資料分析與學習基石 hw3-1

統計 109 H24051053 陳知遙

一、資料前處理

由於在這作業的目標我設定為由「前一天的開、高、收、低價及成交量」來預測今天的「收盤價」相較於昨天是「漲或跌」，因此在資料前處理的階段，我先使用 pandas 的內建函數 diff() 來取得一天的差分值，也就是每一筆資料都和前一筆相減後的結果，根據這個值是否大於零來判斷漲跌，再往前平移一天，就能夠讓我們的 train_y 是根據「前一天的資料」而得。

```
train_data['Diff'] = train_data['Close Price'].diff(periods=1)
train_data.head()
```

	Open Price	Close Price	High Price	Low Price	Volume	Diff
0	902.99	931.80	934.73	899.35	4048270080	NaN
1	929.17	927.45	936.63	919.53	5413910016	-4.35
2	931.17	934.70	943.85	927.28	5392620032	7.25
3	927.45	906.65	927.45	902.37	4704940032	-28.05
4	905.73	909.73	910.00	896.81	4991549952	3.08

▲先得到每一筆資料和前一天相減的值

```
train_y = (train_data['Close Price'].diff(periods=1) > 0) * 1
train_y
```

```
0       0
1       0
2       1
3       0
4       1
       ..
2258    1
2259    0
2260    0
2261    1
2262    1
Name: Close Price, Length: 2263, dtype: int32
```

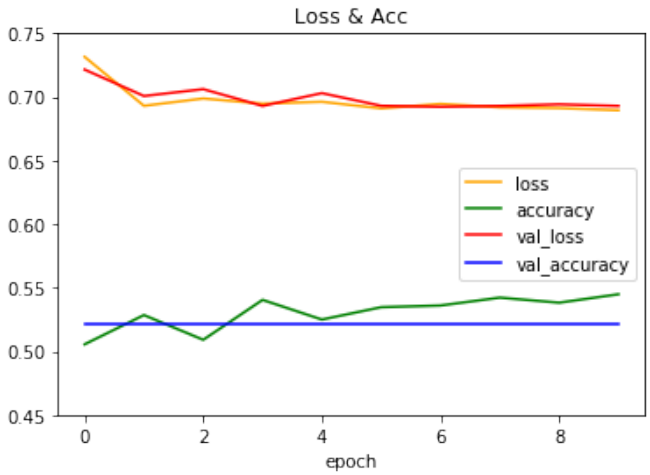
▲用不等式的條件得到 True/False，再乘以 1 轉換為 0/1，
就得到了我們訓練的標籤 train_y

二、模型探討

在這份作業中我實作了三種不同模型，分別是(1)Logistic Regression (2)Neural Network 及 (3)Random Forest，神奇的是這三種模型在測試集上的表現都一樣！而在訓練集上，Random Forest 取得了最好的準確度，達 0.5842，將三種模型的準確度整理成表格如下：

模型名稱	準確度(訓練集)	準確度(測試集)
Logistic Regression	0.5462	0.5219
Neural Network	0.5449	0.5219
Random Forest	0.5842	0.5219

其中 NN 的模型我也嘗試了 LSTM，不過由結果來看並沒有比較好，而且畫出來的 loss 起伏很大，感覺 train 不太起來，因此最後的結果還是由三層 Dense 構成的基本 NN。我也把 loss 和 accuracy 的圖畫了出來，如下：



奇怪的是這三種模型以及 NN 在每一個 epoch 所預測的準確度都是 0.5219，為了改善這個問題，我試著將資料進行標準化，結果 Random Forest 在測試集上的表現進步到了 0.5697！重新整理一次表格如下：

模型名稱	準確度(訓練集)	準確度(測試集)
Logistic Regression	0.5462	0.5219
Neural Network	0.5449	0.5219
Random Forest	0.5842	0.5697

我認為 Random Forest 能取得最好的表現，很可能是因為它是樹的模型，本身就比較抗噪音一點，加上又是投票過後的結果，使得它能具有不錯的預測能力，而且可以透過調整參數來幫助得到更好的預測。不過我覺得不一定對於所有的資料集都會是隨機森林最好，每個資料集的特性不同，可能也會讓結果有很大的不同，也許做價格預測這樣的迴歸問題而不是分類問題，隨機森林的表現就有機會不如類神經網路，但也都很難說，實際上還是要看到資料集後嘗試看看才能確定。

三、參數調整

基本上我讓分類器能夠進步的方式，除了上面提到的標準化以外，就剩調

整參數了，Logistic Regression 能調的參數不多、NN 我嘗試了 LSTM、不同的排列組合及深度，還有 activation function 的使用，最後保留了三層 Dense 的模型，而 Random Forest 我將 n_estimators 調整至 200、max_depth 調整至 3，在我試過的組合中能達到最好的準確度。