

Experiment 02

Aim:

To implement **Supervised Learning** using the **Linear Regression** algorithm in Python with scikit-learn.

Theory:

Supervised Learning is a type of machine learning where the model is trained on labeled data, meaning the input comes with corresponding correct outputs. The model learns to map inputs to outputs using historical data and can then predict outcomes for unseen inputs.

Linear Regression is one of the simplest algorithms in supervised learning. It is used to predict the value of a dependent variable y based on the value of at least one independent variable X . The relationship between variables is assumed to be linear. The equation of a simple linear regression line is:

$$y = mx + c$$

Where:

- y is the predicted output,
- m is the slope (coefficient),
- x is the input feature,
- c is the intercept.

In Python, the `LinearRegression` class from the `sklearn.linear_model` module can be used to implement this algorithm easily.

Linear regression is widely used for:

- Predicting sales based on past data
- Analyzing trends
- Making forecasts
- Evaluating relationships between variables

It is a foundational technique used before diving into more complex machine learning models.

PROCEDURE:

1. Import necessary libraries (sklearn, matplotlib).
 2. Prepare training data: independent variable X and dependent variable y.
 3. Create an instance of the LinearRegression model.
 4. Fit the model to the training data using .fit().
 5. Predict the output using .predict().
 6. Visualize the actual data and the predicted regression line using matplotlib.
-

Code:

```
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# Example data
X = [[1], [2], [3], [4], [5]] # independent variable
y = [2, 4, 6, 8, 10]          # dependent variable

# Create and train the model
model = LinearRegression()
model.fit(X, y)

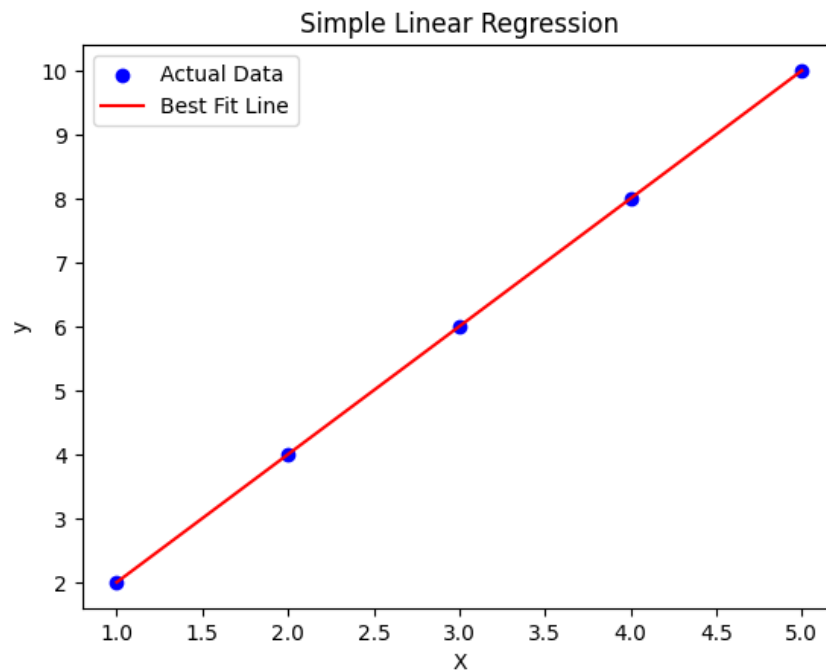
# Predict values for plotting the line
y_pred = model.predict(X)

# Plot original data points
plt.scatter(X, y, color='blue', label='Actual Data')

# Plot regression line
plt.plot(X, y_pred, color='red', label='Best Fit Line')

# Labels and title
plt.xlabel('X')
plt.ylabel('y')
plt.title('Simple Linear Regression')
plt.legend()

# Show plot
plt.show()
```

Output:**Conclusion:**

The implementation of the Linear Regression algorithm successfully demonstrates how supervised learning can be used to model and predict the relationship between variables. The plotted best-fit line accurately represents the pattern in the dataset. This experiment forms the basis for understanding more complex regression and prediction models in machine learning.