**(基本題)**

**(一)資料前處理**

**1.缺失值處理:**

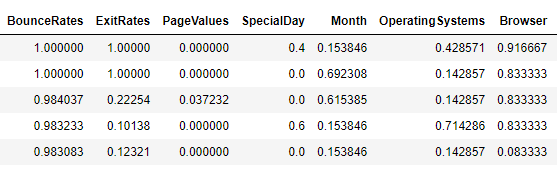
在train.csv檔中，並無發現缺失值的存在，如有發現使用fillna填補該欄位平均值

**2.類別資料的處理:**

此次並未發現類別資料，如有發現，直接利用數值替換，而等價關係則可以使用One-hot encoding來解決

**3.資料特徵縮放:**

是使用函式做資料的Normalization，之所以沒有使用其他套件來做，是為了保持dataframe的屬性，好讓後面的流程做使用



(Normalization後的資料)

**(二)資料切分**

**1.train.csv切分:**

先將train.csv讀進來，並將所要分類的Revenue欄位從中擷取出來，作為訓練模型的預測對象(y)，剩下的欄位則為x

**2. 切分比例選擇:**

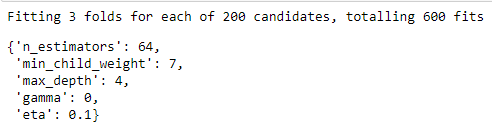
首先對訓練與測試資料以9:1、8:2、7:3、6:4的方式隨機切分，並且以stratify設置為目標 (y) 在訓練和測試集中盡量產生相同的分佈，再以測試集的分數高低，選出8:2的切分方法

**(三)模型選擇**

本次使用XGBClassifier來做分類的預測，主要是因為它的訓練時間較其他分類模型來的短，但又保留蠻高的準確度

**(四)超參數調整**

關於超參數的調整，主要是著重在Booster參數上的調整，有n\_estimator、learning\_rate、max\_depth等等，而此次使用RandomizedSearchCV的調整方法，打算先快速跑出一個基準，在做超參數的微調，而評判的標準是以測試集的score為主



(3-folds，200-n\_iter所跑出的超參數結果)

**(五)欄位選擇**

利用feature\_importance配合訓練與測試集的分數，最後drop掉了SpecialDay這個欄位，也在實驗中發現PageValues這欄位對於此次分類是很重要的一欄

**(六)輸出預測**

將所要預測的資料從test.csv讀入，投入利用train.scv訓練好的模型，輸出對應的HasRevenue的預測到面試題目資料夾中的Answer.csv

**(加分題)**

利用所給的數據得到ModelA、ModelB的TP、TN、FP、FN，進而推出AB兩模型的Sensitivity和Specificity，並且也利用此次模型的測試資料得出相關的數據，下面會放上三者的數值比較:

|  |  |  |
| --- | --- | --- |
|  | 敏感度(Sensitivity) | 特異度(Specificity) |
| Model A | 0.5969209237228832 | 0.9549597147008321 |
| Model B | 0.5920223932820154 | 0.9582617884031172 |
| 此次訓練  的模型 | 0.9626373626373627 | 0.6352941176470588 |

通常是以達到100%的敏感度，及100%的特異度，為理想去判斷一個好模型的基準，而可以看到上述的模型A與B的敏感度與特異度其實非常的相近，但與此次訓練的模型數值相差甚遠，所以我判斷兩者都不是適合本案例的情境