資料探勘研究與實務 HW2

0853426 陳紹雲

1. 學生直接用 read_excel 讀入資料。
2. 發現 Data 並不是 365 天都有紀錄，所以先切 Data 分成 df_train df_test
3. 刪['日期', '測站', '測項']因為要處理時間序列 CONCATE
4. Replace NR => 0
5. 異常值都是 String，用 isinstance 挑出，賦予空值以利後續處理

```
if isinstance(df_test.iloc[col, i], str):
    df_test.iloc[col, i] = np.nan
```

6. 將每 18 項合併

```
for i in range(18, len(df_train), 18):
    a = np.concatenate((a, df_train[i:i+18]), axis=1)
```

7. 做成 DataFrame，再用 bfill,ffill 找出前後值/2

```
df_train = (df_train.ffill(axis=1) + df_train.bfill(axis=1)) /2
df_test = (df_test.ffill(axis=1) + df_test.bfill(axis=1)) /2
```

8. 做 x_train, x_test, y_train, y_test
9. 降維

```
x_train = np.array(x_train).reshape(-1, 108)
x_test = np.array(x_test).reshape(-1, 108)
```

程式結果:

LinearRegression

```
In [24]:
lr = LinearRegression(normalize=True)
lr.fit(x_train, y_train)

Out[24]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=True)

In [25]:
pred = lr.predict(x_test)

In [26]:
from sklearn.metrics import mean_squared_error, mean_absolute_error
print("MSE: %.2f"% mean_squared_error(pred, y_test))
print("MAE:  %.2f" %mean_absolute_error(pred, y_test))

MSE: 8.24
MAE:  2.12
```

MSE: 8.24

MAE: 2.12

RandomForestRegressor

```
In [27]:
from sklearn.ensemble import RandomForestRegressor

In [28]:
rf = RandomForestRegressor(n_estimators=100)
rf.fit(x_train, y_train)
pred = rf.predict(x_test)

In [29]:
print("MSE: %.2f"% mean_squared_error(pred, y_test))
print("MAE:  %.2f" %mean_absolute_error(pred, y_test))

MSE: 13.76
MAE:  2.60
```

MSE: 13.76

MAE: 2.60

RandomForestRegressor(加深度)

```
rf = RandomForestRegressor(n_estimators=250, max_depth=5)
rf.fit(x_train, y_train)
pred = rf.predict(x_test)
```

```
print("MSE: %.2f"% mean_squared_error(pred, y_test))
print("MAE:  %.2f" %mean_absolute_error(pred, y_test))
```

```
MSE: 14.34
MAE:  2.64
```

RandomForestRegressor (min-max 標準化)

```
rf = RandomForestRegressor(n_estimators=250, max_depth=5)
rf.fit(x_train_mm, y_train_mm)
pred = rf.predict(x_test_mm)
```

```
print("MSE: %.4f"% mean_squared_error(pred, y_test_mm))
print("MAE:  %.4f" %mean_absolute_error(pred, y_test_mm))
```

```
MSE: 0.0032
MAE:  0.0432
```

結論:

在使用單一 PM2.5 預測時與使用全部特徵預測時下線性回歸下的表現都足以有可信度對未來的 PM2.5 值提供預測。

在線性回歸上有較小的 MAE 值,表示預測出來的結果與實際上的結果幾乎吻合,只有很小的誤差,可以顯示出越接近第七個小時的變化,越能影響到第七小時的 PM2.5 值。

隨機森林回歸法略遜於線性回歸,我認為隨機森林次因為只挑選少數特徵值,跟線性回歸一樣第六小時特徵的重要程度最高,足以代表第七小時的數值,所以其他五小時的特徵對於建樹沒什麼效果,對第七小時沒什麼影響,而線性迴歸還會因為變化而造成小幅的變動,此變動造成實際情況誤差的縮小。