**FDA HW3-1 Report**

**學號：F64051164**

**系級：109資訊系**

**姓名：張嘉榮**

1. **簡介**

這次作業的目標是利用S&P 500的股市資料預測往後收盤價的漲跌，而圖(一)是2009年1月到2017年12月之間收盤價的漲跌情形，整體而言收盤價呈現上漲的趨勢，但是我認為讓模型來學習這些資料之間的短期變化特徵才是關鍵，而非只是一個大方向，更進一步來說，就是把一段期間的漲跌(像是1天、1個星期)取差值讓模型學習。

不同的模型也會使結果有所差異，因此我比較Logistic Regression、DecisionTreeClassifier和NN模型最終結果的差異，比較方式是以訓練精準度以及測試精準度的高低為依據。

****

圖(一)

1. **資料預處理**
2. **讀取csv檔案並存成DataFrame格式。**

training\_set = pd.read\_csv('training\_set.csv')

1. **將每天的收盤價相減後存成整數，1代表漲，0代表跌，然後擷取需要的資料。**

training\_set['up\_down'] = ((training\_set['Close Price'].diff().shift(-1)) >= 0).astype('int')

train\_x = training\_set[['Open Price', 'Close Price','High Price','Low Price','Volume']]

train\_y = training\_set['up\_down']

1. **用StandardScaler將資料標準化，並轉回原本的維度。**

scaler.fit(train\_x)

train\_x = scaler.transform(train\_x)

train\_x = np.transpose(train\_x)

1. **將資料變成DataFrame格式**

train\_x = pd.DataFrame({

'Open Price': train\_x[0],

'Close Price': train\_x[1],

'High Price': train\_x[2],

'Low Price': train\_x[3],

'Volume': train\_x[4],

})

**處理前：**



**處理後：**

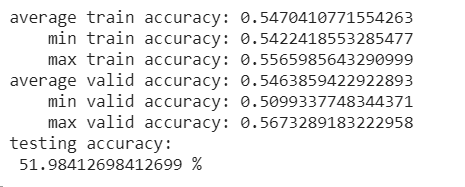


1. **模型實驗結果**

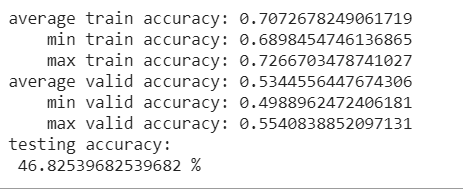
實驗過程的超參數設定理由與模型改進的方式請參考main.ipynb檔案，驗證模型的演算法採取KFold-5，NN模型hidden layer層數設定在(5,50,20,2)，層數太少就要使用更多node達到相近的精準度，太多層則會有overfitting的情況，而整體而言模型改進的方式便是從training accuracy和average accuracy之間的差異與testing accuracy高低去調整超參數，至於最終測試精準度直接以答對題數所佔的百分比來比較模型之間的差異。

**實驗結果：**

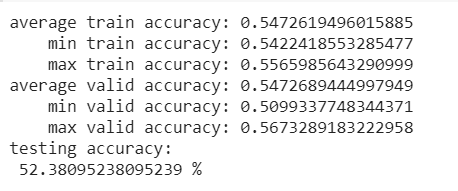
**Logistic Regression**



**DecisionTreeClassifier**



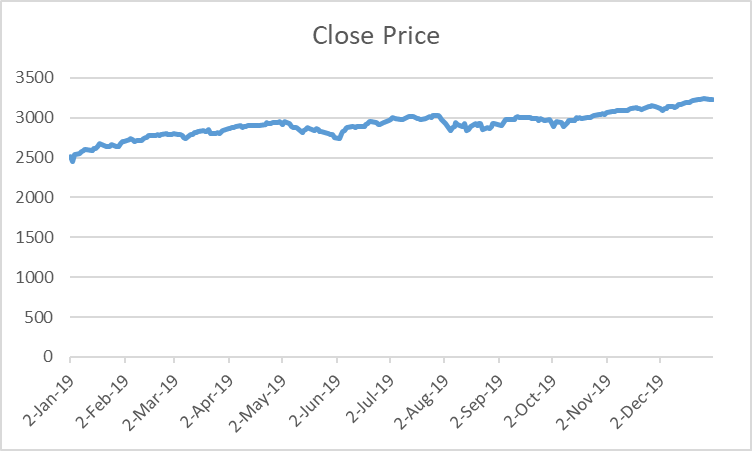
**NN模型**



根據實驗的結果來看，DecisionTreeClassifier有明顯的overfitting現象，且DecisionTreeClassifier精準度最低，而NN模型的精準度最高，達到52.4%。

1. **執行結果與討論**

**圖(二)是2019年收盤價折線圖，改用2019年的資料測試結果與之前測試結果相似，但整體精準度變高：**

****

**圖(二)**

1. **DecisionTreeClassifier的結果仍然是3個模型之間最低的(54.8%)，且仍然有overfitting的情況。**
2. **LogisticRegression與NN模型精準度相同(59.12%)。**

**結論：**

從這次的結果來看，在資料量沒有非常多的情況下(僅2000多筆)，Logistic Regression與層數不多的NN模型的強度非常接近(可能是因為LogisticRegression算是ANN的前身)，但是NN模型在兩次的測試之中皆保持第一名的成績，就這次的訓練成果而言，NN模型還是比Logistic Regression好一點。

DecisionTreeClassifier有overfitting的現象，若減少太多leaf node數量會使模型亂猜，像是全猜上漲或全猜下跌，原因可能是因為沒辦法學到更深層股市漲跌的特徵，無法有效進行feature analysis，但是也不代表DecisionTreeClassifier比較差，只是這些股市的資料可能並不適合DecisionTreeClassifier學習。

我有嘗試使用1~2層的hidden layer，但是因為要使用更多的node才能達到相等的精確度，因此我採取多一點層數的策略以減少運算資源，但是層數太多又會導致overfitting，所以我將hidden layer層數設為4。

嚴格來說NN模型在這次的資料表現最好，其次是LogisticRegression，最後是DecisionTreeClassifier。