In [1]:
```
import pandas as pd
from sklearn.datasets import load_boston
from sklearn.linear_model import LinearRegression
```

In [2]:
```
boston = load_boston()
```

```
C:\Users\HP\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWar
ning: Function load_boston is deprecated; `load_boston` is deprecated in 1.0 and w
ill 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)
```

In [3]:
```
df = pd.DataFrame(boston.data, columns = boston.feature_names)
```

In [4]:
```
df['MEDV'] = boston.target
```

In [5]:
```
X = df.drop(['MEDV'], axis=1)
y = df['MEDV']
```

In [6]:
```
model = LinearRegression()
```

In [7]:
```
model.fit(X, y)
```

Out[7]:
```
LinearRegression()
```

In [8]:
```
print("Intercept: ", model.intercept_)
print(pd.DataFrame({'features':X.columns, 'coeficients': model.coef_}))
```

```
Intercept:  36.459488385089806
   features  coeficients
0      CRIM    -0.108011
1        ZN     0.046420
2     INDUS     0.020559
3      CHAS     2.686734
4       NOX   -17.766611
5        RM     3.809865
6       AGE     0.000692
7       DIS    -1.475567
8       RAD     0.306049
9       TAX    -0.012335
10  PTRATIO    -0.952747
11        B     0.009312
12    LSTAT    -0.524758
```

In [ ]: