import pandas as pd

df=pd.read\_csv("https://raw.githubusercontent.com/selva86/datasets/master/BostonHousing.csv")

df

df.head()

#input data

x = df.drop("medv", axis = 1)

#output data

y = df["medv"]

print("Shape of medv: ", x.shape)

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

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, random\_state = 0, test\_size = 0.25)

x\_train

x\_train.head()

print("shape of train data: ", x\_train.shape)

print("Shape of the test data: ", x\_test.shape)

from sklearn.linear\_model import LinearRegression

reg = LinearRegression()

reg.fit(x\_train, y\_train)

print("Regression coeff: \n", reg.coef\_)

print("\n Regression intercept: \n", reg.intercept\_)

# predictions

y\_pred = reg.predict(x\_test)

print("Shape of y\_pred: ", y\_pred.shape)

result = pd.DataFrame({"Actual": y\_test, "Produced": y\_pred})

print("Comparing Results: \n\n", result)

residual\_err = abs(y\_test - y\_pred)

print("Residual Error \n\n", residual\_err)

print("\n \n Mean Absolute Error: ", sum(residual\_err) / len(residual\_err))

from sklearn.metrics import mean\_absolute\_error

meanAbs = mean\_absolute\_error(y\_pred, y\_test)

print("\n\nMean Absolute Error from SkLearn: ", meanAbs)

from sklearn.metrics import mean\_absolute\_percentage\_error

print("Mean absolute percentage error: ", mean\_absolute\_percentage\_error(y\_test, y\_pred))

from sklearn.metrics import r2\_score

print("R2 Score from SK\_learn: ", r2\_score(y\_test, y\_pred))