Import

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.datasets import load_boston
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

Data

```
boston = load_boston()
```

```
print(boston.DESCR)
```

```
.. _boston_dataset:

Boston house prices dataset
---------------------------

**Data Set Characteristics:**

    :Number of Instances: 506

    :Number of Attributes: 13 numeric/categorical predictive. Median Value (attribute

    :Attribute Information (in order):
        - CRIM     per capita crime rate by town
        - ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
        - INDUS    proportion of non-retail business acres per town
        - CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwis
        - NOX      nitric oxides concentration (parts per 10 million)
        - RM       average number of rooms per dwelling
        - AGE      proportion of owner-occupied units built prior to 1940
        - DIS      weighted distances to five Boston employment centres
        - RAD      index of accessibility to radial highways
        - TAX      full-value property-tax rate per $10,000
        - PTRATIO  pupil-teacher ratio by town
        - B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
        - LSTAT    % lower status of the population
        - MEDV     Median value of owner-occupied homes in $1000's

    :Missing Attribute Values: None

    :Creator: Harrison, D. and Rubinfeld, D.L.

This is a copy of UCI ML housing dataset.
https://archive.ics.uci.edu/ml/machine-learning-databases/housing/


This dataset was taken from the StatLib library which is maintained at Carnegie Mellon

The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic
```

```
prices and the demand for clean air', J. Environ. Economics & Management,
vol.5, 81-102, 1978.   Used in Belsley, Kuh & Welsch, 'Regression diagnostics
...', Wiley, 1980.   N.B. Various transformations are used in the table on
pages 244-261 of the latter.

The Boston house-price data has been used in many machine learning papers that address
problems.

.. topic:: References

   - Belsley, Kuh & Welsch, 'Regression diagnostics: Identifying Influential Data and
   - Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedi
```

```
X = boston.data
Y = boston.target
```

```
fnames = boston.feature_names
fnames
```

```
array(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
       'TAX', 'PTRATIO', 'B', 'LSTAT'], dtype='<U7')
```

```
df = pd.DataFrame(X, columns=fnames)

df
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	3
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	3
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	3
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	3
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	3
...	...	...	...	...	...	...	...	...	...	...	...	
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1.0	273.0	21.0	3
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1.0	273.0	21.0	3
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1.0	273.0	21.0	3
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1.0	273.0	21.0	3
505	0.04741	0.0	11.93	0.0	0.573	6.030	80.8	2.5050	1.0	273.0	21.0	3

506 rows × 13 columns

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3)
```

Model

```
model = LinearRegression()
model.fit(X_train, Y_train)
```

```
model.fit(X_train, Y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
model.coef_
```

```
array([-1.24575891e-01,  3.92667679e-02,  7.14250776e-02,  2.03295124e+00,
       -1.44513410e+01,  5.66149460e+00, -1.95314048e-02, -1.21111167e+00,
        2.53374359e-01, -1.32456171e-02, -8.81067876e-01,  1.30547660e-02,
       -3.55284244e-01])
```

```
model.intercept_
```

```
18.86359991598138
```

Analysis

```
Y_test_pred = model.predict(X_test)
```

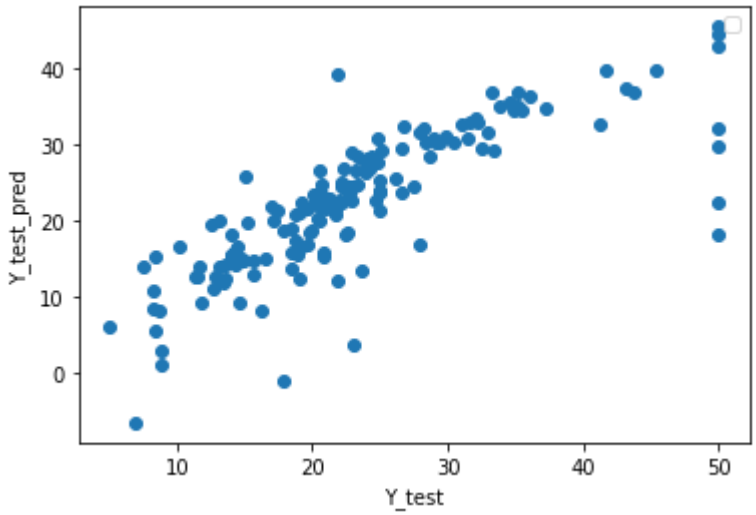
```
Y_test_pred
```

```
array([30.32514359, 23.74077176, 21.76719469, 28.97068424, 23.66342409,
       27.8321864 , 39.37338161, 33.48516036, 36.32008861, 30.81583227,
       25.04129225, 29.230053  , 21.8996646 , 11.10654353, 19.52757803,
       14.75098475, 19.10274702, 21.50066639, 22.03350489, 16.44163446,
        8.34793366, 16.08027641, 19.88247596, 35.66769837, 22.54937383,
       20.10775551, 32.75024198,  8.46897004, 26.94116549, 34.50630881,
        3.88231635, 24.43824005, 17.49979909, 26.6379865 ,  8.31598735,
       22.8721461 , 25.78117104, 16.71439958, 27.79295512, 28.58852428,
       18.66948565, 15.2911278 , 30.78657144, 25.24475987, 12.75141543,
       -0.82158391, 23.42390653, 14.79866359, 36.78680636, 36.80888798,
       35.17616094, 22.83831135, 26.7846726 , 16.69732777, 24.21866753,
       24.77649251, 21.51041654, 32.42037996, 20.21888242, 22.6263824 ,
        9.29914925, 18.53575671, 29.28295359, 15.16562037, 33.06257153,
       21.75580991, 22.77533424,  3.12321259, 20.98442226, 44.60268428,
       24.71033686, 39.80741053, 29.42474002, 30.91137457, 12.12392731,
        6.25394666, 34.90806141, 13.63410178, 29.70268695, 28.04076904,
       34.62009838, 14.2706418 , 15.57610629, 15.94284629,  5.71283876,
       25.49077643, 32.58932057, 15.31357257, 30.28011894, 16.92107418,
       18.30545835, 11.01725913, 26.41804604, 21.6885711 , 42.97582453,
       12.62932015, 14.17806401, 23.73920966, 24.67482881, 35.16707311,
       28.3784346 , 30.24816052, 29.49986838, 28.23534608, 37.55959414,
       20.87754134, 18.42353801, 15.24759481, 31.62015052, 31.65968885,
       32.1634374 , 14.96329214,  9.22941603, 12.62938769, 13.99408352,
       20.16810444, 12.44173586, 24.45380306, 21.83211862, 28.54342561,
       15.7558299 , 22.49754487, 16.87173361, 32.14478271, 13.9695373 ,
       24.63614464, 22.59114099, 13.71658146, 22.4079691 , 21.17746589,
       24.11627228, 12.94474089, 45.48998335, 12.06647205, 30.99818954,
       18.19219687, 39.81785776, 18.20401056, 23.65554193, 18.77938331,
       20.16507089, 15.5884013 , 32.95644926, 23.28893701,  1.0540298 ,
       12.42567485, -6.45859297, 26.92561916, 14.10492784, 30.31145784,
       21.79797789, 36.87057957])
```

```
plt.scatter(Y_test, Y_test_pred)
plt.xlabel("Y_test")
plt.ylabel("Y_test_pred")
plt.legend()
```

```
plt.show()
```

```
No handles with labels found to put in legend.
```

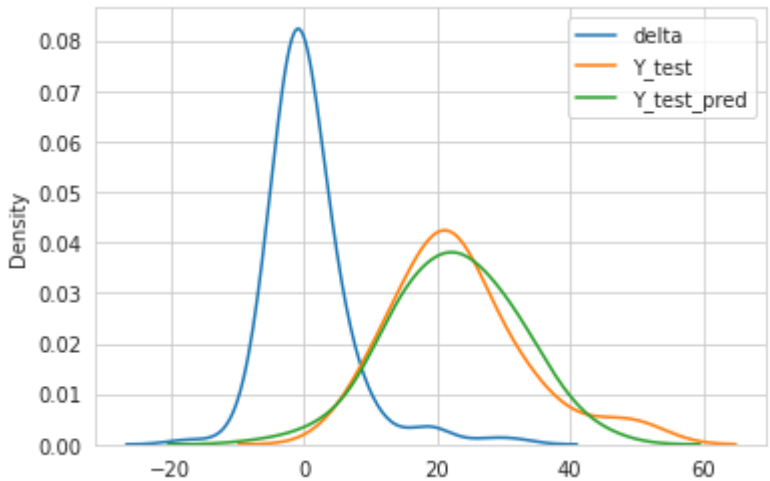


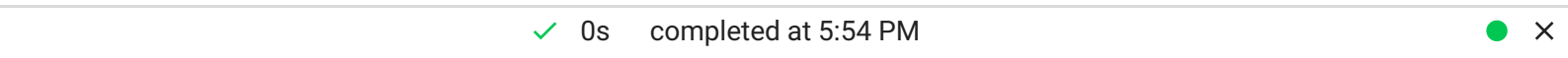

```
print("Squared error", np.sum((Y_test_pred-Y_test)**2))
```

```
Squared error 5742.982148481666
```

```
delta = Y_test - Y_test_pred
sns.set_style('whitegrid')
sns.kdeplot(delta, bw=0.5, label='delta')
sns.kdeplot(Y_test, bw=0.5, label='Y_test')
sns.kdeplot(Y_test_pred, bw=0.5, label='Y_test_pred')
plt.legend()
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:1657: FutureWarning: T
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:1657: FutureWarning: T
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:1657: FutureWarning: T
  warnings.warn(msg, FutureWarning)
```




0s
completed at 5:54 PM