊戲行為軌跡

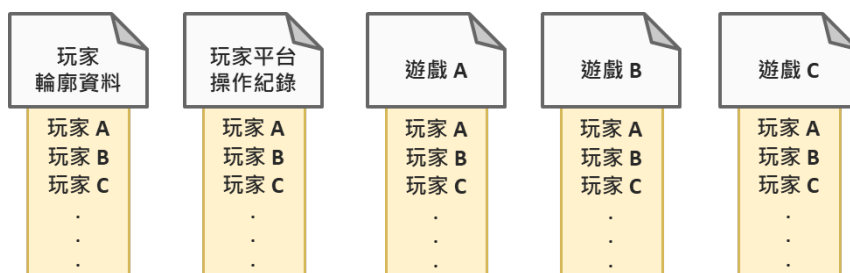


圖 3.4: 原始資料集示意圖

### 3.1.2 資料過濾

此步驟將針對兩大議題：刪除空缺值與無價值玩家資料處理。


為了要在資料分析及訓練機器學習時，能夠更加準確的了解及預測真實遊戲玩家之特性與是否流失，需要透過上述之處理，來過濾掉潛在的無用資料，使得整體研究能夠聚焦於更有價值的資料上。

#### 3.1.2.1 刪除空缺值

為了後續資料特徵重要性分析，希望能保持著資料間的真實性，將採用直接刪去具有空缺值樣本的方式，而不對資料集填入經過處理之假數值，每筆樣本只要在任意資料特徵中擁有一空缺值，即視為欲刪除之對象。

圖 3.5 為刪除空缺值之示意圖，從圖中可以看出樣本 ID 1 及 2 分別在特徵 3 及 1 擁有空缺值，將對兩者予以刪除，故最後只留下樣本 ID 0 及 3 之資料。

ID	特徵 1	特徵 2	特徵 3	特徵 4
0	12.5	198	TW	97
1	6.8	1300	-	70
2	-	1788	US	100
3	45.2	699	JP	65



ID	特徵 1	特徵 2	特徵 3	特徵 4
0	12.5	198	TW	97
3	45.2	699	JP	65

圖 3.5: 刪除空缺值示意圖

### 3.1.2.2 無價值玩家資料處理

對於遊戲領域巨量資料進行研究時，普遍會對所有玩家進行篩檢，以挑選出有價值之玩家族群，可使整體分析與預測更加貼近於真實遊戲情景。定義一時間框架於新進玩家創帳號後，又將其切分為三個時期：

- 觀察期：玩家創立帳號後前  $O$  天。觀察玩家在此時期的行為軌跡，並將其進行特徵工程，因此，也將觀察期視為資料特徵探勘期。
- 挽留期：觀察期之後前  $R$  天。作為市場操作人員實施挽留策略的時間。
- 表現期：挽留期之後前  $P$  天。決定玩家是否流失，於玩家在觀察期有登入紀錄的前提下，如果玩家在此時期有任一登入紀錄則視為非流失玩家，反之將視為流失玩家。

如果該玩家創建帳號的日子較晚，尚未完整擁有上述三個時期的資料，將會造成後續特徵提取與目標值準備的不正確性，進而影響機器學習預測的準確度，因此將其視為無價值玩家，刪除該玩家及其所有行為軌跡。

圖 3.6 為判別有價值與無價值玩家之示意圖。從圖中可以看出，玩家 1 及玩家 2 已在創帳號後完整經歷觀察期、挽留期與表現期，故視為有價值玩家；而玩家 3，因缺少完整的表現期資料，容易被誤判為流失玩家，故視為無價值玩家；玩家 4，除了表現期的資料，觀察期資料也並不完整，若將其視為有價值玩家，其錯誤的資料特徵將會影響機器學習之訓練，故也視為無價值玩家。

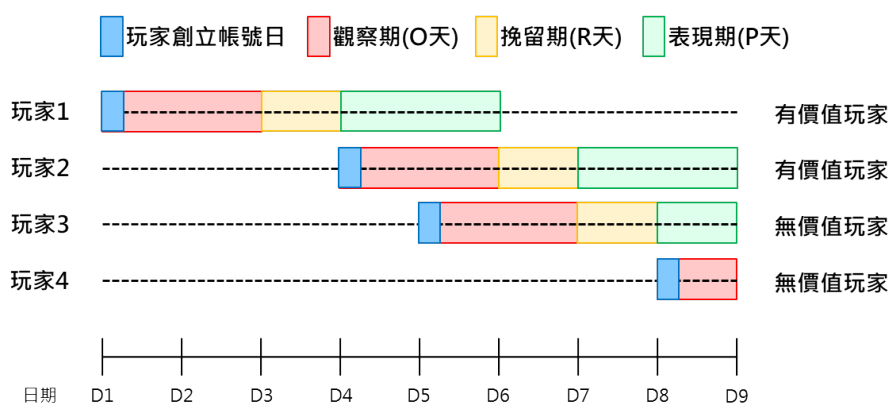


圖 3.6: 判別有價值與無價值玩家之示意圖 (以  $O$ 、 $R$ 、 $P$  分別為 2、1、2 為例)



經過前兩小節 3.1.2.1 及 3.1.2.2 之處理後，最終此步驟將產出有價值之原始資料集，提供給後續分析及訓練機器學習使用。

### 3.1.3 目標值準備

此步驟將準備供機器學習使用之目標值，即為後續預測所需之 *class*。定義目標值「非流失玩家」與「流失玩家」，分別代表 *class 0* 及 *class 1*：

- 非流失玩家 (*class 0*)：觀察期有登入紀錄的玩家中，表現期間有登入紀錄者，視為非流失玩家。
- 流失玩家 (*class 1*)：觀察期有登入紀錄的玩家扣除非流失玩家，剩餘者皆為流失玩家；表現期後才有登入紀錄者同樣視為流失玩家。

圖 3.7 為定義非流失玩家與流失玩家之示意圖。從圖中可以看出玩家 1 及玩家 2 於觀察期及表現期中皆有登入紀錄，故定義為非流失玩家 (*class 0*)；而玩家 3 及玩家 4 則在表現期中無登入紀錄，故定義為流失玩家 (*class 1*)，即使玩家 4 在表現期後有登入紀錄，依舊將其視為流失玩家 (*class 1*)。

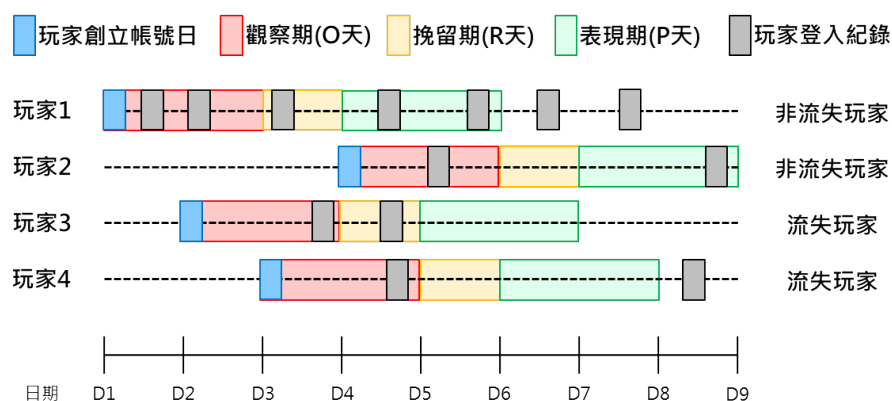


圖 3.7: 非流失玩家與流失玩家之示意圖 (以 *O*、*R*、*P* 分別為 2、1、2 為例)

本論文將預測目標聚焦於流失的新進玩家，所以透過觀察期與表現期來侷限流失玩家之定義。圖 3.8 為流失玩家與非流失玩家范氏圖 (Venn Diagram)，可以從圖中看出，最外圍之黑圓框代表所有玩家 (於 3.1.2.2 小節中，篩檢後之有價值玩家)，而藍色底之圓形範圍代表所有非流失玩家 (*class 0*)，內圈之綠圓框代表前述流失定義門檻，綠圓框內之紅色底圓型範圍則代表所有流失玩家 (*class 1*)，

其中深紅色底之圓形範圍代表表現期後無登入紀錄的玩家；淺紅色底之圓形範圍代表表現期後有登入紀錄的玩家。可以由上述說明來了解到資料內流失玩家與非流失玩家之分佈狀況及其關係。

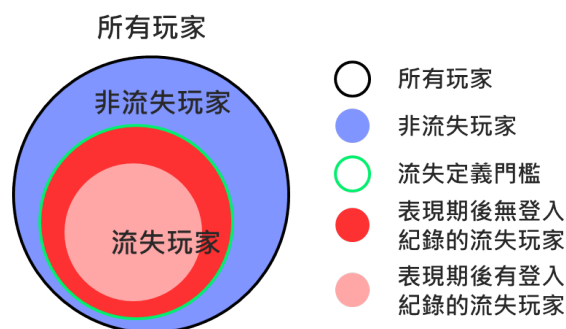


圖 3.8: 流失玩家與非流失玩家范氏圖

### 3.1.4 建立資料特徵

此步驟將探勘供機器學習使用之資料特徵。在遊戲領域巨量資料中，相較於在學習模型上進行深入研究與改進，透過資料特徵之轉化及選擇顯得更為重要且有效。所以對資料集進行不同面向之探勘，除了可以獲取更多的資訊，也能讓後續資料分析以及機器學習更加順利。

本文將觀察期作為資料探勘期，對每位玩家進行資料探勘。資料之探勘面向將參考於 [23] [25] [32] 之探勘想法，主要聚焦於玩家之行為軌跡，並用其建立資料特徵。

圖 3.9 為本文建立資料特徵之示意圖。紅色箭頭為第一層特徵變數建立方式，對資料集以多種統計方式建立資料特徵，並用多個時間框架做拆分，以獲得第一層特徵變數；藍色箭頭為第二層特徵變數建立方式，針對第一層特徵變數做計算，進一步來得到第二層特徵變數，如變化量特徵等。

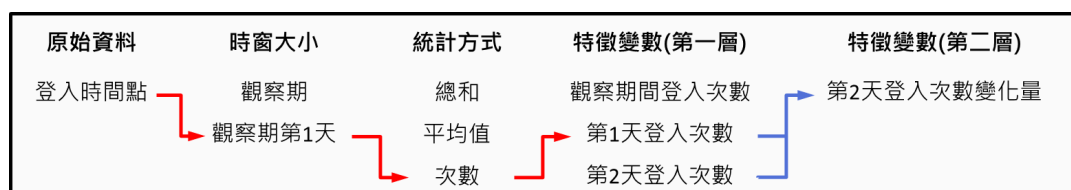


圖 3.9: 建立資料特徵示意圖

## 3.2 資料分析階段

此階段將著重於資料的分析。為求在訓練機器學習前，可以藉由資料分析之方法來了解到資料之特性，以提高後續解讀資料特徵之重要性與其相關之連結。另外，還觀察資料特徵是否可以提供給學習模型較多的資訊。

### 3.2.1 探索性資料分析

採用探索性資料分析，藉由圖表呈現協助了解玩家資料之分布情況。藉由觀察資料是集中還是平均分散，以此來推測是否為高資訊量之資料特徵。此步驟將利用下列兩種圖表來觀察資料特性：

■ 長條圖 ( Bar Chart )：描述離散性資料，如圖 3.10 (a)。

■ 直方圖 ( Histogram )：描述連續性資料，如圖 3.10 (b)。

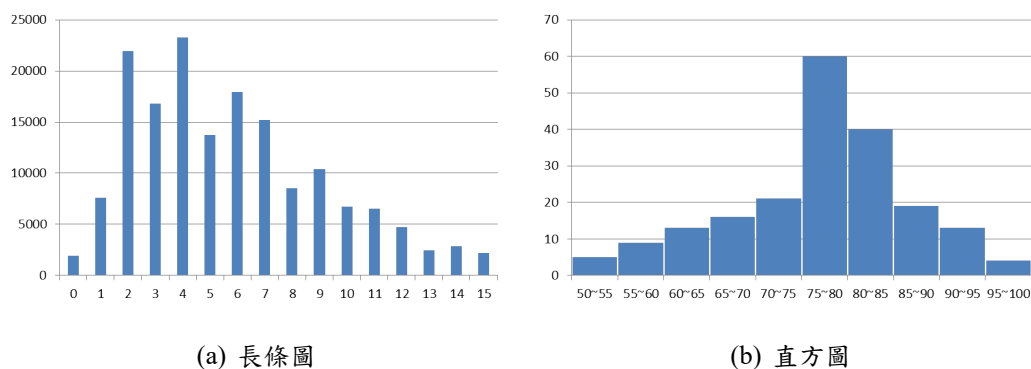


圖 3.10: 探索性資料分析之使用圖表類型

#### 3.2.1.1 高資訊量之資料特徵

藉由觀察資料特徵之分佈是否有明顯差異性，而推測此資料特徵能夠提供給學習模型較多的資訊，將此類資料特徵認為是高資訊量之資料特徵，使得後續資料特徵重要性分析之解釋可以更加順利。

假設，觀察期間玩家登入天數之人數統計如圖 3.11，假設觀察期為 4 天。可以從圖中看出，非流失玩家數會隨著登入天數增加而遞增；流失玩家數則隨著登

入天數增加而遞減，能夠容易地區分出流失玩家與非流失玩家，屬於高資訊量之資料特徵。

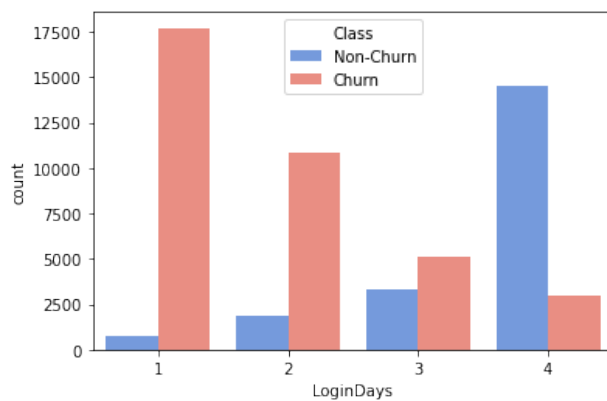


圖 3.11: 高資訊量之資料特徵示意圖

假設，觀察期間玩家登入天數之人數統計如圖 3.12，假設觀察期為 4 天。可以從圖中看出，無論是非流失玩家還是流失玩家，玩家數都隨著登入天數增加而遞增，較難以區分出流失玩家與非流失玩家，屬於低資訊量之資料特徵。

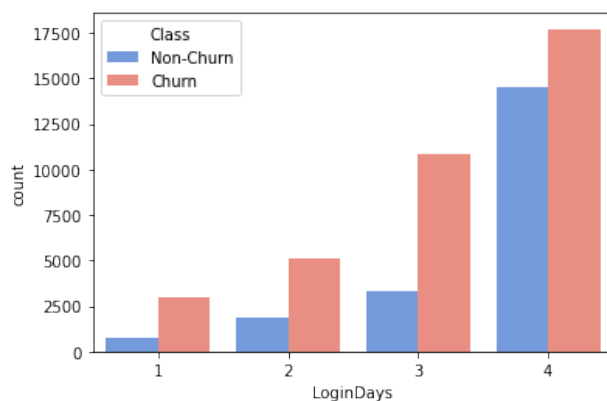


圖 3.12: 低資訊量之資料特徵示意圖

### 3.3 機器學習階段

此階段將著重於機器學習訓練以及不平衡資料權重調整，最後產出最佳模型之預測結果，能提供給後續的預測結果分析以及產業應用分析使用。本文選擇樹狀結構之學習模型進行訓練，樹狀結構之學習模型對於巨量資料分類預測顯得更為合適，並且對於預測結果之解釋也相對清楚，而本論文所挑選之學習模型包含：決策樹、隨機森林與極限梯度提升。

#### 3.3.1 分割訓練與測試資料集

透過前述 3.1.2 小節過濾後之資料集，進行訓練集與測試集分割，並按照  $X:Y$  之比例隨機分配。為了避免隨機切割時，目標類別分布不平衡，將採分類隨機抽樣，即流失玩家與非流失玩家各別以  $X:Y$  之比例隨機抽樣，如圖 3.13。

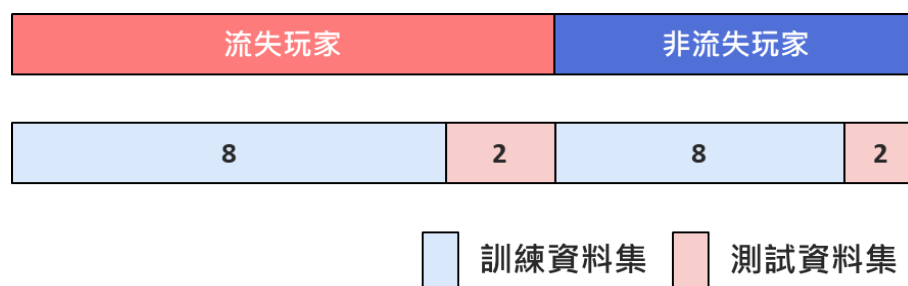


圖 3.13: 分割訓練與測試資料集示意圖 (以  $X$ 、 $Y$  分別為 8、2 為例)

#### 3.3.2 學習模型選擇

本文選擇樹狀結構的學習模型進行訓練。樹狀結構的學習模型對於巨量資料分類預測顯得更為合適，並且對於預測結果之解釋也相對清楚，是研究中常使用的學習模型 [33]，因其模型結構能夠清楚表現每筆樣本之預測路徑，可以協助了解學習模型如何做出決策，如白箱模型 (white box model)；而神經網路結構之學習模型，其模型內的結構難以解析，如黑箱模型 (black box model)，則無法進行更一步的研究。而本論文所挑選之學習模型包含：

- 決策樹：樹狀結構學習模型之最基礎結構，單樹結構，採用 CART ( Classification and Regression Tree ) 演算法進行建樹。
- 隨機森林：多樹結構，採用裝袋算法建樹。
- 極限梯度提升：多樹結構，採用提升方法建樹。

最後將比較上述三種不同學習模型來挑選出最佳之模型，包含裝袋算法與提升方法不同方式之建樹差異。

### 3.3.3 不平衡資料權重調整

進行機器學習訓練於遊戲領域巨量資料時，往往將會遭受資料不平衡之問題，進而影響學習模型之成效與可靠度。

普遍研究中將針對資料集進行預處理，設法解決資料不平衡之問題，如 SMOTE，為了讓少數群之樣本數與多數群相等，會於原始資料集中放入模擬資料。但本文為了確保資料間之真實性，於處理不平衡資料時，不希望針對資料集進行加工，會無法呈現出真實資料集之特性，所以本文將重點放於訓練學習模型時的樣本權重影響，而不對資料集進行直接處理。樣本權重設置如式 3.1，其中  $N_0$  為非流失玩家 ( class 0 ) 之樣本數； $N_1$  為流失玩家 ( class 1 ) 之樣本數。將計算  $N_0$  與  $N_1$  之比例差距，此值則為流失玩家 ( class 1 ) 樣本權重縮小倍數。

$$class\ 0 : class\ 1 = 1 : \frac{N_0}{N_1} \quad (3.1)$$

### 3.3.4 搜尋最佳參數解

此步驟將對前述 3.3.2 小節挑選之學習模型進行搜尋最佳參數解，透過網格搜索與交叉驗證，對所有參數一一進行訓練、測試，以調校出最適合該學習模型之參數。

### 3.3.4.1 模型參數

本文選擇各學習模型之調校參數如表 3.1，針對各學習模型之結構不同，挑選不同的參數進行最佳化，各參數意義說明如表 3.2。

學習模型	決策樹	隨機森林	極限梯度提升
參數調校	max_depth	n_estimators	n_estimators
	min_samples_split	max_depth	max_depth
	min_samples_leaf	min_samples_split	learning_rate
		min_samples_leaf	

表 3.1: 學習模型參數調校表

參數	參數說明
n_estimators	多樹結構之樹總數
max_depth	樹狀結構之最大深度限制
min_samples_split	節點分割之最小樣本數限制
min_samples_leaf	葉節點之最小樣本數限制
learning_rate	學習速率

表 3.2: 學習模型參數說明表

### 3.3.4.2 網格搜索與交叉驗證

網格搜索是在參數列表中進行窮舉搜索，嘗試每一種可能性，找出結果最好的組合，通常會搭配交叉驗證來評估。

針對訓練資料集進行交叉驗證，並且搭配網格搜索找出模型的最佳參數解。參考了 [34] 中所使用之 RepeatedStratifiedKFold 方法，其中使用 Stratified 方式分割，即為在各 Fold 中，流失玩家與非流失玩家之資料比例將會相等；使用 Repeated 方式反覆驗證，即為反覆執行上述之交叉驗證。透過上述之分割方式，可以在每次訓練學習模型時，使真實訓練集保持著原始訓練集的流失玩家與非流失玩家比例。

圖 3.14 為 RepeatedStratifiedKFold 示意圖。可以從圖中看出，首先依照前述 3.3.1 小節，從所有資料初步分割出原始訓練集與測試集，再針對原始訓練集進行 RepeatedStratifiedKFold，進行了兩次的交叉驗證，而每次分割原始訓練集時可以看到淺藍底之真實訓練集與淺橘底之驗證集中的流失玩家與非流失玩家比例與原始測試集相等，並且真實訓練集與驗證集中的流失玩家與非流失玩家比例也相等。

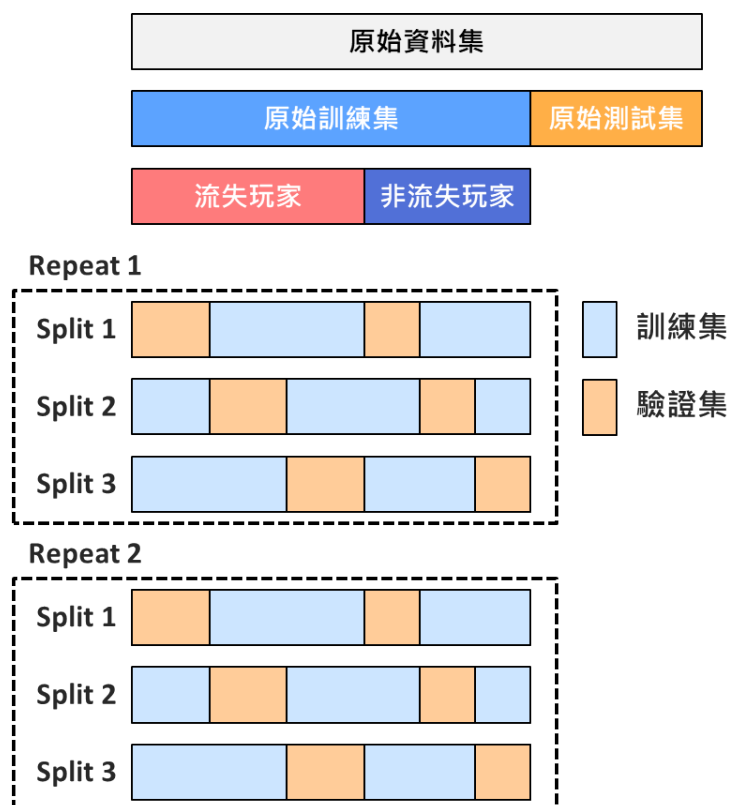


圖 3.14: RepeatedStratifiedKFold 方法示意圖 (以 Repeated、K-Fold 分別為 2、3 為例)

### 3.3.5 評估驗證最佳模型

學習模型之評估將藉由混淆矩陣 ( Confusion Matrix )、精確率 ( Precision )、召回率 ( Recall )、真陽率 ( True Positive Rate, TPR ) 及假陽率 ( False Positive Rate, FPR ) 來說明，計算方式分別如表 3.3、式 3.2、式 3.3、式 3.4 及式 3.5。並會搭配 ROC Curve 和 PR Curve 來觀察。



	Predicted <i>class 1</i>	Predicted <i>class 0</i>
Actual <i>class 1</i>	<i>True Positive ( TP )</i>	<i>False Negative ( FN )</i>
Actual <i>class 0</i>	<i>False Positive ( FP )</i>	<i>True Negative ( TN )</i>

表 3.3: 混淆矩陣

$$Precision = \frac{TP}{TP + FP} \quad (3.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.3)$$

$$True\ Positive\ Rate\ (TPR) = \frac{TP}{TP + FN} = Recall \quad (3.4)$$

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{TN + FP} \quad (3.5)$$

藉由 ROC Curve 來觀察學型模型的真陽率及假陽率。圖 3.15 為 ROC Curve 示意圖，x 軸為假陽率，y 軸為真陽率。當 x 為 0 時，表示模型在非流失玩家的預測上沒有錯誤；而當 y 為 1 時，表示模型在流失玩家的預測上完全正確。因此，曲線越趨近於左上角則越佳。再用 AUC (Area Under Curve) 來衡量此曲線，為計算該曲線面積。

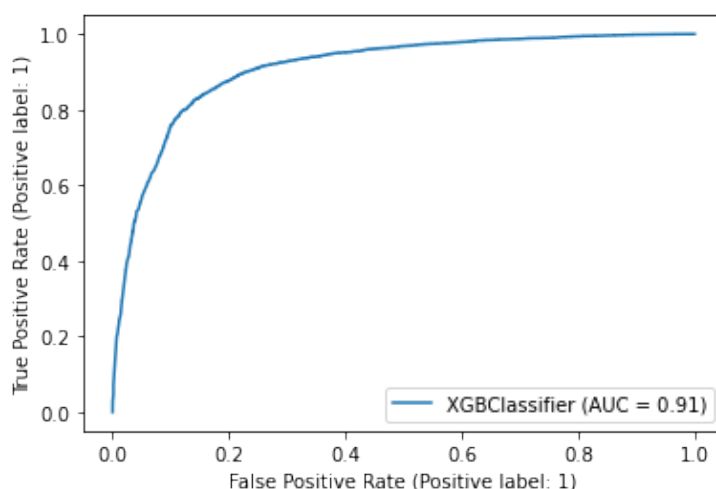


圖 3.15: ROC Curve 示意圖 (x 軸為假陽率；y 軸為真陽率；AUC 為其曲線面積)

藉由 PR Curve 來觀察學型模型的精確率及召回率，圖 3.16 為 PR Curve 示意圖。當  $x$  為 1 時，表示模型在流失玩家的預測上完全正確；而當  $y$  為 1 時，表示模型預測的流失玩家完全正確。因此，曲線越趨近於右上角則越佳。再用 AP (Average Precision) 來衡量此曲線，皆為計算該曲線面積。

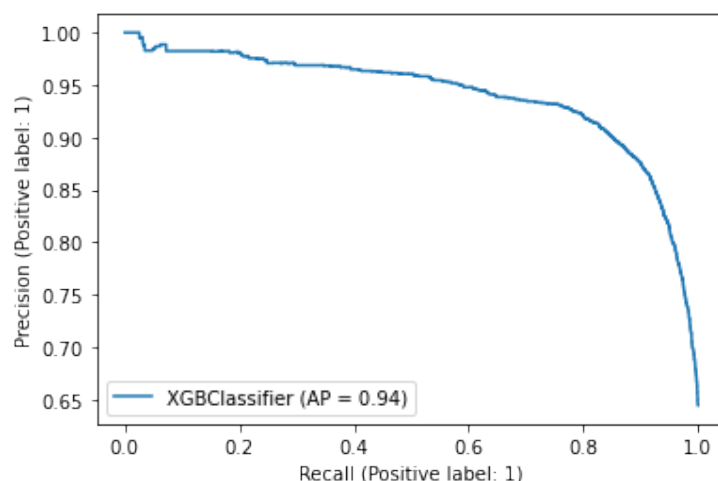


圖 3.16: PR Curve 示意圖 (  $x$  軸為召回率； $y$  軸為精確率；AP 為其曲線面積 )

本論文將以評估 PR Curve 為重。因在不平衡資料集上進行評估時，ROC Curve 將無法準確的呈現出學習模型的好壞，常有在 ROC Curve 上表現良好，但其 PR Curve 卻不如預期，導致此情況原因為多數群之評估遠大於少數群之評估，故可在 ROC Curve 上擁有好的數值，卻在 PR Curve 中表現不佳 [35]，如圖 3.17。

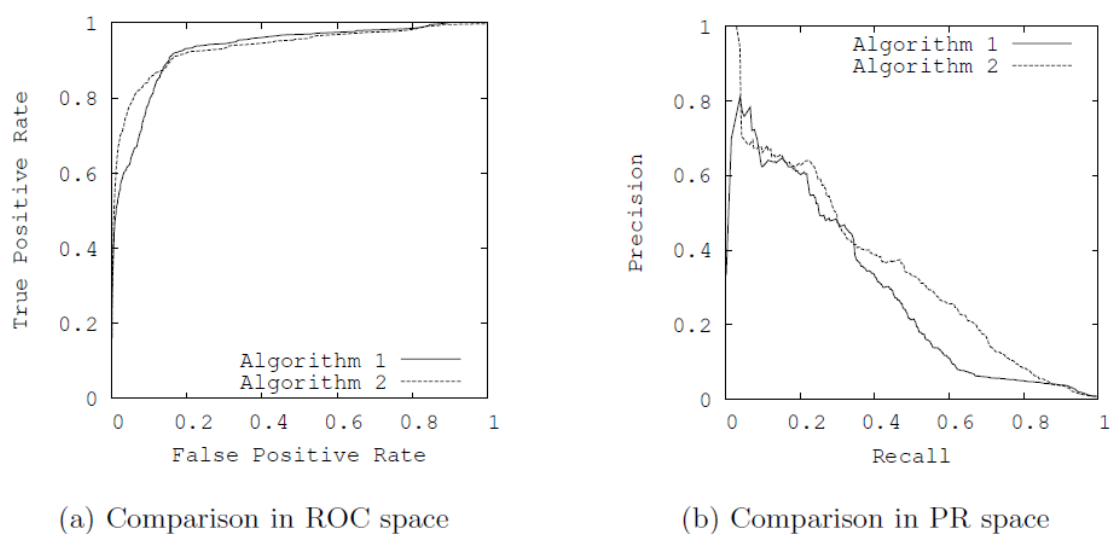


圖 3.17: 不平衡資料中 ROC Curve 失準示意圖 ( 此圖取自 [35] )

前述 3.3.4.2 小節中，交叉驗證搭配 3.3.4 小節中的參數調校表所使用之評估值為 Weighted  $F_\beta$  - Score，擇其最高值之學習模型，選定為最佳模型。Weighted  $F_\beta$  - Score 為在  $F_\beta$  - Score 評估值上導入樣本數權重概念，如式 3.6 與式 3.7，其中  $N_1$  為流失玩家之樣本數、 $N_0$  為非流失玩家之樣本數、 $F_{\beta 1}$  為流失玩家之  $F_\beta$  - Score、 $F_{\beta 0}$  為非流失玩家之  $F_\beta$  - Score。適合使用在評估資料不平衡之資料集中。

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{(\beta^2 \times precision) + recall} \quad (3.6)$$

$$Weighted F_\beta = \frac{N_1}{N_0 + N_1} \times F_{\beta 1} + \frac{N_0}{N_0 + N_1} \times F_{\beta 0} \quad (3.7)$$

其中  $\beta$  則為精確率 (Precision) 與召回率 (Recall) 之間的比重，如表 3.4。本論文預測新進玩家是否會流失，將著重於召回率，即為將所有可能流失的新進玩家預測出來，因為新進玩家有可能是經由廣告吸引而來，而該玩家身上即帶有廣告投放之成本，故希望能將有可能會流失的新進玩家全部預測出來，使得遊戲商能盡可能地提高留存率。

$\beta$ 數值範圍	說明
$0 < \beta < 1$	評估著重於精確率
$\beta = 1$	精確率與召回率比重相當
$1 < \beta$	評估著重於召回率

表 3.4:  $\beta$  數值意義表

### 3.4 預測結果分析階段

此階段將著重於資料特徵重要性之分析。透過前述 3.3.5 小節所產出之預測結果，計算其資料特徵於各學習模型中各樹之重要性，並加總後正規化，產出之分析結果將與前述 3.2.1.1 小節中推測之資料特徵進行探討，並藉由最終結果對遊戲中的遊玩體驗進行評估與建議。

### 3.4.1 資料特徵重要性分析

基尼不純度 ( Gini Impurity,  $G$  ) [36] 表示一個隨機選中的樣本被分類錯誤的可能性，如式 3.8，其中  $D$  為樹中的某個節點、 $c$  為 *Class*、 $p(i)$  為被分類為  $i$  類之發生機率。透過基尼不純度 ( $G$ ) 來衡量建樹時之分類準則，值越小，表示其分類的結果較好，以此來挑選出最適合用來進行分割之資料特徵及數值。

$$Gini\ Impurity(D) = G(D) = \sum_{i=1}^c p(i)(1 - p(i)) = 1 - \sum_{i=1}^c p(i)^2 \quad (3.8)$$

將資料特徵重要性 ( Feature Importance,  $fi$  ) 定義為加總各樹中各資料特徵於節點分割時所提供之基尼不純度 ( $G$ ) ( 見式 3.8 )，稱為基尼重要性 ( Gini Importance,  $GI$  )，再將其正規化至區間  $[0,1]$  中。

式 3.9 為計算樹中各節點之基尼重要性 ( $GI$ )，其中  $D_p$  為父節點、 $N_p$  為父節點之樣本數、 $D_{left}$  為左子節點、 $N_{left}$  為左子節點之樣本數、 $D_{right}$  為右子節點、 $N_{right}$  為右子節點之樣本數。首先計算  $D_p$ 、 $D_{left}$  及  $D_{right}$  之基尼不純度 ( $G$ )，並計算  $D_{left}$  及  $D_{right}$  之樣本數權重比例，最後將  $D_p$  之基尼不純度 ( $G$ ) 減去兩權重值。

$$Gini\ Importance(D_p) = GI(D_p) = G(D_p) - \frac{N_{left}}{N_p} \times G(D_{left}) - \frac{N_{right}}{N_p} \times G(D_{right}) \quad (3.9)$$

式 3.10 為計算資料特徵於單樹中之重要性，其中  $x$  為欲求其重要性之資料特徵、 $k$  為節點分割時所用資料特徵為  $x$  之所有節點、 $l$  為樹中所有節點。首先加總所有  $k$  之基尼重要性 ( $GI$ )，並加總  $l$  之基尼重要性 ( $GI$ )，最後將其進行正規化計算，落於區間  $[0,1]$  中，並總和為 1。

$$fi(t, x) = \frac{\sum_{k \in \text{node split based on } x} GI(D_k)}{\sum_{l \in \text{all nodes}} GI(D_l)} \quad (3.10)$$

式 3.11 為計算資料特徵於多樹中之重要性，其中  $x$  為欲求其重要性之資料特徵、 $t$  為學習模型中的所有樹、 $N_{trees}$  為樹總數。首先加總所有  $t$  中  $x$  的  $fi(t, x)$ ，並取其平均於  $N_{trees}$  中，最後即計算出  $x$  於學習模型內之資料特徵重要性 ( $fi$ )。

$$f_i(x) = \frac{\sum_{t \in \text{all trees}} f_i(t, x)}{N_{\text{trees}}} \quad (3.11)$$

## 3.5 產業應用分析階段

此階段將著重於建立代理人模型。代理人模型為一種優化方法，當模型較為複雜、計算量較大時，可以用一簡化模型來替代，如決策樹。

### 3.5.1 代理人模型

代理人模型表示用一簡單的模型來解釋一複雜的模型，如：用決策樹作為極限梯度提升之代理人模型。結構上較為簡單，在解釋玩家流失的行為規則上也較為清楚，以此讓市場操作人員的挽留策略更加明確。

圖 3.18 為代理人模型示意圖，sample 為樣本比、value 為分類為 Class0 及分類為 Class1 的機率、class 為類別。此圖將數據集劃分為 2 個類別，其分別各占數據集的 0.36 與 0.64，所以最上面的根節點認為屬於 Class1 的可能性較大。透過根節點可以知道，當特徵 1 小於或等於 3.5 時，有 69.4% 的樣本數，且屬於 Class1 的可能性較大；反之，當特徵 1 大於 3.5 時，有 30.6% 的樣本數，且屬於 Class0 的可能性較大。繼續向下通過其他節點，分類結果就能更加確定。

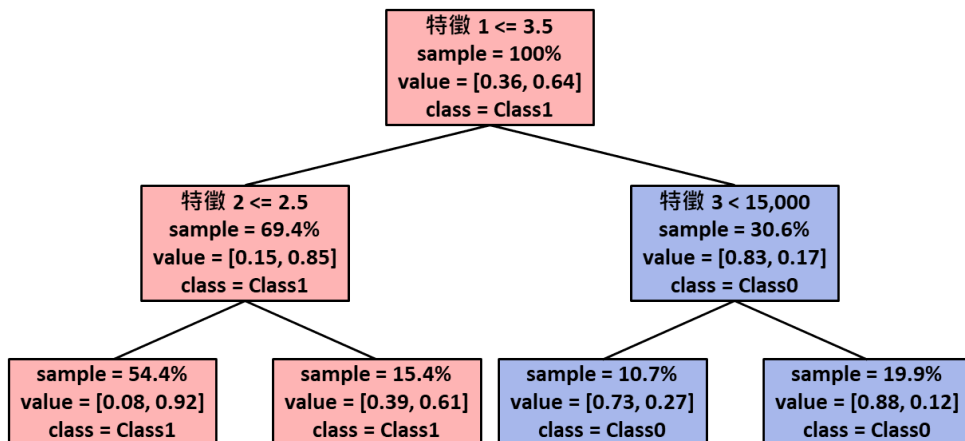


圖 3.18: 代理人模型示意圖

## 第 4 章 實驗結果與分析

此章節中，我們將針對前述第 3 章之研究方法進行實驗結果評估與分析，第 4.1 小節說明實驗系統架構、第 4.2 小節說明資料前處理評估、第 4.3 小節說明資料分析評估、第 4.4 小節說明機器學習評估、第 4.5 小節說明預測結果分析評估，以及第 4.6 小節說明產業應用分析評估。

### 4.1 實驗系統架構

本論文實驗系統架構將分為資料前處理端、資料分析端與機器學習端，如 4.1，並均以 Python 3.9.7 做為開發語言。

- 資料前處理端：以 Apache Spark 3.2.1 [37] 處理資料之前處理以及分割資料集所使用。
- 資料分析端：以 missingno 0.5.1 [38]、Seaborn 0.11.2 [39] 協助以圖表方式呈現資料特性。
- 機器學習端：以 pandas 1.3.4 [40] [41]、scikit-learn 0.24.2 [42] [43]、XGBoost 1.6.1 [10] 處理機器學習訓練與評估。

系統端點	資料前處理端	資料分析端	機器學習端
	<i>Apache Spark</i>	<i>missingno</i>	<i>pandas</i>
研究環境		<i>Seaborn</i>	<i>scikit-learn</i>
			<i>XGBoost</i>

表 4.1: 實驗系統架構之研究環境表

本文的資料集取自一博弈遊戲，此遊戲於全球發行，遊戲內容包含了老虎機 (Fruit Machine)、魚機 (Fish Hunter) 和其他小遊戲。

## 4.2 資料前處理評估

此階段將評估前章 3.1 小節之資料前處理。4.2.1 小節為預測受眾評估，將說明 3.1.2 小節之資料過濾；4.2.2 小節為預測特徵評估，將說明 3.1.3 小節之目標值準備及 3.1.4 小節之資料特徵探勘與特徵工程。

本文收集了 2022/03/01 至 2022/05/31 的資料，共計三個月。此資料集是以天為單位，記錄了玩家於遊戲平台上和各項遊戲內的行為軌跡。總容量約為 44 GB。

### 4.2.1 預測受眾評估

本文將日本新進玩家作為預測受眾，因為日本玩家相較於其他國家較少有不良的帳號紀錄（例如：同一位玩家多次創建新帳號等），其遊戲資料相對地會較有可信度。此外，本文特別排除了等級 10 以下的玩家，確保收集到的玩家資料皆是有通過新手教學的，讓後續產生的特徵更有價值。共有 60,469 位玩家作為預測受眾，來進行後續實驗。

接著進行資料過濾，以獲得有價值玩家。本文直接鎖定預測受眾於日本等級 10 以上之新進玩家，用到的資料集沒有空缺值存在，因此，會直接進行無價值玩家資料處理。本文根據市場操作人員的建議，將觀察期 ( $O$ )、挽留期 ( $R$ ) 及表現期 ( $P$ ) 分別設為 4 天、1 天及 2 天，將無價值玩家排除後，最後剩下 57,170 位有價值玩家，如表 4.2。

新進玩家總數	日本等級 10 以上 新進玩家數	空缺值玩家數	無價值玩家數	有價值玩家數
4,750,383	60,469	0	3,299	57,170
有價值玩家數 = 日本等級 10 以上新進玩家數 - 空缺值玩家數 - 無價值玩家數				

表 4.2: 有價值玩家觀察表

### 4.2.2 預測特徵評估

圖 4.1 為新進玩家之流失長條圖，用來觀察有價值玩家的流失速度。從圖中可以看出，隨著天數增加，有登入遊戲之玩家數量會跟著減少，創帳號後隔天有登入紀錄的玩家有 6 成左右，創帳號後第 3 天有登入紀錄者只剩下不到 4 成。故本論文聚焦於新進玩家之流失預測，希望能快速進行挽留策略，把握住新進玩家。

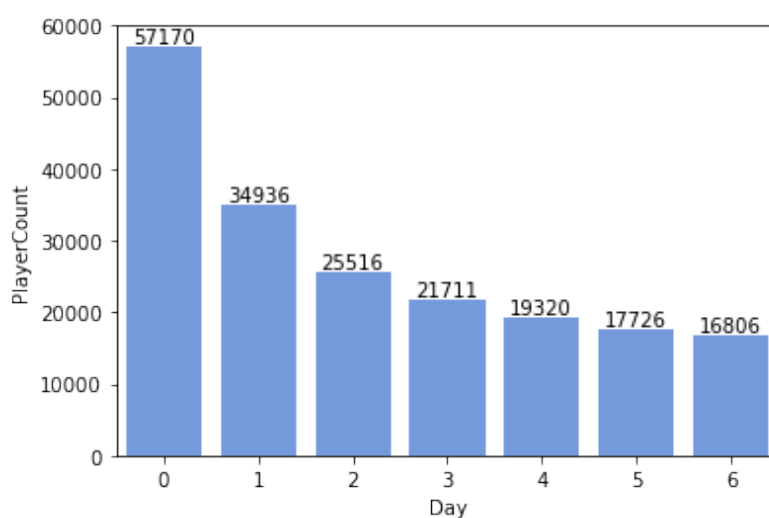


圖 4.1: 新進玩家之流失長條圖 (x 軸為玩家登入日減去創立帳號日；y 軸為有登入之玩家數)

本文將定義新進玩家 (於 4.2.1 小節中，篩選後之有價值玩家) 中的非流失玩家與流失玩家。觀察期有登入紀錄的玩家中，表現期間也有登入紀錄者視為非流失玩家，反之視為流失玩家。如表 4.3，會有 20,488 位非流失玩家及 36,682 位流失玩家來做後續實驗。

有價值玩家數	非流失玩家	非流失玩家占比	流失玩家數	流失玩家占比
57,170	20,488	35.84 %	36,682	64.16 %


表 4.3: 非流失玩家及流失玩家定義表



從原始資料集中探勘出  $O$  天 ( 資料特徵探勘期，即觀察期 ) 內之玩家資料，並以此來建立資料特徵。以計次、加總與平均，並以天為單位拆分時間段，產生第一層特徵變數，再對第一層特徵變數做計算，進而得到第二層特徵變數。最後，第一層特徵有 156 個，第二層特徵有 93 個，共得到了 249 個特徵變數。

另外，因類別型資料特徵不適用於樹狀結構之學習模型，故在其應用於機器學習前，將此類資料特徵透過 One Hot Encoding [44] 進行轉化，以利機器學習訓練，將此類特徵也歸類至第一層特徵，如圖 4.2 之作業系統。因此，第一層特徵數為 158 個，總特徵數為 251 個。

ID	作業系統			
0	1			
1	2			



ID	Android	iOS
0	1	0
1	0	1

圖 4.2: 作業系統 One Hot Encoding (作業系統 1 為 Android ; 2 為 iOS)

為了避免特徵之間存在高度相關性，會影響到模型預測的準確性，本文對上述 251 個特徵變數進行篩選。透過支持向量機 ( Support Vector Machine ) 找出高共線性的冗贅特徵，並將其排除，最後只剩下 52 個特徵變數。此外，去掉高共線性的特徵也可以讓模型的可解釋性更好。表 4.4 為資料特徵總數表。圖 4.3 為第一層變數圖、圖 4.4 為第二層變數圖，而紅色底為篩選後之特徵。

第一層特徵數	第二層特徵數	篩選前特徵數	篩選後特徵數
158	93	251	52

表 4.4: 資料特徵總數表

為了避免部分特徵之數值過大或過小，彼此間差距較大，會影響到模型訓練精度，本文對所有特徵數值進行歸一化 ( Normalization )。式 4.1 為本文歸一化的方式，會將所有數值映射到  $[0,1]$  區間。

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (4.1)$$

第一層特徵變數							
Android	iOS	登入天數	登入次數	第1天登入次數	第2天登入次數	第3天登入次數	第4天登入次數
儲值次數	第1天儲值次數	第2天儲值次數	第3天儲值次數	第4天儲值次數	儲值總金額	第1天儲值總金額	第2天儲值總金額
第3天儲值總金額	第4天儲值總金額	儲值平均金額	第1天儲值平均金額	第2天儲值平均金額	第3天儲值平均金額	第4天儲值平均金額	撿錢次數
第1天撿錢次數	第2天撿錢次數	第3天撿錢次數	第4天撿錢次數	撿錢總金額	第1天撿錢總金額	第2天撿錢總金額	第3天撿錢總金額
第4天撿錢總金額	撿錢平均金額	第1天撿錢平均金額	第2天撿錢平均金額	第3天撿錢平均金額	第4天撿錢平均金額	客訴次數	第1天客訴次數
第2天客訴次數	第3天客訴次數	第4天客訴次數	活動完成次數	第1天活動完成次數	第2天活動完成次數	第3天活動完成次數	第4天活動完成次數
老虎機總押注次數	第1天老虎機總押注次數	第2天老虎機總押注次數	第3天老虎機總押注次數	第4天老虎機總押注次數	老虎機平均押注次數	第1天老虎機平均押注次數	第2天老虎機平均押注次數
第3天老虎機平均押注次數	第4天老虎機平均押注次數	老虎機有贏分總次數	第1天老虎機有贏分總次數	第2天老虎機有贏分總次數	第3天老虎機有贏分總次數	第4天老虎機有贏分總次數	老虎機有贏分平均次數
第1天老虎機有贏分平均次數	第2天老虎機有贏分平均次數	第3天老虎機有贏分平均次數	第4天老虎機有贏分平均次數	老虎機總贏分	第1天老虎機總贏分	第2天老虎機總贏分	第3天老虎機總贏分
第4天老虎機總贏分	老虎機平均贏分	第1天老虎機平均贏分	第2天老虎機平均贏分	第3天老虎機平均贏分	第4天老虎機平均贏分	魚機總押注次數	第1天魚機總押注次數
第2天魚機總押注次數	第3天魚機總押注次數	第4天魚機總押注次數	魚機平均押注次數	第1天魚機平均押注次數	第2天魚機平均押注次數	第3天魚機平均押注次數	第4天魚機平均押注次數
魚機有贏分總次數	第1天魚機有贏分總次數	第2天魚機有贏分總次數	第3天魚機有贏分總次數	第4天魚機有贏分總次數	魚機有贏分平均次數	第1天魚機有贏分平均次數	第2天魚機有贏分平均次數
第3天魚機有贏分平均次數	第4天魚機有贏分平均次數	魚機總贏分	第1天魚機總贏分	第2天魚機總贏分	第3天魚機總贏分	第4天魚機總贏分	魚機平均贏分
第1天魚機平均贏分	第2天魚機平均贏分	第3天魚機平均贏分	第4天魚機平均贏分	小遊戲次數	第1天小遊戲次數	第2天小遊戲次數	第3天小遊戲次數
第4天小遊戲次數	小遊戲總贏分	第1天小遊戲總贏分	第2天小遊戲總贏分	第3天小遊戲總贏分	第4天小遊戲總贏分	小遊戲平均贏分	第1天小遊戲平均贏分
第2天小遊戲平均贏分	第3天小遊戲平均贏分	第4天小遊戲平均贏分	遊戲卡次數	第1天遊戲卡次數	第2天遊戲卡次數	第3天遊戲卡次數	第4天遊戲卡次數
遊戲卡總贏分	第1天遊戲卡總贏分	第2天遊戲卡總贏分	第3天遊戲卡總贏分	第4天遊戲卡總贏分	遊戲卡平均贏分	第1天遊戲卡平均贏分	第2天遊戲卡平均贏分
第3天遊戲卡平均贏分	第4天遊戲卡平均贏分	武器卡次數	第1天武器卡次數	第2天武器卡次數	第3天武器卡次數	第4天武器卡次數	武器卡總贏分
第1天武器卡總贏分	第2天武器卡總贏分	第3天武器卡總贏分	第4天武器卡總贏分	武器卡平均贏分	第1天武器卡平均贏分	第2天武器卡平均贏分	第3天武器卡平均贏分
第4天武器卡平均贏分	技能卡次數	第1天技能卡次數	第2天技能卡次數	第3天技能卡次數	第4天技能卡次數		

圖 4.3: 第一層特徵變數 (紅色底為篩選後之特徵)

第二層特徵變數							
第2天登入 次數變化量	第3天登入 次數變化量	第4天登入 次數變化量	第2天儲值 次數變化量	第3天儲值 次數變化量	第4天儲值 次數變化量	第2天儲值 總金額變化量	第3天儲值 總金額變化量
第4天儲值 總金額變化量	第2天儲值 平均金額變化量	第3天儲值 平均金額變化量	第4天儲值 平均金額變化量	第2天撿錢 次數變化量	第3天撿錢 次數變化量	第4天撿錢 次數變化量	第2天撿錢 總金額變化量
第3天撿錢 總金額變化量	第4天撿錢 總金額變化量	第2天撿錢 平均金額變化量	第3天撿錢 平均金額變化量	第4天撿錢 平均金額變化量	第2天客訴 次數變化量	第3天客訴 次數變化量	第4天客訴 次數變化量
第2天活動完成 次數變化量	第3天活動完成 次數變化量	第4天活動完成 次數變化量	第2天老虎機總押注 次數變化量	第3天老虎機 總押注次數變化量	第4天老虎機 總押注次數變化量	第2天老虎機 平均押注次數變化量	第3天老虎機 平均押注次數變化量
第4天老虎機 平均押注次數變化量	第2天老虎機 有贏分總次數變化量	第3天老虎機 有贏分總次數變化量	第4天老虎機 有贏分總次數變化量	第2天老虎機 有贏分平均次數 變化量	第3天老虎機 有贏分平均次數 變化量	第4天老虎機 有贏分平均次數 變化量	第2天老虎機 總贏分變化量
第3天老虎機 總贏分變化量	第4天老虎機 總贏分變化量	第2天老虎機 平均贏分變化量	第3天老虎機 平均贏分變化量	第4天老虎機 平均贏分變化量	第2天魚機 總押注次數變化量	第3天魚機 總押注次數變化量	第4天魚機 總押注次數變化量
第2天魚機 平均押注次數變化量	第3天魚機 平均押注次數變化量	第4天魚機 平均押注次數變化量	第2天魚機 有贏分總次數變化量	第3天魚機 有贏分總次數變化量	第4天魚機 有贏分總次數變化量	第2天魚機 有贏分平均次數 變化量	第3天魚機 有贏分平均次數 變化量
第4天魚機 有贏分平均次數 變化量	第2天魚機 總贏分變化量	第3天魚機 總贏分變化量	第4天魚機 總贏分變化量	第2天魚機 平均贏分變化量	第3天魚機 平均贏分變化量	第4天魚機 平均贏分變化量	第2天小遊戲 次數變化量
第3天小遊戲 次數變化量	第4天小遊戲 次數變化量	第2天小遊戲 總贏分變化量	第3天小遊戲 總贏分變化量	第4天小遊戲 總贏分變化量	第2天小遊戲 平均贏分變化量	第3天小遊戲 平均贏分變化量	第4天小遊戲 平均贏分變化量
第2天遊戲卡 次數變化量	第3天遊戲卡 次數變化量	第4天遊戲卡 次數變化量	第2天遊戲卡 總贏分變化量	第3天遊戲卡 總贏分變化量	第4天遊戲卡 總贏分變化量	第2天遊戲卡 平均贏分變化量	第3天遊戲卡 平均贏分變化量
第4天遊戲卡 平均贏分變化量	第2天武器卡 次數變化量	第3天武器卡 次數變化量	第4天武器卡 次數變化量	第2天武器卡 總贏分變化量	第3天武器卡 總贏分變化量	第4天武器卡 總贏分變化量	第2天武器卡 平均贏分變化量
第3天武器卡 平均贏分變化量	第4天武器卡 平均贏分變化量	第2天技能卡 次數變化量	第3天技能卡 次數變化量	第4天技能卡 次數變化量			

圖 4.4: 第二層特徵變數 (紅色底為篩選後之特徵)

## 4.3 資料分析評估

此階段將評估前章 3.2.1.1 小節之高資訊量資料特徵推測，於 4.3.1 小節 探索性資料分析評估說明。

### 4.3.1 探索性資料分析評估

利用長條圖及直方圖觀察在資料集中哪些特徵可能屬於高資訊量資料特徵。

圖 4.5 為觀察期間玩家登入天數之人數統計圖。可以從圖中看出，非流失玩家數會隨著登入天數增加而遞增，流失玩家數則隨著登入天數增加而遞減。推測玩家登入天數可以帶給學習模型很好的分類資訊。

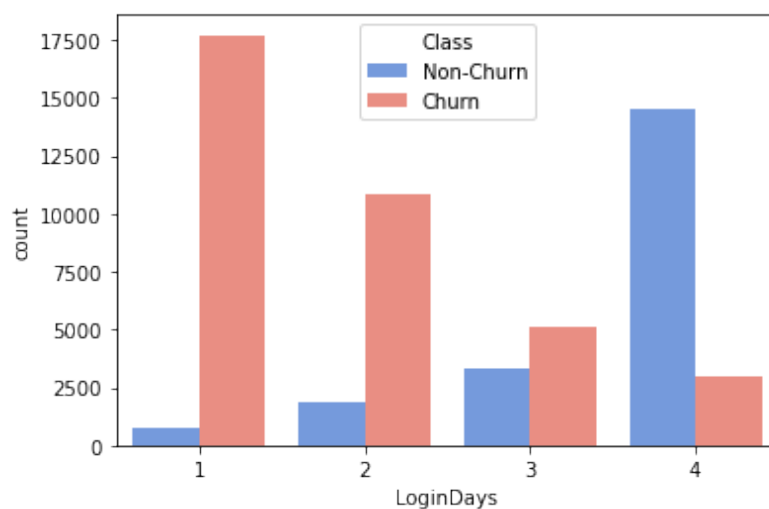


圖 4.5: 觀察期間玩家登入天數之人數統計圖 (x 軸為登入天數；y 軸為玩家數量)

圖 4.6 為觀察期間玩家完成活動任務之次數統計圖。可以從圖中看出，除了創帳號當天以外，非流失玩家完成活動任務的次數皆遠大於流失玩家，甚至於創帳號後第三天，其次數變化量還為正值。推測活動任務完成次數之相關特徵可以帶給學習模型很好的分類資訊。

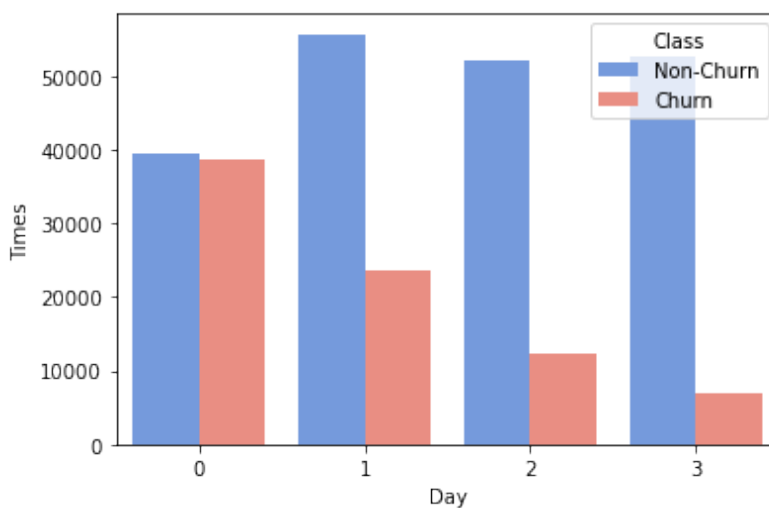


圖 4.6: 觀察期間玩家完成任務之次數統計圖 (x 軸為觀察期第 x 天；y 軸為活動任務完成次數)

圖 4.7 為玩家創帳號後第三天之老虎機有贏分平均次數統計圖。可以從圖中看出，到了創帳號後第三天，流失玩家幾乎都沒有獲得贏分。推測此特徵可以帶給學習模型很好的分類資訊。

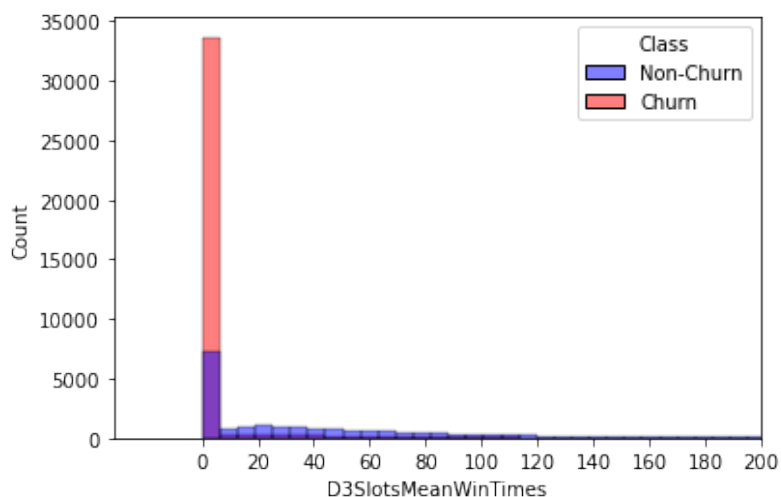


圖 4.7: 玩家創帳號後第三天之老虎機有贏分平均次數 (x 軸為平均有贏分次數；y 軸為玩家數量) (紫色區域為非流失玩家與流失玩家重疊區域)

圖 4.8 為作業系統各玩家統計圖。可以從圖中看出，不論是何種作業系統，流失玩家與非流失玩家的比例都差不多。推測作業系統的差異無法帶給學習模型很好的分類資訊。

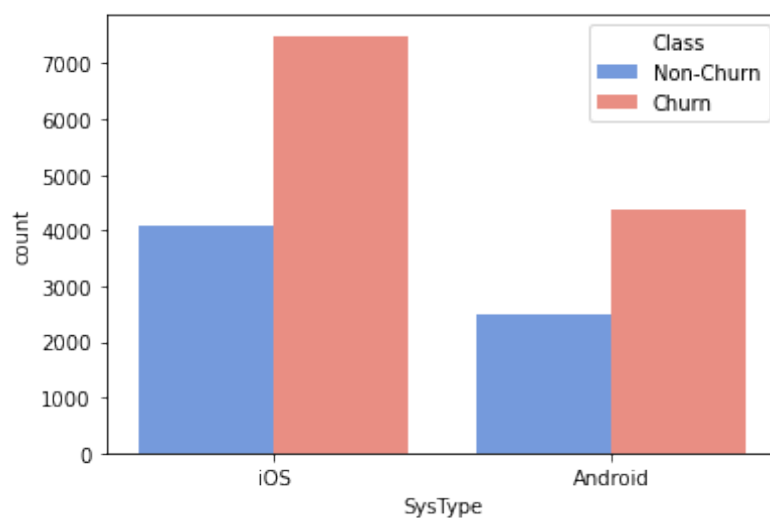


圖 4.8: 作業系統各玩家統計圖 (x 軸為作業系統；y 軸為玩家數量)

## 4.4 機器學習評估

此階段將評估前章 3.3 小節之機器學習。4.4.1 小節為分割訓練與測試資料集評估，將說明 3.3.1 小節之分割訓練與測試資料集；4.4.2 小節為不平衡資料權重調整評估，將說明 3.3.3 小節之不平衡資料權重調整；4.4.3 小節為最佳模型評估，將說明 3.3.4 小節之搜尋最佳參數解及 3.3.5 小節之評估驗證最佳模型；4.4.4 小節為時間框架評估，將比較觀察期、表現期不同的天數，會對模型的輸出有甚麼影響。

### 4.4.1 分割訓練與測試資料集評估

將資料集依照 8：2 之比例分割。如圖 3.13，採分類隨機抽樣，即流失玩家與非流失玩家各別以 8：2 之比例隨機抽樣。表 4.5 為分割完資料集後之流失玩家數與非流失玩家數，訓練資料集與測試資料集之流失玩家與非流失玩家比例皆與原資料集相等，約為 1.79 倍。

資料集 \ 玩家數	玩家數	
	流失玩家	非流失玩家
訓練集	29,346	16,390
測試集	7,336	4,098

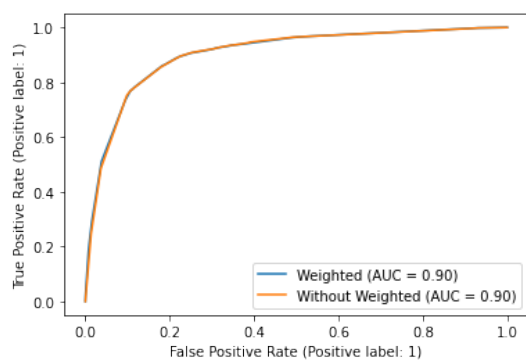
表 4.5: 訓練與測試資料集玩家數表

### 4.4.2 資料不平衡處理評估

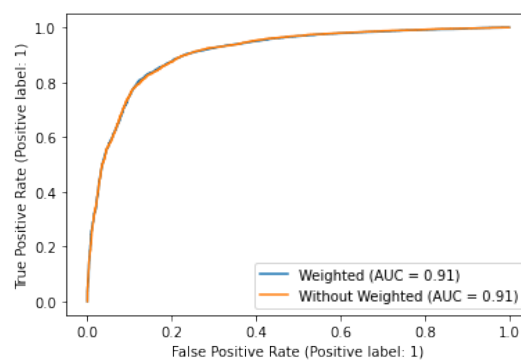
依照式 3.1 計算流失玩家樣本之縮小權重，如式 4.2，最後將流失玩家之樣本權重縮小 0.56 倍。

$$class\ 0 : class\ 1 = 1 : \frac{16,390}{29,346} = 1 : 0.56 \quad (4.2)$$

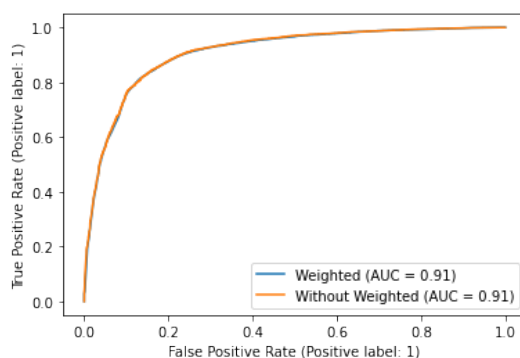
圖 4.9 為三種學習模型之 ROC Curve，並比較不平衡資料處理前後之差異，(a) 為決策樹、(b) 為隨機森林、(c) 為極限梯度提升，黃色線為未加入權重值、藍色線為加入權重值。從圖組中可以看出，在流失玩家之樣本權重上進行縮小，於本文並沒有明顯的作用，推測是因為本文預測的受眾資料並沒有極大的不平衡。



(a) 決策樹 ROC Curve 圖



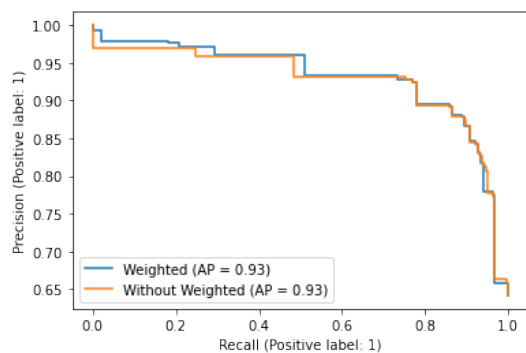
(b) 隨機森林 ROC Curve 圖



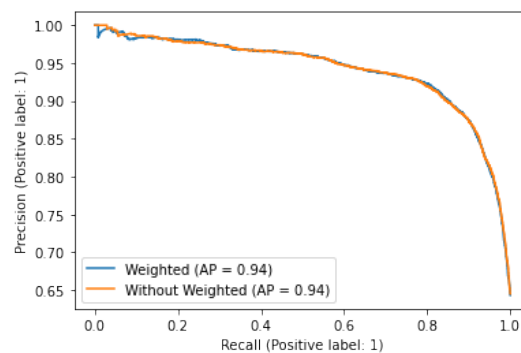
(c) 極限梯度提升 ROC Curve 圖

圖 4.9: 不平衡資料處理前後比較之 ROC Curve 圖 (x 軸為假陽率；y 軸為真陽率)

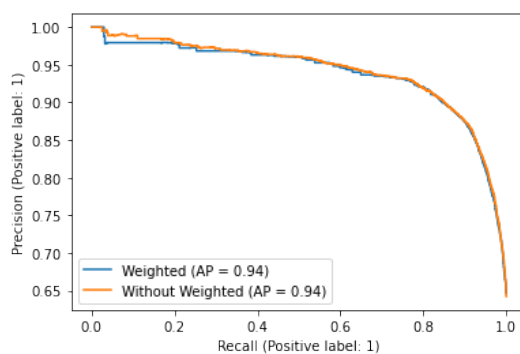
圖 4.10 為三種學習模型之 PR Curve，並比較不平衡資料處理前後之差異，(a) 為決策樹、(b) 為隨機森林、(c) 為極限梯度提升，黃色線為未加入權重值、藍色線為加入權重值。從圖組中可以看出，在流失玩家之樣本權重上進行縮小，於本文並沒有明顯的作用，推測是因為本文預測的受眾資料並沒有極大的不平衡。



(a) 決策樹 PR Curve 圖



(b) 隨機森林 PR Curve 圖



(c) 極限梯度提升 PR Curve 圖

圖 4.10: 不平衡資料處理前後比較之 PR Curve 圖 (x 軸為召回率；y 軸為精確率)



### 4.4.3 最佳模型評估

利用網格搜索與交叉驗證來調校出最佳模型。如圖 3.14，我們將採用 Repeated 2 次及 5-Fold，最後使用測試資料集進行評估驗證，將說明 7,336 位流失玩家 (*class 1*) 及 4,098 位非流失玩家 (*class 0*)。驗證結果如表 4.6，從表中可以看出，極限梯度提升的結果表現為三種模型中最好的，因為其 Weighted  $F_\beta$  - Score 為 0.849 是三者最高。

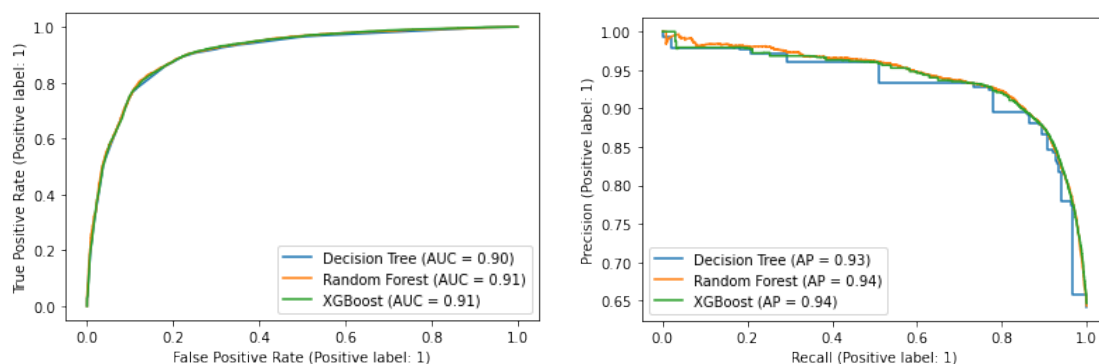
學習模型 \ 評估	$precision^+$	$recall^+$	$F_\beta^+$	$Weighted F_\beta$
	$precision^-$	$recall^-$	$F_\beta^-$	
決策樹	0.892	0.863	0.870	0.846
	0.769	0.813	0.802	
隨機森林	0.888	0.874	0.877	0.848
	0.781	0.804	0.798	
極限梯度提升	0.887	0.876	0.878	0.849
	0.783	0.801	0.796	

＋：以正例 (流失玩家  $class\ 1$ ) 為評估對象進行計算

－：以反例 (非流失玩家  $class\ 0$ ) 為評估對象進行計算

表 4.6: 最佳模型評估表

圖 4.11 為三種學習模型之曲線比較圖，(a) 為三種學習模型之 ROC Curve 比較圖、(b) 為三種學習模型之 PR Curve 比較圖，並且都為加入權重值之結果。從圖組中可以看出，隨機森林與極限梯度提升並沒有明顯的差異，原因可能在於特徵數量還不足夠，如果持續的擴增特徵，或許能凸顯出極限梯度提升的優點。



(a) 三種學習模型之 ROC Curve 比較圖

(b) 三種學習模型之 PR Curve 比較圖

圖 4.11: 三種學習模型之曲線比較圖

綜合上述得到的實驗結果，可以發現本文挑選的三種模型表現都不錯，Weighted  $F_\beta$  - Score 都有接近 0.85，且曲線面積也都有 0.9 以上。可見本文建立的資料特徵能適用於各種模型中，並都有不錯的結果。

此外，極限梯度提升是採用梯度下降法 ( Gradient Descent ) 來加速學習模型之收斂，減少建樹時間成本，也較不會有過擬合 ( Overfitting ) 的情形出現。在訓練極限梯度提升模型時，可以將資料格式轉為極限梯度提升專用的 DMatrix 格式，以此來提升演算的效率。

表 4.7 為三種學習模型之最佳參數解，參數搜尋範圍如表 4.8。決策樹因其為單樹結構，相較之下需要生成更深的樹；而隨機森林及極限梯度提升則因其為多樹結構，希望能以廣度發展，而非深度，相較之下需要生成更多的樹。

學習模型	決策樹	隨機森林	極限梯度提升
參數調校	max_depth=5	n_estimators=60	n_estimators=25
	min_samples_split=2	max_depth=14	max_depth=5
	min_samples_leaf=20	min_samples_split=6	learning_rate=0.05
		min_samples_leaf=5	

表 4.7: 最佳模型參數解表

參數名	搜尋範圍
n_estimators	20, 25, 30, 35, 40, 45, 50, 55, 60
max_depth	5, 6, 7, 8, 9, 10, 11, 12, 13, 14
min_samples_split	2, 4, 6, 8, 10
min_samples_leaf	1, 5, 10, 15, 20
learning_rate	0.05, 0.07, 0.08, 0.1, 0.2

表 4.8: 參數搜尋範圍表

#### 4.4.4 時間框架評估

將比較觀察期與表現期在不同天數的情況下，會對模型的輸出結果有甚麼影響，並用極限梯度提升來進行驗證。

表 4.9 為觀察期評估表，將挽留期及表現期設為 1 天及 2 天，並比較觀察期為 2 天、4 天與 6 天的模型驗證結果。從表中可以看出，當觀察期越長時，模型的  $Weighted F_{\beta}$  - Score 越高，表示模型的表現結果越好。推測原因在於本文建立特徵的方式，因為能拆分更多的時間段，以此來獲得更多的玩家資料與特徵。

<div>觀察期</div> <div>評估</div>	$sample^{+}$	$precision^{+}$	$recall^{+}$	$F_{\beta}^{+}$	$Weighted F_{\beta}$
	$sample^{-}$	$precision^{-}$	$recall^{-}$	$F_{\beta}^{-}$	
2 天	6,570	0.834	0.767	0.790	0.780
	4,905	0.718	0.796	0.766	
4 天	7,336	0.887	0.876	0.878	0.849
	4,098	0.783	0.801	0.796	
6 天	7,701	0.913	0.907	0.908	0.878
	3,589	0.803	0.815	0.813	
+：以正例 (流失玩家 $class\ 1$ ) 為評估對象進行計算					
-：以反例 (非流失玩家 $class\ 0$ ) 為評估對象進行計算					

表 4.9: 觀察期評估表

表 4.10 為表現期評估表，將觀察期及挽留期設為 4 天及 1 天，並比較表現期為 2 天、4 天與 6 天的模型驗證結果。從表中可以看出，當表現期越長時，模型的  $Weighted F_{\beta}$  - Score 越低，表示模型的表現結果越差。推測原因在於，定義是否流失的時間太長，原本會被定義為流失玩家的人，可能會變成非流失玩家，導致模型的預測能力受到影響。

<div>表現期</div> <div>評估</div>	$sample^{+}$	$precision^{+}$	$recall^{+}$	$F_{\beta}^{+}$	$Weighted F_{\beta}$
	$sample^{-}$	$precision^{-}$	$recall^{-}$	$F_{\beta}^{-}$	
2 天	7,336	0.887	0.876	0.878	0.849
	4,098	0.783	0.801	0.796	
4 天	6,779	0.867	0.845	0.852	0.828
	4,449	0.773	0.802	0.793	
6 天	6,463	0.849	0.846	0.847	0.821
	4,548	0.782	0.786	0.785	
+：以正例 (流失玩家 $class\ 1$ ) 為評估對象進行計算					
-：以反例 (非流失玩家 $class\ 0$ ) 為評估對象進行計算					

表 4.10: 表現期評估表

從上述兩個表格中可以發現，當觀察期越長時，模型表現會越好，因為得到的特徵變多了；而當表現期越長時，模型表現會越差，因為原本該被定義為流失玩家的人，可能被定義成非流失玩家。但最後還是要根據市場需求來定義時間框架。

## 4.5 預測結果分析評估

此階段將評估前章 3.4 小節之預測結果分析。4.5.1 小節為資料特徵重要性評估，將說明 3.4.1 小節之資料特徵重要性分析。

### 4.5.1 資料特徵重要性評估

將利用式 3.9、式 3.10 及式 3.11 計算之各資料特徵於各模型之資料特徵重要性。如圖 4.12、圖 4.13 及圖 4.14，分別為決策樹、隨機森林及極限梯度提升之資料特徵重要性比較圖。

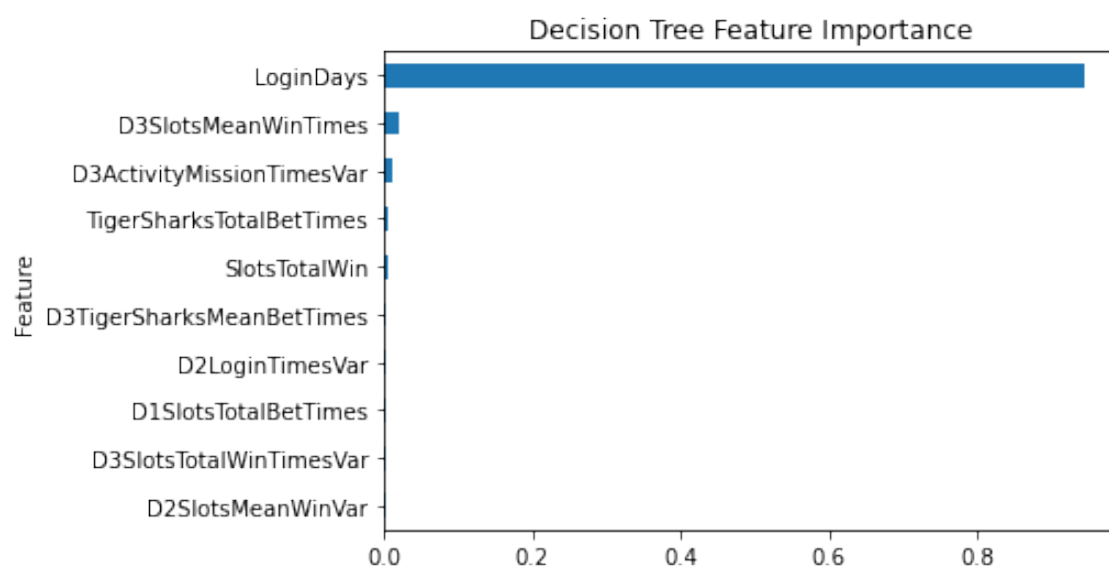


圖 4.12: 決策樹資料特徵重要性比較圖

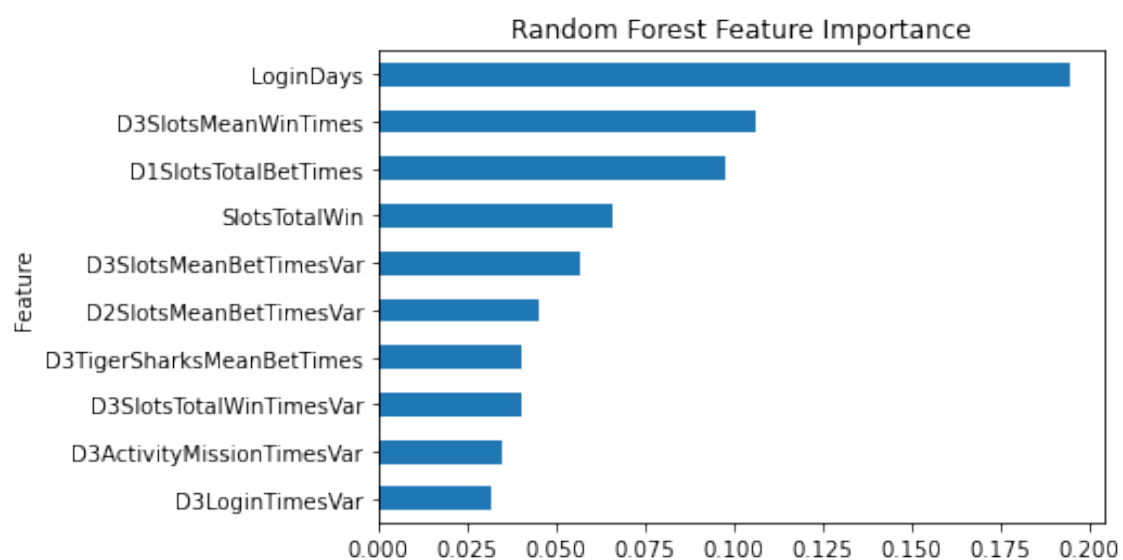


圖 4.13: 隨機森林資料特徵重要性比較圖

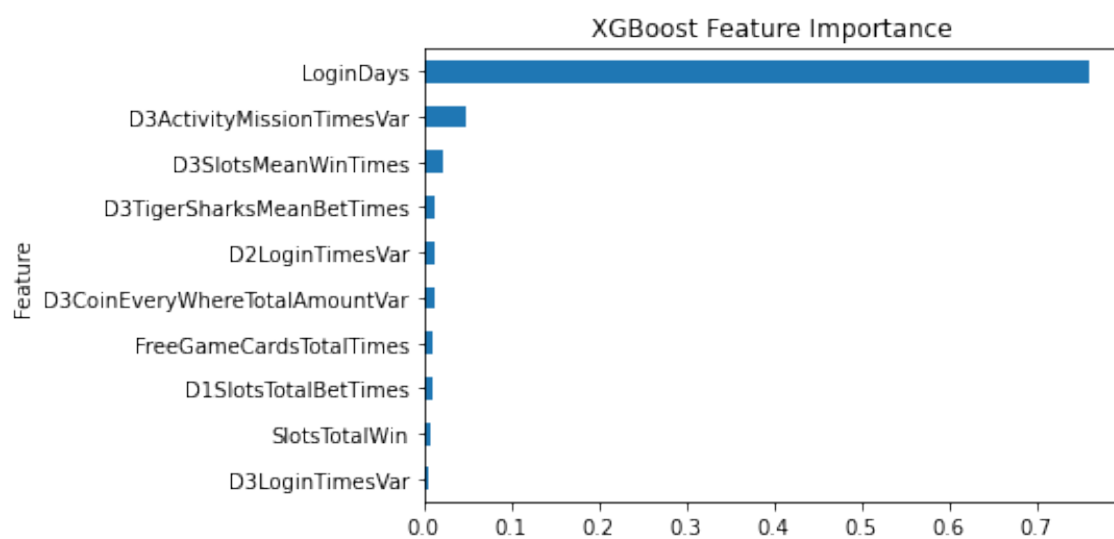


圖 4.14: 極限梯度提升資料特徵重要性比較圖

從三圖中可以看出，觀察期間的登入天數 (LoginDays) 皆是三種學習模型的第一名，可以說明登入天數的多寡確實與玩家是否流失有極大的關係。整合來看，在前十名中可以看到許多與變化量相關的特徵，如表 4.11，雖不是主要原因，但可以從老虎機相關變化量特徵中說明，玩家是否流失確實與遊玩遊戲時的體驗起伏有關係。

參數名	代表意思
D2LoginTimesVar	創帳號後第 2 天登入次數變化量
D3LoginTimesVar	創帳號後第 3 天登入次數變化量
D2SlotsMeanWinVar	創帳號後第 2 天老虎機平均贏分變化量
D2SlotsMeanBetTimesVar	創帳號後第 2 天老虎機平均押注次數變化量
D3SlotsMeanBetTimesVar	創帳號後第 3 天老虎機平均押注次數變化量
D3SlotsTotalWinTimesVar	創帳號後第 3 天老虎機有贏分總次數變化量
D3ActivityMissionTimesVar	創帳號後第 3 天活動任務完成次數變化量
D3CoinEveryWhereTotalAmountVar	創帳號後第 3 天撿錢總金額變化量

表 4.11: 特徵重要性前十名 (變化量相關特徵)

另外，三種學習模型之前十名資料特徵皆為數值型的資料特徵，而無類別型資料特徵，因其資料特徵重要性之評估以計算基尼重要性為主，數值型將會比類別型來得更為顯著，未來將可在計算重要性分析中，對於不同類型的資料特徵加入權重值，使得類別型的資料特徵能夠突出，讓整體分析更加準確。

## 4.6 產業應用分析評估

此階段將評估前章 3.5 小節之產業應用分析。4.6.1 小節為代理人模型評估，將說明 3.5.1 小節之代理人模型。

### 4.6.1 代理人模型評估

代理人模型，以決策樹來說明極限梯度提升，能清楚地知道模型判斷是否流失的門檻，了解玩家流失的行為規則，在後續市場操作的策略上能有很好的建議。

圖 4.15 為本文的代理人模型圖。從圖中來看，本文將數據集劃分為 2 個類別，即非流失玩家與流失玩家，其分別各占數據集的 0.36 與 0.64，所以最上面的根節點認為流失玩家的可能性較大。透過根節點可以知道，當玩家於觀察期間之登入天數 (LoginDays) 小於 3 天時，有 69.4% 的樣本數，且屬於流失玩家的可能性較大，因此會被判斷為流失玩家。繼續向下通過其他節點，分類結果就能更加確定。

此外，可以從代理人模型中發現，當登入天數小於 1 天時，流失機率已經高達 0.92。因此，可以建議遊戲商多推出留存規格，或是多舉辦與登入相關的活動，讓玩家增加其遊玩天數。

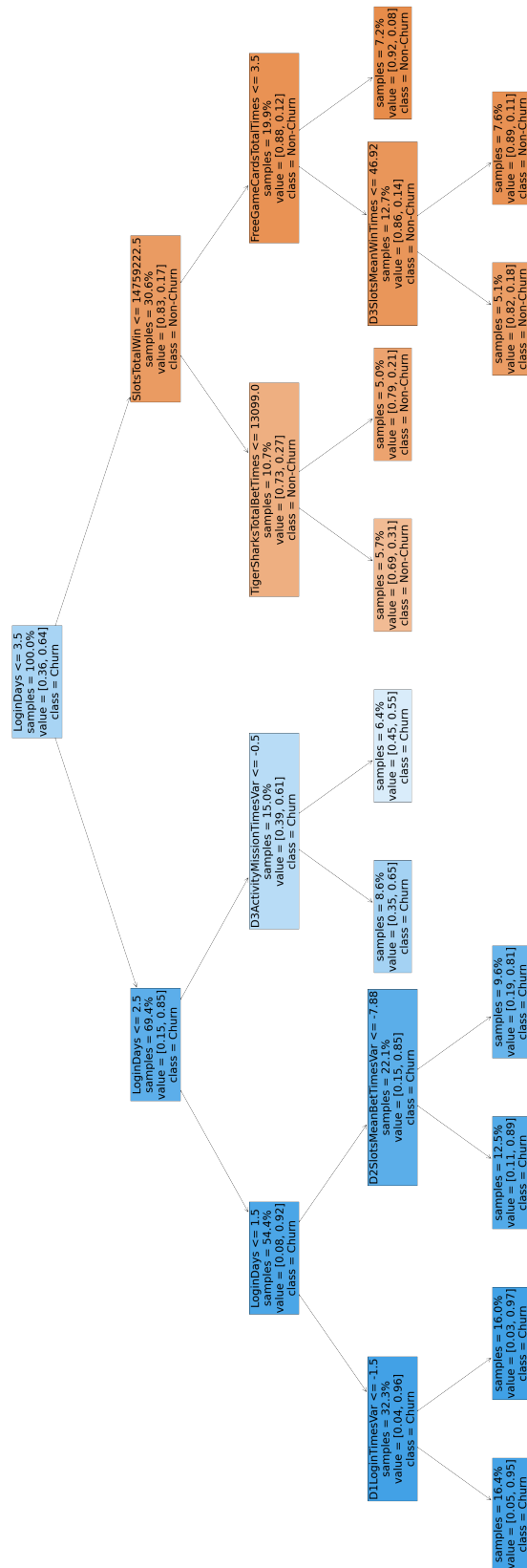


圖 4.15: 代理人模型圖



## 第 5 章 結論與未來研究

### 5.1 結論

本文提出一巨量資料探勘框架，此框架由五大階段組成：資料前處理、資料分析、機器學習、預測結果分析及產業應用分析。於資料前處理階段進行資料的整合與過濾，以篩選出有價值玩家資料，來提高整體分析成效，接著進行目標值準備、資料特徵探勘與特徵，透過一時間框架，定義流失及非流失玩家，並建立多個特徵變數，使得後續機器學習更加順利；於探索性資料分析階段分析資料特徵，提早推測資訊量較高的特徵；於機器學習階段預測結果，在訓練機器之前會調整權重值來解決資料不平衡問題；於預測結果分析階段計算特徵重要性，可整理出資料特徵突出的原因，例如：玩家登入天數等；最後於產業應用分析階段了解流失玩家的行為規則，可以做為市場操作人員實施挽留策略的依據。

此框架可以應用於遊戲領域且著重於新進玩家。透過資料特徵重要性分析，可以發現本文的第二層特徵變數（即變化量特徵）確實對模型的預測上有很大的幫助，讓實驗上有極好的數值表現；也能透過代理人模型快速地知道玩家流失的原因，藉由準確地採取策略，以強化遊戲玩家的留存，並進一步提高營收。

### 5.2 未來研究

由於本文探勘的資料特徵種類受限於遊戲平台所提供的原始資料集，如能獲得更進一步的詳細資料，或是盡可能地擴增特徵變數，將可更加準確的預測出流失玩家。此外，本文選擇的學習模型只有三種樹狀結構之模型，未來可進行更多實驗於不同類型的學習模型。於模型驗證上，未來也可以透過 A/B 測試，觀察哪個模型的結果比較可以提高玩家的留存率。雖然關於流失預測已有許多研究，但於產業應用上尚有許多議題可以探討。

## 參 考 文 獻

- [1] P. Miller, “GDC 2012: How Valve made Team Fortress 2 free-to-play,” *Gamasutra. Haettu*, vol. 7, 2012.
- [2] F. Reichheld and W. Sasser, “Zero defects: quality comes to service,” *Harvard Business Review*, September/October, pp. 105–111, 1990.
- [3] The Team at Swrve, “The April 2014 New Players Report.” <https://www.swrve.com/resources/weblog/the-april-2014-new-players-report>, 2014. [Online; accessed 30-June-2022].
- [4] K. Mustač, K. Bačić, L. Skorin-Kapov, and M. Sužnjević, “Predicting Player Churn of a Free-to-Play Mobile Video Game Using Supervised Machine Learning,” *MDPI*, 2022.
- [5] J. W. Tukey, *Exploratory data analysis*, vol. 2. Reading, MA, 1977.
- [6] Wikipedia contributors, “Hyperparameter optimization — Wikipedia, The Free Encyclopedia.” [https://en.wikipedia.org/wiki/Hyperparameter\\_optimization](https://en.wikipedia.org/wiki/Hyperparameter_optimization). [Online; accessed 30-June-2022].
- [7] Wikipedia contributors, “Cross-validation — Wikipedia, The Free Encyclopedia.” [https://en.wikipedia.org/wiki/Cross-validation\\_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics)). [Online; accessed 30-June-2022].
- [8] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.
- [9] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [10] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.
- [11] M. N. W. Stefano Nembrini, Inke R König, “The revival of the Gini importance?,” *Bioinformatics*, vol. 34, pp. 3711–3718, 2018.

- [12] Wikipedia contributors, “Surrogate model — Wikipedia, The Free Encyclopedia.” [https://en.wikipedia.org/wiki/Surrogate\\_model](https://en.wikipedia.org/wiki/Surrogate_model). [Online; accessed 30-June-2022].
- [13] T. Fawcett, “An introduction to ROC analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [14] D. Powers, “Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness and Correlation,” *Mach. Learn. Technol.*, vol. 2, 01 2008.
- [15] C. Goutte and É. Gaussier, “A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation,” in *ECIR*, 2005.
- [16] Wikipedia contributors, “Free-to-play — Wikipedia, The Free Encyclopedia.” <https://en.wikipedia.org/wiki/Free-to-play>. [Online; accessed 30-June-2022].
- [17] E. Lee, Y. Jang, D. M. Yoon, J. Jeon, S. i. Yang, S. K. Lee, D. W. Kim, P. P. Chen, A. Guitart, P. Bertens, Á. Periañez, F. Hadiji, M. Müller, Y. Joo, j. Lee, I. Hwang, and K. J. Kim, “Game data mining competition on churn prediction and survival analysis using commercial game log data,” *IEEE Transactions on Games*, vol. 11, no. 3, pp. 215–226, 2018.
- [18] R. Flunger, A. Mladenow, and C. Strauss, “Game Analytics on Free to Play,” in *Big Data Innovations and Applications* (M. Younas, I. Awan, and S. Benbernou, eds.), (Cham), pp. 133–141, Springer International Publishing, 2019.
- [19] B. Gregory, “Predicting Customer Churn: Extreme Gradient Boosting with Temporal Data,” *arXiv preprint arXiv: 1802.03396*, 2018.
- [20] M. Tamassia, W. Raffè, R. Sifa, A. Drachen, F. Zambetta, and M. Hitchens, “Predicting player churn in destiny: A Hidden Markov models approach to predicting player departure in a major online game,” in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, IEEE, 2016.
- [21] Á. Periañez, A. Saas, A. Guitart, and C. Magne, “Churn prediction in mobile social games: Towards a complete assessment using survival ensembles,” in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 564–573, IEEE, 2016.

- [22] J. Runge, P. Gao, F. Garcin, and B. Faltings, “Churn prediction for high-value players in casual social games,” in *2014 IEEE conference on Computational Intelligence and Games*, pp. 1–8, IEEE, 2014.
- [23] R. Sifa, F. Hadiji, J. Runge, A. Drachen, K. Kersting, and C. Bauckhage, “Predicting purchase decisions in mobile free-to-play games,” in *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.
- [24] H. Xie, S. Devlin, D. Kudenko, and P. Cowling, “Predicting player disengagement and first purchase with event-frequency based data representation,” in *2015 IEEE conference on Computational Intelligence and Games*, pp. 230–237, IEEE, 2015.
- [25] S. K. Lee, S. J. Hong, S. I. Yang, and H. Lee, “Predicting churn in mobile free-to-play games,” in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1046–1048, IEEE, 2016.
- [26] F. Hadiji, R. Sifa, A. Drachen, C. Thureau, K. Kersting, and C. Bauckhage, “Predicting player churn in the wild,” in *2014 IEEE Conference on Computational Intelligence and Games*, pp. 1–8, 2014.
- [27] L. Breiman, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [28] Y. Freund, R. Schapire, and N. Abe, “A short introduction to boosting,” *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.
- [29] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
- [30] N. Chinchor and B. M. Sundheim, “MUC-5 evaluation metrics,” in *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*, 1993.
- [31] M. Kubat, R. Holte, and S. Matwin, “Learning when negative examples abound,” in *European Conference on Machine Learning*, pp. 146–153, Springer, 1997.
- [32] A. Martínez, C. Schmuck, S. Pereverzyev Jr, C. Pirker, and M. Haltmeier, “A machine learning framework for customer purchase prediction in the non-contractual

- setting,” *European Journal of Operational Research*, vol. 281, no. 3, pp. 588–596, 2020.
- [33] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, “Top 10 algorithms in data mining,” *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37, 2008.
- [34] J. Brownlee, *Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, Cost-Sensitive Learning*. Machine Learning Mastery, 2020.
- [35] J. Davis and M. Goadrich, “The relationship between Precision-Recall and ROC curves,” in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240, 2006.
- [36] Wikipedia contributors, “Decision tree learning — Wikipedia, The Free Encyclopedia.” [https://en.wikipedia.org/wiki/Decision\\_tree\\_learning](https://en.wikipedia.org/wiki/Decision_tree_learning). [Online; accessed 30-June-2022].
- [37] M. Armbrust, R. S. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi, and M. Zaharia, “Spark sql: Relational data processing in spark,” in *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pp. 1383–1394, 2015.
- [38] A. Bilogur, “Missingno: a missing data visualization suite,” *Journal of Open Source Software*, vol. 3, no. 22, p. 547, 2018.
- [39] M. Waskom, O. Botvinnik, J. Ostblom, M. Gelbart, S. Lukauskas, P. Hobson, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. de Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, C. Swain, A. Miles, T. Brunner, D. O’Kane, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, and Brian, “mwaskom/seaborn: v0.10.1 (April 2020),” Apr. 2020.
- [40] J. Reback, W. McKinney, jbrockmendel, J. V. den Bossche, T. Augspurger, P. Cloud, gfyong, Sinhrks, A. Klein, M. Roeschke, S. Hawkins, J. Tratner, C. She, W. Ayd, T. Petersen, M. Garcia, J. Schendel, A. Hayden, MomIsBestFriend, V. Jancauskas, P. Battiston, S. Seabold, chris b1, h vetinari, S. Hoyer, W. Overmeire, alimcmaster1,

- K. Dong, C. Whelan, and M. Mehyar, “pandas-dev/pandas: Pandas 1.0.3,” Mar. 2020.
- [41] Wes McKinney, “Data Structures for Statistical Computing in Python,” in *Proceedings of the 9th Python in Science Conference* (Stéfan van der Walt and Jarrod Millman, eds.), pp. 56 – 61, 2010.
- [42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [43] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, “API design for machine learning software: experiences from the scikit-learn project,” in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.
- [44] Wikipedia contributors, “One-hot — Wikipedia, The Free Encyclopedia.” <https://en.wikipedia.org/wiki/One-hot>. [Online; accessed 30-June-2022].