資料科學導論競賽報告

H34071039 工資系 111 李若瑜
H34074087 工資系 111 何彩綺
H34076039 工資系 111 黃振維

競賽敘述與目標

本次競賽以銀行客戶資料做為資料集，分析其中欄位並預測銀行是否會流失該客戶，即未來不再於銀行進行交易。使用的資料集分別為：訓練數據集 (train.csv) 和測試數據集 (test.csv)，預測目標為 Exited 欄位。上傳 upload.csv 檔案至競賽網站，最後評分以三個評估指標 Accuracy (30%)、Precision (30%) 和 F-Score (40%) 為標準，數值越高者為佳。

資料前處理

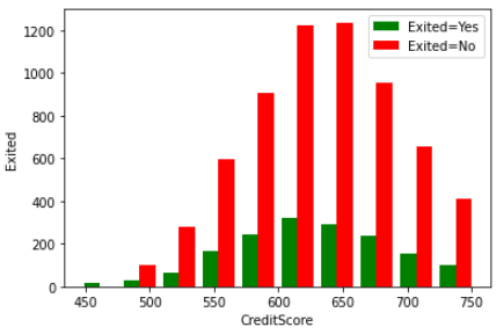
我們運用了 MinMaxScaler 以及 OneHotEncoder 方法。使用 MinMaxScaler 給定一個明確的最大值與最小值，每個特徵中的最小值變成了 0，最大值變成了 1，數據會縮放到到[0,1]之間。OneHotEncoder 的編碼邏輯為將類別拆成多個 column，每個列中的數值由 1、0 替代，當某一列的資料存在的該行的類別則顯示 1，反則顯示 0。

特徵處理與分析

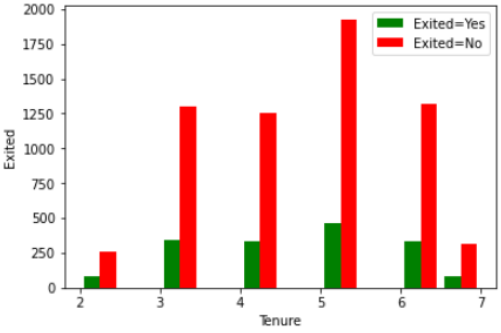
(1) 刪除不影響 Exited 的三個欄位 RowNumber、CustomerId、Surname。此三種 attribute 並不影響任何結果，都僅僅代表一個編號而已，故刪除。

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	720	Spain	Male	38	5	114051.97	2	0	1	107577.29	0
1	682	France	Female	54	4	62397.41	1	1	0	113088.60	1
2	672	France	Female	31	5	119903.67	1	1	1	132925.17	0
3	592	Spain	Female	40	4	104257.86	1	1	0	110857.33	0
4	753	Spain	Male	42	5	120387.73	1	0	1	126378.57	0
...	...	...	...	...	...	...	...	...	...	...	...
7995	568	France	Female	35	6	121079.60	2	1	1	124890.50	1
7996	602	Germany	Female	45	7	145846.07	1	1	0	99276.02	0
7997	679	Spain	Female	43	5	132810.01	1	1	0	130780.85	1
7998	715	France	Male	38	4	118729.45	1	0	0	95484.52	0
7999	600	France	Female	42	5	62397.41	1	0	0	66315.00	0

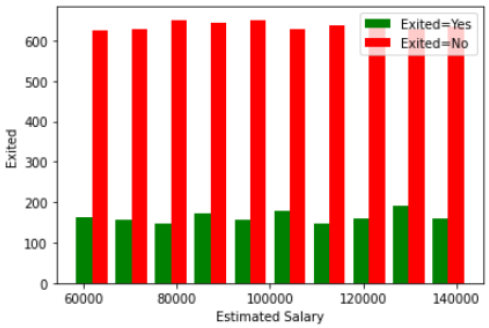
(2) 分析剩餘欄位與 Exited 的關係分佈，以直方圖呈現。若是 No 與 Yes 大致上成比例則與 x 軸之 attribute 無關，若不成比例則可能有關。綜合下面的直方圖，發現 age 跟 balance 比較有關，較有可能影響到 exited 的值。



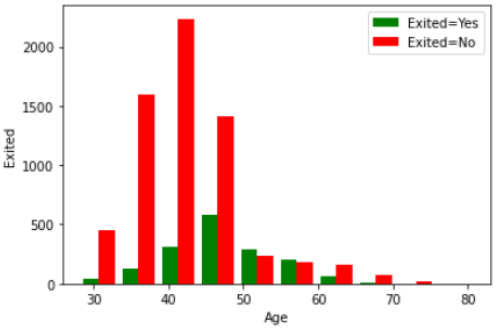
▲ Credit vs. Exited



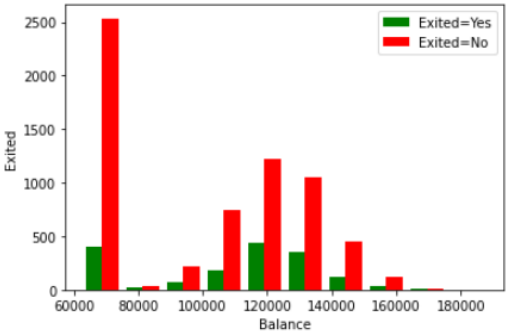
▲ Tenure vs. Exited



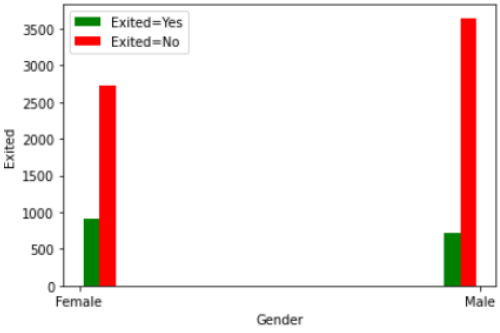
▲ Estimated Salary vs. Exited



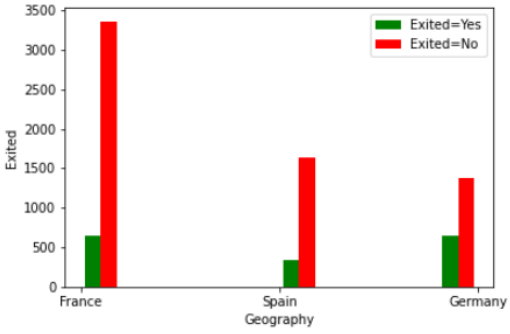
▲ Age vs. Exited



▲ Balance vs. Exited

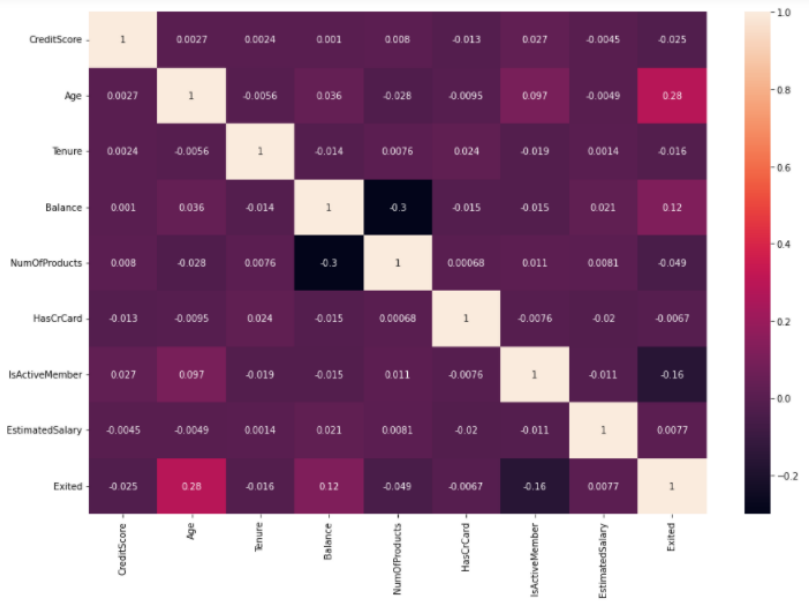


▲ Gender vs. Exited



▲ Geography vs. Exited

(3)以 heatmap 的方式呈現



▲ 各欄位之間共變異數

(4)將每一行最大值設為 1，最小值設為 0

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_Germany	Geography_Spain	Gender_Female	Gender_Male
0	0.892508	0.188679	0.6	0.409621	0.333333	0	1	0.589163	0	0	1	0	1
1	0.768730	0.490566	0.4	0.000000	0.000000	1	0	0.654032	1	0	0	1	0
2	0.736156	0.056604	0.6	0.456025	0.000000	1	1	0.887514	0	0	0	1	0
3	0.475570	0.226415	0.4	0.331954	0.000000	1	0	0.627770	0	0	1	1	0
4	1.000000	0.264151	0.6	0.459864	0.000000	0	1	0.810459	0	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7995	0.397394	0.132075	0.8	0.465350	0.333333	1	1	0.792944	1	0	0	1	0
7996	0.508143	0.320755	1.0	0.661749	0.000000	1	0	0.491455	0	1	0	1	0
7997	0.758958	0.283019	0.6	0.558373	0.000000	1	0	0.862275	1	0	1	1	0
7998	0.876221	0.188679	0.4	0.446714	0.000000	0	0	0.446828	0	0	0	0	1
7999	0.501629	0.264151	0.6	0.000000	0.000000	0	0	0.103495	0	0	0	1	0

▲ 清理資料後之結果

(5)drop 掉 Exited

(6)訓練模型

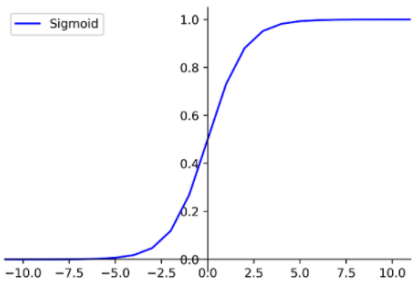
```
Epoch 1/50
200/200 [==============================] - 0s 609us/step - loss: 0.5598 - accuracy: 0.7270
Epoch 2/50
200/200 [==============================] - 0s 630us/step - loss: 0.4782 - accuracy: 0.7975
Epoch 3/50
200/200 [==============================] - 0s 702us/step - loss: 0.4672 - accuracy: 0.8023
Epoch 4/50
200/200 [==============================] - 0s 605us/step - loss: 0.4598 - accuracy: 0.8014
Epoch 5/50
200/200 [==============================] - 0s 632us/step - loss: 0.4525 - accuracy: 0.8072
Epoch 6/50
200/200 [==============================] - 0s 633us/step - loss: 0.4461 - accuracy: 0.8081
Epoch 7/50
200/200 [==============================] - 0s 642us/step - loss: 0.4404 - accuracy: 0.8120
Epoch 8/50
200/200 [==============================] - 0s 663us/step - loss: 0.4348 - accuracy: 0.8152
Epoch 9/50
200/200 [==============================] - 0s 624us/step - loss: 0.4318 - accuracy: 0.8145
Epoch 10/50
200/200 [==============================] - 0s 633us/step - loss: 0.4256 - accuracy: 0.8178
Epoch 11/50
200/200 [==============================] - 0s 749us/step - loss: 0.4210 - accuracy: 0.8195
Epoch 12/50
200/200 [==============================] - 0s 704us/step - loss: 0.4161 - accuracy: 0.8213
```

▲ 部分訓練結果

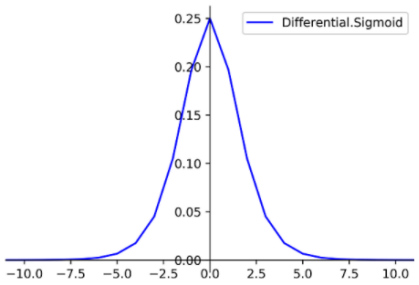
預測訓練模型

(一) 最終模型—ANN

本組使用 Python 的 Keras、Tensorflow 模組建立神經網路以預測最終結果。由於訓練資料量不算很大，本組僅用一層 input layer、一層 hidden layer 和一層 output layer。Activation function 分別使用 relu 與 sigmoid，Sigmoid 函數是深度學習領域開始時使用頻率最高的 activation function，但容易出現梯度消失，ReLU 是近年來最頻繁被使用的激勵函數，因其存在以下特點，包含解決梯度爆炸問題、計算數度相當快、收斂速度快等特性。我們使用的優化器為 Adam，Adam 擁有收斂速度快、調參容易的優點，由於 exited 結果為二項分類，因此 loss 函數使用 binary crossentropy。在建立神經網路模型後便開始訓練的程序，epoch 代表一個訓練的 iteration，我們指定此參數為 50。在調整參數的過程中我們發現當層數及經元數太多會出現 overfitting 的狀況，使用的 activation function 與優化器的不同也會影響準確率。

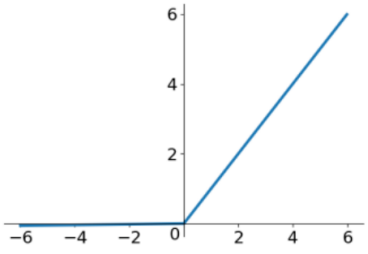


Sigmoid

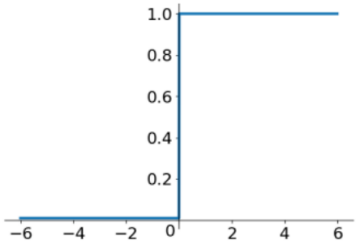


Sigmoid導數

$\mathrm{ReLU} = \max(0.01x, x)$



ReLU



ReLU導數

(二) 最初模型—sklearn

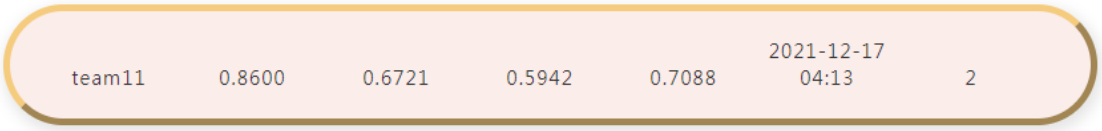
本組一開始使用 sklearn 作為預測模型。運用 LabelEncoder 和 StandardScaler 做前處理，並用 LogisticRegression 處理二元分類問題。首先計算 Logit(Odds)，勝算比取對數 log，產生 y 值，經過函數轉換器，像是 Sigmoid 函數、Arctan(X)等等，再將 y 值帶入函數轉換公式化，並產生最終結果比，大於 50%機率的會被預測為 1，小於 50%會被預測為 0。但由於 final score 結果不慎理想，調整參數並嘗試新的模型。

預測結果分析

在執行檔案中我們用 sklearn 的 confusion_matrix 和 classification_report，計算出 Exited 中 0、1 的預測結果，precision、recall、f1-score 的值。而競賽網站中的最終結果如下圖，雖然 accuracy 達 0.86，但是 precision 和 fScore 的值不佳，以致最後 final 加權分數未如我們預期般高。對應檔案中分析出的 Exited 中 0、1 結果，0 的預測結果比 1 的預測結果準確許多。我們認為可以再針對每個欄位做深入分析，加上多層神經網路，或嘗試其他模型，期以後續能增加預測準確性。

	precision	recall	f1-score	support
0	0.88	0.95	0.91	1262
1	0.73	0.50	0.59	338
accuracy			0.85	1600
macro avg	0.80	0.72	0.75	1600
weighted avg	0.85	0.85	0.84	1600

▲ 執行檔預測結果



▲ 競賽網站預測結果

感想與心得

(一) 組員：何彩綺

剛開始的組內分工是分成 sklearn 和 ANN，等撰寫完成後再討論預測的成效如何改善。我是負責 sklearn 的撰寫，但由於我先前並沒有實際使用過預測模型，所以花了一些時間研究前處理、引入 package 等等，並參考網路上的 Youtube 教學。但是 sklearn 比起 ANN 的預測結果差，所以我們後來改成用 ANN 的方式。

我們在建模時遇到最大的瓶頸決定刪除哪個欄位，因為刪除掉多餘的欄位，才會讓預測結果更準確，但關鍵是要選擇哪些欄位。我們上網找了很多分析方法，例如共變異數分析，資料視覺化找出關鍵特徵，最後保留適合欄位。

經過一學期的課程，幸運能在學期末接觸到資料科學競賽，讓我多了實務運用的經驗，和組員討論與自學的過程中獲益良多，也希望老師能在課堂中有更多的 code 講解和案例分析。最後，期許自己能在寒假自主參加 Kaggle 的競賽，複習本學期教學內容，也謝謝老師的精闢講解！

(二) 組員：黃振維

這次的 churn prediction 競賽，我使用 Neural Network 輸入訓練的資料來預測該成員是否 Exited。由於資料科學導論課程尚未介紹神經網路，我是透過網路上的資源來學習相關知識，如 Youtube 上的教學影片。我認為建構神經網路的語法相對簡單，我是使用 Keras 模組來建立，且 Keras 有官方的文本可

以查看，其步驟也與 scikit learn 相似。

我認為較困難的地方是如何挑選 layers、activation function、batch size 和 epoch，這些層面需要較深厚的統計知識且需要分析原資料來決定使用哪個函式，因此花費最多時間來理解。要讓準確率提高對資料的前處理十分重要，但在進行競賽時我對這部分的知識較為缺乏，因此建議可以提供更多前處理的程式範例以及實際例子，像是如何篩掉 outlier。我覺得整個競賽十分充實且特別，可以透過實際的練習加強機器學習的能力，其他的課程也比較少體驗過班內競賽，謝謝教授的教導！

(三) 組員：李若瑜

這次作業是要使用 excel 資訊去預測，我也做過預測的作業，不過不一樣的是那份是去預測圖片屬於什麼樣的動物，由於預測的東西不同，一個是數據一個是圖片，因此我一開始以為這是兩種雖然為不一樣的東西，但還是可以透過修改程式碼去完成，不過似乎修改過還是無法運作，可是做完後發現又有一點相似，我嘗試使用的方法是 tfkeras 的方法，其中包含 tenserflow 的檔案，不過可能還是有些許不同，因此這些檔案即使在修改過之後還是不太能用在 excel 的處理上面，後來看到同學做的 keras 的方法發現其實並沒有我想像中的複雜，只有訓練的那一部分跟 tfkeras 有點像而已。這次競賽讓我學到了另一種資料的預測，不但會了影像的預測，還會了資料的預測，加強了我機器學習的能力，實屬有幫助。

GitHub Repository：
https://github.com/charis0811/charis0811

GitHub Page：
https://charis0811.github.io/charis0811/main.html