Supervised Learning-Regression

Multiple Linear Regression using sample dataset

Multiple Linear Regression is an extension of Simple Linear Regression where we model the relationship between a dependent variable (Y) and multiple independent variables (X1, X2, ..., Xn). The goal is to predict the value of the dependent variable based on several features.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \epsilon$$

Where:

- Y is the dependent variable (what you are trying to predict).
- X1,X2,...,Xn are the independent variables (predictors).
- β0 is the intercept.
- β1,β2,...,βn are the coefficients for the independent variables.
- ϵ is the error term (captures deviations from the predicted linear relationship).

Use Case: Predicting House Prices with Multiple Features

Consider a real-world example where we want to predict house prices (Y) based on multiple factors such as:

- Size of the house (X1)
- Number of bedrooms (X2)
- Age of the house (X3)

These factors influence the price of the house, and by using multiple linear regression, we can predict prices more accurately than using just one factor.

Sample Dataset

Let's assume the following dataset with three independent variables:

Size (sq.ft)	Bedrooms	Age (years)	Price (thousands of dollars)
750	2	10	150
800	3	8	160
850	3	5	180
900	4	2	200
1000	4	1	220

```python
# Step 1: Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

- `numpy`: Used for creating and handling arrays (e.g., the dataset for the house sizes, bedrooms, and age).
- `matplotlib.pyplot`: Used for plotting graphs to visualize data.
- `sklearn.linear_model.LinearRegression`: A machine learning model for performing linear regression (both simple and multiple regression).

```python
# Step 2: Define the dataset
# Independent variables (Size in sq.ft, Bedrooms, Age in years)
X = np.array([
    [750, 2, 10],
    [800, 3, 8],
    [850, 3, 5],
    [900, 4, 2],
    [1000, 4, 1]
])

# Dependent variable (Price in thousands of dollars)
Y = np.array([150, 160, 180, 200, 220])
```

- **Independent Variables (X)**:
  - Here, `X` represents the features or predictors. It is a 2D array, where each row contains three values:
    - Size of the house in square feet (1st column).
    - Number of bedrooms (2nd column).
    - Age of the house in years (3rd column).
  - Example: `[750, 2, 10]` represents a house with 750 sq.ft, 2 bedrooms, and 10 years old.
- **Dependent Variable (Y)**:
  - `Y` is the target variable, which represents the house prices (in thousands of dollars).
  - Example: 150 represents a house price of $150,000.

```
# Step 3: Create a multiple linear regression model and fit the data
model = LinearRegression()
model.fit(X, Y)
```

- `model = LinearRegression()`: Initializes the Linear Regression model from `sklearn`.
- `model.fit(X, Y)`: This is where the model learns the relationship between the independent variables (size, bedrooms, age) and the dependent variable (price). The model identifies the coefficients for each of the features in `X` and the intercept, forming the regression equation:

  Regression Equation:

  $$\text{Price} = (\text{coefficient}_1 \times \text{Size}) + (\text{coefficient}_2 \times \text{Bedrooms}) + (\text{coefficient}_3 \times \text{Age}) + \text{intercept}$$

```
# Step 4: Make predictions using the model on existing data
Y_pred = model.predict(X)
```

- `model.predict(X)`: The model uses the learned regression equation to predict prices based on the existing data in `X`.
- `Y_pred`: This array contains the predicted prices for the houses in the original dataset.

```
# Step 5: New data for prediction
new_house_data = np.array([
    [1100, 4, 1],  # Size: 1100 sq.ft, 4 bedrooms, 1 year old
    [1200, 5, 0],  # Size: 1200 sq.ft, 5 bedrooms, new house
    [950, 3, 5],   # Size: 950 sq.ft, 3 bedrooms, 5 years old
    [1050, 4, 2],  # Size: 1050 sq.ft, 4 bedrooms, 2 years old
    [1150, 3, 3],  # Size: 1150 sq.ft, 3 bedrooms, 3 years old
    [980, 2, 7]    # Size: 980 sq.ft, 2 bedrooms, 7 years old
])
```

- **New House Data**: This is a 2D array representing new houses with different sizes, bedrooms, and ages. We want to use the trained model to predict the prices for these new houses.

```
# Step 6: Predict prices for new house data
predicted_prices = model.predict(new_house_data)
```

- `model.predict(new_house_data)`: The model takes the new house data as input and uses the learned relationship from the training data to predict the prices for these new houses.
- `predicted_prices`: This array stores the predicted prices for the new house data.

```
# Step 7: Display the predicted prices for the new data
print("Predicted Prices for New Houses:")
for data, price in zip(new_house_data, predicted_prices):
    print(f"Size: {data[0]} sq.ft, Bedrooms: {data[1]}, Age: {data[2]} years -> Predicted Price: {price:.2f} thousands of dollars")
```

- **For loop**:
  - `zip(new_house_data, predicted_prices)` pairs each house data entry with its predicted price.

```python
# Step 8: (Optional) Visualizing the regression model with existing and new data

# Plot for existing data
plt.scatter(X[:, 0], Y, color="blue", label="Actual Data (Price vs Size)")
plt.plot(X[:, 0], Y_pred, color="red", label="Predicted Price (Existing Data)")

# Plot for new data
plt.scatter(new_house_data[:, 0], predicted_prices, color="green", marker="x", label="Predicted Price (New Houses)", s=100)

# Labels and title
plt.title("House Size vs Price (Including New Predictions)")
plt.xlabel("Size (sq.ft)")
plt.ylabel("Price (thousands of dollars)")
plt.legend()
plt.show()
```

1. **Visualization for New Data**:
   - We added another **scatter plot** for the **new house data** predictions using `plt.scatter()`.
   - The `new_house_data[:, 0]` represents the sizes of the new houses, and `predicted_prices` contains their predicted prices.
   - The marker style is changed to "x" and colored green to differentiate it from the original data.
2. **New Plot Customization**:
   - The green "x" markers represent the **predicted prices for new houses**.
   - The title and labels are updated to make the plot more informative.