Practical:4

Aim : Study and implement simple linear regression.

What is Simple Linear Regression?

Simple Linear Regression is a statistical method that models the relationship between a dependent variable (YYY) and a single independent variable (XXX) using a straight-line equation:

Y=b0+b1X+ϵY = b_0 + b_1X + \epsilonY=b0+b1X+ϵ

Where:

- YYY = Dependent variable (response)
- XXX = Independent variable (predictor)
- b0b_0b0 = Intercept (value of YYY when X=0X = 0X=0)
- b1b_1b1 = Slope (change in YYY for each unit increase in XXX)
- ϵ\epsilonϵ = Error term (accounts for randomness)

Steps to Implement Simple Linear Regression

1. **Collect Data**: Gather XXX (independent variable) and YYY (dependent variable) values.
2. **Estimate Coefficients**: Compute b0b_0b0 (intercept) and b1b_1b1 (slope) using the least squares method.
3. **Make Predictions**: Use the regression equation to predict YYY for given XXX.
4. **Visualize the Model**: Plot data points and the fitted regression line.
5. **Evaluate the Model**: Measure performance using metrics like R-squared.

Source code:

```
import numpy as np

import matplotlib.pyplot as plt

def estimate_coef(x, y):

    # Convert to NumPy arrays

    x = np.array(x)

    y = np.array(y)

    # number of data points

    n = len(x)

    # mean of x and y

    mean_x = np.mean(x)

    mean_y = np.mean(y)

    # calculate cross-deviation and deviation about x

    SS_xy = np.sum(x * y) - n * mean_x * mean_y

    SS_xx = np.sum(x * x) - n * mean_x * mean_x

    # calculate regression coefficients

    b_1 = SS_xy / SS_xx

    b_0 = mean_y - b_1 * mean_x

    return b_0, b_1

# User input for X and Y

x = np.array(list(map(float, input("Enter the independent variable values (X), separated by spaces: ").split())))
```

```
y = np.array(list(map(float, input("Enter the dependent variable values (Y), separated by spaces: ").split())))

# Estimate the coefficients

b_0, b_1 = estimate_coef(x, y)

# Print the results

print(f"Intercept (b_0): {b_0}")

print(f"Slope (b_1): {b_1}")

# Predict values (optional)

Y_pred = b_0 + b_1 * x

print(f"Predicted Y values: {Y_pred}")

# Plot the data points and the regression line

plt.scatter(x, y, color='blue', label='Data points')  # Plot the data points

plt.plot(x, Y_pred, color='red', label='Regression line')  # Plot the regression line

plt.xlabel('X')  # Label for the x-axis

plt.ylabel('Y')  # Label for the y-axis

plt.legend()  # Show legend

plt.title('Simple Linear Regression')  # Title of the plot

plt.grid(True)  # Optional: show grid lines for better readability

plt.show()  # Display the plot
```

output :

```
Enter the independent variable values (X), separated by spaces: 1 2 3 4
Enter the dependent variable values (Y), separated by spaces: 3 4 5 7
Intercept (b_0): 1.5
Slope (b_1): 1.3
Predicted Y values: [2.8 4.1 5.4 6.7]
```