作業概述：

本次作業將使用新竹地區 2019 年 10~12 月之空氣品質資料，進行時間序列分析&迴歸預測 pm2.5 值。

雖然一共有 18 種屬性的汙染物，但只需預測 PM2.5 即可(預測未來第一個小時、預測未來第六個小時)，不過其他十七種汙染物也不可以隨意丟棄，因為有兩種 X_train 的取值方式(一種是只用 PM2.5 預測 PM2.5、另一種則是用所有汙染物預測 PM2.5)。

因為此作業為分類問題，模型則是採用線性回歸以及隨機森林中的回歸森林來進行預測，利用 MAE 來評估模型好壞。

結果：

線性回歸建模

```
from sklearn.linear_model import LinearRegression
regr = LinearRegression()
```

第一種

```
regr.fit(X1_train, Y1_train)
Y1_predict = regr.predict(X1_test)
```

```
MAE_Y1_0 = 0
for i in (Y1_predict-Y1_test):
    MAE_Y1_0 += abs(i)
MAE_Y1_0 = MAE_Y1_0/len(Y1_predict)
MAE_Y1_0
```

```
2.613432740957174
```

第二種

```
regr.fit(X2_train, Y2_train)
Y2_predict = regr.predict(X2_test)
```

```
MAE_Y2_0 = 0
for i in (Y2_predict-Y2_test):
    MAE_Y2_0 += abs(i)
MAE_Y2_0 = MAE_Y2_0/len(Y2_predict)
MAE_Y2_0
```

```
4.839436524253567
```

第三種

```
regr.fit(X3_train, Y3_train)
Y3_predict = regr.predict(X3_test)
```

```
MAE_Y3_0 = 0
for i in (Y3_predict-Y3_test):
    MAE_Y3_0 += abs(i)
MAE_Y3_0 = MAE_Y3_0/len(Y3_predict)
MAE_Y3_0
```

```
3.72385855036525
```

第四種

```
regr.fit(X4_train, Y4_train)
Y4_predict = regr.predict(X4_test)
```

```
MAE_Y4_0 = 0
for i in (Y4_predict-Y4_test):
    MAE_Y4_0 += abs(i)
MAE_Y4_0 = MAE_Y4_0/len(Y4_predict)
MAE_Y4_0
```

```
6.561998734388232
```

隨機森林建模

```
from sklearn.ensemble import RandomForestRegressor
```

```
# 回歸森林：樹木設100棵、最大深度設為8、因為作業是問mae所以設定criterion='mae'
regr = RandomForestRegressor(n_estimators=100, max_depth=8, criterion='mae', random_state=9487)
```

第一種

```
regr.fit(X1_train, Y1_train)
Y1_predict = regr.predict(X1_test)
```

```
MAE_Y1_1 = 0
for i in (Y1_predict-Y1_test):
    MAE_Y1_1 += abs(i)
MAE_Y1_1 = MAE_Y1_1/len(Y1_predict)
MAE_Y1_1
```

```
2.8004302168021695
```

第二種

```
regr.fit(X2_train, Y2_train)
Y2_predict = regr.predict(X2_test)
```

```
MAE_Y2_1 = 0
for i in (Y2_predict-Y2_test):
    MAE_Y2_1 += abs(i)
MAE_Y2_1 = MAE_Y2_1/len(Y2_predict)
MAE_Y2_1
```

```
5.045965211459755
```

第三種

```
regr.fit(X3_train, Y3_train)
Y3_predict = regr.predict(X3_test)
```

```
MAE_Y3_1 = 0
for i in (Y3_predict-Y3_test):
    MAE_Y3_1 += abs(i)
MAE_Y3_1 = MAE_Y3_1/len(Y3_predict)
MAE_Y3_1
```

```
2.927682926829268
```

第四種

```
regr.fit(X4_train, Y4_train)
Y4_predict = regr.predict(X4_test)
```

```
MAE_Y4_1 = 0
for i in (Y4_predict-Y4_test):
    MAE_Y4_1 += abs(i)
MAE_Y4_1 = MAE_Y4_1/len(Y4_predict)
MAE_Y4_1
```

```
6.497252728512956
```

總結

```
print(f'只有取PM2.5當作X，將未來第一個小時PM2.5當預測目標，用線性回歸建模，其MAE為{MAE_Y1_0:.2f}')
print(f'只有取PM2.5當作X，將未來第六個小時PM2.5當預測目標，用線性回歸建模，其MAE為{MAE_Y2_0:.2f}')
print(f'全部汙染物當作X，將未來第一個小時PM2.5當預測目標，用線性回歸建模，其MAE為{MAE_Y3_0:.2f}')
print(f'全部汙染物當作X，將未來第六個小時PM2.5當預測目標，用線性回歸建模，其MAE為{MAE_Y4_0:.2f}')
print(f'只有取PM2.5當作X，將未來第一個小時PM2.5當預測目標，用隨機森林建模，其MAE為{MAE_Y1_1:.2f}')
print(f'只有取PM2.5當作X，將未來第六個小時PM2.5當預測目標，用隨機森林建模，其MAE為{MAE_Y2_1:.2f}')
print(f'全部汙染物當作X，將未來第一個小時PM2.5當預測目標，用隨機森林建模，其MAE為{MAE_Y3_1:.2f}')
print(f'全部汙染物當作X，將未來第六個小時PM2.5當預測目標，用隨機森林建模，其MAE為{MAE_Y4_1:.2f}')
```

```
只有取PM2.5當作X，將未來第一個小時PM2.5當預測目標，用線性回歸建模，其MAE為2.61
只有取PM2.5當作X，將未來第六個小時PM2.5當預測目標，用線性回歸建模，其MAE為4.84
全部汙染物當作X，將未來第一個小時PM2.5當預測目標，用線性回歸建模，其MAE為3.72
全部汙染物當作X，將未來第六個小時PM2.5當預測目標，用線性回歸建模，其MAE為6.56
只有取PM2.5當作X，將未來第一個小時PM2.5當預測目標，用隨機森林建模，其MAE為2.80
只有取PM2.5當作X，將未來第六個小時PM2.5當預測目標，用隨機森林建模，其MAE為5.05
全部汙染物當作X，將未來第一個小時PM2.5當預測目標，用隨機森林建模，其MAE為2.93
全部汙染物當作X，將未來第六個小時PM2.5當預測目標，用隨機森林建模，其MAE為6.50
```

討論報告：

因為這次作業總共有八個輸出結果，所以有不少變量可以做一定程度的比較。比方說比較一種是只用 PM2.5 預測 PM2.5、另一種則是用所有汙染物預測 PM2.5，從結果發現平均而言前者的預測效果較高，推測原因可能是因為後者輸入太多的 feature 導致模型過於複雜而泛化能力不高。

另一方面，如果比較兩種模型在此問題上何種較適用，我會選擇線性回歸模型。在不考慮繼續調整最佳的超參數的情況下，兩者的表現幾乎是一樣的，但本人在建構隨機森林時本身就是用 MAE 最佳化了，而線性回歸是利用最小平方法來最佳化，但在 MAE 的表現幾乎一樣，因此可以推測在此問題上線性回歸可能較好。

最後一點，比較預測未來第一個小時與預測未來第六個小時的 MAE，前者的預測較佳，也蠻符合常識的，時間隔的越遠，其 PM2.5 可能跑動的範圍更大，更難預測。