FDA_HW3

交管 109 劉冠廷 學號：H54051261

- Please implement **3 classifiers** to predict the stock movement.
    - Logistic Regression:

```
average train accuracy: 0.5475044769380489
    min train accuracy: 0.5383765875207068
    max train accuracy: 0.5552486187845304
average valid accuracy: 0.5475219285393347
    min valid accuracy: 0.5165562913907285
    max valid accuracy: 0.584070796460177
test accuracy: 0.5258964143426295
```

    - Neural Network:

```
test accuracy: 0.5258964143426295
```

    - Random forest:

```
average train accuracy: 1.0
    min train accuracy: 1.0
    max train accuracy: 1.0
average valid accuracy: 0.5037547129266053
    min valid accuracy: 0.4756637168141593
    max valid accuracy: 0.5287610619469026
test accuracy: 0.4940239043824701
```

    - Naive bayes:

```
average train accuracy: 0.5470629761036758
    min train accuracy: 0.5350635008282717
    max train accuracy: 0.5574585635359116
average valid accuracy: 0.5430961730059192
    min valid accuracy: 0.5209713024282561
    max valid accuracy: 0.577433628318584
test accuracy: 0.5258964143426295
```

- How did you preprocess this dataset ?
  1. 首先，利用 info()函式得知 train data 裡的資料筆數及型態，每個欄位皆有 2264 筆的資料，並無缺失，因此不須填補資料。

```
RangeIndex: 2264 entries, 0 to 2263
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Date         2264 non-null   object
 1   Open Price   2264 non-null   float64
 2   Close Price  2264 non-null   float64
 3   High Price   2264 non-null   float64
 4   Low Price    2264 non-null   float64
 5   Volume       2264 non-null   int64
```

  2. 再來，由於 date 的資料型態為 object 因此利用 LabelEncoder()給予類別數值資料。

```
le = LabelEncoder()
le.fit(train_x['Date'])
train_x['Date'] = le.transform(train_x['Date'])
```

  3. 將 Close Price 欄位的值取出，把當天與隔天互相比較，若當天較高則跌設為 0，若隔天較高則漲設為 1，並設為答案資料。

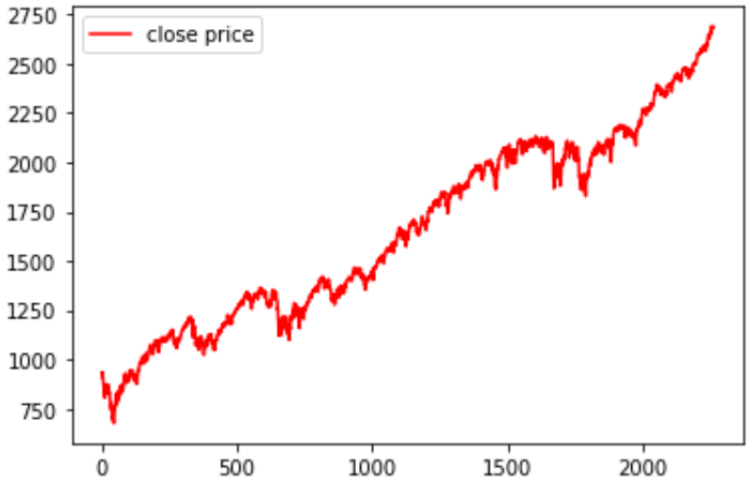
```
for i in range(0, train_datanumber-1):
  if df['Close Price'][i] > df['Close Price'][i+1]:
    train_trend = train_trend.append(pd.Series(0))
  else:
    train_trend = train_trend.append(pd.Series(1))

train_y = train_trend.to_frame()
```

  4. 因為並無最後一天之隔天資料，所以將最後一天的資料丟棄。

```
train_x = train_x.drop(index=[train_datanumber-1])
```

  5. 畫出收盤價之時間走勢圖



- Which classifier reaches the highest classification accuracy in this dataset ?

    - Why:
      利用羅吉斯回歸模型、簡單貝氏模型以及神經網路預測後的準確度皆相同且為最高，但觀察後其預測結果皆為隔天收盤價會漲的情況，調整部分參數後結果並無太大變化，除隨機森林模型為 overfitting 的狀況準確度較低外，得出其他三者模型之預測準確率及結果皆相同之結論，並未找出結果相同之主因。

    - Can this result remain if the dataset is different:
      結果應為不同，股市變動雖有一定的相似性，但不可能為完全相同，因為股市會受到太多變數所影響，而造成變動的變數並非皆為數值資料，可能為隨機突發事件。如 2007 年的美國次貸危機，由於信用緊縮問題，導致股市大幅受到衝擊，此種狀況難以預測，因此資料集的不同會使預測之結果產生不同的變化。

- How did you improve your classifiers ?
  一開始使用羅吉斯回歸模型作為第一個模型，預測結果皆為隔天股價會漲，再來使用隨機森林模型作為第二個模型，發現會有 overfitting 的情形發生，因此再拿第三個簡單貝氏模型進行實驗，發現竟然與羅吉斯回歸模型有相同的預測結果。最後才使用神經網路進行預測，在實驗中不斷調整神經網路的相關超參數，如不同層數的隱藏層、不同啟動函數、不同的優化方式以及學習率的調整。最後發現，除啟動函數的不同有較明顯差異外，調整其他超參數的效果並不顯著，採其中能達成較高準確率之情況作為最後呈現結果。