**Practical Number 1**

**Aim:** Linear Regression using scikit-learn built-in dataset, Boston Housing Prices.

**Theory:**

 Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

**Code:**

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

#from sklearn.datasets import fetch\_california\_housing

#from sklearn.datasets import fetch\_california\_housing

from sklearn.linear\_model import LinearRegression

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

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

# Load the Boston Housing dataset

#boston = fetch\_california\_housing()

boston = pd.read\_csv('C:\\Users\\Sanket\\OneDrive\\Documents\\Desktop\\SEM 8\\ML & AI\\Practicals\\BostonHousing.csv')

# Split the data into training and testing sets

x= boston.drop(['medv'],axis=1)

y= boston['medv']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x,y, test\_size=0.2)

# Train the model

model = LinearRegression()

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

# Make predictions on the test set

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

# Evaluate the model using mean squared error and R-squared

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

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

# Highlight the more accurate predictions

highlight = abs(y\_pred - y\_test) < 5

# Plot the data and the model's predictions

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 5))

ax1.scatter(y\_test[highlight], y\_pred[highlight], color='red', marker='o', label="Accurate")

ax1.set\_xlabel("True Values")

ax1.set\_ylabel("Predictions")

ax1.legend()

ax2.scatter(y\_test, y\_pred, color='blue', marker='x', label="Inaccurate")

ax2.set\_xlabel("True Values")

ax2.set\_ylabel("Predictions")

ax2.legend()

plt.show()

# Print the evaluation metrics

print("Mean Squared Error:", mse)

print("R-Squared:", r2)

**Results:**

