⌄ Boston Housing Data

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

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]

feature_names = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRATIO', 'B', 'LSTAT']
df = pd.DataFrame(data, columns=feature_names)
df['PRICE'] = target


X = df.drop('PRICE', axis=1)
y = df['PRICE']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'R-squared: {r2}')

results = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
display(results.head())
```

```
Mean Squared Error: 24.291119474973478
R-squared: 0.6687594935356326
```

	Actual	Predicted
173	23.6	28.996724
274	32.4	36.025565
491	13.6	14.816944
72	22.8	25.031979
452	16.1	18.769880

```python
print(df.head())
```

```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  \
0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900  1.0  296.0
1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671  2.0  242.0
2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671  2.0  242.0
3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622  3.0  222.0
4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622  3.0  222.0

   PTRATIO       B  LSTAT  PRICE
0     15.3  396.90   4.98   24.0
1     17.8  396.90   9.14   21.6
2     17.8  392.83   4.03   34.7
3     18.7  394.63   2.94   33.4
4     18.7  396.90   5.33   36.2
```

```python
input_data = {}
print("Please enter values for the following features:")
for feature in feature_names:
    try:
        value = float(input(f"{feature}: "))
        input_data[feature] = value
    except ValueError:
```

```
        print("Invalid input. Please enter a numerical value.")


input_df = pd.DataFrame([input_data])

predicted_price = model.predict(input_df)
54
print(f"\nPredicted PRICE: {predicted_price[0]:.2f}")
```

```
Please enter values for the following features:
CRIM: 0.05
ZN: 16
INDUS: 2.2
CHAS: 0
NOX: 0.5
RM: 6
AGE: 60
DIS: 5
RAD: 2
TAX: 300
PTRATIO: 16
B: 300
LSTAT: 5

Predicted PRICE: 25.07
```