

```
# Importing required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
df = pd.read_csv('/home/charanko/Downloads/HousingData.csv')
df
```

	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	296
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2	242
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2	242
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3	222
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3	222
..
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1	273
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1	273
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1	273
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1	273
505	0.04741	0.0	11.93	0.0	0.573	6.030	NaN	2.5050	1	273

	PTRATIO	B	LSTAT	MEDV
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	NaN	36.2
..
501	21.0	391.99	NaN	22.4
502	21.0	396.90	9.08	20.6
503	21.0	396.90	5.64	23.9
504	21.0	393.45	6.48	22.0
505	21.0	396.90	7.88	11.9

```

[506 rows x 14 columns]
df.isnull().sum()
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

df.fillna(df.median(), inplace=True)

X = df.drop(columns=["MEDV"]) # Features
y = df["MEDV"] # Target variable

# Split dataset into training (80%) and testing (20%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

model = LinearRegression()

# Train the model using the training data
model.fit(X_train, y_train)

LinearRegression()

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error (MSE): {mse}')
print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'R-squared (R²): {r2}')

Mean Squared Error (MSE): 24.999384790103203
Root Mean Squared Error (RMSE): 4.999938478631832
R-squared (R²): 0.6591013893903532

```

```
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='blue')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)],
color='red', lw=2)
plt.xlabel('Act')
Text(0.5, 0, 'Act')
```

