資料科學導論hw4

(1)小組各成員的姓名、系級與學號。

賴榆方 統計114 H24104034、林孟漩 統計114 H24101159、林佑融 測量111 F64071122

(2) 競賽敘述與目標:

競賽的目的是要利用多種的特徵去預測銀行的客戶是否會流失。首先我們需要把TRAIN.CVS已經有結果的檔案拿去訓練模型並預測TEST.CVS裡的顧客是否會流失。

(3) 資料前處理:

我們認為"RowNumber"、"CustomerId"、"Surname"這些特徵對於預測結果是沒有幫助的，所以在一開始就先將他們從訓練資料中移除。

有試過利用RFE來過濾資料，但認為只使用部分特徵的效果沒有比較好，所以還是採取了原本選定的所有特徵。

(4) 特徵處理與分析:

接著，我們將"Geography"、"Gender"這兩種文字型特徵轉變成數字型資料(詳情請看下圖中的程式碼)，以利模型的進行。

之後，我們有對資料進行過標準化和PCA處理，但是認為對於預測結果來說，他們並沒有提高準確率，所以我們還是使用最初的格式來做為訓練資料和測試資料。

最終選定特徵:

```
import pandas as pd
import numpy as np
```

```
train_data = pd.read_csv('train.csv')
train_data['Gender'] = np.where(train_data['Gender']=='Male', 1, 2)
train_data['Geography'] = np.where(train_data['Geography']=='Spain', 1, np.where(train_data['Geography']=='France', 2, 3))
test_data = pd.read_csv('test.csv')
test_data['Gender'] = np.where(test_data['Gender']=='Male', 1, 2)
test_data['Geography'] = np.where(test_data['Geography']=='Spain', 1, np.where(test_data['Geography']=='France', 2, 3))
```

```
X_train = train_data.drop(['Exited', 'Surname', 'RowNumber', 'CustomerId'],axis=1)
y_train = train_data['Exited']
X_test = test_data.drop(['Surname', 'RowNumber', 'CustomerId'],axis=1)
```

```
X_train
```

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	720	1	1	38	5	114051.97	2	0	1	107577.29
1	682	2	2	54	4	62397.41	1	1	0	113088.60
2	672	2	2	31	5	119903.67	1	1	1	132925.17
3	592	1	2	40	4	104257.86	1	1	0	110857.33
4	753	1	1	42	5	120387.73	1	0	1	126378.57
...	...	...	...	...	...	...	...	...	...	...

(5) 預測訓練模型:

我們嘗試過使用RandomForestClassifier、XGBClassifier分類模型來進行訓練與預測。有單純用模型來進行預測，也有調整模型參數來進行預測。在調整參數的部分，我們分別使用RandomizedSearchCV搭配RandomForest、以及利用在網上搜尋到的optuna(快速調整適當的超參數)和RandomizedSearchCV搭配xgboost，而在經由分數排名後，最後我們所選擇的模型及參數為:

```
#使用隨機森林
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=20,
                               bootstrap= True,
                               criterion= 'entropy',
                               max_depth= None,
                               max_features= 2,
                               min_samples_split= 10)

#從訓練組資料中建立隨機森林模型
model.fit(X_train,y_train)

#預測測試組結果是否Exited
y_pred = model.predict(X_test)
```

(6) 預測結果分析:

我們有試過利用同樣的參數來進行預測，但由於每次的結果都不太一致且有落差，所以只能大概的分析哪些模型與哪些特徵會產生較高與較低的效果。

RandomForestClassifier:

參數為使用RandomizedSearchCV跑出來的結果，然後挑選分數最佳的一個，並以此為基礎，做些微調(底線部分)。

1.{n_estimators=20, 'bootstrap': True, 'criterion': 'entropy', 'max_depth': None, 'max_features': 2, 'min_samples_split': 10}

2.{n_estimators=20, 'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': 2, 'min_samples_split': 10}

3.{n_estimators=20, 'bootstrap': True, 'criterion': 'entropy', 'max_depth': None, 'max_features': 3, 'min_samples_split': 10}

4.{n_estimators=15, 'bootstrap': True, 'criterion': 'entropy', 'max_depth': None, 'max_features': 2, 'min_samples_split': 10}

5.{n_estimators=60, 'bootstrap': True, 'criterion': 'entropy', 'max_depth': None, 'max_features': 2, 'min_samples_split': 10}

6.{n_estimators=20, 'bootstrap': True, 'criterion': 'entropy', 'max_depth': None, 'max_features': 2, 'min_samples_split': 9}

7.{n_estimators=20, 'bootstrap': True, 'criterion': 'entropy', 'max_depth': None, 'max_features': 2, 'min_samples_split': 11}

8.(剔除"Geography"特徵)：{n_estimators=20, 'bootstrap': True, 'criterion': 'entropy', 'max_depth': None, 'max_features': 2, 'min_samples_split': 10}

9.(另一組用RandomizedSearchCV找出的參數)：{n_estimators=30, 'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': 4, 'min_samples_split': 8}

	Accuracy	Precision	F Score
預設值(無調參)	0.8625	0.6897	0.5926
1(最佳)	0.8850	0.7925	0.6462
2	0.8825	0.8125	0.6240
3	0.8750	0.7547	0.6154
4	0.8725	0.7167	0.6277
5	0.8750	0.7213	0.6377
6	0.8675	0.7069	0.6074
7	0.8600	0.7234	0.5484
8	0.8550	0.8519	0.4423
9	0.8825	0.7885	0.6357

XGBClassifier：(只選分數最佳的一次)

1.搭配optuna，參數為：XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None, colsample_bylevel=1, colsample_bynode=1, colsample_bytree=0.7666666666666666, early_stopping_rounds=None, enable_categorical=False, eval_metric=None, feature_types=None,gamma=0, gpu_id=-1, grow_policy='depthwise', importance_type=None,interaction_constraints='', learning_rate=0.42894736842105263, max_bin=256, max_cat_threshold=64, max_cat_to_onehot=4, max_delta_step=0, max_depth=2, max_leaves=0, min_child_weight=6, missing=np.nan, monotone_constraints='()', n_estimators=92, n_jobs=0,num_parallel_tree=1, predictor='auto', random_state=0)

2.搭配RandomizedSearchCV，參數為：XGBClassifier(n_estimators= 17, min_child_weight= 5, max_depth= 6, learning_rate= 0.4, colsample_bytree= 1)

	Accuracy	Precision	F Score
預設值(無調參)	0.8500	0.6232	0.5890
1	0.8750	0.7755	0.6032

2	0.8675	0.7069	0.6074

在一開始，除了過濾掉的特徵外，我們有嘗試過每次剔除一種特徵來進行訓練，爾後發現特徵"Geography"的變異最低。而預測結果中呈現出，剔除"Geography"與使用全部特徵相比較，他的分數中Precision會較高，但F Score會較低。不過在總分上，使用全部特徵(不含過濾掉的特徵)會高一點，所以我們還是使用全部特徵來進行訓練與預測。

而在我們使用的兩種分類模型中，不論是使用預設值還是經過調參後的模型，都是RandomForestClassifier的分數大於XGBClassifier的分數。

所以我們最終使用到的特徵有：「"CreditScore"、"Geography"、"Age"、"Tenure"、"Balance"、"NumOfProducts"、"HasCrCard"、"IsActiveMember"、"EstimatedSalary"」。而搭配的分類模型為：RandomForestClassifier，模型的參數為：{'n_estimators'=20, 'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': 2, 'min_samples_split': 8}。

(註：避免太過繁瑣，此僅挑選我們覺得較具代表性的幾個例子而已。)

(7) 感想與心得：

賴榆方：

我覺得我從這場競賽中，學到的最重要的東西是「自學」的能力與重要性。像是在一開始我們訓練模型時，其實沒有調參數的狀況下就已經有基礎的分數了，不過想要知道如何去提高分數、調整參數，就需要大量的上網尋找資料，因為實際上不太可能一個一個的去試參數結果。比如我是因為GridSearchCV的耗時太長，所以選擇使用RandomizedSearchCV，以及學到了一個之前沒聽說過的套件optuna。此外，這也是一個很好的實作練習經驗，最初我們的排名是在倒數的位置，不過經過多次嘗試，原本都只提升零點零零幾的分數，突然變到了20名，雖然不是很厲害的成績，但看到排名往前的那一刻，心中還是很高興且有成就感！

然後我對競賽分數網站有一個建議的小地方，就是網站上的時間好像比現實快了8小時左右，希望可以把他調回正確的時間，因為有繳交期限的限制，這樣比較不容易誤會要依照哪個時間來判斷。

林孟漩：

我覺得這次的競賽相當有趣，除了能更加理解課堂上講過不同種模型的原理，也能有機會實際應用在真實數據上。

雖然學到了許多東西，在競賽期間還是遇到不少的困難，還未開始做的時候以為最棘手的部分是該如何把資料丟到模型裡去預測，但實際去操作時才發現資料的前處理和參數的調整才是最耗時間的環節，調參的時候我會去參考網路上一些學習曲線，或者網格搜索的代碼去找出該模組最合適的參數，然而每次都會出一些錯誤，系統報錯後上網

找解決方法也是看的一頭霧水，到最後只能手動調參，不過看到精確值一點點上升時都會有種莫名的感動。

林佑融：

從第七周政德老師開始講授scikit-learn的套件到第十四周講授Keras套件，我本以為沒有機會嘗試老師上課所講授的各種機器學習的套件，一直到了這次的作業四才有機會課後嘗試套件中那些模型最能準確預測行。過程中嘗試了好多方法如：PCA，也嘗試了老師在第十三、十四周講授的feature selection如：RFE，進行資料特徵的挑選，但是都沒有提升準確率。最後使用了Random forest和XGboost才顯著提高了準確率，從最一開始上傳的準確度只有27%，跟亂猜沒兩樣，到了最後一次準確度達到88%，private leaderboard(final)的分數和public leaderboard的88%有大約10%的落差。這次的作業四成果讓我了解到並不是用愈多的方法準確率就會愈好，找到一個合適的模型準確率才會有顯著提升。這次也驗證了老師在上課時說的：想要提高準確率贏得比賽，用ensemble learning就對了 (To win? Ensemble!)

GitHub 連結：https://github.com/cathylinnn/-HW4_DL.git