```
 numpy as np
 pandas as pd
 sklearn
klearn.datasets import load_boston
d_boston()
s()
df.DESCR)
=pd.DataFrame(df.data, columns=df.feature_names)
.head()
['MEDV']= df.target
.head()
.isnull()
.isnull().sum ()
klearn.model_selection import train_test_split
ston.drop('MEDV' , axis=1)
ston['MEDV']
n, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.15, random_state= 5)
X_train,)
X_test,)
Y_train,)
Y_test,)
klearn.linear_model import LinearRegression
klearn.metrics import mean_squared_error
del = LinearRegression()
del.fit(X_train, Y_train)
n_predict= lin_model.predict(X_train)
(np.sqrt(mean_squared_error(Y_train, y_train_predict)))
"The model performance for training set")
'RMSE is {}'.format(rmse))
"\n")
_predict = lin_model.predict(X_test)
(np.sqrt(mean_squared_error(Y_test, y_test_predict)))
"The model performance for testing set")
'RMSE is {}'.format(rmse))
     495     23.1
     206     24.4
     355     20.6
     Name: MEDV, Length: 430, dtype: float64
     226     37.6
     292     27.9
     90      22.6
     373     13.8
     273     35.2
             ...
     311     22.1
     49      19.4
     409     27.5
     407     27.9
     353     30.1
     Name: MEDV, Length: 76, dtype: float64
     The model performance for training set
     RMSE is 4.710901797319796


     The model performance for testing set
     RMSE is 4.687543527902972
     /usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWa

        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::

```python
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::

```python
from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()
```

for the California housing dataset and::

```python
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)