In [25]:
```
import numpy as np
import pandas as pd
```

In [26]:
```
data = pd.read_csv("HousingData.csv")
data.head()
```

Out[26]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LS
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	

In [27]:
```
data.tail()
```

Out[27]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	L
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	
505	0.04741	0.0	11.93	0.0	0.573	6.030	NaN	2.5050	1	273	21.0	396.90	

In [28]:
```
print("The shape of the data is: ")
data.shape
```

```
The shape of the data is:
```

Out[28]:
```
(506, 14)
```

In [29]:
```
data.isnull().sum()
```

Out[29]:
```
CRIM       20
ZN         20
INDUS      20
CHAS       20
NOX         0
RM          0
AGE        20
DIS         0
RAD         0
TAX         0
PTRATIO     0
B           0
LSTAT      20
MEDV        0
dtype: int64
```

In [30]:
```
X = data.iloc[:,0:13]
y = data.iloc[:,-1]
```

In [31]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,ra
```

In [32]:
```
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(404, 13)
(102, 13)
(404,)
(102,)
```

In [33]:
```
from sklearn.linear_model import LinearRegression
```

In [37]:
```
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import make_pipeline

# Create a pipeline that includes imputation, scaling, and regression
model = make_pipeline(
    SimpleImputer(strategy='mean'),  # Replace 'mean' with 'median', 'most_
    StandardScaler(with_mean=False),
    LinearRegression()
)


model.fit(X_train, y_train)
```

Out[37]:
```
Pipeline(steps=[('simpleimputer', SimpleImputer()),
                ('standardscaler', StandardScaler(with_mean=False)),
                ('linearregression', LinearRegression())])
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [40]:
```
model.score(X_test,y_test)
```

Out[40]:
```
0.6590604241860214
```

In [ ]:

In [ ]: