資訊四乙 F74052316 朱浩澤

資料分析HW3_1 Report

資料處理：

這次的資料所要預測的是股票的漲跌，拿到的資料可以說是滿完善的，大致上裡面有六個欄位：Date(日期）、Open Price（開盤價）、Close Price（收盤價）、High Price（當日最高點）、Low Price（當日最低點）、Volume（交易量）。如圖所示。

Out[3]:

	Date	Open Price	Close Price	High Price	Low Price	Volume
0	02-Jan-2009	902.99	931.80	934.73	899.35	4048270080
1	05-Jan-2009	929.17	927.45	936.63	919.53	5413910016
2	06-Jan-2009	931.17	934.70	943.85	927.28	5392620032
3	07-Jan-2009	927.45	906.65	927.45	902.37	4704940032
4	08-Jan-2009	905.73	909.73	910.00	896.81	4991549952

並且進一步去看說這份資料有沒有缺失值等等，基本上完全沒有，因此針對這一部分沒有要做更進一步的處理，不過由於這次是分析漲跌，因此要為這份資料多新增一個欄位“Movement”，去比較前一天的收盤價，若是比昨天高，則標記為1，反之則標記為0。

在選擇特徵的時候，就將全部的欄位拿來計算，由於Date是字串不太好處理，將這個欄位剃除，且稍微觀察這份資料，剩下的五個欄位有一項成交量他的尺度滿明顯與其他欄位不太相同，有可能會造成預測的結果不佳，所以選擇將這份資料標準化，讓他們彼此之間的尺度差距不會太大。不過做這樣的處理並不絕對會讓預測的結果變好，因此還是有試著將無標準化的資料和有標準化的資料都做看看，但結果是有標準化的資料在所有模型的輸出上表現都較為優異。

由於股票的漲跌跟前幾天的趨勢其實有相關，所以嘗試以前五天的當日收盤價作為特徵餵進去模型，不過沒有帶來比較好的結果。

模型的選擇及使用：

這次作業規定的是LR以及NN，我另外選擇的模型是RF（random forest），達到最高預測率的是LR模型，如圖。

```
Test Accuracy: 0.8214285714285714
[[ 98  23]
 [ 22 109]]
```

本來對於沒標準化的資料，準確率只有0.54左右，不過改成有標準化過的資料後神奇的準確率飆升到0.82，下面的矩陣則是混淆矩陣。若是全部猜1的話其實準確率也有0.52左右。

稍微值得一提的是在做NN的時候我只做了兩層全連接linear：5->14->1、學習率：0.001、loss function：BCEloss，最後用sigmoid將值對應到0-1之間，一開始沒有做任何微調的時候，大部分全部都會猜成1，不過稍微觀察資料後發現，一開始收盤價900多，到最後2000多，其實全部都猜漲也

是蠻正常的，而且這份資料中本來漲的時候所佔的比例就比較高，因此嘗試將不同類別設定不同權重，也就是讓猜1的權重低一點點，跳到0.4，狀況就變得好一點，不再只會猜1，且準確旅遊提高一些。如圖是有測到比較好的模型：

```
[1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 1
 0 0 0 0 1 0 1 1 1 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1
 1 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 0 1 0 0 1
 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
 0 0 0 1 1 1 1 0 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1]
Test Accuracy: 0.5753968253968254
[[50 71]
 [36 95]]
```

討論：

這次的股票預測跟上次的鐵達尼號比起來，雖然在處理上比較輕鬆，因為沒什麼缺失值，但是準確率就下降了很多，而且不能只單單看準確率來判斷是否準確，因為本來整份資料中佔“漲”的資料比重就較高，因此只要預測的時候只要都猜1準確率就會大於5成，因此透過混淆矩陣或AUC來決定是否較好是比較OK的。而且股票的預測不準就比較正常，不然的話都去玩股票就可以賺大錢！