Supervised Learning-Regression

Simple Linear Regression using sample dataset.

Simple Linear Regression is a statistical technique that models the relationship between two variables: a dependent variable (Y) and an independent variable (X). It assumes that the relationship between the variables is linear, which means that we can represent it with a straight line of the form:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Where:

- Y is the dependent variable (what you are trying to predict).
- X is the independent variable (the predictor).
- β0 is the intercept of the regression line (the value of Y when X = 0).
- β1 is the slope of the regression line (how much Y changes for a unit change in X).
- ϵ is the error term (captures any deviations from the exact linear relationship).

Methodology

1. **Hypothesis Setup**: Simple linear regression aims to find the line that best fits the data points in a two-dimensional plane.
2. **Data Collection**: Collect data where you have observations for both the dependent and independent variables.
3. **Fitting the Model**: The goal is to estimate the parameters β0 & β1 such that the sum of squared residuals (the difference between the actual value and the predicted value) is minimized.
4. **Model Evaluation**: After fitting the model, evaluate how well the model predicts by using metrics such as **R-squared** and the **mean squared error (MSE)**.
5. **Prediction**: Once the model is trained, we can use the estimated coefficients to predict new data.

Use Case: Predicting House Prices

Consider a real-world scenario where you want to predict house prices (Y) based on the size of the house (X). This is a simple linear regression problem because house price can be predicted using one feature: house size.

Dataset

For simplicity, let's use a sample dataset of house sizes (in square feet) and house prices (in thousands of dollars):

Size (sq.ft)	Price (thousands of dollars)
750	150
800	160
850	180
900	200
1000	220

- **Import Libraries**: The necessary libraries to be imported (numpy, matplotlib, and sklearn).
- **Define the Dataset**: The initial dataset of house sizes and prices
- **Model Creation**: A LinearRegression object is created, and the model is trained on the existing dataset using the .fit() method.
- **New Data for Prediction**: A new dataset (new_house_sizes) is created with house sizes of 1100, 1200, and 1300 square feet. This data will be used to make new price predictions using the .predict() method.
- **Plotting**: The plot includes the original data points, the regression line, and the predicted prices for the new house sizes (displayed in green).

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

# Step 2: Define the dataset
# House size (independent variable X)
X = np.array([750, 800, 850, 900, 1000]).reshape(-1, 1)

# House prices (dependent variable Y)
Y = np.array([150, 160, 180, 200, 220])
```

- `numpy` (**np**): Used for handling arrays and reshaping data.
- `matplotlib.pyplot` (**plt**): Used for plotting graphs, to visualize the relationship between house size and price.
- `LinearRegression`: The regression model from the **scikit-learn** library.

- `X`: This array contains the sizes of houses in square feet. The `reshape(-1, 1)` converts the 1D array into a 2D array as required by the regression model. Each row represents a different house size.

- `Y`: This array contains the corresponding prices (in thousands of dollars) for the house sizes in `X`.

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

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

- `LinearRegression()` : Initializes a linear regression model.
- `model.fit(X, Y)` : Fits (or trains) the model using the data provided (`X` as input and `Y` as output). The model learns the relationship between house size and price, finding the best-fitting line.

- `model.predict(X)` : Predicts the house prices for the given house sizes (`X`) using the trained model.
- `Y_pred` : Stores the predicted prices for the house sizes in `X` (original data). These are the values lying on the regression line.

```
# Step 5: New data for prediction
new_house_sizes = np.array([1100, 1200, 1300]).reshape(-1, 1)  # New house sizes (in sq.ft)

# Step 6: Predict prices for new house sizes
predicted_prices = model.predict(new_house_sizes)

# Step 7: Display results
print("Predicted Prices for New House Sizes:")
for size, price in zip(new_house_sizes, predicted_prices):
    print(f"Size: {size[0]} sq.ft -> Predicted Price: {price:.2f} thousands of dollars")
```

- `new_house_sizes` : This array contains the new house sizes (1100 sq.ft, 1200 sq.ft, and 1300 sq.ft) for which we want to predict the prices. Again, it's reshaped into a 2D array for input into the regression model.

- `model.predict(new_house_sizes)` : Uses the trained model to predict the prices of houses based on the new house sizes provided.
- `predicted_prices` : Stores the predicted prices for the new house sizes.

- `for size, price in zip(new_house_sizes, predicted_prices)` : Loops over the new house sizes and their corresponding predicted prices.
- `print()` : Displays the predicted prices for each of the new house sizes, formatted to two decimal places.

```
# Step 8: Plotting the data and regression line
plt.scatter(X, Y, color="blue", label="Actual Data")
plt.plot(X, Y_pred, color="red", label="Regression Line")
plt.scatter(new_house_sizes, predicted_prices, color="green", label="Predicted Prices for New Sizes")
plt.title("House Size vs Price")
plt.xlabel("Size (sq.ft)")
plt.ylabel("Price (thousands of dollars)")
plt.legend()
plt.show()
```

- `plt.scatter(X, Y, color="blue")` : Plots the original data points (house sizes and actual prices) in blue.
- `plt.plot(X, Y_pred, color="red")` : Plots the regression line, which represents the predicted prices based on house size. It shows the best-fit linear relationship between size and price for the original data.
- `plt.scatter(new_house_sizes, predicted_prices, color="green")` : Plots the predicted prices for the new house sizes in green.
- `plt.title(), plt.xlabel(), plt.ylabel()` : Add a title and labels to the graph for clarity.
- `plt.legend()` : Adds a legend to distinguish between the actual data, the regression line, and the new predictions.
- `plt.show()` : Displays the plot.

```
Predicted Prices for New House Sizes:
Size: 1100 sq.ft -> Predicted Price: 252.70 thousands of dollars
Size: 1200 sq.ft -> Predicted Price: 282.16 thousands of dollars
Size: 1300 sq.ft -> Predicted Price: 311.62 thousands of dollars
```