```python
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
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(r"C:\Users\ASUS\Documents\pythonStack\DS_PR\
HousingData.csv")

df.head()
```

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

        B  LSTAT  MEDV
0  396.90   4.98  24.0
1  396.90   9.14  21.6
2  392.83   4.03  34.7
3  394.63   2.94  33.4
4  396.90    NaN  36.2
```

```python
df=df.dropna()
df.head()
```

```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX
PTRATIO  \
0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296
15.3
1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242
17.8
2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242
17.8
3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222
18.7
5  0.02985   0.0   2.18   0.0  0.458  6.430  58.7  6.0622    3  222
18.7

        B  LSTAT  MEDV
0  396.90   4.98  24.0
```

```
1   396.90    9.14   21.6
2   392.83    4.03   34.7
3   394.63    2.94   33.4
5   394.12    5.21   28.7
```

```python
X = df.drop("MEDV", axis=1)  # Inputs
y = df["MEDV"]               # Output (Price)

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

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

```
LinearRegression()
```

```python
pred = model.predict(X_test)
```

```python
print("Mean Squared Error:", mean_squared_error(y_test, pred))
```

```
Mean Squared Error: 21.16807298027489
```

```python
plt.plot(y_test.values, label='Actual', color='blue')
plt.plot(pred, label='Predicted', color='orange')
```

```
[<matplotlib.lines.Line2D at 0x2127cc3b290>]
```