

11

LinearRegression 실습

** 회귀분석에 대한 개념이 기억이 안나시면, 다시 통계적 회귀분석 강의를 복습하고 들으시기를 바랍니다.

[mglearn 데이터 설정]

예제 데이터를 통해 보스턴 데이터셋을 불러오겠습니다.

00:00

The screenshot shows a Jupyter Notebook interface with the title '11.2_LinearRegression (autosaved)'. The notebook contains the following code and output:

```
[예제]
boston 데이터로 LinearRegression을 사용하여 선형 회귀분석을 실시한 후, 추정된 회귀모형에 대해 해석하라
```

```
In [1]: import pandas as pd
import numpy as np
import mglearn
X, y = mglearn.datasets.load_extended_boston()
```

```
In [2]: X
```


```
Out[2]: array([[0.00000000e+00, 1.80000000e-01, 6.78152493e-02, ...,
1.00000000e+00, 8.96799117e-02, 8.04248696e-03],
[2.35822539e+04, 0.00000000e+00, 2.42302053e-01, ...,
1.00000000e+00, 2.04470199e-01, 4.18090621e-02],
[2.35697744e-04, 0.00000000e+00, 2.42302053e-01, ...,
9.79579831e-01, 6.28144504e-02, 4.02790570e-03],
...,
[6.11892474e+04, 0.00000000e+00, 4.20454545e-01, ...,
1.00000000e+00, 1.07891832e-01, 1.16406475e-02],
[1.16072990e+03, 0.00000000e+00, 4.20454545e-01, ...,
9.82676820e-01, 1.29930407e-01, 1.71795127e-02],
[4.61841693e-04, 0.00000000e+00, 4.20454545e-01, ...,
1.00000000e+00, 1.69701987e-01, 2.87987643e-02]])
```

```
In [3]: print(X.shape)
print(y.shape)
```

[훈련 데이터 셋 성능평가]

성능을 보고 모델이 과대적합 된 것을 파악 하실 수 있어야
합니다.

03:01

jupyter 11.2_LinearRegression (unsaved changes)  Logout

File Edit View Insert Cell Kernel Navigate Widgets Help Trusted Python 3 (pykernel)

```

3.74315386, 1.53781147, -2.2785726, -0.09450673,
1.18011524, -2.53839853, 0.97201093, 1.6017904,
2.6937499, 0.89561635, -1.75353745, 0.57738392,
4.44192311, 0.22454182, -0.58752863, -2.14748428,
-3.341989, 1.05177041, -1.3495073, 4.32084074,
0.61217328, -1.00102953, -3.78083157, 3.08335417])

In [10]: ## R2
print("훈련 데이터 셋 성능 : ", lr.score(X_train,y_train))
print("테스트 데이터 셋 성능 : ", lr.score(X_test,y_test))

훈련 데이터 셋 성능 : 0.9517246762476054
테스트 데이터 셋 성능 : 0.6486839499987773

In [11]: mse
(resid**2).sum()/len(y_test)

Out[11]: 29.25250713919926

In [12]: ## rmse
np.sqrt((resid**2).sum()/len(y_test))

Out[12]: 5.40656893330346

In [13]: from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test,y_pred)
rmse = np.sqrt(mse)

print(mse)
print(rmse)

29.25250713919926
5.40656893330346

```