

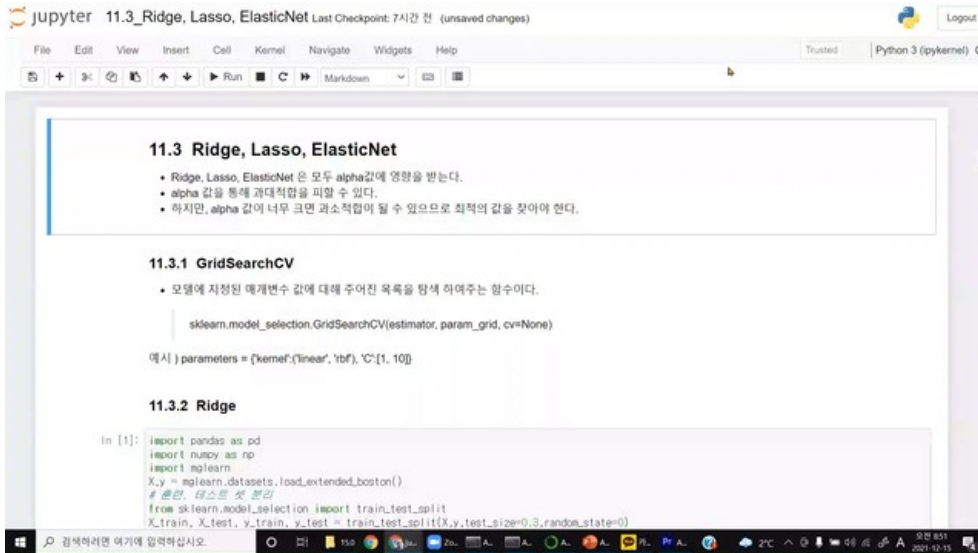
11

## Lidge, Rasso, ElasticNet 실습

[Ridge, Lasso, Elastic]

# 다중 공선성을 제거하기 위해 패널티를 적용시키는  $\alpha$  값을 찾는 것이 중요합니다.

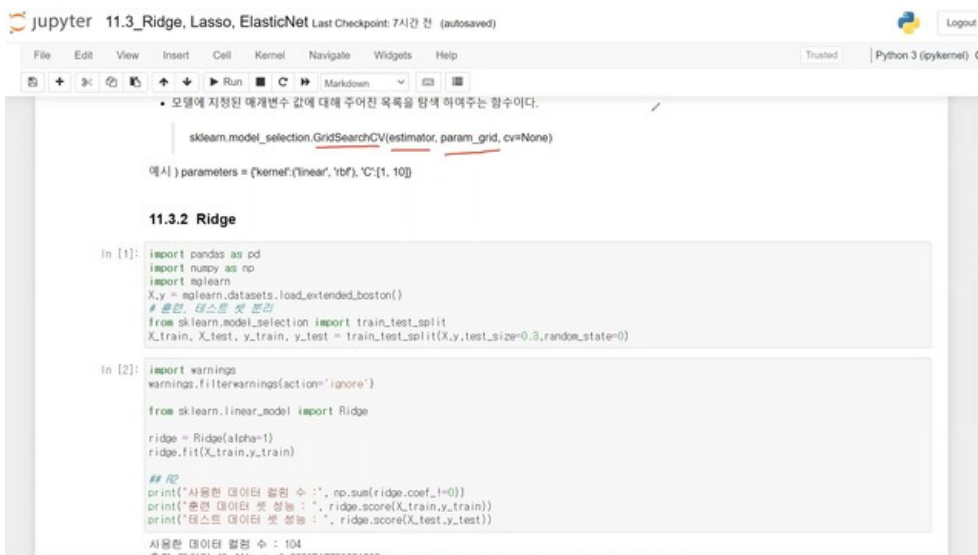
00:00



[Ridge]

# Ridge회귀는 L2패널티를 적용하여 다중공선성이 높은 변수의 계수를 0에 가깝게 만들어줍니다.

01:11



[Lasso]

# L1패널티를 사용하여 다중공선성 문제가 있는 변수를 0으로도 만들어줍니다.

04:33



Jupyter Notebook interface showing code for Ridge regression. The notebook title is "11.3\_Ridge, Lasso, ElasticNet". The code includes imports for pandas, numpy, and sklearn, followed by data loading and model fitting. The output shows the R-squared value and scores for training and testing data.

```
sklearn.model_selection.GridSearchCV(estimator, param_grid, cv=None)

예시 ) parameters = {'kernel':('linear','rbf'), 'C':[1, 10]}

11.3.2 Ridge

In [1]: import pandas as pd
import numpy as np
import sklearn
X,y = mglearn.datasets.load_extended_boston()
# 훈련, 테스트 셋 분리
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=0)

In [2]: import warnings
warnings.filterwarnings(action='ignore')

from sklearn.linear_model import Ridge

ridge = Ridge(alpha=1)
ridge.fit(X_train,y_train)

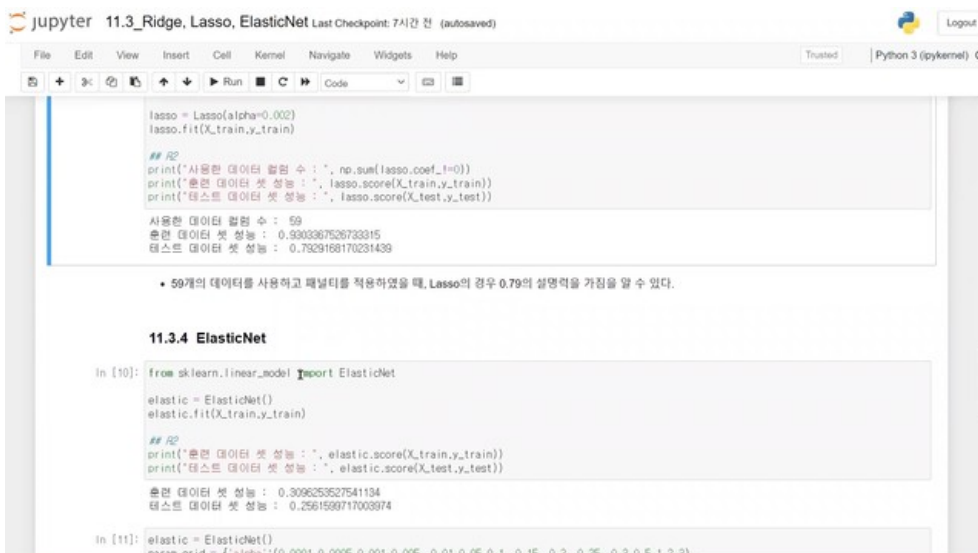
## R2
print("사용한 데이터 합성 수 :", np.sum(ridge.coef_!=0))
print("훈련 데이터 셋 성능 :", ridge.score(X_train,y_train))
print("테스트 데이터 셋 성능 :", ridge.score(X_test,y_test))

사용한 데이터 합성 수 : 104
훈련 데이터 셋 성능 : 0.8202517700991202
```

[Elastic]

# L1, L2패널티와 L1\_ratio를 사용합니다.

06:14



Jupyter Notebook interface showing code for Lasso and ElasticNet regression. The notebook title is "11.3\_Ridge, Lasso, ElasticNet". The code includes imports for sklearn, followed by Lasso model fitting and ElasticNet model fitting. The output shows the R-squared value and scores for training and testing data.

```
lasso = Lasso(alpha=0.002)
lasso.fit(X_train,y_train)

## R2
print("사용한 데이터 합성 수 :", np.sum(lasso.coef_!=0))
print("훈련 데이터 셋 성능 :", lasso.score(X_train,y_train))
print("테스트 데이터 셋 성능 :", lasso.score(X_test,y_test))

사용한 데이터 합성 수 : 59
훈련 데이터 셋 성능 : 0.930386752673315
테스트 데이터 셋 성능 : 0.7929168170231439

59개의 데이터를 사용하고 패널티를 적용하였을 때, Lasso의 경우 0.79의 설명력을 가짐을 알 수 있다.

11.3.4 ElasticNet

In [10]: from sklearn.linear_model import ElasticNet

elastic = ElasticNet()
elastic.fit(X_train,y_train)

## R2
print("훈련 데이터 셋 성능 :", elastic.score(X_train,y_train))
print("테스트 데이터 셋 성능 :", elastic.score(X_test,y_test))

훈련 데이터 셋 성능 : 0.3096253527541134
테스트 데이터 셋 성능 : 0.2561590717003974

In [11]: elastic = ElasticNet()
param_grid = {'alpha':(0.0001,0.0005,0.001,0.005, 0.01,0.05,0.1, 0.15, 0.2, 0.25, 0.3,0.5,1,2,3),
```