**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。

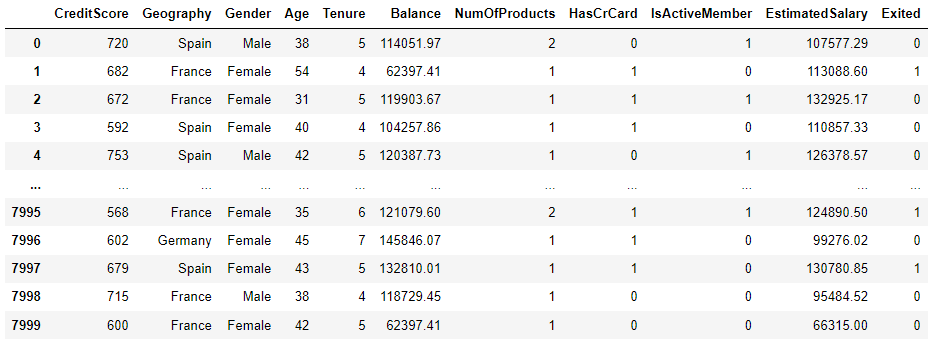
**資料前處理**

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。

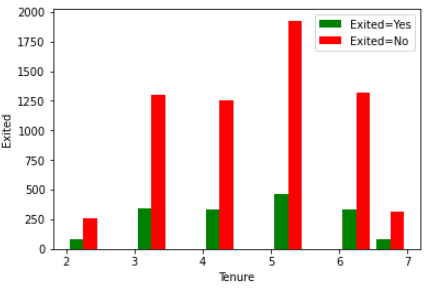
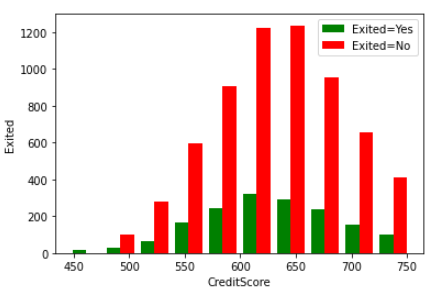
**特徵處理與分析**

1. 刪除不影響Exited的三個欄位RowNumber、CustomerId、Surname。

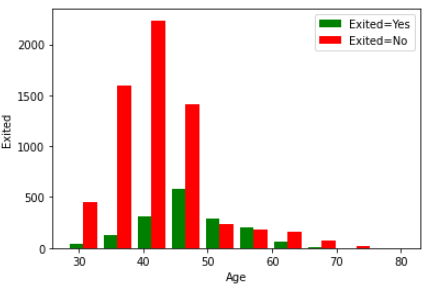
此三種attribute並不影響任何結果，都僅僅代表一個編號而已，故刪除。



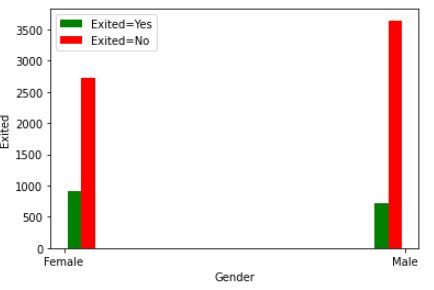
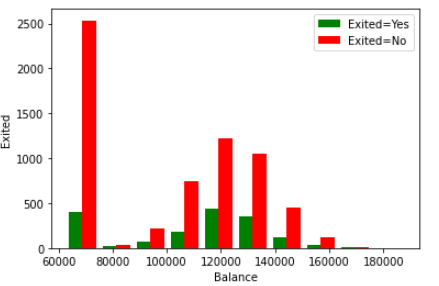
1. 分析剩餘欄位與Exited的關係分佈，以直方圖呈現。若是No與Yes大致上成比例則與x軸之attribute無關，若不成比例則可能有關。綜合下面的直方圖，發現age 跟balance比較有關，較有可能影響到exited的值。



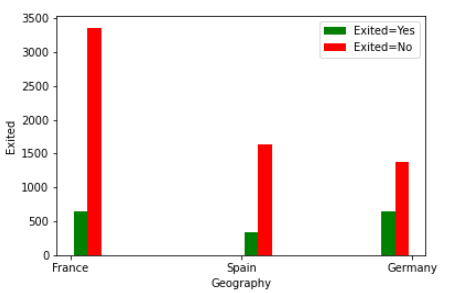
▲ Credit vs. Exited ▲ Tenure vs. Exited



▲ Estimated Salary vs. Exited ▲ Age vs. Exited

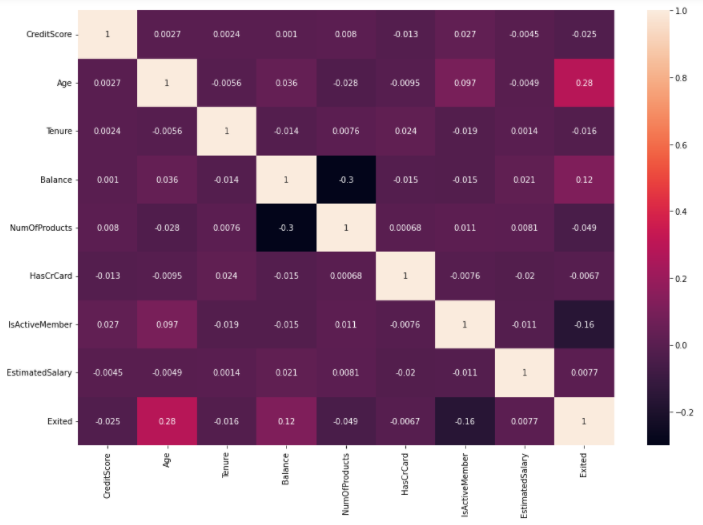


▲ Balance vs. Exited ▲ Gender vs. Exited



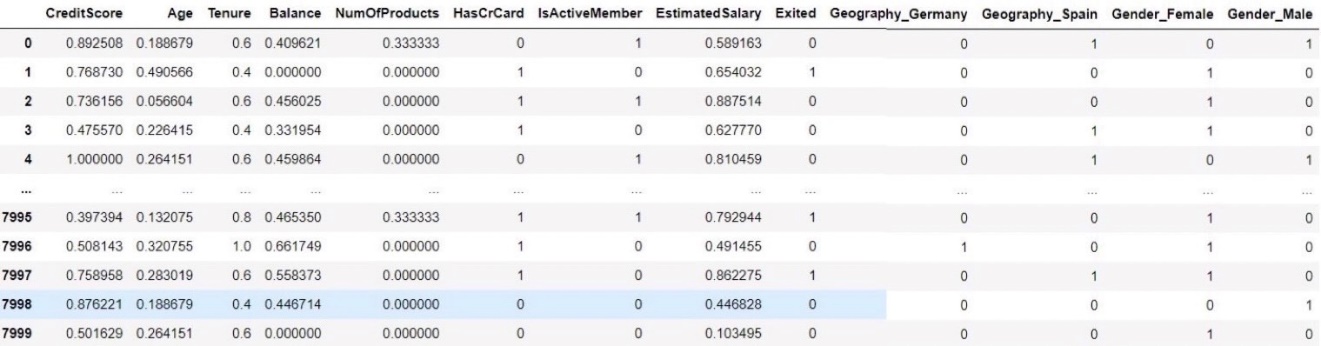
▲ Geography vs. Exited

(3)以heatmap的方式呈現



▲ 各欄位之間共變異數

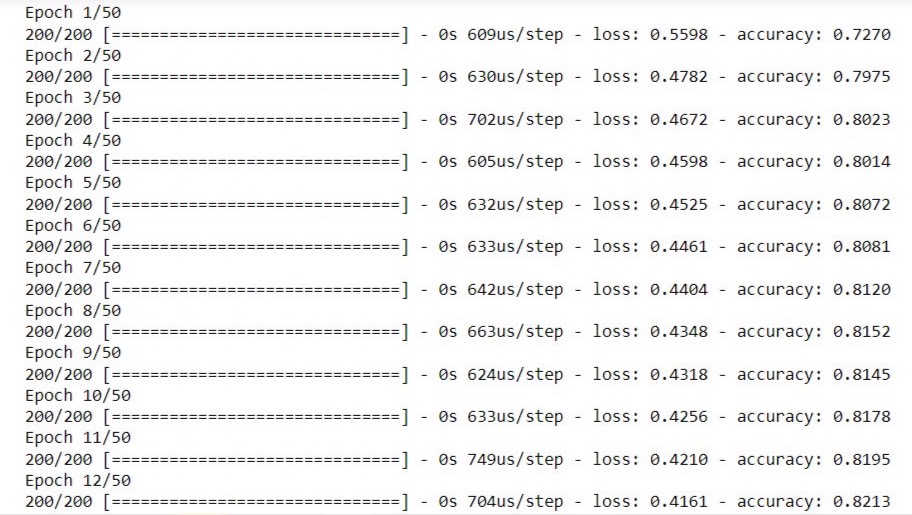
(4)將每一行最大值設為1，最小值設為0



▲ 清理資料後之結果

(5)drop掉Exited

(6)訓練模型

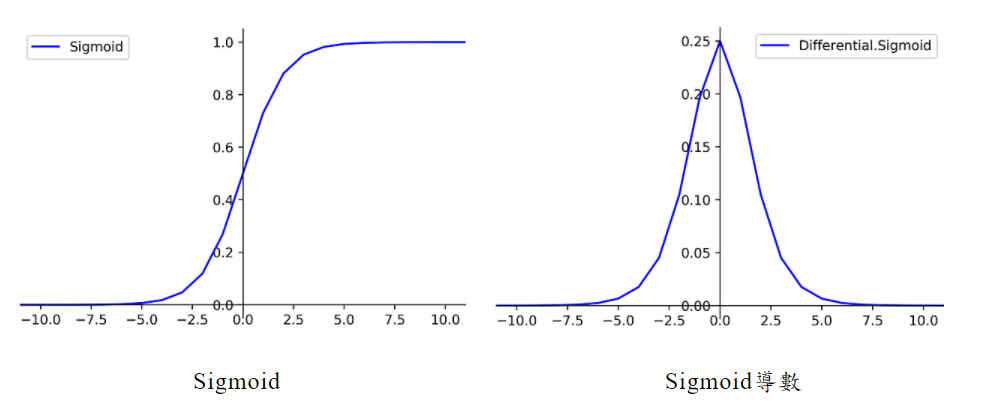
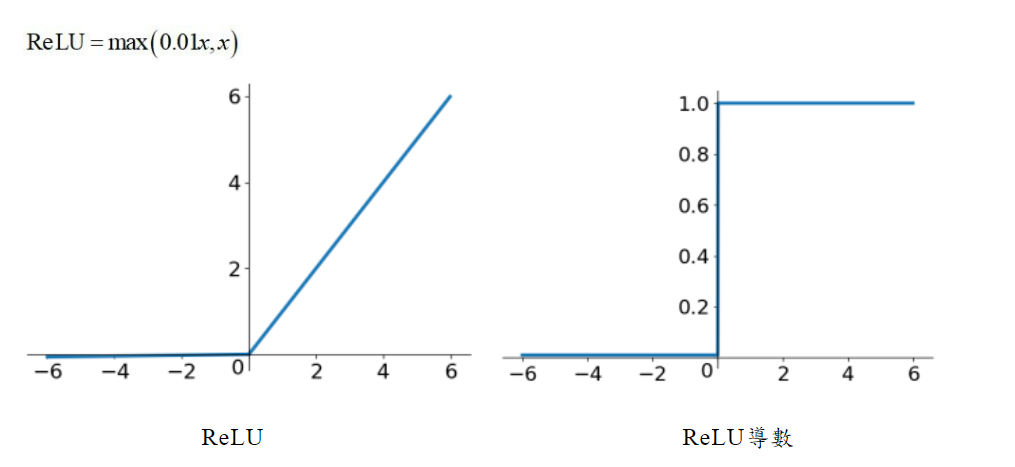


▲ 部分訓練結果

**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。

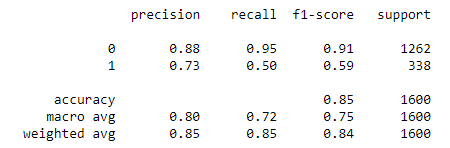


(二) 最初模型—sklearn

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。

****

**▲** 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：git@github.com:cindylee0313/Data-Science.git

GitHub Page：<https://cindylee0313.github.io/Data-Science/>