國泰大數據競賽
複賽分析說明書

隊伍名稱：QoO
參賽者姓名：張家豪、曾立豪、林筱庭、李修逸

簡介

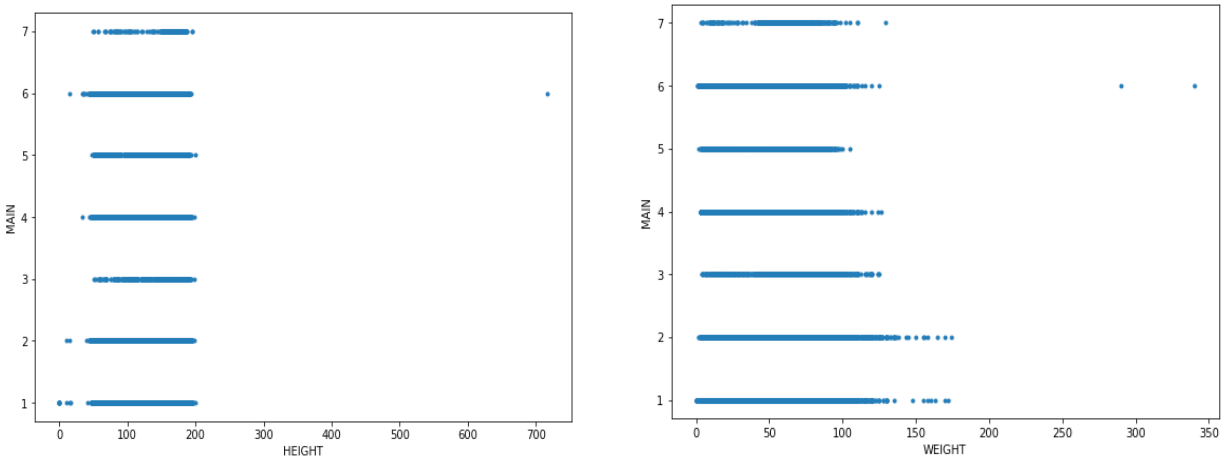
資料共分為六個部分，分別是訓練組被保人主約投保紀錄、訓練組被保人附約投保紀錄、測試組被保人主約投保紀錄、測試組被保人附約投保紀錄、被保人最新一筆投保保單當下資訊以及提交預測資料表格，而我們的建模目標即為透過客戶的個人資訊(身高、體重、職業、年收入...等)，再加上客戶曾經購買商品的歷史記錄來預測最新一期的購買大類，假如能夠透過這些資訊準確預測客戶很可能會買什麼商品的話，將其應用在銷售上，便可以達到精準行銷的目的，此即為將資料科學應用在在行銷上的方法。

資料處理與特徵選取

一、Data cleaning

1. EDA

(1)WEIGHT、HEIGHT：若身高和體重變數分開看，則只能個別看出異常值，但若從身高體重的組合(BMI)來看，則會找出從單一變數中看不出來的異常值。



2. 移除重複資料

在主約的訓練資料集中，發現有 1,642 筆一模一樣的資料，因此先將重複的資料移除。

3. 遺失值及異常值處理

首先檢查客戶資料集中是否有遺失值，整理如下表

變數名稱	NA 個數(比例)	變數名稱	NA 個數(比例)
HEIGHT	41893(11.5%)	PURPOSE_SAFE	5509(1.5%)
WEIGHT	41893(11.5%)	PURPOSE_LOAN	5509(1.5%)
DISEASE1	335654(92.1%)	PURPOSE_EDU	5509(1.5%)
DISEASE2	335654(92.1%)	PURPOSE_SAVING	5509(1.5%)
DISEASE3	335654(92.1%)	PURPOSE_TAX	5509(1.5%)

DISEASE4	335654(92.1%)	PURPOSE_OTHER	5509(1.5%)
SMOKES	5509(1.5%)	EDUCATION	27246(7.5%)
ARCEA	5509(1.5%)	CAREER	13(0.004%)
LIQUEUR	5509(1.5%)	NOW_INCOME	145048(39.8%)
VETERAN_STATUS	13899(3.8%)		

- HEIGHT、WEIGHT：新增 BMI 變數，利用 HEIGHT、WEIGHT 的資料，套入 BMI = 體重 / (身高/100)^2 公式計算可得，利用此變數主要目的在於找出異常的 BMI 值，並將其轉成遺失值，判定 BMI 是異常值的條件為: BMI 大於 50 或 BMI 小於等於 10。經過轉換後我們暫不處理身高與體重的遺失值，因為在後續即將使用的 XGBOOST 方法中會自動幫我們補齊遺失值。
- DESEASE1 ~ DESEASE4：資料中有高達 92.1%的遺失值，另外發現在有值的資料中全部都是 0，所以我們處理的方式為直接將 NA 值當作一個新的類別 1。
- PURPOSE_SAFE ~ PURPOSE_OTHER：我們發現在 PURPOSE_SAFE 全部的值皆為 1，而 PURPOSE_LOAN~PURPOSE_OTHER 全部的值皆為 0，所以一樣將 NA 值當作新的類別。
- SMOKES、ARCEA、LIQUEUR、VETERAN_STATUS、EDUCATION、CAREER：沒有特別作轉換，但是在後面 dummy 化的時候會自動將 NA 當作一個新的類別。
- NOW_INCOME：發現有一筆資料為-10000 的異常值，直接將其當作 NA，接下來和身高體重一樣暫不處理，直接交由 XGBOOST model 幫助我們補值。

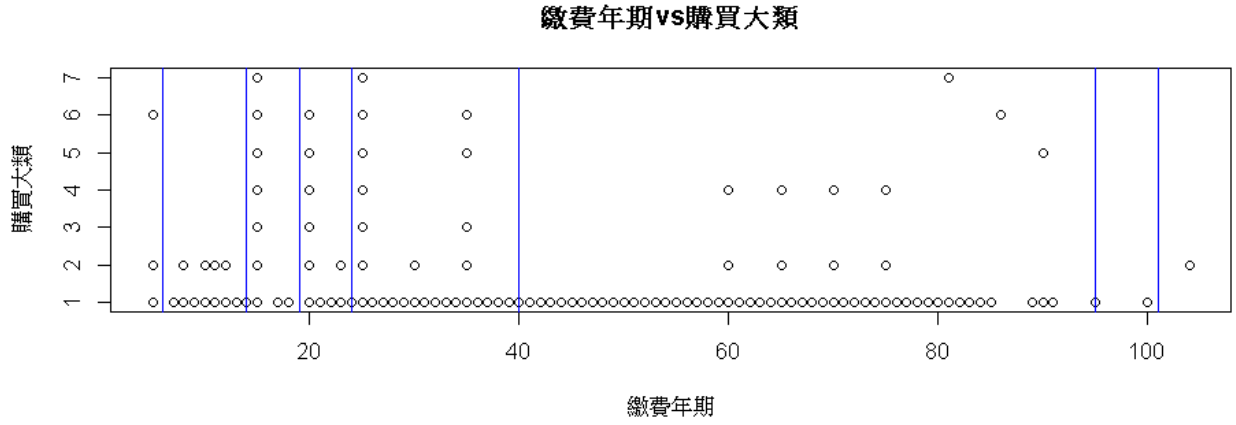
接下來檢查主約以及附約的訓練集是否有遺失值，發現只有郵遞區號這個變數有非常少的遺失值，由於這個變數類別數量過多，所以不作處理。

4. 變數轉換

(1)主約變數處理

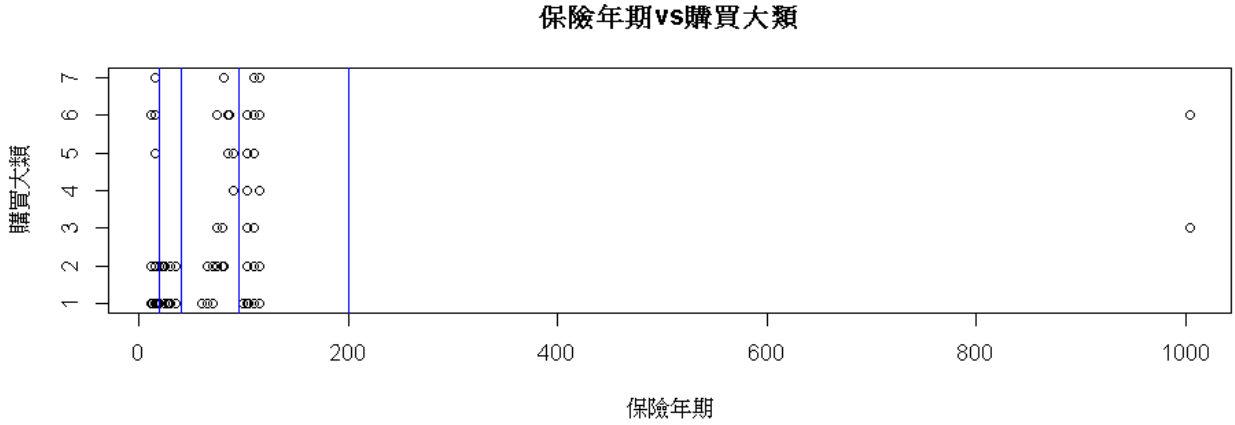
- 將主約中類代碼轉為主約大類代碼，資料中共記錄 16 種主約中類(1 到 17，不含 11)，屬於(1,2,5,13,14,15,16)中類者，轉為大類類別 1；屬於(3,4,7,8)中類者，轉為大類類別 2；中類代碼為 6 者轉為大類類別 3；中類代碼為 9 者轉為大類類別 4；中類代碼為 10 者轉為大類類別 5；中類代碼為 17 者轉為大類類別 6；中類代碼為 12 者轉為大類類別 7。
- CURRENCY：變更資料，將各幣別先查好匯率之後再轉為幣值，澳幣轉為 21.5；人民幣轉為 4.3；台幣轉為 1；美元轉為 30.5
- MAIN_AMOUNT_UNIT：變更資料，將保額單位轉為實際單位金額，元保留為 1；百元轉為 100；千元轉為 1000，將單位代號轉為 0 做另外處理。

- MAIN_PAY_PERIOD：由於原本資料中的主約繳費年期包含 85 種，所以我們嘗試看看主約繳費年期與購買大類之間的關係，嘗試把資料切割成比較少類別，透過下圖來看兩者間的散佈圖



根據觀察散布圖的結果，我們將原始 MAIN_PAY_PERIOD 小於 6 者轉為類別 1；大於等於 6 且小於 14 者轉為類別 2；大於等於 14 且小於 19 者轉為類別 3；大於等於 19 且小於 24 者轉為類別 4；大於等於 24 且小於 40 者轉為類別 5；大於等於 40 且小於 95 者轉為類別 6；大於等於 95 且小於 101 者轉為類別 7； 大於等於 101 者轉為類別 8。

- MAIN_PERIOD：與 MAIN_PAY_PERIOD 處理方法相同，原本資料中的主約保險年期共有 29 種，先以下圖觀察保險年期與購買大類之間的關係，最後將原始 MAIN_PERIOD 小於 20 者轉為類別 1；大於等於 20 且小於 40 者轉為類別 2；大於等於 40 且小於 95 者轉為類別 3；大於等於 95 且小於 200 者轉為類別 4；大於等於 200 者轉為類別 5。



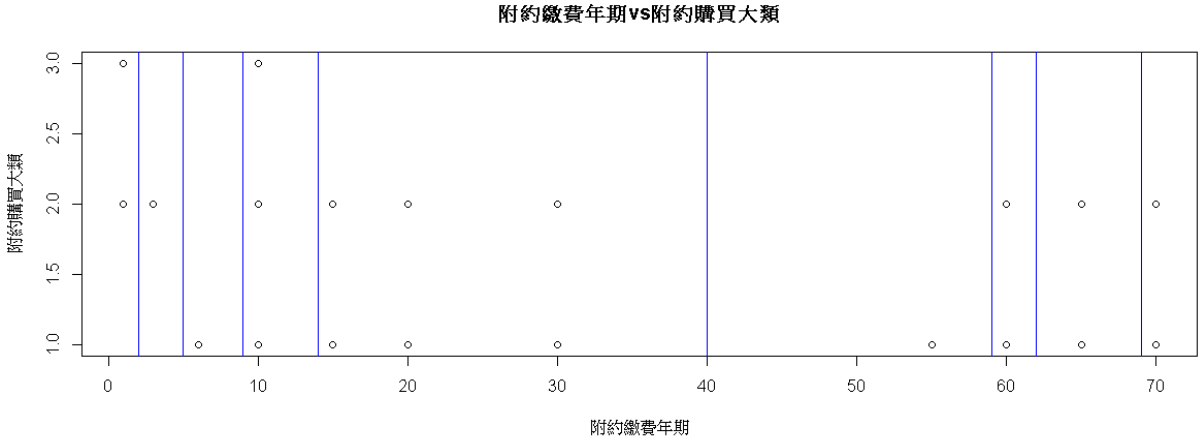
- 保額修正 ：利用原本資料上的 MAIN_AMOUNT 乘上剛剛有作轉換的保額單位以及幣別匯率作修正，避免資料單位不同的問題。
- 保費修正 ：利用原本資料上的 MAIN_PREMIUM 乘上幣別匯率作修正，避免資料單位不一樣的問題。

(2)附約變數處理

- 將附約中類代碼轉為附約大類代碼，資料中共記錄 6 種附約中類(6,7,9,10,12,17)，屬於(6,7,9,10)中類者，轉為大類類別 1；屬於 17 中類者，轉為大類類別 2；中類代碼為 12 者轉為大類類別 3。
- ADD_AMOUNT_UNIT：變更資料，將保額單位轉為實際單位金額，元保

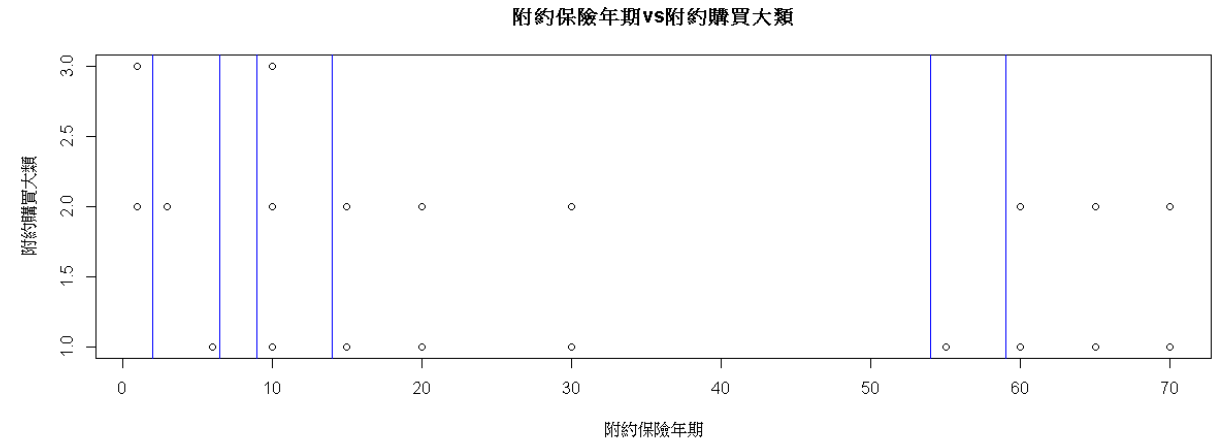
留為 1；百元轉為 100；千元轉為 1000，將單位代號轉為 0 做另外處理。

- ADD_PAY_PERIOD：由於原本資料中的附約繳費年期包含 11 種，所以我們嘗試看看主約繳費年期與購買大類之間的關係，嘗試把資料切割成比較少類別，透過下圖來看兩者間的散佈圖



根據觀察散布圖的結果，我們將原始 ADD_PAY_PERIOD 小於 2 者轉為類別 1；大於等於 2 且小於 5 者轉為類別 2；大於等於 5 且小於 9 者轉為類別 3；大於等於 9 且小於 14 者轉為類別 4；大於等於 14 且小於 40 者轉為類別 5； 大於等於 40 且小於 59 者轉為類別 6；大於等於 59 且小於 62 者轉為類別 7；大於等於 62 且小於 69 者轉為類別 8；大於等於 69 者轉為類別 9。

- ADD_PERIOD：與 ADD_PAY_PERIOD 處理方法相同，原本資料中的附約保險年期共有 14 種，先以下圖觀察保險年期與購買大類之間的關係，最後將原始 ADD_PERIOD 小於 2 者轉為類別 1；大於等於 2 且小於 6.5 者轉為類別 2；大於等於 6.5 且小於 9 者轉為類別 3；大於等於 9 且小於 14 者轉為類別 4；大於等於 14 且小於 54 者轉為類別 5；大於等於 54 且小於 59 者轉為類別 6；大於等於 59 且小於 90 者轉為類別 7；大於等於 90 且小於 101 者轉為類別 8；大於等於 101 者轉為類別 9。



- 附約保額修正：利用原本資料上的 ADD_AMOUNT 乘上剛剛有作轉換的保額單位作修正，避免資料單位不同的問題。

二、Data processing

(1)主約產生新變數

先利用 CUST_ID 將主約保單資料與客戶資料集做合併，此時被保人可能會有無主約的情況，或者一筆至多筆主約資料，我們處理的方式是將每個被保人的所有保單資料合併成一個，最終每個被保人僅會保留一筆整理後的主約資訊。

在合併過程中，會產生許多新變數如下：

- PAST_BUY_TYPE1 ~ PAST_BUY_TYPE7：截至目前為止，累計每個客戶在過去的購買記錄中，買過第 i 大類(i=1,2,...,7)的總個數。此外，沒有歷史記錄的客戶(無投保紀錄的新客戶)的值記為 0。
- AMOUNT_SUM1 ~ AMOUNT_SUM7：將已經用幣別及保額單位修正過後的保額資料，計算每個客戶曾經購買第 i 類商品(i=1,2,...,7)之保額總和。此外，沒有歷史記錄的客戶(無投保紀錄的新客戶)的值記為 0。
- PREMIUM_1 ~ PREMIUM_7：將已經用幣別修正過後的保費資料，計算每個客戶曾經購買第 i 類商品(i=1,2,...,7)之保費總和。此外，沒有歷史記錄的客戶(無投保紀錄的新客戶)的值記為 0。
- CURRENCY_1 ~ CURRENCY_4：分別為每個客戶在該幣別的累計總數，CURRENCY_1 為累計台幣別總數；CURRENCY_2 為累計人民幣別總數； CURRENCY_3 為累計美金別總數； CURRENCY_4 為累計澳幣別總數。
- PAY_FREQ1 ~ PAY_FREQ6：分別為每個客戶在該繳費頻率的累計總數，PAY_FREQ1 為月繳；PAY_FREQ2 為季繳；PAY_FREQ3 為半年繳；PAY_FREQ4 為年繳；PAY_FREQ5 為躉繳；PAY_FREQ6 為彈性繳。
- PAY_WAY1 ~ PAY_WAY5：新增的變數，將原始的繳費管道類別，轉為被保人分別在這 1,2,3,4,6(共有 5 類)方式的繳費管道中的累積總個數。PAY_WAY1 為人工件；PAY_WAY2 為扣薪件；PAY_WAY3 為轉帳；PAY_WAY4 為信用卡；PAY_WAY5 為自繳。
- SALE_CHANNEL1 ~ SALE_CHANNEL5：將原始的 5 種購買管道(0,1,2,3,4)，轉為被保人分別在這 5 種購買管道中的累積總個數。SALE_CHANNEL1 為業務通路；SALE_CHANNEL2 為 OUTBOUND；SALE_CHANNEL3 為 INBOUND；SALE_CHANNEL4 為銀行保代；SALE_CHANNEL5 為其他。
- BUY_YEAR1 ~ BUY_YEAR13：把被保人的購買年記錄之次數累積，BUY_YEAR1 為被保人在 2006 年有紀錄的累積次數，BUY_YEAR2 為被保人在 2007 年有紀錄的累積次數，依此類推至，BUY_YEAR13 為被保人在 2018 年有紀錄的累積次數。
- MAIN_PERIOD1 ~ MAIN_PERIOD5：把切割後的 MAIN_PERIOD 變數，重新拆成對被保人個別累計 1,2...,5 類的 MAIN_PERIOD 個數。也就

是說，MAIN_PERIOD1 是類別 1 的累積總個數，MAIN_PERIOD2 是類別 2 的累積總個數，MAIN_PERIOD3 至 MAIN_PERIOD5 依此類推。

- MAIN_PAY_PERIOD1 ~ MAIN_PAY_PERIOD8：把切割後的 MAIN_PAY_PERIOD 變數，重新拆成對被保人個別累計 1,2...,8 類的 MAIN_PAY_PERIOD 個數。也就是說， MAIN_PAY_PERIOD1 是類別 1 的累積總個數，MAIN_PAY_PERIOD2 是類別 2 的累積總個數，MAIN_PAY_PERIOD3 至 MAIN_PAY_PERIOD8 依此類推。
- Unit7：將保額單位類別為 7 的獨立出來，再計算每個客戶累計保額單位為 7 的個數。

(2)附約產生新變數：

附約方面一樣先利用 CUST_ID 將主約保單資料與客戶資料集做合併，將每個被保人的所有保單資料合併成一個，最終每個被保人僅會保留一筆整理後的主約資訊。在合併過程中，會產生許多新變數如下：

- ADD_AMOUNT_SUM1 ~ ADD_AMOUNT_SUM3：將已經用保額單位修正過後的附約保額資料，計算每個客戶曾經購買第 i 類附約商品(i=1,2,3)之保額總和。此外，沒有歷史記錄的客戶(無投保紀錄的新客戶)值記為 0。
- ADD_PREMIUM_SUM1 ~ ADD_PREMIUM_SUM3：計算每個客戶曾經購買第 i 類附約商品(i=1,2,3)之保費總和。此外，沒有歷史記錄的客戶(無投保紀錄的新客戶)的值記為 0
- ADD_PAY_PERIOD1 ~ ADD_PAY_PERIOD9：把切割後的 ADD_PAY_PERIOD 變數，重新拆成對被保人個別累計 1,2...,9 類的 ADD_PAY_PERIOD 個數。也就是說， ADD_PAY_PERIOD1 是類別 1 的累積總個數，ADD_PAY_PERIOD2 是類別 2 的累積總個數，ADD_PAY_PERIOD3 至 ADD_PAY_PERIOD9 依此類推。
- ADD_PERIOD1 ~ ADD_PERIOD9：把切割後的 ADD_PERIOD 變數，重新拆成對被保人個別累計 1,2...,9 類的 ADD_PERIOD 個數。也就是說， ADD_PERIOD1 是類別 1 的累積總個數， ADD_PERIOD2 是類別 2 的累積總個數， ADD_PERIOD3 至 ADD_PERIOD9 依此類推。
- ADD_Unit7： 將保額單位類別為 7 的獨立出來，再計算每個客戶累計保額單位為 7 的個數。

三、Feature selection
(1)主約模型最後使用變數

變數名稱	資料型態
POLICY_ID	連續型
BUY_MONTH	連續型
BUY_YEAR	連續型
HEIGHT	連續型
WEIGHT	連續型
DISEASE1 ~ DISEASE4	類別型
MARRIAGE	類別型
SMOKES、ARCEA、LIQUEUR	類別型
VETERAN_STATUS	類別型
EDUCATION	類別型
CAREER	類別型
PURPOSE_SAFE、PURPOSE_LOAN、PURPOSE_EDU、PURPOSE_SAVING、PURPOSE_TAX	類別型
NOW_INCOME	連續型
NOW_CHILD	連續型
PARENTS_DEAD	類別型
REAL_ESTATE_HAVE	類別型
IS_MAJOR_INCOME	類別型
AGE	連續型
SEX	類別型
PAST_BUY_TYPE1 ~ PAST_BUY_TYPE7	連續型
unit7	類別型
AMOUNT_SUM1 ~ AMOUNT_SUM7	連續型
CURRENCY_1 ~ CURRENCY_4	連續型
PREMIUM_1 ~ PREMIUM_7	連續型
PAY_FREQ1 ~ PAY_FREQ6	連續型
PAY_WAY1 ~ PAY_WAY5	連續型
SALE_CHANNEL1 ~ SALE_CHANNEL5	連續型

BUY_YEAR1 ~ BUY_YEAR13	連續型
MAIN_PERIOD1 ~ MAIN_PERIOD5	連續型
MAIN_PAY_PERIOD1 ~ MAIN_PAY_PERIOD8	連續型

(2)附約模型最後使用變數

利用上述主約的資料再加上以下變數作為附約的解釋變數。

變數名稱	資料型態
ADD_AMOUNT_SUM1 ~ ADD_AMOUNT_SUM3	連續型
add_unit7	連續型
ADD_PREMIUM_SUM1 ~ ADD_PREMIUM_SUM3	連續型
ADD_PAY_PERIOD_1 ~ ADD_PAY_PERIOD_9	連續型
ADD_PERIOD_1 ~ ADD_PERIOD_9	連續型

(3)將上述屬於類別型的變數改成 dummy variables，即為我們的最終解釋變數。

模型選擇與驗證成效說明

一、模型選擇

(1)模型簡介

我們在此次分析中選擇使用 Extreme Gradient Boosting(XGBOOST)當作我們的預測方法。

XGBoost 屬於一種改良的 Gradient tree boosting。

Gradient tree boosting 是一種 additive function，使用多個弱分類器來建構一個強分類器，建構的方式為使用前 k-1 次迭代結果的負梯度(negative gradient)當作新的反應變數進行下一次迭代(使用 tree 的方式)，產生新的弱分類器(tree)並加到前一次的估計函數上當新的估計函數，把眾多迭代結果相加即為最終估計函數。

而 XGBoost 對 gradient tree boosting 的改進有以下幾項:

- 避免 overfitting : 在目標函數(loss function)中加上懲罰項，用來權衡目標函數的下降與模型的複雜程度，藉此避免 overfitting，其中懲罰項和樹的葉節點數量以及葉節點的值有關。
- 使用二階 Taylor expansion : 傳統的 gradient boosting 只有利用一階的導數資訊，而 XGBoost 對 loss function 做了二階的 Taylor expansion，精度更高。
- 樹節點分裂優化：在尋找最佳分割點時，考慮傳統的列舉每個特徵的所有可能分割點的貪心法效率太低，XGboost 利用了一種近似的演算法，即：根據百分位法列舉幾個可能成為分割點的候選人，然後從候選人中根據求分割點的公式計算找出最佳的分割點。

- XGboost 內置處理缺失值的規則。

(2) xgboost function 參數說明 (in R)：

- eta : learning rate ，減少每次迭代移動的步長，可以避免 overfitting
- min_child_weight : 決定最小葉節點樣本權重和
- max_depth : 樹的最大深度
- subsample : 控制對於每棵樹隨機抽樣的比例
- colsample_bytree : 控制每棵隨機抽樣的行數的占比
- objective : 定義需要被最小化的 loss function，在主約分析中用 multi:softmax(用於多類別)；在附約分析中用 binary:logistic(用於二元分類)
- eval_metric : 對於模型的評估指標，在主約分析中用 merror(多分類錯誤率)；在附約分析中用 error(二元分類錯誤率)

(3)主約模型

使用 R 的 xgboost package，解釋變數見上一節所述，反應變數為訓練集中最新一筆主約購買資訊的購買大類，共有 7 種類別，由於是多類別的反應變數，所以設定 xgb function 之參數 objective 為 multi:softmax；eval_metric 則為 merror。

(4)附約模型

在附約中，由於我們想要預測的是在最新一筆購買紀錄中是否有購買第 i 大類附約(i=1,2,3)，所以我們分為 3 個 model 處理，這三個 model 的解釋變數相同(見上一節)，反應變數則取最新一筆附約購買紀錄是否有購買第 i 類附約，所以我們會有三個反應變數為二元分類的模型，分別對他們做預測，在此部分設定 xgb function 的參數 objective 為 binary:logistic；eval_metric 的參數為 error。

二、驗證成效說明

(1)建立驗證集資料

首先，我們先從訓練集中切出一組驗證集，切的方式為，先從訓練集資料找出每個客戶的最新一筆主約購買資訊，再從這 354345 個客戶中隨機挑選百分之三十作為驗證用的 testing data。

(2)搜尋最佳參數之方法以及驗證結果

利用 grid search 加上 K-fold 交叉驗證法(設定 K=8)，測試多種參數組合，選出一組最佳的參數組合(CV error 最小)，再套入驗證集確認模型的準確度。

主約嘗試調整的參數包含 ：max_depth、eta、min_child_weight

- Cross-Validation : 嘗試過的參數組合結果請詳見附錄
- 最後選擇參數 ： max_depth = 10、eta = 0.04 、 min_child_weight = 4

- Confusion matrix

	PREDICT							
TRUE		**1**	**2**	**3**	**4**	**5**	**6**	**7**
	1	42171	8364	243	218	33	518	1
	2	9904	21341	53	413	22	534	2
	3	398	16	1697	0	1	2	0
	4	4260	3176	7	889	19	304	0
	5	1070	657	4	47	59	50	0
	6	4121	3361	6	238	25	1804	0
	7	148	104	2	3	0	15	4

預測準確率 ： 63.94%

附約嘗試調整的參數包含 ：max_depth、eta

Model 1

- 最後選擇參數 ：max_depth = 15、 eta = 0.03

	PREDICT		
TRUE		**1**	**0**
	1	99602	1097
	0	4095	1510

預測準確率 ： 95.12%

Model 2

- Cross-Validation：嘗試過的參數組合結果請詳見附錄
- 最後選擇參數 ：max_depth = 9、 eta = 0.05
- Confusion matrix

	PREDICT		
TRUE		**1**	**0**
	1	70166	7772
	0	8628	19738

預測準確率 ： 84.57%

Model 3

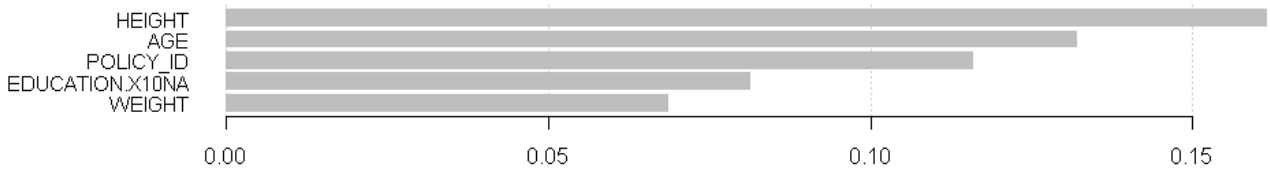
- 最後選擇參數 : max_depth = 15 、 eta = 0.04
- Confusion matrix

	PREDICT		
TRUE		**1**	**0**
	1	70399	7498
	0	11578	16829

預測準確率 ： 82.06%

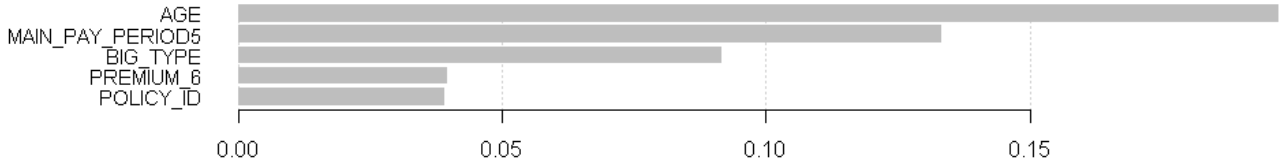
(3)變數重要性
最後我們利用 XGBoost package 的 xgb.importance 函數來看各模型中解釋變數的重要性

- 主約



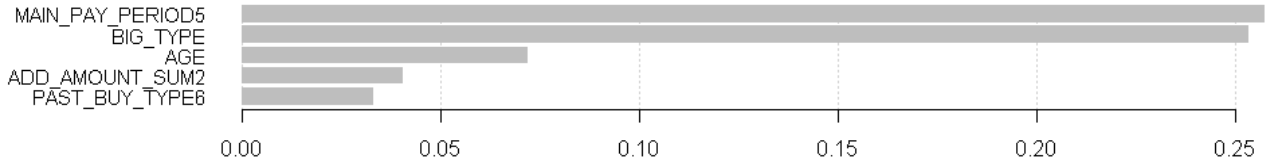
在主約模型中前五重要的變數為身高、年齡、保單 ID、教育程度的 NA 類、體重，保單號碼由於與購買時間有關(保單號碼越高表示越晚購買)，所以出現在裡面還算合理。

- 附約第一大類



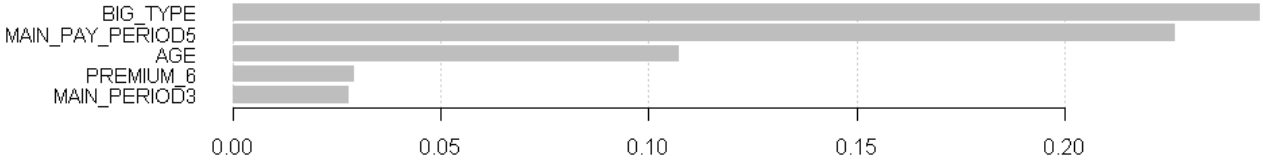
在附約第一大類模型中，前五重要的變數分別為年齡、主約繳費年期第五類累計、主約大類、主約第六大類保費累計、保單號碼。

- 附約第二大類



附約第二大類模型中，前五重要的變數包含主約繳費年期第五類累計、主約大類、年齡、附約第二類保額累計、曾購買主約第六大類總數。

- 附約第三大類



附約第二大類模型中，變數重要性前五的變數有主約大類、主約繳費年期第五類累計、年齡、 主約第六大類保費累計、主約保險年期第三類累計

結論

從主約的預測結果我們可以看到，購買大類 4 到購買大類 7 的預測結果是比較差的，原因可能是因為前 3 類就佔原本訓練集比例的 80%左右，而第 4 類到第 7 類僅有 20%，因為時間不足所以我們還沒做到修正資料比例不平衡後是否能得到更佳的預測結果。至於附約因為是二元分類，所以我們的 model 都可以達到不錯的預測準確率。

附錄

1. 主約 cross validation 嘗試組合整理

```
[1] "eta: 0.03 ,max_depth: 4 min_child_weight: 1 | test: 0.387472625 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 4 min_child_weight: 2 | test: 0.3877295 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 4 min_child_weight: 3 | test: 0.388034125 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 4 min_child_weight: 4 | test: 0.387749375 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 5 min_child_weight: 1 | test: 0.383061875 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 5 min_child_weight: 2 | test: 0.383169 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 5 min_child_weight: 3 | test: 0.38325925 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 5 min_child_weight: 4 | test: 0.38301375 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 6 min_child_weight: 1 | test: 0.378371375 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 6 min_child_weight: 2 | test: 0.37888225 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 6 min_child_weight: 3 | test: 0.378489625 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 6 min_child_weight: 4 | test: 0.378758125 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 7 min_child_weight: 1 | test: 0.3749565 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 7 min_child_weight: 2 | test: 0.374756125 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 7 min_child_weight: 3 | test: 0.3748155 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.03 ,max_depth: 7 min_child_weight: 4 | test: 0.37467975 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 4 min_child_weight: 1 | test: 0.383087125 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 4 min_child_weight: 2 | test: 0.38323675 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 4 min_child_weight: 3 | test: 0.383341125 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 4 min_child_weight: 4 | test: 0.3830505 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 5 min_child_weight: 1 | test: 0.379709 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 5 min_child_weight: 2 | test: 0.3798445 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 5 min_child_weight: 3 | test: 0.379742875 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 5 min_child_weight: 4 | test: 0.379821875 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 6 min_child_weight: 1 | test: 0.37612775 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 6 min_child_weight: 2 | test: 0.375949875 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 6 min_child_weight: 3 | test: 0.37602075 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 8 min_child_weight: 4 | test: 0.370418625 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 9 min_child_weight: 4 | test: 0.368544875 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.04 ,max_depth: 10 min_child_weight: 4 | test: 0.36761625 nround: 100"
[1] "-------------------------------------------------------------"
[1] "eta: 0.05 ,max_depth: 8 min_child_weight: 4 | test: 0.368911625 nround: 99"
[1] "-------------------------------------------------------------"
```

2. 附約模型二 cross validation 嘗試組合整理

eta	max_depth	CV error	eta	max_depth	CV error
0.02	4	0.1630	0.04	4	0.1605
0.02	5	0.1591	0.04	5	0.1570
0.02	6	0.1579	0.04	6	0.1552
0.02	7	0.1560	0.04	7	0.1535
0.02	8	0.1543	0.04	8	0.1519
0.02	9	0.1526	0.04	9	0.1511
0.03	4	0.1617	0.05	4	0.1598
0.03	5	0.1581	0.05	5	0.1565
0.03	6	0.1560	0.05	6	0.1542
0.03	7	0.1541	0.05	7	0.1527
0.03	8	0.1527	0.05	8	0.1514
0.03	9	0.1514	**0.05**	**9**	**0.1504**