

Lasso on some values

```
In [1]: import numpy as np
from sklearn.linear_model import Lasso

# Creating a toy dataset
X = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
y = np.array([10, 11, 12])

# Creating Lasso model with alpha=0.1
lasso = Lasso(alpha=0.1)

# Fitting the model on the dataset
lasso.fit(X, y)

# Printing the coefficients and intercept
print("Coefficients:", lasso.coef_)
print("Intercept:", lasso.intercept_)

Coefficients: [3.16666667e-01 2.46716228e-17 0.00000000e+00]
Intercept: 9.733333333333333
```

LASSO ON DATASET

```
In [2]: from sklearn.datasets import load_boston
from sklearn.linear_model import Lasso
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

# Loading the Boston housing dataset
boston = load_boston()

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(boston.data, boston.target, test_size=0.2, random_state=42)

# Creating a Lasso model with alpha=0.1
lasso = Lasso(alpha=0.1)

# Fitting the model on the training set
lasso.fit(X_train, y_train)

# Predicting on the testing set
y_pred = lasso.predict(X_test)

# Calculating the mean squared error of the predictions
mse = mean_squared_error(y_test, y_pred)

# Printing the mean squared error and the coefficients of the Lasso model
print("Mean Squared Error:", mse)
print("Coefficients:", lasso.coef_)

Mean Squared Error: 25.15559375393417
Coefficients: [-0.10415691  0.03489335 -0.01678527  0.91995182 -0.          4.31168655
 -0.01512583 -1.15148729  0.23923695 -0.01296223 -0.73224678  0.01309057
 -0.56467442]

c:\users\91798\appdata\local\programs\python\python39\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Function load_boston is deprecated; `load_boston` is deprecated in 1.0 and will be removed in 1.2.

The Boston housing prices dataset has an ethical problem. You can refer to the documentation of this function for further details.

The scikit-learn maintainers therefore strongly discourage the use of this dataset unless the purpose of the code is to study and educate about ethical issues in data science and machine learning.

In this special case, you can fetch the dataset from the original source::

import pandas as pd
import numpy as np

data_url = "http://lib.stat.cmu.edu/datasets/boston"
raw_df = pd.read_csv(data_url, sep="s+", skiprows=22, header=None)
data = np.hstack([raw_df.values[::2, :], raw_df.values[1::2, :2]])
target = raw_df.values[1::2, 2]

Alternative datasets include the California housing dataset (i.e. :func:`~sklearn.datasets.fetch_california_housing`) and the Ames housing dataset. You can load the datasets as follows::

from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()

for the California housing dataset and::

from sklearn.datasets import fetch_openml
housing = fetch_openml(name="house_prices", as_frame=True)

for the Ames housing dataset.
warnings.warn(msg, category=FutureWarning)
```

RIDGE ON SOME VALUES

```
In [3]: import numpy as np
from sklearn.linear_model import Ridge

# Creating a toy dataset
X = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
y = np.array([10, 11, 12])

# Creating Ridge model with alpha=0.1
ridge = Ridge(alpha=0.1)

# Fitting the model on the dataset
ridge.fit(X, y)

# Printing the coefficients and intercept
print("Coefficients:", ridge.coef_)
print("Intercept:", ridge.intercept_)

Coefficients: [0.11090573 0.11090573 0.11090573]
Intercept: 9.33641404805915
```

RIDGE ON DATASET

```
In [4]: from sklearn.datasets import load_boston
from sklearn.linear_model import Ridge
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

# Loading the Boston housing dataset
boston = load_boston()

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(boston.data, boston.target, test_size=0.2, random_state=42)

# Creating a Ridge model with alpha=0.1
ridge = Ridge(alpha=0.1)

# Fitting the model on the training set
ridge.fit(X_train, y_train)

# Predicting on the testing set
y_pred = ridge.predict(X_test)

# Calculating the mean squared error of the predictions
mse = mean_squared_error(y_test, y_pred)

# Printing the mean squared error and the coefficients of the Ridge model
print("Mean Squared Error:", mse)
print("Coefficients:", ridge.coef_)

Mean Squared Error: 24.30102550019275
Coefficients: [-1.12399694e-01  3.04593914e-02  3.48958400e-02  2.75033318e+00
 -1.59244585e+01  4.44577949e+00 -7.30474388e-03 -1.42960751e+00
 2.60042840e-01 -1.07802286e-02 -9.00771040e-01  1.24004789e-02
 -5.10902332e-01]

c:\users\91798\appdata\local\programs\python\python39\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Function load_boston is deprecated; `load_boston` is deprecated in 1.0 and will be removed in 1.2.

The Boston housing prices dataset has an ethical problem. You can refer to the documentation of this function for further details.

The scikit-learn maintainers therefore strongly discourage the use of this dataset unless the purpose of the code is to study and educate about ethical issues in data science and machine learning.

In this special case, you can fetch the dataset from the original source::

import pandas as pd
import numpy as np

data_url = "http://lib.stat.cmu.edu/datasets/boston"
raw_df = pd.read_csv(data_url, sep="s+", skiprows=22, header=None)
data = np.hstack([raw_df.values[::2, :], raw_df.values[1::2, :2]])
target = raw_df.values[1::2, 2]

Alternative datasets include the California housing dataset (i.e. :func:`~sklearn.datasets.fetch_california_housing`) and the Ames housing dataset. You can load the datasets as follows::

from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()

for the California housing dataset and::

from sklearn.datasets import fetch_openml
housing = fetch_openml(name="house_prices", as_frame=True)

for the Ames housing dataset.
warnings.warn(msg, category=FutureWarning)
```

