1. 小組成員:

C44066040 苗鈺 統計110

H24074077 劉柏鈞 統計111

H24076053 李弘哲 統計111

1. 競賽目的

藉由老師提供之資料進行預測，期望能預測出顧客到底會不會離開此銀行

1. 資料前處理

有先對資料進行單變數的分析以及雙變數的分析(檔案upupup)，僅在最後一據分析結果將某些明顯的成員進行分類ex: NumOfProducts>=4時一定會離開

1. 預測訓練模型、預測結果分析:

有測試過的模型:LinearSVC、KNeighbors Classifier、SVC、DecisionTreeClassifier、DecisionTreeRegressor、RandomForestClassifier、BaggingClassifier以及自行嘗試不使用任何pakage去做分類(untitled1)，最後取最高值，其中測試時為了避免因overfitting產生誤差，所以有使用combinations這個函數將所有排列組合都做一次最後取預測正確率最高的前幾筆資料來確認正確率(DecisionTree\_iris內，不過因為我是直接改模型並未留下所有模型測試的檔案)，因為試了方法後得出使用DecisionTreeClassifier預測正確率最高，因此都在對這個資料的參數進行調整，有調整過內部大部分認為可能影響之參數(如:)，但發現預設參數是正確率最高的，其中還發現DecisionTreeClassifier一個奇特的特性，當時我測試正確率最高的前10名參數組合中，我選擇使用參數數量最多(五個)的排列組合以避免overfitting發生，所以我選擇使用['Age', 'NumOfProducts', 'IsActiveMember', 'Geography', 'Gender']這幾個參數，後來發現若把所有測試資料fit進去後去預測測試資料，正確率竟然不是100%(約86%)，然後我又把其中分類正確的重新fit一個模型(以下簡稱f2)，分類錯誤的在fit出另一個模型(以下簡稱f3)，將測試資料分別透過3種模型去預測產生結果然後比較這個結果，發現fit所有測試資料的模型出來的結果與f2產生的結果完全一致，但若與f3做比較會出現幾乎完全相反的預測結果，其中僅有68筆資料仍然一樣，所以推測DecisionTreeClassifier在嘗試fit模型時會將某些少數資料視為相反資料並忽視他們，才會產生這樣的結果；發現這個結果後有嘗試利用這個特性去交叉比較再對資料做預測，但正確率並沒有太大差別，且發現若考慮所有參數不僅使用其中那五個參數(['Age', 'NumOfProducts', 'IsActiveMember', 'Geography', 'Gender'])時，所有資料fit出來的模型再次預測所有資料時正確率會變100%，推測是overfitting，所以似乎用這個方法也可以檢查到底有沒有overfitting的問題。以上為僅使用一種模型預測的結果，另外有嘗試使用其他模型相互配合來提高預測正確率，不過沒有顯著效果，所以最後依然專注於嘗試DecisionTreeClassifier。

1. 心得:

苗鈺

這次的報告我似乎太過於磚牛角尖DecisionTreeClassifier的一些奇怪用法(像是前面提到的忽略少數資料)，而且我也沒有想嘗試調整所有參數，只有調整其中幾個(如最低樣本數的參數)，後來聽過前幾名組別的報告後發現了自己的許多盲點以及正確應該如何去測試這個樣本，希望之後的cp2能夠做出較理想的成績，而對CP2的部分我自己是期望我自己能夠嘗試利用複數個分類器互相配合，以使預測正確率更高，也希望這次能在學習到更多的相關知識，例如這次上台分享的其中一組就有使用神經元網路(應該是這個名子)，我這次應該也會嘗試使用這個方法。

劉柏鈞

這次的競賽，就像是人生，很多事情沒有標準答案，也像微積分，我們只能嘗試著去逼近我們找的值，在這字裡行間的程式碼中去探索，在這各式各樣的演算法中去摸索誰最合適，就像愛情。

雖然我對於程式的理解如同管中窺天，非常片面，但是我內心卻沒有絲毫迷惘，因為我相信著我的隊友，我也堅信團隊的力量，1+1+1一定遠大於3。在起初嘗試的過程中，跟大家一樣，都把較無關的column給去除，把一些用值0、1來表示，也進行過標準化，試著去提高那三個指標，我跟隊友也一直嘗試著不同的演算法，從Decision Tree決策樹到Random Forest﹐分數不升反降，讓我在摸索的過程中感到失望，也透過一些更改維度的方式看能不能找到最佳維度，但總是處處碰壁，最後的成績也不如我們所期望，但是我還是堅信能在下次CP2的競賽中有所進步，我也會盡我所能自學，在競賽中自我突破，帶領團隊拔得頭籌。

李弘哲

這個競賽跟我原本想像的很不一樣。原以為只要寫好程式碼，一切分數都可以簡單拿到，沒想到不同的演算法會讓分數有這麼大的差別。

由於本人資質愚昧，試了很久卻連程式碼都寫不好，甚至python的環境也一直有問題，因此我用了隊友的程式碼做修改及嘗試不同演算法。一開始用Decision Tree法，不斷嘗試改變參數值卻無法衝高分數，後來改用Random Forest，但在參數相同的情況下，分數都比Decision Tree低。最後嘗試先以PCA降維後再進行分類，準確度雖高，但Precision和F-score卻比其他方法低很多，或許是降維的方法及觀念不太正確。

我會持續在cp2努力，也會先了解更多演算法的內涵及使用方法，希望能在cp2為團隊衝高分數！