**import** numpy **as** np

**import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.linear\_model **import** LinearRegression

**from** sklearn.metrics **import** mean\_squared\_error, r2\_score

boston **=** pd**.**read\_csv("boston.csv")

print("------------Dataframe Info------------------")

print(boston**.**info())

print("\n")

print("----------Dataframe Describe------------")

print(boston**.**describe())

print("\n")

print("-----------Dataframe 5 Rows---------------")

print(boston**.**head())

print("\n")

print("-----------Dataframe Columns List----------------")

print(boston**.**columns)

print("\n")

X **=** boston[['RM', 'LSTAT', 'PTRATIO']]

y **=** boston['MEDV']

print("-------------Splitting data into training and test sets-------------------")

*# Splitting the data into training and testing sets*

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.2, random\_state**=**42)

print("X\_train shape:", X\_train**.**shape)

print("X\_test shape:", X\_test**.**shape)

print("y\_train shape:", y\_train**.**shape)

print("y\_test shape:", y\_test**.**shape)

print("\n")

**for** i, feature **in** enumerate(X**.**columns):

plt**.**subplot(1, 3, i **+** 1)

plt**.**scatter(X[feature], y, marker**=**'o', s**=**5)

plt**.**title(feature)

plt**.**xlabel(feature)

plt**.**ylabel('MEDV')

plt**.**tight\_layout()

plt**.**show()

model **=** LinearRegression()

model**.**fit(X\_train, y\_train)

y\_pred **=** model**.**predict(X\_test)

print("-------------Visualization after fitting model---------------")

*# Visualization after fitting the model*

plt**.**figure(figsize**=**(8, 6))

plt**.**scatter(y\_test, y\_pred)

plt**.**plot([y\_test**.**min(), y\_test**.**max()], [y\_test**.**min(), y\_test**.**max()], 'k--', lw**=**2)

plt**.**xlabel('Actual MEDV')

plt**.**ylabel('Predicted MEDV')

plt**.**title('Actual MEDV vs Predicted MEDV')

plt**.**show()

print("\n")

mse **=** mean\_squared\_error(y\_test, y\_pred)

r2 **=** r2\_score(y\_test, y\_pred)

print("-----------Evaluation Result----------------")

print("Mean Squared Error:", mse)

print("R^2 Score:", r2)

print('\n')