

機器學習導論期末報告

巨資 4A 謝誠閔 06170142

Code 比較多，我拆成 3 個 ipynb。

wsdm_data.ipynb = 預處理
wsdm_obj_int.ipynb = dummy / labelencoder
wsdm_algo.ipynb = 跑演算法

資料預處理:

資料都合併之後，處理缺失值跟異常值。

缺失值

先看類別資料:

(source_system_tab/source_screen_name/source_type/language)

基本上缺值比例比較少的，都是用眾數來補，比較不會影響本來的分布。

比較不一樣的是 source_screen_name，他缺失值的比例比較多(5%)，我用原本占比最多的 4 個類別隨機填充。

然後我去掉了三個有點問題的欄位

性別(genre_ids)的缺值太多了，而且也不能用平均值來補，用起來感覺不太 OK。另外的兩項是作曲和作詞(composer/lyricist)，本來我的猜想是他沒有特別附上這兩格，而且歌手(artist_name)的欄位都沒有問題，那這種空值的歌可能是歌手、作曲、作詞都是同一個人，但我去用 isrc 稍微查了一下，發現很像不是這樣，所以我最後不打算用這兩個欄位。

聽歌的時間(song_length)，單純用平均值來補。

歌手(artist_name)的缺值沒有很多，一開始我直接把空值的資料給刪掉，後來發現這樣做 test 最後的結果會比數變少，後來改用 'No_name' 來補

異常值

裡面唯一有異常值的是年齡(bd)欄位，有很多負值，而且最大竟然有到 1000 多。最後我只留 5~100 歲的資料，然後再用這些資料的平均值來補原本的異常值。

新欄位

資料中有註冊的時間和到期的時間，如果單單比較註冊或到期時間，不如用註冊到到期的時間來觀察，所以加入 'time' 來放使用時間。

類別 >>> 數值

演算法沒辦法直接算類別資料，所以跑演算法之前要先轉換格式。

Dummy or LabelEncoder

正常來說 dummy 會比較好，但如果全部 dummy，電腦跑不太動。先用 LabelEncoder 來看相關程度，只把相關程度比較高的 feature 做 dummy。其他的 feature 用 LabelEncoder。

algorithm

最後把準備好的資料套入演算法，這裡又有刪掉一些 feature。雖然前面有盡量減少維度了，但 55 個 feature 可能真的太多，電腦跑不動，後來把資料刪到只剩 19 個維度終於可以用了。

演算法的部分，選擇用 KNN、LogisticRegression、RandomForest 三種來預測。

KNN score

Your most recent submission				
Name	Submitted	Wait time	Execution time	Score
KNN.csv	a few seconds ago	1 seconds	13 seconds	0.57731
Complete				
Jump to your position on the leaderboard				

會選用 KNN 主要原因是 KNN 比較好解釋。

KNN 的概念就是把最近的分在一起，很容易理解。

原本的猜想是 KNN 沒有用 label，所以可能會比較不準，結果有出乎意料之外，沒想到是三個裡 score 最高的。

LogisticRegression score

Your most recent submission				
Name	Submitted	Wait time	Execution time	Score
LR.csv	a few seconds ago	1 seconds	14 seconds	0.57590
Complete				
Jump to your position on the leaderboard ▼				

會選用羅吉斯回歸主要原因是，最後的結果是 1 或 0，我最直接想到就是上課預測下雨的 ppt。

羅吉斯回歸也是分類，但是跟 KNN 比較不同的是，概念比較像預測目標的機率，是有還是沒有、會還是不會的這種感覺。

RandomForest score

Your most recent submission				
Name	Submitted	Wait time	Execution time	Score
RFC.csv	a few seconds ago	1 seconds	15 seconds	0.57640
Complete				
Jump to your position on the leaderboard ▼				

會選用 RandomForest 主要原因是，我很喜歡這個方法，一次不夠好，那就多試幾次，最後統整起來截長補短。而且上課老師也有提到，到部分的比賽最後都是用隨機森林，準確率跟活躍程度還不錯。

原本的猜想是 score 最高的，沒想到不是，有點意外。

待改善

應該有可以改進執行效率跟空間的方法，如果學會就可以更有效率。

受限於執行空間，沒辦法用完整的資料先 dummy 再挑適合的 dummy 有點可惜。

參考資料

<https://chih-sheng-huang821.medium.com/%E6%A9%9F%E5%99%A8-%E7%B5%B1%E8%A8%88%E5%AD%B8%E7%BF%92-%E7%BE%85%E5%90%89%E6%96%AF%E5%9B%9E%E6%AD%B8-logistic-regression-aff7a830fb5d>

<https://www.itsfun.com.tw/%E6%8A%98%E9%95%B7%E8%A3%9C%E7%9F%AD/wiki-2687912-7767702>

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.concat.html>

<https://www.kaggle.com/c/kkbox-music-recommendation-challenge/discussion>

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.reset_index.html

<https://stackoverflow.com/questions/20107570/removing-index-column-in-pandas-when-reading-a-csv>

<https://datatofish.com/numpy-array-to-pandas-dataframe/>

https://blog.csdn.net/weixin_39223665/article/details/79935467

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>