Machine Learning (Report)

404410077 林楷博

1. Explain your methods :

Step 1：load data

一開始先將 TraData.csv 用 pd.read_csv 的方式讀進程式，並將一些看起來像雜訊的欄位做刪除(像是 dclkVerticals、ip 之類的)，接著用 labelencoder 的方式處理每個欄位。

Step 2：build train and test set

將要判斷的欄位和答案的欄位分成 X 和 y 並用 cross_validation.train_test_split 的方式做訓練，其中切 2 成當作測試集；並把某些欄位增加權重。

Step 3：build random forest model

這裡我先用 K-fold 的方式將資料切成 5 份來做數據的分析，用來檢視數據的標準差穩不穩定，並用 RandomForestClassifier 來當作 model。RandomForest 就字面上的意思就是用隨機的方式建立一個森林，森林裡面由各種 Decision trees 組成，Decision trees 之間沒有任何關聯。森林生成後，每當一個新樣本輸入時，就讓森林中的 Decision trees 分別判斷，判斷該樣本應屬哪一類的算法，並且判斷哪一類被選擇的最多數，就預測該樣本為那一類。這種 Model 主要用在回歸、分類。

這裡 Model 有下了 class_weight、min_samples_split、

min_samples_leaf、max_depth、n_estimators 的參數，下列一一做介紹：

1. class_weight：click 是 1 的資料，我將他的權重設成 11 倍。

2. min_samples_split：當下 samples 最小必須分裂出 n 個 internal node。我設成 5。

3. min_samples_leaf：當下 samples 最小必須分裂出 n 個 leaf node。我設成 4。

4. max_depth：樹的最大深度。我設成 7。

5. n_estimators：在森林裡的樹的數量。我設成 1。

Step 4：exam for unknown data.

和 Step 1 的方式相同，只是再讀進一次要測試的資料。

並用 rf.predict(data_X)來用 train 的 data 作預測。

p.s：

#end exam#以下的只是印出 test_size 所做預測的各精準度，當作參考。

而每次跑出來的數據都會不一樣，可能要多測試幾次才會有非全 0 的數據產生，只是當測資還沒釋出時我們 train 的 f1 可以到達 20%左右，而跑 test data 的資料發現全 0，以及得到第一次 test 的結果 f1 只有 1.x%的時候覺得身心俱疲。之後再不斷的調參數才得到更多非全 0 的 ouput，但是同樣的 model 和同樣的參數有時能測出幾百個到幾千個 1 的 output，甚至 1 萬多個 1 的 output，但之後就可能是全 0 的 output，不知道是不是正常現象。

個人心得(404410077 林楷博):

其實寫出一個機械學習的程式碼就像老師說的一樣，入門門檻其實很低很好寫，網路上也有很多的資源。但我想差別就在於如何善用每一種方法，並充分了解每一個參數怎麼去下，才是關鍵。

這次的Project我實作了很多的方法，並且參考了許多網站，其中https://ithelp.ithome.com.tw/articles/10187452這個網站幫助了我很多。照著這個網站的範例，我很快地就寫出了一個用**AdaboostClassifier**的**DecisionTreeClassifier**，但一開始的f1score總是0趴。之後再與組員討論和修改參數，我們最終決定採用**AdaboostClassifier**的**RandomTreeClassifier**，至於參數則是**RandomTreeClassifier**的(1) class_weight = {0:1, 1:10} (2) min_samples_split = 4 (3) n_estimators = 44 (4) max_depth = 7，至於**AdaboostClassifier**的(1) n_estimators = 3 (2) learning_rate = 0.05。在不斷的調整當中，我大概發現了幾個定則:

1. class_weight: 把出現1的設為10倍會是最佳狀態。
2. n_estimators (候選人) : **AdaboostClassifier**大約設在不多2~3會是不錯的選擇。至於**RandomTreeClassifier**則設在大約default(50)附近的44上下
3. max_depth (Tree深度) : 大約七層會是最佳值。
4. min_samples_split : 每層資料至少都分為3或4個。
5. learning_rate = 0.05 : 預設為1。但要盡量再設低一些，免得使其下不去

local min的值。不過根據測試，若是再低於0.05就沒有太大的幫助了。

至於各筆Attribute的權重，我則是利用Excel先將Click為1得先取出來分析，經過觀察之後，我發現ip和dclkVerticals這兩項每一個為1的都不一樣，因此我就直接不讓我的Training Data引入這兩筆Attribute。再來是publisherId，我們發現把他的權重設大一點會有較好的結果，至於adx和spaceType維持一倍，剩下的則設為兩倍即可。