# 資料科學導論 HW5 競賽書面報告

## TEAM 4

組員：經濟 111 雷以誠

D44071135

一、 競賽敘述與目標：

這次競賽主要目的是藉由透過舉辦一場課程內的小競賽，來驗收大家對於課堂所學之內容之熟悉度，比賽內容是透過給予諸多去識別化後的銀行客戶資料當作 X 的訓練資料，然後以該客戶是否有在辦卡後離開作為 LABEL。

## 二、 資料前處理：

由於此次競賽起初所打算使用的方法為 KNN 及決策樹二擇一，因此在特徵的選取上，一開始就有先將一些不連續及無關的特徵刪去，最終刪除了「姓名」、「國家」、「性別」這三項不連續，且和本次競賽較無關聯的數據。

以程式碼呈現如下：

```
train=pd.read_csv('train.csv')
Finaltest=pd.read_csv('test.csv')
Finaltestinput=test[['CreditScore','Age','Balance','EstimatedSalary','Tenure','NumOfProducts','HasCrCard','IsActiveMember']].valu
```

三、　預測訓練模型：

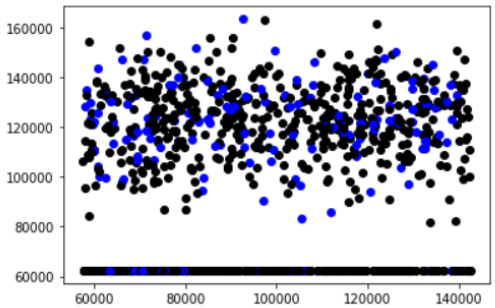
這次比賽採用了 KNN 及決策樹兩種方法來訓練模型，以下分別闡述：

KNN：

最初選用 KNN 的想法為認為有相似特徵的人，也很有可能做出相似的決斷，因此可以透過找出最近的幾位人(KNN)來判斷出其 LABEL 為何。

而在選用後遇到的第一個問題為「要選用何種特徵來作為判斷依據？」由於 KNN 模型本身的限制，通常在訓練時也只會有 2-D 的維度，然而在此次比賽所提供的特徵有非常的多，因此在一開始時是多測試了很多種來試驗，然後大部分特徵皆無法輕億的直接分群，會呈現出如下圖中較複雜的狀況：

```
In [120]: train=pd.read_csv('train.csv')
          train=train.head(1000)
          for i in range(len(train['Exited'])):
              if train['Exited'][i]==1:
                  plt.scatter(train['EstimatedSalary'][i],train['Balance'][i],color='blue')
              else:
                  plt.scatter(train['EstimatedSalary'][i],train['Balance'][i],color='black')
```



因此再將數據丟入模型訓練後，雖然帳面效果似乎不錯，然在考慮 Overfitting 後等因素…，仍決定不採用此模型。

```
In [140]: X=np.array(train[['EstimatedSalary','Balance']])
          #X=X[:,np.newaxis]
          Y=np.array(train['Exited'])
          #trainY=trianY[:,np.newaxis]

In [167]: trainX, testX, trainY, testY = train_test_split(X, Y,
                                                          test_size=0.5,
                                                          stratify=Y)

In [168]: knn = KNeighborsClassifier(n_neighbors=5)
          knn.fit(trainX,trainY)
          predY=classifier.predict(textX)
          testY

Out[168]: array([0, 1, 1, ..., 0, 0, 0], dtype=int64)

In [169]: print(np.sum(predY==testY)/float(len(testY)))
          len(testY)

          0.7595

Out[169]: 4000
```

決策樹：

在使用 KNN 效果不理想後，我便將思考模式轉換成找尋能夠同時處理多項變數的模型，而在這之中最讓我認為有效率且效果良好的模型即為決策樹，由於決策樹能夠透過其獨特的計算方式(亂度 IG)來衡量每個特徵，因此相對來說會更具有參考性，因此首先將前處理過的資料直接丟入模型中：

```
In [6]: DT=DecisionTreeClassifier()
        DT.fit(trainInput,trainClass)
        DT.score(testInput,testClass)

Out[6]: 0.7670833333333333
```

由於此次題目變數繁多，考慮到在分類到後面的特徵時可能參考性已大幅下降，因此限制模型最多只能分類至第 5 層，結果如下：

```
In [165]: DT=DecisionTreeClassifier(max_depth=5)
          DT.fit(trainInput,trainClass)
          DT.score(testInput,testClass)

Out[165]: 0.8554166666666667
```

Score 大幅上升，故在考慮後決定採用此模型。

四、　預測結果分析：

Leaderboard 結果：


在上傳排行榜後與上面實際結果對照，發現實際預測結果與在本地端訓練時有一小段落差，不過由於模型在資料轉換中有落差本就是非常正常的狀況，因此在評估之後認為仍屬可接受範圍，而第一次測試之 KNN 資料由於上傳時並無截圖，因次在後續就被蓋過去了…QQ，但從上傳中的 KNN 成績也確實如上面預期的一般，單純簡易的分群對於此類多維的數據來說仍有難度，因此跑出來的數據也不甚理想，故最後仍是採用深度=5 之決策樹最為最終結果。

五、 感想與心得

這次比賽我認為對於我來說是非常有意義，因為即使在課堂上學習到了非常多種分類與分群的模型和技巧後，在第一次要實際應用時仍會感到非常陌生與不知道如何起頭，因此這次競賽就剛好提供了一個非常棒的機會浪讓大家可以藉此來學習與練習。

我認為之後可以考慮將競賽的形式、規模近一步擴大，然後可能可以降低或減少些許作業的份量來平衡，讓修習這堂課程的新生能夠花費更多時間來實際操作與切換各式模型來訓練比賽題目，也可以讓更多人提前了解自身是否對資料科學領域有興趣以利其近一步評估自身後續生涯的發展。

GITHUB 個人網址：

https://github.com/BrextonLei

HW5 網址：

https://github.com/BrextonLei/DataScienceLesson