

EXPERIMENT-8

CODE:-

```
# Linear Regression from Scratch (Single Program)
```

```
import numpy as np
```

```
# Function to train Linear Regression using Gradient Descent
```

```
def train_linear_regression(X, y, learning_rate=0.01, epochs=1000):
```

```
    m = len(y)
```

```
    w = 0 # weight
```

```
    b = 0 # bias
```

```
    for _ in range(epochs):
```

```
        # Predicted values
```

```
        y_pred = w * X + b
```

```
        # Compute gradients
```

```
        dw = (-2 / m) * np.sum(X * (y - y_pred))
```

```
        db = (-2 / m) * np.sum(y - y_pred)
```

```
        # Update parameters
```

```
        w = w - learning_rate * dw
```

```
        b = b - learning_rate * db
```

```
    return w, b
```

```
# Prediction function
```

```
def predict(X, w, b):
```

```
return w * X + b

# Sample dataset
X = np.array([1, 2, 3, 4, 5])
y = np.array([2, 4, 6, 8, 10])

# Train the model
w, b = train_linear_regression(X, y, learning_rate=0.01, epochs=1000)

# Make predictions
y_pred = predict(X, w, b)

# Output results
print("Weight (w):", w)
print("Bias (b):", b)
print("Predicted Values:", y_pred)
print("Actual Values: ", y)
```

OUTPUT:-

```
Weight (w): 1.9951803506719779
Bias (b): 0.017400463340610635
Predicted Values: [2.01258081 4.00776116 6.00294152 7.99812187 9.99330222]
Actual Values: [ 2 4 6 8 10]
```