FDA HW 3-2

Online shoppers intention
資料分析及預測

系級:統計四
姓名:陳宥任

資料介紹

資料來源:
https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset

資料筆數: 12330筆
(為避免特殊活動或節日造成線上購物的影響，資料都是以一年為一個區間做紀錄)

資料變數: 18個變數，包含10個數值型及8個類別型變數
Administrative、Administrative_Duration、Informational、Informational_Duration、ProductRelated、ProductRelated_Duration、BounceRates、ExitRates、PageValues、SpecialDay、Month、OperatingSystems、Browsers、Region、TrafficType、VisitorType、Weekend、Revenue

問題描述

預測的目標為Revenue，即透過其餘變數像是逛網站的時間、特殊節日等等對Revenue進行預測，檢視在何種情況下會造成Revenue的不同

Revenue的分類是一個不平衡的狀態，因為他的True約佔了整個資料的84.5%而False只佔了15.5%。

因此想要透過不同的分類器來檢視哪一種分類器較準確，能夠幫助未來實際的應用

資料預處理

缺失值處理: 檢視資料後發現資料並無缺失值，不須執行填補缺失值的動作

類別變數處理: 資料內含有許多類別變數，都必須經過轉換才可以使用

1.Dummy variable: 因為這裡的類別變數有許多都不是Binary的，像是日期就有0-11，如果使用Label encoding這樣會造成數字間含有距離關係。因此在這裡對Month、Browser、Region、OperatingSystems及TrafficType使用pandas中的get_dummies函數

資料預處理

2.Label encoding: 其餘Binary的類別變數像是VisitorType、Weekend及Revenue就使用Label encoding即可完成變數轉換

切割資料: 資料原先還沒經過切割的動作，因此這裡就已問題描述的定義將y設為Revenue，其餘資料則為x

拆分訓練、測試資料: 要預測資料的準確性，必須將x及y再拆分為train和test資料，這邊使用sklearn裡面的train_test_split函數將資料以7:3的比例做拆分

預測方法及結果

Logistic Regression

```
Train Accuracy : 0.8907426717645696
Test Accuracy : 0.8726682887266829
```

可以看到原始的Logistic Regression分類其實準確率就非常高，train來到89%而test也有87%，且沒有overfitting的狀態。不過為了使準確度更高，稍微調整一下參數

預測方法及結果

Logistic Regression(tune class_weight)

```
Train Accuracy : 0.8916695632024099
Test Accuracy : 0.8778048121113815
```

思考因Revenue是一個不平衡的狀態，因此這裡做權重的調整，把True的權重調高，因此True在資料集中是數量相對來說較少的。不過得到的結果並沒有太大的提升效果，train和test都稍微提升了0.1%左右

預測方法及結果

Support Vector Classifier

```
Train Accuracy : 0.8525083999536555
Test Accuracy : 0.8340091916734252
```

可以看到SVC相對來說就沒有Logistic Regression那麼高，不過還是有著train 85%及test 83%的準確度。這邊也是稍微調動一下參數看是否能夠顯著提升

預測方法及結果

Support Vector Classifier(tune C = 250)

```
Train Accuracy : 0.8936392075078207
Test Accuracy : 0.8759124087591241
```

SVC的懲罰項C原始是1，這邊將他調至250，能夠產生出夠好的預測結果並且也沒有overfitting的產生，不過這邊要注意的是，懲罰項也不能調得過大，不然就會產生overfitting的結果。可以看到最後tune出來的結果比Logistic Regression的結果好一些，train 89%及test 87.6%，是目前最好的結果

預測方法及結果

K-Neighbors Classifier

```
Train Accuracy : 0.8963040203916116
Test Accuracy : 0.852392538523925 4
```

原始的K-Neighbors Classifier的參數n_neighbors是3，結果會達到不錯的train accuracy 89%，但是與前面兩種模型比較他的test相對來說就不高了，只有85%，所以我們也試著將他的參數調整尋找更好的準確度

預測方法及結果

K-Neighbors Classifier

這邊以迴圈的方式將範圍設定再1-30來尋找n_neighbors為多少的時候，他的train和test accuracy會最高，結果如下:

Train

```
best score: 1.0
best k: 1
```

Test

```
best score: 0.8561773452284401
best k: 8
```

結果為在train accuracy中，當K=1時，他的準確度會到100%。
在test accuracy中，當K=8時，準確度來到86%
那這邊我是直接選擇test的結果來調整參數，因為很明顯的train的結果肯定會造成overfitting

預測方法及結果

K-Neighbors Classifier

```
Train Accuracy : 0.8812420345267 06
Test Accuracy : 0.8561773452284401
```

最後的結果為train 88%，test 86%，是三個模型當中準確度最低的。不過我已經將K從1-30檢視過一遍，既然8是最好的準確度，那肯定是這個模型在這個Data中的最高準確度了