

# Machine Learning Final Project

Team : Lab1126

李孟哲、李丞彥、呂雅芳

R10525102、R10525067、R10525062

## 1. 資料集探索

### 1.1. 介紹

本次機器學習期末專案使用電信公司顧客的數據資料，總共 7043 筆資料、50 項欄位；經過預處理刪除 CustomerID、Count、Total Long Distance Charges、Total Charges 欄位後，共 46 項欄位，訓練集資料 5634 筆，測試集資料 1409 筆。

資料集包含顧客基本資料、住處地區資料、服務滿意度資料、使用電信服務項目與體驗滿意度資料，其中欄位「Churn Category」為預測顧客繼續/停用本公司服務的原因；「No Churn」為繼續使用本公司服務，後四種類別為停用的原因，「Competitor」、「Dissatisfaction」、「Attitude」、「Price」、「Other」，總共五個類別。

### 1.2. 類別分布

首先將欄位「Churn Category」類別轉成數值 [0,5]，利用長條圖顯示如下圖：

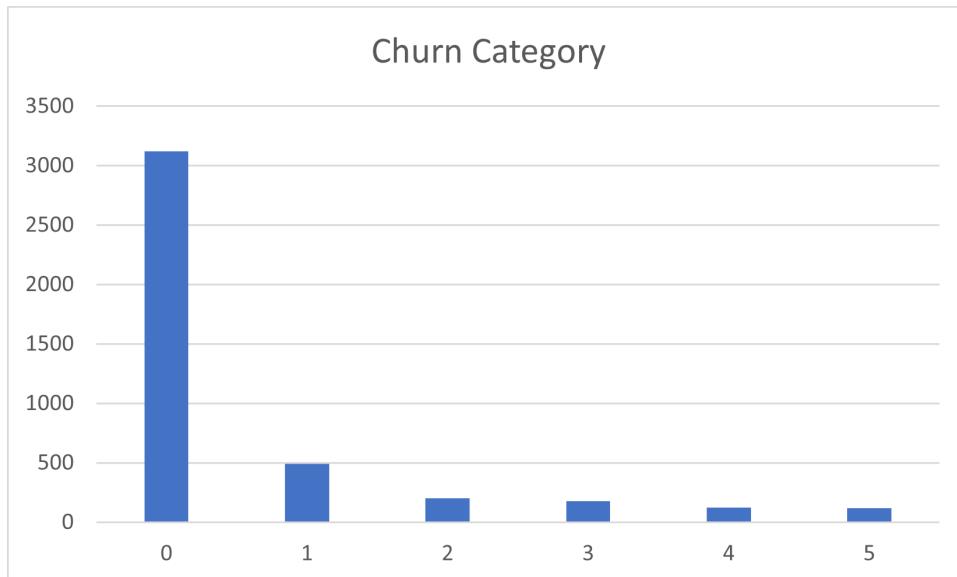


Table 1 Churn Category 各類別長條圖統計圖 ( 數值型態 )

可從圖中得知類別 0：「No Churn」總共有 3118 筆，顧客中最多的類別，為了訓練模型學習到每種類別，故將訓練集資料的類別數量調正為相同筆數，本組使用 Over Sample 處理資料類別不平衡問題

### 1.3. 預處理

預處理過後的資料可以提高模型訓練效果，可以降低訓練時間；資

料集有部分欄位缺值，大部分欄位可以透過資料的眾數以及平均數來補值；此外，本組會針對特定欄位進行條件式補值，以下列出本組補值的辦法：

### 1.3.1. Demographics

此表共有 6163 列 9 欄，紀錄顧客基本資料與人口統計數據。

- 「年齡」確認「是否小於 30 歲」欄位，如為是，則補 30 歲以下資料平均年齡，如為否；檢查「是否為年長者」欄位，如為是，則補 65 歲以上資料平均年齡，如為否；檢查「是否結婚」欄位，如為是，則補 30 歲以上且 65 歲以下結婚人口之平均年齡，如為否則補 30 歲以上且 65 歲以下未結婚人口平均年齡。
- 「結婚與否」判斷「年齡」欄位是否大於已接婚人口均年齡加上為結婚人口均年齡之平均。如是，填入已結婚，如否則填入為結婚
- 「是否有同居人」判斷「是否結婚」欄位，如為是則補是，如為否則補值為否；「同居人數」如遇缺值則檢查「結婚」，如為是則補同居人數之眾數，如為否則補 0

### 1.3.2. Location

資料集共有 9 個欄位，並皆用以表示顧客之居住地；在此資料集的前處理上，本組將原始資料集既存之 Zip Code 與 City 之對應關係和 Lat Long 與 City 之歸屬關係整理出來，並搭配 [BigDataCloud](#) 提供之 API 服務，以填補原始資料集之缺值；最後選擇了 Latitude 和 Longitude 作為居住地資訊代表進行訓練。

### 1.3.3. Services

資料集共有 20 個欄位，皆為顧客使用本公司服務項目與體驗滿意度資料，Internet Service 欄位則先確認其他相關網路欄位是否有值，如果有值則為 Yes，否則為 No。

經過觀察，本組發現 Services 資料集中的部分欄位存在相依性，其關聯如下：

$$MonthlyCharge \times TenureInMonths = TotalCharges$$

$$AvgMonthlyLongDistanceCharges \times TenureInMonths = \\ TotalLongDistanceCharges$$

$$TotalCharges + TotalLongDistanceCharges + TotalExtraDataCharges - \\ TotalRefunds = TotalRevenue$$

## 2. 實驗思路

### 2.1. Logistic Regression

於訓練之初，本組採用邏輯回歸演算法 (Logistic Regression) 搭配一對多 (OVA)、一對一 (OVO) 概念進行分類，試圖找出特徵 (features) 與類別 (label) 之間是否存在簡單的線性關係。

有鑑於部分顧客資料缺人口資料表 (demographics) 或 服務資料表 (services) 中的資料，本組嘗試使用三種特徵 (features) 組合進行訓練，分別為人口資料表 (demographics) 之特徵，服務資料表 (services) 之特徵，所有資料集之特徵，然而效果不彰，皆只有約 0.2 之準確度，**顯示出特徵與類別之關係並不全然線性可分或是存在雜質。**

### 2.2. SVM

根據邏輯回歸演算法 (Logistic Regression) 所得出之推論：資料可能含不少雜質或是不容易線性分割且資料特徵數較多，因此嘗試採用支撐向量分類演算法 (SVM)，搭配逕向基核函數核 (Radial Basis Kernel Function) 用意在於提高維度，試著求出較佳之分線。再者，為了避免過度擬合 (overfitting)，C 值設為 0.1，gamma 值則交由套件自動算出。且欲預測之類別為多類別之分類，分類上選擇一對多 (OVA) 產生  $N * (N - 1)/2$  共 10 種分類器選擇該資料最適合的分類。實驗結果如下：

$$E_{in} = 0.257, E_{val} = 0.168$$

預測類別	數量
0	1392
3	10
5	5
2	1
4	1

乍看之下似乎獲得不錯結果，Kaggle 上之 Fscore 却只有 0.14，且實際預測 TestID 類別下顯示出訓練出之支撐向量分類 (SVM) 模型在類別 0 以外之分類並沒有如預期中學習。

### 2.3. AdaBoost-Decision-Tree

由於支撐向量分類演算法 (SVM) 之分類方式或許過於強大，以致於無法學習到類別數量較少之分類，本組嘗試採用具條件式判斷的樹狀結構搭配自適應增強演算法 (AdaBoost) 方法聚合一些較弱之分類。

透過自適應增強演算法 (AdaBoost) 中的機制取出樣本 (sample)，

並使用決策樹 (decision tree) 訓練後再交由自適應增強演算法 (AdaBoost) 中的機制更新資料權重，迭代數輪後算出所需之數值 (alpha)。其中決策樹 (decision tree) 長得越高會導致  $E_{in}$  值等於 0 的情況，也代表著  $\varepsilon_t = 0 \Rightarrow \alpha = \infty$ ，故須限制決策樹 (decision tree) 高度，讓樹 (tree) 變成弱決策樹 (weak decision tree)，這些弱決策樹 (weak decision tree) 將會以投票 (voting) 方式來決定預測的結果。

在經過實驗後得到  $E_{in} = 0.145$ 、 $E_{val} = 0.275$ ，而  $Fscore$  上升至 0.29，Test ID 類別預測如下：

預測類別	數量
0	1122
1	222
2	28
3	25
5	6
4	6

預測分布上比起前述兩種演算法似乎更靠近了原訓練資料之分布狀況，雖準確度提昇但仍不盡理想，**惟自適應增強演算法 (AdaBoost) 對雜質 (noise) 影響極為敏感**，所以推測資料預處理需要更完善或是須改善演算法對雜質的容忍度。

#### 2.4. XGBoost

極度梯度增強演算法 (Extreme Gradient Boost) 為一種梯度增強樹狀結構算法 (Gradient Boost Decision tree) 的實現，不同於自適應增強演算法 (AdaBoost) 在訓練新一輪的樹時調整樣本錯誤權重，極度梯度增強演算法 (XGBoost) 目標函式中加入前 ( $t - 1$ ) 輪樹枝預測結果以保證新一輪的樹將更擬合資料集，此外，其在訓練上引入 Bagging 概念並在目標函式上加入了正則項以改善自適應增強演算法 (AdaBoost) 對雜質 (noise) 過度敏感的問題。

在實驗結果上，只須三層樹即可達到非常優異的  $E_{in} = 0$ ， $E_{val} = 0.001$ 。 $F1Score$  也略好於上述自適應增強演算法 (AdaBoost)。

### 3. 結果分析

經過上述實驗，以下我們將分析並比較其中三種較佳之算法。

#### 3.1. AdaBoost

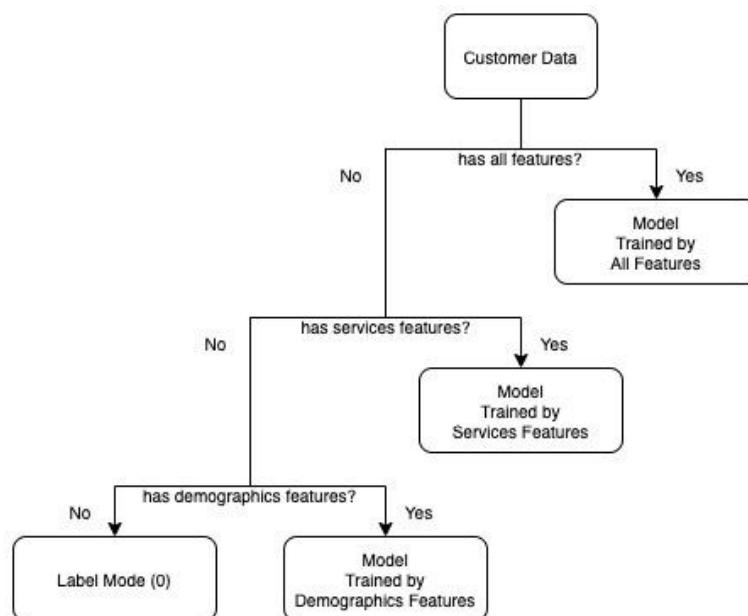
在使用 AdaBoost 算法的實驗中，本組找到最佳的超參數配置為  $algorithm = SAMME$ 、 $learning\_rate = 1$ 、 $max\_depth = 3$ 、 $n\_estimators = 1000$ 、（其餘使用預設值），此算法基於之弱分類器——Decision Stump 為 Decision Tree 的一種實現，因此有較佳的 Interpretability。

#### 3.2. XGBoost

在使用 XGBoost 算法的實驗中，本組找到最佳的超參數配置為  $max\_depth = 2$ 、（其餘使用預設值），此算法基於之弱分類器——Regression Decision Tree 同為 Decision Tree 的一種實現，因此也有較佳的 Interpretability；此外，**此算法支援平行化運算**，即使資料及較大，在訓練速度上依然表現優異。

#### 3.3. Self-Defined Decision Tree

此模型為本組自定義之 ensemble 概念，應用算法可任意抽換，其概念是發想自原始資料集中缺乏任一表格所有欄位資料之顧客，由於這些顧客缺乏過多資料，本組認為單純透過預處理補值也無法從模型得到好的預測，因此使用自定義三層樹架構以處理上述狀況，其架構如下圖所示：



在模型之決策上，由於透過實驗發現 Services 資料集與 label 關聯較強，因此條件設置為 Services 相關 features 之使用將優先於 Demographics 相關 feature。

	AdaBoost	XGBoost	Self-Defined Decision Tree (+ XGBoost)
$F1Score_{in}$	0.86931	0.90881	0.62954
$F1Score_{val}$	0.33603	0.38634	0.33555
$F1Score_{public}$	0.29665	0.28918	0.29499
$F1Score_{private}$	0.28275	0.32004	0.30939
<i>Efficiency (Training Time)</i>	20.3s	1.1s	1.7s
<i>Scalability</i>	X	O	O
<i>Interpretability</i>	O	O	O

於 Kaggle 競賽尾聲，本組選擇送出 AdaBoost 之預測結果，然而於 private leaderboard 公布後，其  $F1Score$  却不如預期，反觀基於 XGBoost 之模型分數上升，本組認為在解決方案之推薦上會選擇**基於 XGBoost 之 Self-Defined Decision Tree** 以進行預測以避免過度擬合與應對不同程度之資料缺失。

#### 4. 參考文獻

- [1] [AdaBoost](#)
- [2] [Logistic Regression](#)
- [3] [XGBoost](#)
- [4] [SVM](#)
- [5] [XGBoost: A Scalable Tree Boosting System](#)

#### 5. 工作分配

- 李孟哲(R10525102): 資料預處理(Demographics)、SVM 模型、Logistic Regression 模型、撰寫報告
- 李丞彥(R10525067): 資料預處理(Satisfaction、Services、Location)、Logistic Regression 模型、XGBoost 模型、撰寫報告
- 呂雅芳(R10525062): 資料預處理(Services)、Adaboost模型、Logistic Regression 模型、撰寫報告