

V2018113_???_???

October 30, 2018

1 Ridge, Lasso, Ordinary Least Squares

1.0.1

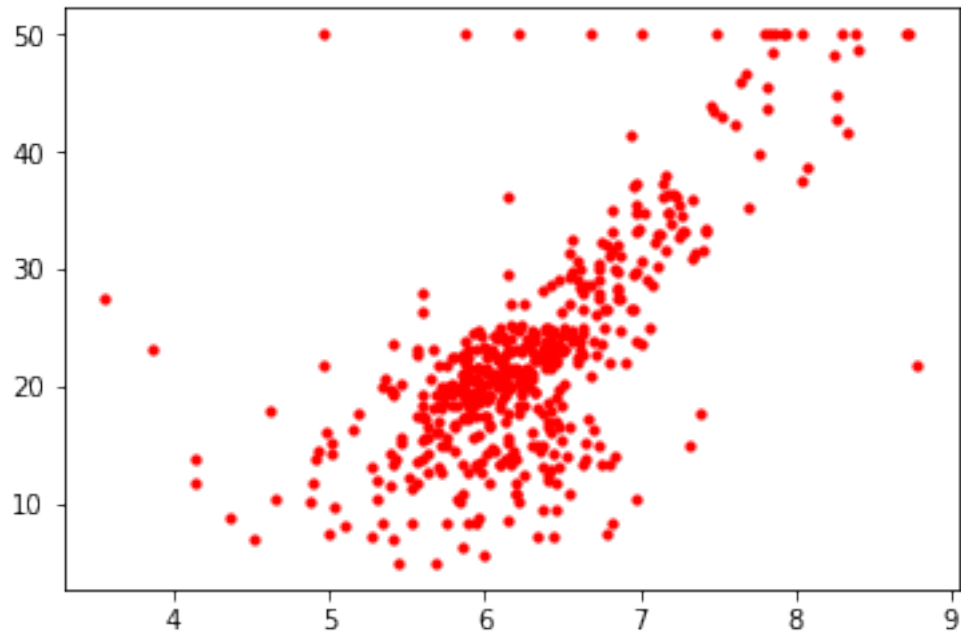
```
In [13]: import numpy as np
import numpy as np
from sklearn.model_selection import train_test_split
import mglearn

from sklearn.datasets import load_boston
boston = load_boston()
print("data shape : {}".format(boston.data.shape))
```

data shape : (506, 13)

```
In [14]: import matplotlib.pyplot as plt
# x: average number of rooms per dwelling( ), y:
plt.scatter(boston.data[:, 5], boston.target, color='r', s=10)
```

Out[14]: <matplotlib.collections.PathCollection at 0x2c06224aac8>



1.1 Linear Regression

Linear Regression, y (mean squared)
 $w b . w \quad b$.

1.1.1 Linear Regression

```
In [72]: from sklearn.linear_model import LinearRegression
X, y = mglearn.datasets.load_extended_boston()

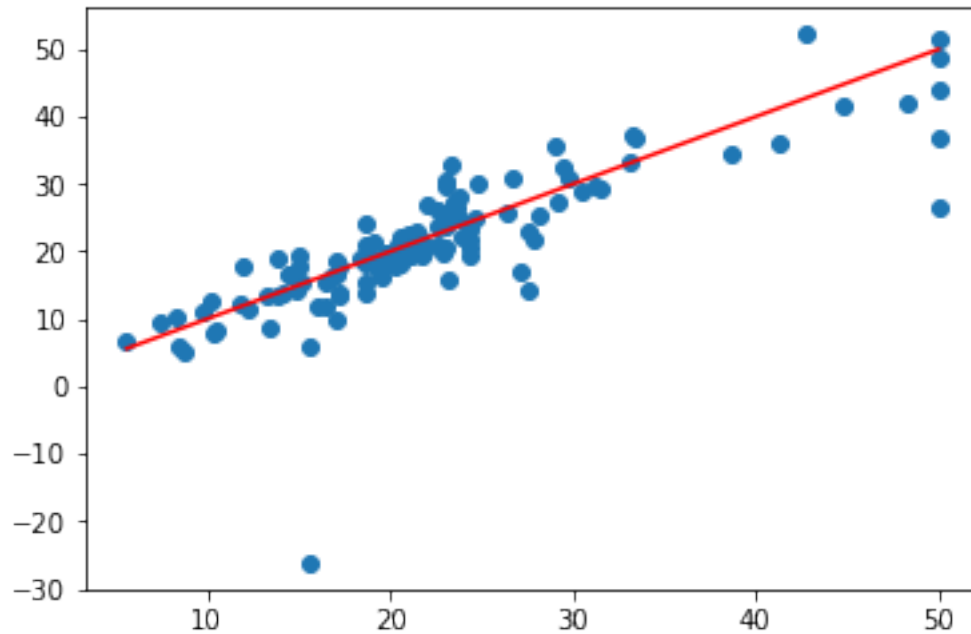
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
lr = LinearRegression().fit(X_train, y_train)

print(" : {:.2f}".format(lr.score(X_train, y_train)))
print(" : {:.2f}".format(lr.score(X_test, y_test)))

y_pred = lr.predict(X_test)
plt.scatter(y_test, y_pred)
line = np.linspace(min(y_test), max(y_test), 1000)
plt.plot(line, line, color = 'r')

: 0.95
: 0.61
```

Out[72]: [



1.2 Ridge Regression

, w_0 .
(regularization) , .

1.2.1 Ridge Regression

In [24]: `from sklearn.linear_model import Ridge`

```
ridge = Ridge().fit(X_train, y_train)
print(" : {:.2f}".format(ridge.score(X_train, y_train)))
print(" : {:.2f}".format(ridge.score(X_test, y_test)))
```

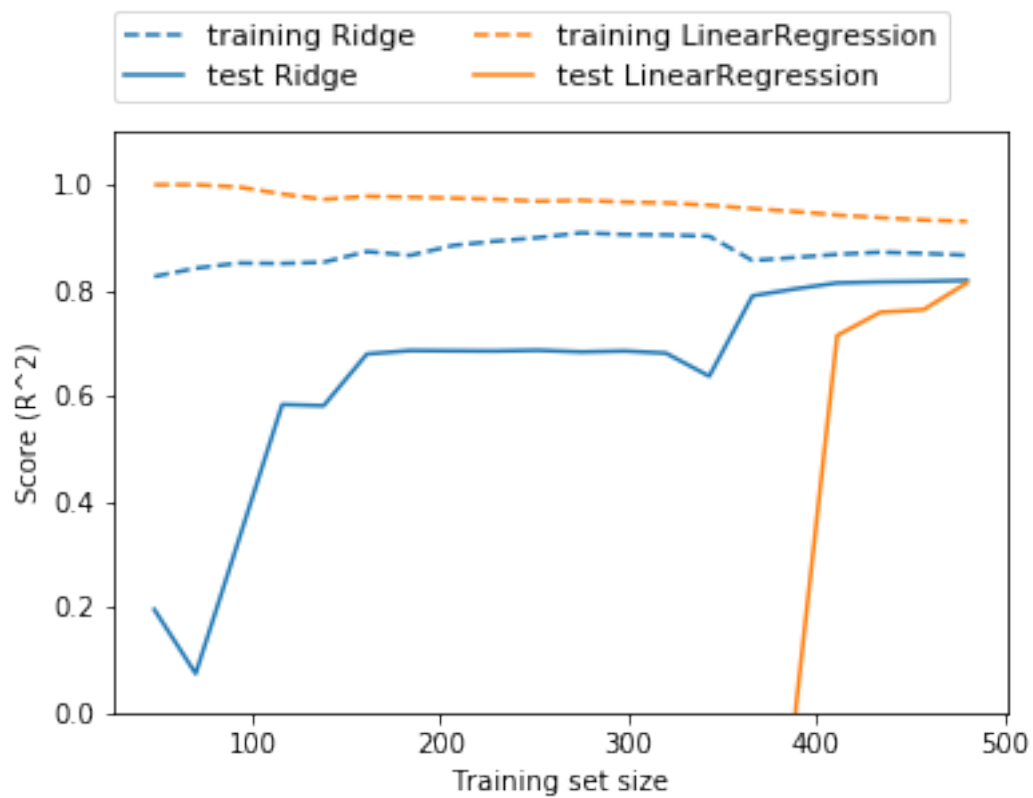
: 0.89

: 0.75

```
In [27]: ridge01 = Ridge(alpha=0.1).fit(X_train, y_train)
print(" : {:.2f}".format(ridge01.score(X_train, y_train)))
print(" : {:.2f}".format(ridge01.score(X_test, y_test)))
```

```
: 0.93  
: 0.77
```

```
In [33]: mglearn.plots.plot_ridge_n_samples()
```



1.3 Rasso Regression

Ridge 0 L1 .

1.3.1

```
In [73]: from sklearn.linear_model import Lasso  
  
lasso = Lasso().fit(X_train, y_train)  
print(" : {:.2f}".format(lasso.score(X_train, y_train)))
```

```

print(" : {:.2f}".format(lasso.score(X_test, y_test)))
print(" : {}".format(np.sum(lasso.coef_ != 0)))

: 0.29
: 0.21
: 4

```

104 4 alpha

```

In [35]: # "max_iter"    max_iter
lasso001 = Lasso(alpha=0.01, max_iter=100000).fit(X_train, y_train)
print(" : {:.2f}".format(lasso001.score(X_train, y_train)))
print(" : {:.2f}".format(lasso001.score(X_test, y_test)))
print(" : {}".format(np.sum(lasso001.coef_ != 0)))

: 0.90
: 0.77
: 33

```

alpha LinearRegression .

```

In [37]: # "max_iter"    max_iter
lasso001 = Lasso(alpha=0.0001, max_iter=100000).fit(X_train, y_train)
print(" : {:.2f}".format(lasso001.score(X_train, y_train)))
print(" : {:.2f}".format(lasso001.score(X_test, y_test)))
print(" : {}".format(np.sum(lasso001.coef_ != 0)))

: 0.95
: 0.64
: 94

```