Regression

It is an estimation of one independent variable in terms of the other.

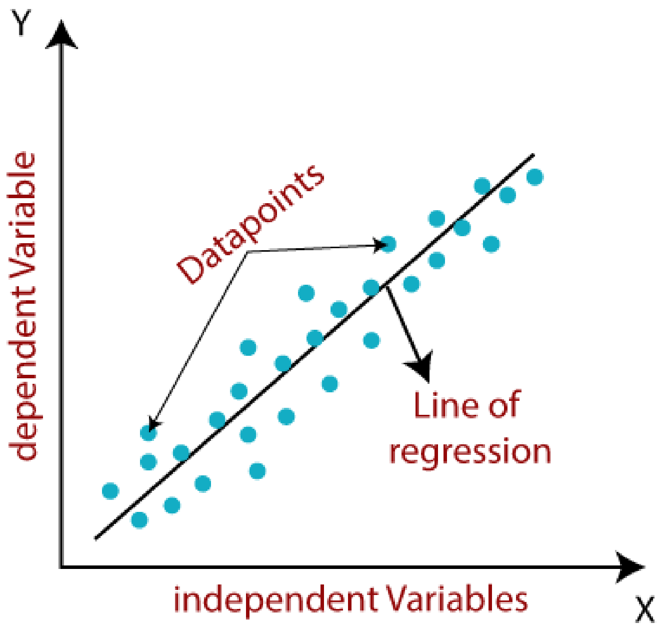
The best fitting straight line of the form $y = a + bx$ is called the regression of y on x and $x = a + by$ is called the regression of x on y.

Linear Regression

Linear regression is a supervised algorithm that draws a linear relationship model between the independent variables (inputs, sometimes it is called regressors or predictors) and the dependent variable (output or outcome). In simple words, the idea of Linear regression is to find the line that fits the observed data and use it to predict the future outcome from new inputs or unseen data. The observed data can be a historical data that has been collected for a specific purpose within a certain duration.

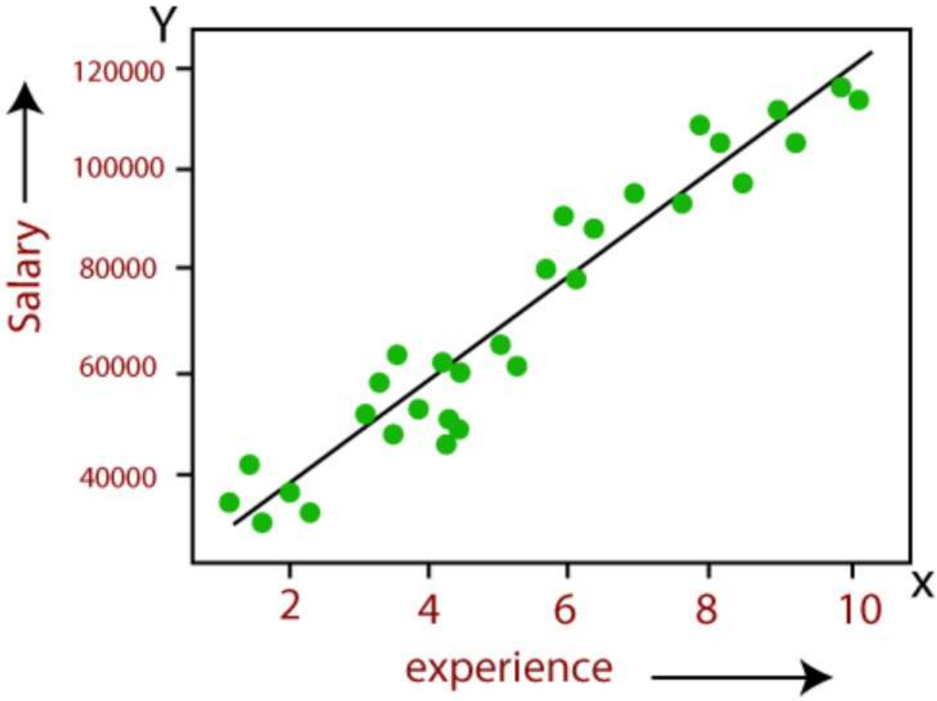
Linear regression is widely used in financial and business analysis fields including to predict stocks and commodity prices, sales forecasting, employee performance analysis, etc.

The simple Linear regression consists of only one independent variable (x) and one dependent variable (y) as illustrated in the following figure.

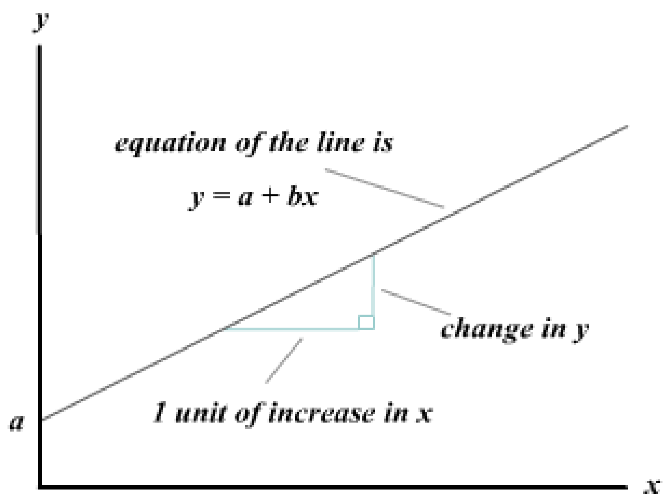


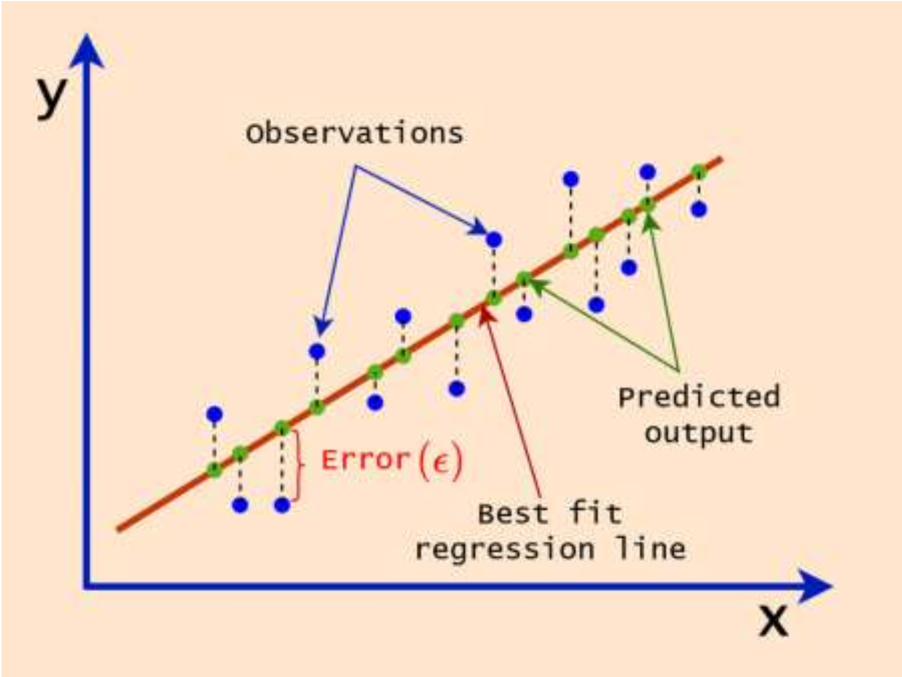
Example

The following figure represents a simple linear regression model where salary is modelled using experience.



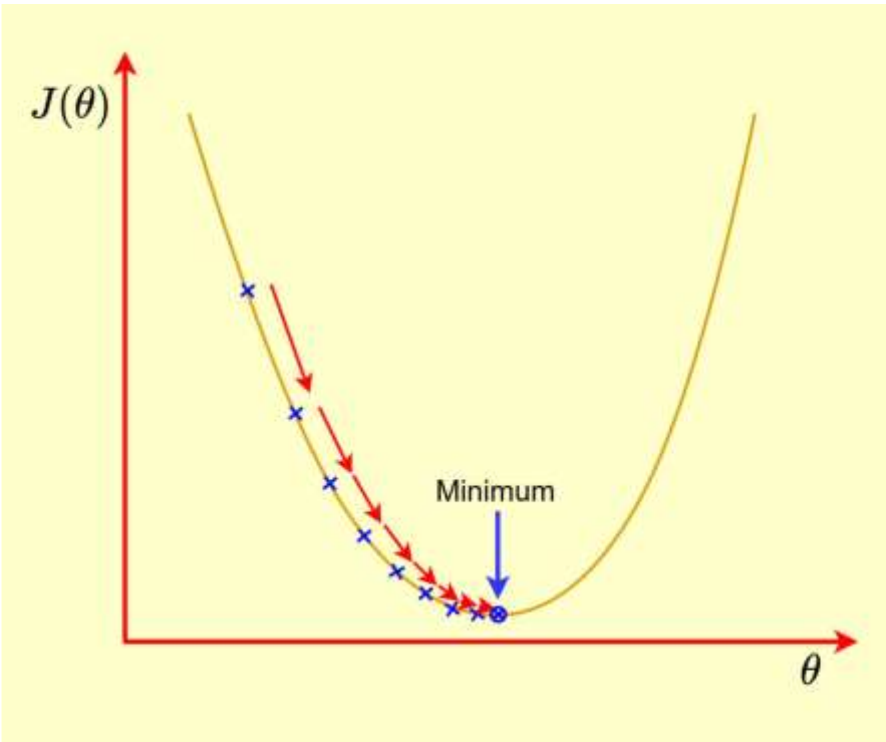
What we actually do in Linear regression is basically to find the best parameters that fits the straight line to the observations by minimizing the errors between the predicted outcomes and the observations. Meaning that, the best model in Linear regression is the linear equation model with the best parameters.





Gradient Descent Method

Gradient Descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In machine learning, we use gradient descent to update the parameters of our model.



Working Procedure to Fit a Linear Regression Line to Data Using the Gradient Descent Method

(1) The following steps are used to fit a linear regression line $\hat{y} = a + bx$ for the given data:

Step 1: Define the Model and Cost Function

The linear regression model you want to fit is given by:

$$\hat{y} = a + bx$$

where $\hat{y}$ is the predicted value, *a* and *b* are the parameters (weights) of the model and *x* is the input feature.

The cost function (error function) for linear regression is the Mean Squared Error (MSE):

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n} (y_i - \hat{y})^2$$

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n} (y_i - (a + bx_i))^2$$

Where *n* is the number of data points.

Step 2: Initialize Weights and Hyperparameters

Initialize the weights *a* and *b* to some arbitrary values. Start with *a* = 0 and *b* = 0. Also need to set hyperparameters:

- Learning rate (α): A small positive value that controls the step size in each iteration.
- Number of iterations: The number of times we update the weights.

Step 3: Gradient Descent

For each iteration, calculate the gradients of the cost function with respect to the weights (*a* and *b*) and update the weights accordingly.

The gradients are given by:

$$\Delta a = -\frac{2}{n}\sum_{i=1}^{n}\left(y_i - (a + bx_i)\right)$$

$$\Delta b = -\frac{2}{n}\sum_{i=1}^{n}x_i \times \left(y_i - (a + bx_i)\right)$$

Update the weights using the gradients:

$$a_{new} = a - \alpha \times \Delta a$$

$$b_{new} = b - \alpha \times \Delta b$$

Repeat this process for the specified number of iterations.

Step 4: Predict

After training, use the final values of *a* and *b* to make predictions:

$$\hat{y} = a + bx$$

Step 5: Evaluate and visualize

Evaluate the quality of the linear regression model by calculating the final MSE on your training data:

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n}(y_i - \hat{y})^2$$

Also, visualize the linear regression line by plotting it alongside the data points.

Step 6: Iterate as Needed

We need to adjust the learning rate and the number of iterations to find the best-fitting line. If the cost is not converging or fluctuating, you may need to modify the hyperparameters.

This process allows you to iteratively update the weights to minimize the cost function, resulting in a linear regression line that best fits the given data.

Examples

1. We have recorded the weekly average price of a stock over 6 consecutive days. Y shows the weekly average price of the stock and x shows the number of the days. Try to fit the best possible function ' f ' to establish the relationship between the number of the day and conversion rate. (Applying Gradient descent) where f(x) = y = a + b * x.

X	1	2	3	4	5	6
Y	10	14	18	22	25	33

The initial values of a & b are a = 4.9 & b = 4.401. The learning rate is mentioned as .05. The error rate of a & b should be less than .01. Plot the predicted and actual data in a graph.

Solution:

Given data:

X = x_i = 1, 2, 3, 4, 5, 6

Y = y_i = 10, 14, 18, 22, 25, 33

n = 6

Initialization: a = 4. 9 and b = 4.401

Learning rate α = 0.05

Maximum allowable error for a and b = 0.01

The goal is to minimize the MSE (mean squared error) defined as

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n} (y_i - (a + bx_i))^2$$

Iteration - 1

Compute the gradients of a and b with respect to the MSE

$$\Delta a = -\frac{2}{n} \sum_{i=1}^{n} \left(y_i - (a + bx_i)\right)$$

$$\Delta b = -\frac{2}{n} \sum_{i=1}^{n} x_i \times \left(y_i - (a + bx_i)\right)$$

x_i	y_i	a	b	$a + bx_i$	$\left(y_i - (a + bx_i)\right)$	$x_i\left(y_i - (a + bx_i)\right)$
1	10	4.9	4.401	9.301	0.699	0.699
2	14	4.9	4.401	13.702	0.298	0.596
3	18	4.9	4.401	18.103	–0.103	–0.309
4	22	4.9	4.401	22.504	–0.504	–2.016
5	25	4.9	4.401	26.905	–1.905	–9.525
6	33	4.9	4.401	31.306	1.694	10.164
				Total	0.179	–0.391

$$\Delta a = -\frac{2}{6}\{0.179\} = -0.0597$$

$$\Delta b = -\frac{2}{6}\{-0.391\} = 0.1303$$

Update a and b using the gradients and the learning rate

$$a = a - \alpha \times \Delta a = 4.9 - 0.05 \times -0.0597 = 4.903$$

$$b = b - \alpha \times \Delta b = 4.401 - 0.05 \times 0.1303 = 4.394$$

Iteration – 2

Compute the gradients of a and b with respect to the MSE

$$\Delta a = -\frac{2}{n} \sum_{i=1}^{n} \left(y_i - (a + bx_i)\right)$$

$$\Delta b = -\frac{2}{n}\sum_{i=1}^{n} x_i \times \left(y_i - (a + bx_i)\right)$$

x_i	y_i	a	b	$a + bx_i$	$\left(y_i - (a + bx_i)\right)$	$x_i\left(y_i - (a + bx_i)\right)$
1	10	4.903	4.394	9.297	0.703	0.703
2	14	4.903	4.394	13.691	0.309	0.618
3	18	4.903	4.394	18.085	–0.085	–0.255
4	22	4.903	4.394	22.479	–0.479	–1.916
5	25	4.903	4.394	26.873	–1.873	–9.365
6	33	4.903	4.394	31.267	1.733	10.398
				Total	0.308	0.183

$$\Delta a = -\frac{2}{6}\{0.308\} = -0.103$$

$$\Delta b = -\frac{2}{6}\{0.183\} = -0.061$$

Update *a* and *b* using the gradients and the learning rate

$$a = a - \alpha \times \Delta a = 4.903 - 0.05 \times -0.103 = 4.908$$

$$b = b - \alpha \times \Delta b = 4.394 - 0.05 \times -0.061 = 4.397$$

Iteration – 3

Compute the gradients of *a* and *b* with respect to the MSE

$$\Delta a = -\frac{2}{n}\sum_{i=1}^{n}\left(y_i - (a + bx_i)\right)$$

$$\Delta b = -\frac{2}{n}\sum_{i=1}^{n} x_i \times \left(y_i - (a + bx_i)\right)$$

x_i	y_i	a	b	$a + bx_i$	$(y_i - (a + bx_i))$	$x_i(y_i - (a + bx_i))$
1	10	4.908	4.397	9.305	0.695	0.695
2	14	4.908	4.397	13.702	0.298	0.596
3	18	4.908	4.397	18.099	–0.099	–0.297
4	22	4.908	4.397	22.496	–0.496	–1.984
5	25	4.908	4.397	26.893	–1.893	–9.465
6	33	4.908	4.397	31.29	1.71	10.26
				Total	0.215	–0.195

$$\Delta a = -\frac{2}{6}\{0.215\} = -0.072$$

$$\Delta b = -\frac{2}{6}\{-0.195\} = 0.065$$

Update *a* and *b* using the gradients and the learning rate

$$a = a - \alpha \times \Delta a = 4.908 - 0.05 \times -0.072 = 4.91$$

$$b = b - \alpha \times \Delta b = 4.397 - 0.05 \times 0.065 = 4.39$$

These are the final values for a and b.

The best fit of the line is $\hat{y} = 4.91 + 4.39\ x$

Prediction:

If x = 1, then $\hat{y} = 4.91 + 4.39 \times 1 = 9.3$

If x = 2, then $\hat{y} = 4.91 + 4.39 \times 2 = 13.69$

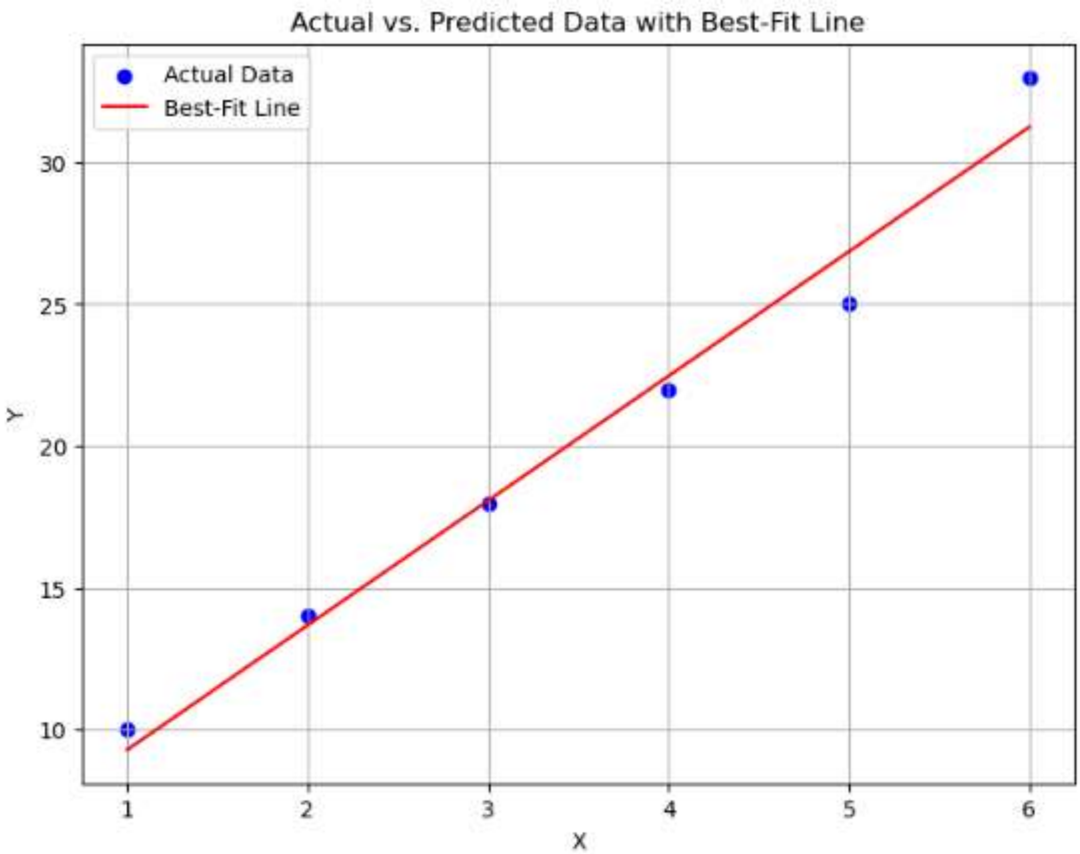
If x = 3, then $\hat{y} = 4.91 + 4.39 \times 3 = 18.08$

If x = 4, then $\hat{y} = 4.91 + 4.39 \times 4 = 22.47$

If x = 5, then $\hat{y} = 4.91 + 4.39 \times 5 = 26.86$

If x = 6, then $\hat{y} = 4.91 + 4.39 \times 6 = 31.25$

If x = 7, then $\hat{y} = 4.91 + 4.39 \times 7 = 35.64$



---

(2) The following steps are used to fit a linear regression line $\hat{y} = ax + b$ for the given data:

Step 1: Define the Model and Cost Function

The linear regression model you want to fit is given by:

$$\hat{y} = ax + b$$

where $\hat{y}$ is the predicted value, *a* and *b* are the parameters (weights) of the model and *x* is the input feature.

The cost function (error function) for linear regression is the Mean Squared Error (MSE):

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n} (y_i - \hat{y})^2$$

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n} (y_i - (ax_i + b))^2$$

Where *n* is the number of data points.

Step 2: Initialize Weights and Hyperparameters

Initialize the weights *a* and *b* to some arbitrary values. Start with *a* = 0 and *b* = 0. Also need to set hyperparameters:

- Learning rate (α): A small positive value that controls the step size in each iteration.
- Number of iterations: The number of times you'll update the weights.

Step 3: Gradient Descent

For each iteration, calculate the gradients of the cost function with respect to the weights (*a* and *b*) and update the weights accordingly.

The gradients are given by:

$$\Delta a \;=\; -\frac{2}{n}\sum_{i=1}^{n} x_i \times \left(y_i - (ax_i + b\,)\right)$$

$$\Delta b \;=\; -\frac{2}{n}\sum_{i=1}^{n} \left(y_i - (ax_i + b\,)\right)$$

Update the weights using the gradients:

$$a \;=\; a \;-\; \alpha \;\times\; \Delta a$$

$$b \;=\; b \;-\; \alpha \;\times\; \Delta b$$

Repeat this process for the specified number of iterations.

Step 4: Predict

After training, use the final values of *a* and *b* to make predictions:

$$\hat{y} \;=\; ax \;+\; b$$

Step 5: Evaluate and visualize

Evaluate the quality of the linear regression model by calculating the final MSE on your training data:

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n} (y_i - \hat{y})^2$$

Also, visualize the linear regression line by plotting it alongside the data points.

Step 6: Iterate as Needed

We need to adjust the learning rate and the number of iterations to find the best-fitting line. If the cost is not converging or fluctuating, you may need to modify the hyperparameters.

This process allows you to iteratively update the weights to minimize the cost function, resulting in a linear regression line that best fits the given data.

2. Consider the data given below and fit a linear regression line y = ax + b using gradient descent method.

X	0	0.4	0.6	1
Y	0	1	0.48	0.95

Initialize the weights a and b to 0.8 and 0.2 respectively. Update the weights such that the error is minimum using gradient descent. Plot the linear regression line after updating the values of a and b in two iterations.

Solution

Given data:

X = x_i = [0, 0.4, 0.6, 1]

Y = y_i = [0, 1, 0.48, 0.95]

Initialize the weights: a = 0.8, b = 0.2

Learning rate (α) is a hyperparameter that controls the step size in gradient descent.

Set it to a small value, e.g., α = 0.01.

The goal is to minimize the MSE (mean squared error) defined as

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n} (y_i - (ax_i + b))^2$$

Iteration - 1

Compute the gradients of a and b with respect to the MSE

$$\Delta a = -\frac{2}{n} \sum_{i=1}^{n} x_i \times \left(y_i - (ax_i + b)\right)$$

$$\Delta b = -\frac{2}{n} \sum_{i=1}^{n} \left(y_i - (ax_i + b)\right)$$

x_i	y_i	a	b	$ax_i + b$	$\left(y_i - (ax_i + b)\right)$	$x_i\left(y_i - (ax_i + b)\right)$
0	0	0.8	0.2	0.2	–0.2	0
0.4	1	0.8	0.2	0.52	0.48	0.192
0.6	0.48	0.8	0.2	0.68	–0.2	–0.12
1	0.95	0.8	0.2	1	–0.05	–0.05
				Total	0.03	0.022

$$\Delta a = -\frac{2}{4} \times \{0.022\} = -0.011$$

$$\Delta b = -\frac{2}{4} \times \{0.03\} = -0.015$$

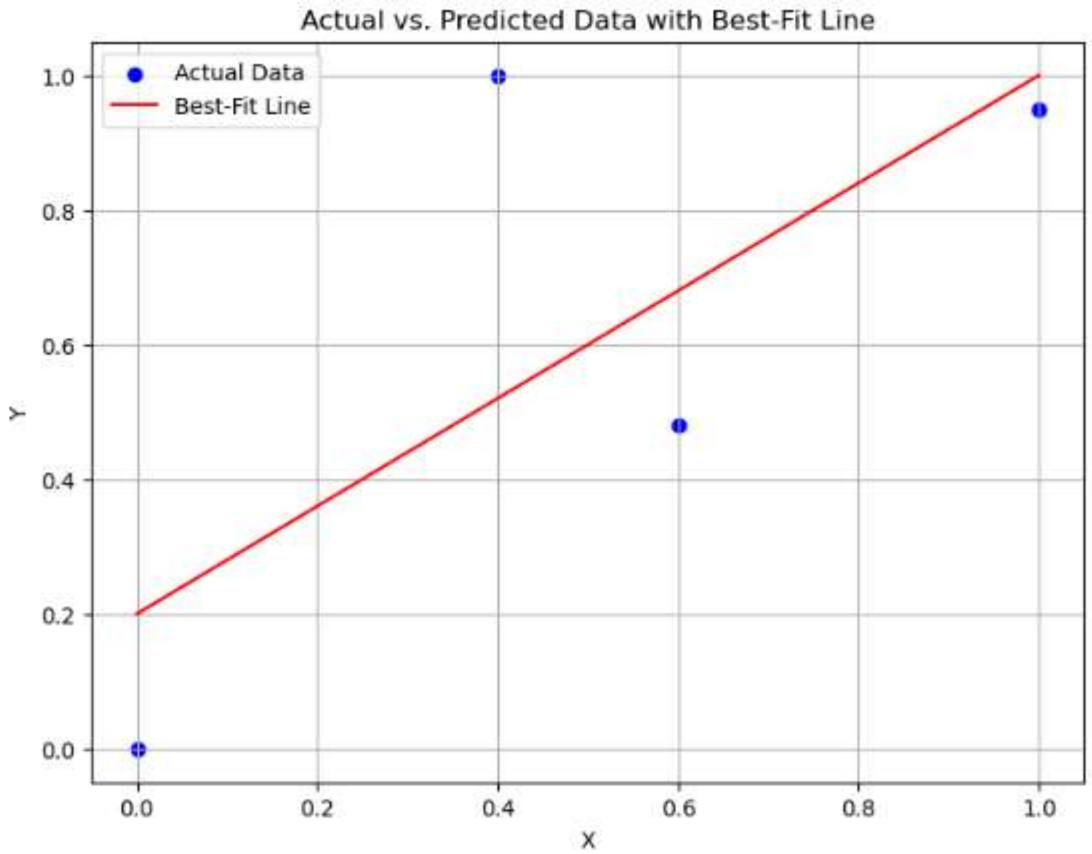
Update a and b using the gradients and the learning rate

$$a = a - \alpha \times \Delta a = 0.8 - 0.01 \times -0.011 = 0.8$$

$$b = b - \alpha \times \Delta b = 0.2 - 0.01 \times -0.015 = 0.2$$

These are the final values for a and b.

The best fit of the line is $\hat{y} = 0.8\, x + 0.2$



Actual vs. Predicted Data with Best-Fit Line
Actual Data
Best-Fit Line
1.0
0.8
0.6
0.4
0.2
0.0
Y
0.0
0.2
0.4
0.6
0.8
1.0
X