In [2]:
```
pip install scikit-learn==1.1.3
```

...

In [3]:
```
from sklearn.datasets import load_boston
boston_dataset = load_boston()
```

...

In [4]:
```
print(boston_dataset.keys())
```

```
dict_keys(['data', 'target', 'feature_names', 'DESCR', 'filename', 'data_modu
le'])
```

In [5]:
```
import pandas as pd
import numpy as np
```

In [6]:
```
boston = pd.DataFrame(boston_dataset.data, columns=boston_dataset.feature_name
boston.head()
```

Out[6]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTA
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.9
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.1
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.0
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.9
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.3

In [7]:
```
boston['MEDV'] = boston_dataset.target
boston.isnull().sum()
```

Out[7]:
```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
MEDV       0
dtype: int64
```

In [8]:
```
X = pd.DataFrame(np.c_[boston['LSTAT'], boston['RM']], columns = ['LSTAT','RM'
Y = boston['MEDV']
```

In [9]:
```
from sklearn.linear_model import LinearRegression
```

In [10]:
```
reg = LinearRegression()
```

In [11]:
```
  from sklearn.model_selection import train_test_split
  x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.33,
  random_state=42)
```

In [12]:
```
reg.fit(x_train, y_train)
#k
```

Out[12]:

▾ LinearRegression

LinearRegression()

In [13]:
```
print(reg.coef_)
#c
```

```
[-0.67873898  4.82035621]
```

In [14]:
```
y_pred = reg.predict(x_test)
print(y_pred)
#a
```

```
[25.74505916 31.13347279 17.52883188 26.44227535 19.58541492 23.09596513
 17.01975517 14.3159995  22.24408223 20.27941136 17.47639133 18.28843618
 -2.36312381 22.61827496 20.4996517  26.68544392 17.36009694  3.69380472
 36.89790451 18.4618195  26.55958698 27.31886979 13.15264811 26.17065665
 19.24831147 14.8012021  23.09170175 20.68560431 17.96095488 19.31237833
 17.84180325 27.04214245 27.08246715 19.88385388 15.41933731 17.75557917
 32.8523954  22.59248268 20.23378068 25.93888672 12.58663197 28.85716926
 37.72542766 19.03383346 25.90598948 16.87892513 15.97944452 27.32455396
 19.62393282 29.26717607 20.91210827 31.42628427 17.85408818 28.43436158
 34.81669984 24.19848407 19.67096965 31.76571727 25.22124093 15.44202903
 27.10068247 32.63148179 29.98436944 19.31885888 28.94030062 11.04135456
 20.52596775 26.77021299 29.81155093 16.85707475 19.23987853 28.23741881
 13.02461894 25.78651245 23.57979189  6.476231   22.3117392  36.22171593
 18.19486773 10.78148919 23.35513475 10.11068577 22.99953162  7.79875088
 22.03163869 27.91132118 22.10976697 27.72215664 26.67771746 22.16123133
 23.10994738  8.43901692 23.12568757 20.78429212 13.28612102 24.1211864
 24.44840581 -0.06619623 19.84653693 18.63972417 21.21931339 25.18628675
  9.61323536 23.32649306 24.71049619 14.56425713 20.9034584  28.50517088
 24.61724531 27.80546388 12.65563939 20.55707777 26.78841367 24.15563928
 31.16488573 20.58175509 33.25225339 16.73664792 20.31157182 27.87740279
 19.42946911 28.21854486 15.67271821 22.98580733 26.74675809 24.84693582
 28.04611526 32.17158091 20.41362599 36.09658112  9.50768882 26.49882587
 21.37190068 18.71941492  6.72993339 23.7590128  24.38392365 31.12798307
 29.81420979 18.44579887 19.87028693 28.16528153 21.89856832 10.72649824
  9.03097093 23.88801424 20.2348839  18.58367103 15.48085377 39.86850058
 20.96812037 19.9270319  28.41211446 22.55195671 21.41279308 25.09616572
 17.21334724 26.50671648 32.69081428 12.75011661 21.68768825 19.70170149
 21.99195405 27.02510851 20.66181227 24.99321844 37.42345819]
```

In [15]:
```
y_pred[2]
y_test[0]
#1
```

Out[15]:
```
24.0
```

In [16]:
```
from sklearn.metrics import mean_squared_error
print(mean_squared_error(y_test, y_pred))
#v
```

```
28.97075021100663
```