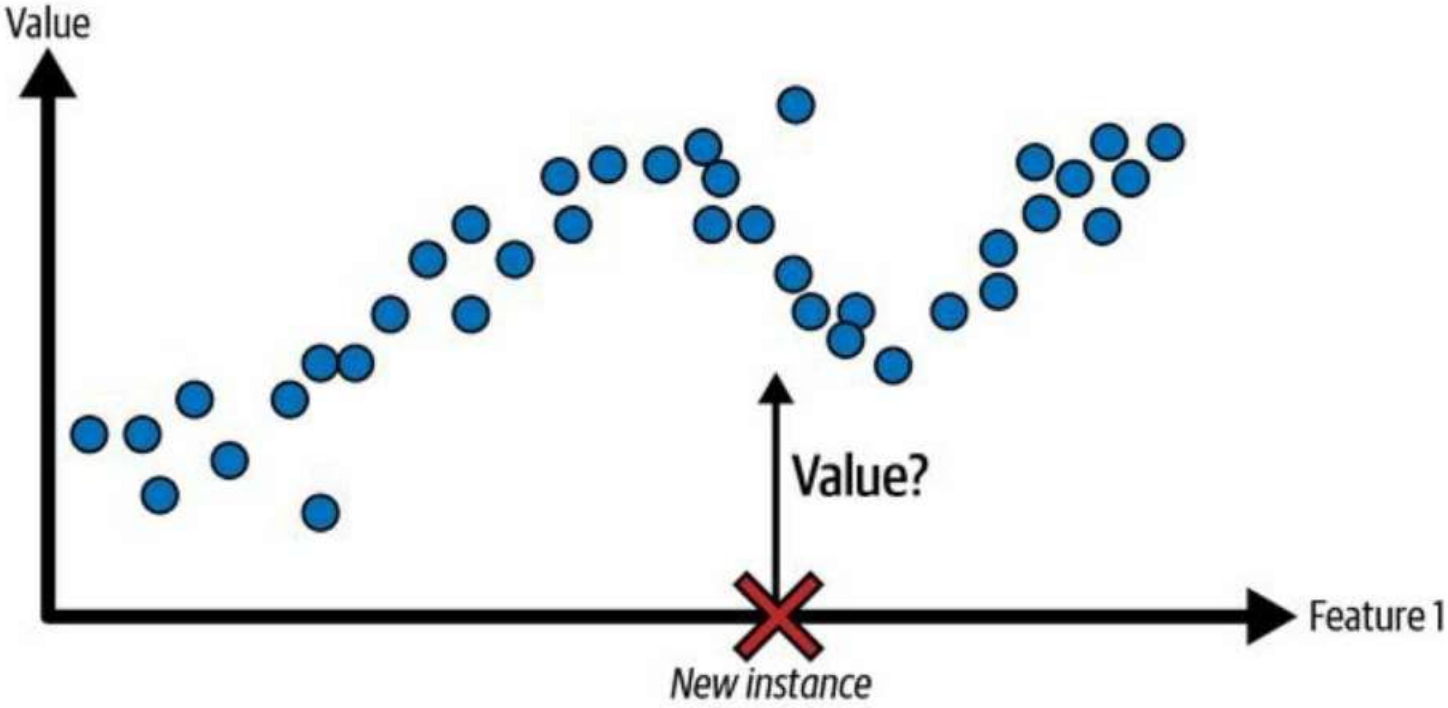
Supervised Learning: Regression

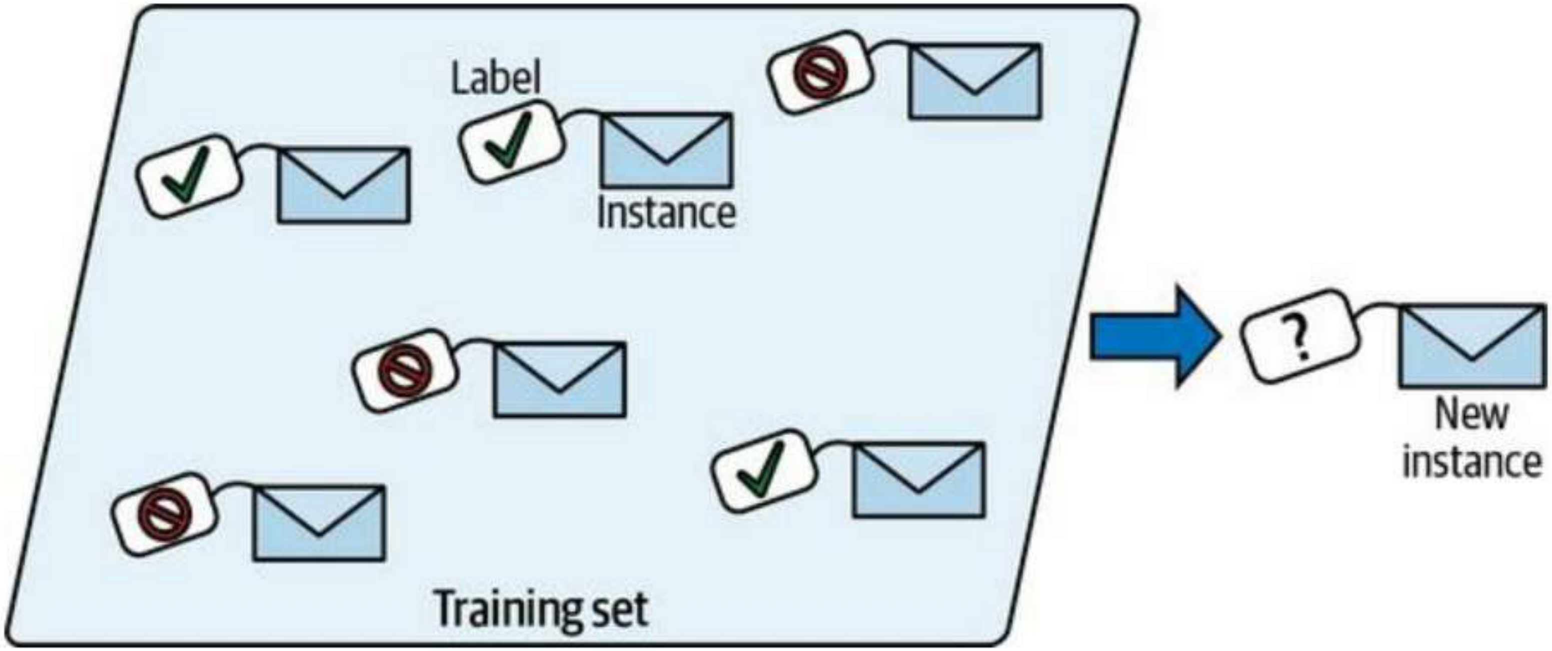
A typical task is to predict a target numeric value, such as the price of a car, given a set of features (mileage, age, brand, etc.). This sort of task is called **regression**.



A regression problem: predict a value, given an input feature (there are usually multiple input features, and sometimes multiple output values)

Supervised Learning

Another typical supervised learning task is **classification**. The spam filter is a good example of this: it is trained with many example emails along with their class (spam or ham), and it must learn how to classify new emails.

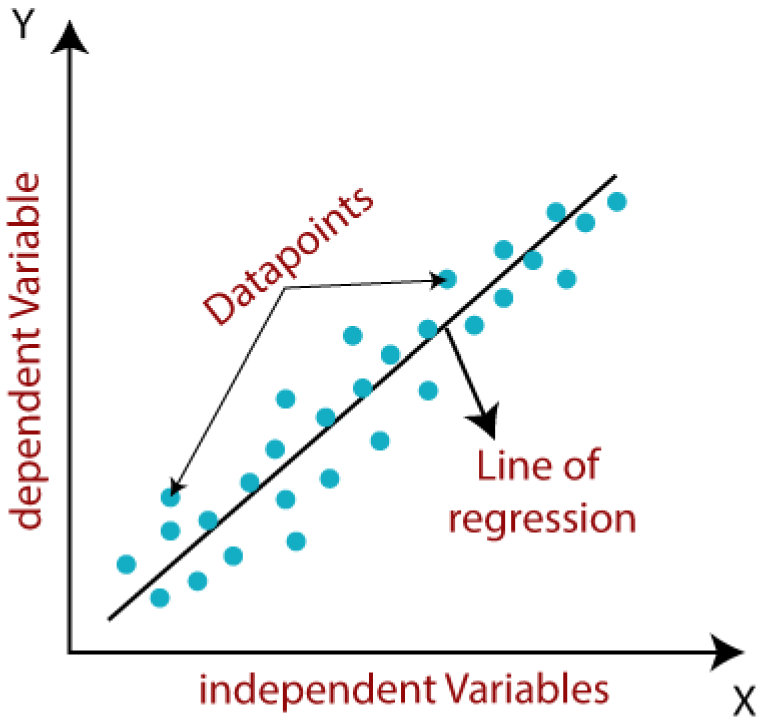


Regression

- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Linear Regression
- Logistic Regression

Linear Regression

Linear regression is a **supervised** machine learning algorithm that models the linear relationship between a **dependent** variable and one or more **independent** variables by fitting a straight line to the observed data.



https://www.javatpoint.com/linear-regression-in-machine-learning

Linear Regression

Linear regression is a **supervised** machine learning algorithm that models the linear relationship between a **dependent** variable and one or more **independent** variables by fitting a straight line to the observed data.

When there is a single independent variable, it is referred to as S**imple Linear Regression**, whereas with multiple independent variables, it is called M**ultiple Linear Regression**.

Polynomial Linear Regression is a form of regression where the relationship between the independent variable(s) and the dependent variable is modeled as a **polynomial**, but it is still considered a type of linear regression because the model is linear in terms of the coefficients.

Simple Linear Regression

This is the most basic form of linear regression, involving a single independent variable and a single dependent variable. The equation for simple linear regression is: $\boldsymbol{y = b_0 + b_1 x}$

where

- y is the dependent variable
- x is the independent variable
- $\boldsymbol{b_0}$ is the intercept
- $\boldsymbol{b_1}$ is the slope

Example

Suppose we are analyzing the relationship between the number of hours a student studies (x) and their exam score (y) using a simple linear regression. After running the regression, we get the following equation: y=5x+50

In this equation:

$y$is the predicted exam score,

$x$is the number of hours studied,

The slope m = 5

m=5 indicates that for each additional hour of studying, the exam score is expected to increase by 5 points.

The intercept b= 50

b=50 suggests that if a student studies 0 hours, the predicted exam score would be 50 points.

Example calculation using the slope:

If a student studies 4 hours (x= 4),

the predicted score is: y=5(4)+50=20+50=70

Thus, according to this regression model, studying for an extra hour would raise a student’s expected exam score by 5 points.

Multiple Linear Regression

This involves multiple independent variables and a single dependent variable. The equation for simple linear regression is:

$$\boldsymbol{y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \ldots \ldots + b_n\, x_n}$$

where

- y is the dependent variable
- $\boldsymbol{x_1, x_2, \ldots, x_n}$ are the independent variables
- $\boldsymbol{b_0}$ is the intercept
- $\boldsymbol{b_1, b_2, \ldots, b_n}$ are the slopes

EXample

Scenario

Imagine we want to predict a person’s monthly salary (y) based on their years of education (x1) and years of work experience (x2). The multiple regression model would look like this:

y=b0+b1x1+b2x2

Where:

Y is the monthly salary (dependent variable),

X_1 is the number of years of education,

x_2 is the number of years of work experience,

b_0 is the intercept (baseline salary if x1 and x2 are zero),

b_1 and b_2 are the coefficients showing how much salary changes with each additional year of education and experience, respectively.

EXample

Example Model Output

Let’s say we use data to fit this model, and the result is:

$y=2000+1500x_1+800x_2$

Interpretation:

Intercept (2000): A baseline salary of $2,000 for someone with zero years of education and experience.

Coefficient for x_1 (1500): Each additional year of education increases the predicted salary by $1,500.

Coefficient for x_2 (800): Each additional year of work experience increases the predicted salary by $800.

EXample

Using the Model

If a person has 16 years of education and 5 years of work experience, we can predict their salary as follows:

$y=2000+1500(16)+800(5)=2000+24000+4000=30{,}000$

So, based on this model, their predicted monthly salary would be $30,000

Linear Regression

The objective of the algorithm is to determine the **best-fit line equation** that can predict values based on the independent variables.

In regression, a dataset with **X** and **Y** values is provided, and these values are used to **train** a function. Once trained, this function can be applied to predict **Y** for an **unknown X**.

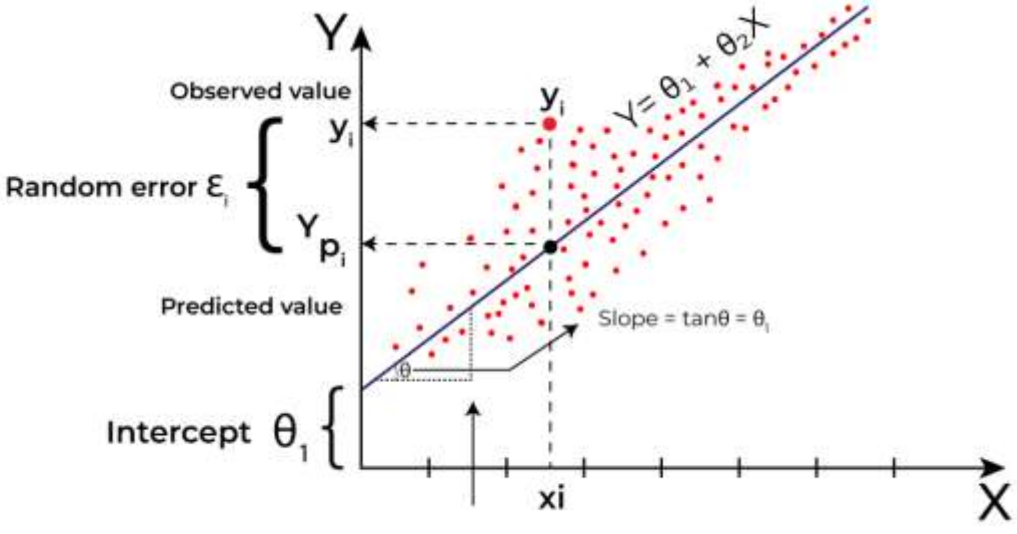
In regression, the goal is to estimate the value of **Y**, meaning a function is needed that predicts the continuous **Y** value given **X** as the independent variable(s).

Linear Regression

What is the best Fit Line?

The main goal of using linear regression is to find the best-fit line, which **minimizes the error** between the predicted and actual values. The best-fit line will have the least amount of error.

The equation of the best-fit line represents the relationship between the dependent and independent variables, with the slope indicating how much the dependent variable changes in response to a unit change in the independent variable(s).


Y
Observed value
y_i
Random error ε_i
Y_{p_i}
Predicted value
Intercept θ_1
y_i
$Y = \theta_1 + \theta_2 X$
Slope = tanθ = θ_i
θ
xi
X

https://www.geeksforgeeks.org/ml-linear-regression/

Linear Regression

Error and Cost Function

Let $\boldsymbol{\hat{y}_i = \theta_1 + \theta_2 x_i}$ be the prediction for input $\boldsymbol{x_i}$
And $\boldsymbol{y_i}$ be the correct value for input $\boldsymbol{x_i}$
So, the error would be $\boldsymbol{\hat{y}_i - y_i}$

The objective the best line would be to minimize the Mean Squared Error (MSE) cost function,

$$\mathbf{J} = \boldsymbol{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - \ y_i)^2}$$

Linear Regression

Gradient Descent for Linear Regression

A linear regression model can be **trained** using the **gradient descent** optimization algorithm, which iteratively adjusts the model's parameters to minimize the mean squared error (MSE) on the training dataset.

To update the values of $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$ and reduce the cost function (by minimizing the RMSE), the model applies Gradient Descent.

The process begins with random values for $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$, and these values are progressively updated to achieve the lowest possible cost, ultimately resulting in the best-fit line.

Linear Regression

Gradient Descent for Linear Regression

A gradient is essentially a derivative that describes how slight changes in the inputs of a function affect its outputs.

If we differentiate the cost function **J** with respect to $\boldsymbol{\theta_1}$

$$J'_{\boldsymbol{\theta_1}} = \frac{\partial}{\partial \boldsymbol{\theta_1}} \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{\widehat{y}_i} - \ \boldsymbol{y_i})^{\mathbf{2}}$$

$$= \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \mathbf{2}(\boldsymbol{\widehat{y}_i} - \ \boldsymbol{y_i}) \frac{\partial}{\partial \boldsymbol{\theta_1}} (\boldsymbol{\widehat{y}_i} - \ \boldsymbol{y_i})$$

$$= \frac{\mathbf{1}}{\boldsymbol{n}} \textstyle\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \mathbf{2}(\boldsymbol{\widehat{y}_i} - \ \boldsymbol{y_i}) \frac{\partial}{\partial \boldsymbol{\theta_1}} (\boldsymbol{\theta_1} + \boldsymbol{\theta_2} \boldsymbol{x_i} - \ \boldsymbol{y_i})$$

$$= \frac{\mathbf{1}}{\boldsymbol{n}} \textstyle\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \mathbf{2}(\boldsymbol{\widehat{y}_i} - \ \boldsymbol{y_i}) \ (\mathbf{1} + \mathbf{0} - \mathbf{0})$$

$$= \frac{\mathbf{2}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{\widehat{y}_i} - \ \boldsymbol{y_i})$$

Linear Regression

Gradient Descent for Linear Regression

If we differentiate the cost function **J** with respect to $\boldsymbol{\theta_2}$

$$J'_{\boldsymbol{\theta_2}} = \frac{\partial}{\partial \boldsymbol{\theta_2}} \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} (\boldsymbol{\hat{y}_i} - \boldsymbol{y_i})^{\mathbf{2}}$$

$$= \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} \mathbf{2}(\boldsymbol{\hat{y}_i} - \boldsymbol{y_i}) \frac{\partial}{\partial \boldsymbol{\theta_2}} (\boldsymbol{\hat{y}_i} - \boldsymbol{y_i})$$

$$= \frac{\mathbf{1}}{\boldsymbol{n}} \textstyle\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} \mathbf{2}(\boldsymbol{\hat{y}_i} - \boldsymbol{y_i}) \frac{\partial}{\partial \boldsymbol{\theta_2}} (\boldsymbol{\theta_1} + \boldsymbol{\theta_2} \boldsymbol{x_i} - \boldsymbol{y_i})$$

$$= \frac{\mathbf{1}}{\boldsymbol{n}} \textstyle\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} \mathbf{2}(\boldsymbol{\hat{y}_i} - \boldsymbol{y_i})\ (\mathbf{0} + \boldsymbol{x_i} - \mathbf{0})$$

$$= \frac{\mathbf{2}}{\boldsymbol{n}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} (\boldsymbol{\hat{y}_i} - \boldsymbol{y_i})\, \boldsymbol{x_i}$$

Linear Regression

Update θ$_1$ and θ$_2$ values in order to reduce the Cost function

The goal of linear regression is to determine the coefficients of a linear equation that best fits the training data. This is achieved by adjusting the coefficients in the direction of the negative gradient of the Mean Squared Error with respect to those coefficients.

The respective intercept and coefficient of **X** will be adjusted by a factor of **α**, where α represents the learning rate.

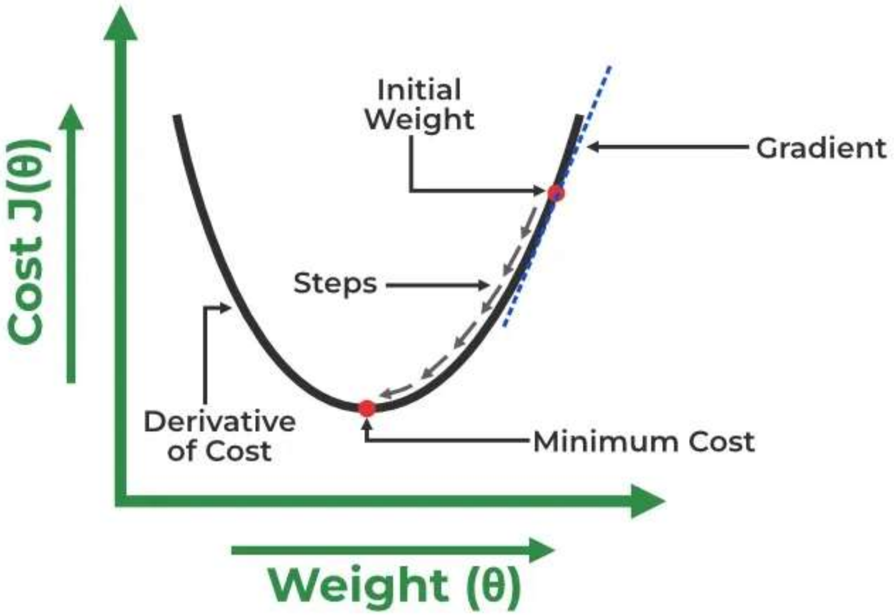
$$\boldsymbol{\theta_1} = \boldsymbol{\theta_1} - \boldsymbol{\alpha}(J'_{\theta_1})$$

$$= \boldsymbol{\theta_1} - \boldsymbol{\alpha}\left(\frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)\right)$$

$$\boldsymbol{\theta_2} = \boldsymbol{\theta_2} - \boldsymbol{\alpha}(J'_{\theta_2})$$

$$= \boldsymbol{\theta_2} - \boldsymbol{\alpha}\left(\frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)\right)$$

Repeat until convergence



https://www.geeksforgeeks.org/ml-linear-regression/

Linear Regression

Numerical Example

$$\boldsymbol{\hat{y}_i = \theta_1 + \theta_2 x_i}$$

Let $\boldsymbol{\theta_1}$=300, $\boldsymbol{\theta_2}$=10

Learning rate, $\alpha = 0.0001$

$$\boldsymbol{\hat{y}_i = 300 + 10x_i}$$

$$\mathbf{J} = \boldsymbol{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - \ y_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(300 + 10x_i - \ y_i)^2}$$

$$= \frac{1}{7}[(300 - 10 * 30 - 800)^2 + (300 - 10 * 37 - 950)^2 + \ldots..$$

$$= 74485.714$$

Age (x)	Salary (y)
30	800
37	950
25	600
43	1050
50	1200
29	740
46	1100

Linear Regression

Numerical Example: Learning (updating θ_1 and θ_2)

Age (x)	Salary (y)
30	800
37	950
25	600
43	1050
50	1200
29	740
46	1100

$$J'_{\theta_2} = \frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)\, x_i$$

$$= \frac{2}{n}\sum_{i=1}^{n}(300 + 10x_i - y_i)\, x_i$$

$$= -20388.57412$$

$$J'_{\theta_1} = \frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)$$

$$= \frac{2}{n}\sum_{i=1}^{n}(300 + 10x_i - y_i)$$

$$= -497.1412$$

Linear Regression

Numerical Example: Learning (updating $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$)

$\boldsymbol{\theta_1}$(new)=$\boldsymbol{\theta_1} - \boldsymbol{\alpha}(J'_{\boldsymbol{\theta_1}})$

$= \mathbf{300 - 0.0001} * (\mathbf{-497.1412})$

$= \mathbf{300.0497}$

$\boldsymbol{\theta_2}$(new)=$\boldsymbol{\theta_2} - \boldsymbol{\alpha}(J'_{\boldsymbol{\theta_2}})$

$= \mathbf{10 - 0.0001} * (\mathbf{-20388.57412})$

$= \mathbf{12.03}$

Age (x)	Salary (y)
30	800
37	950
25	600
43	1050
50	1200
29	740
46	1100

Updated equation be

$$\hat{y}_i = \mathbf{300.0497 + 12.03}x_i$$

$$\mathbf{J(new)} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 = 39084.8289$$

Repeat until convergence

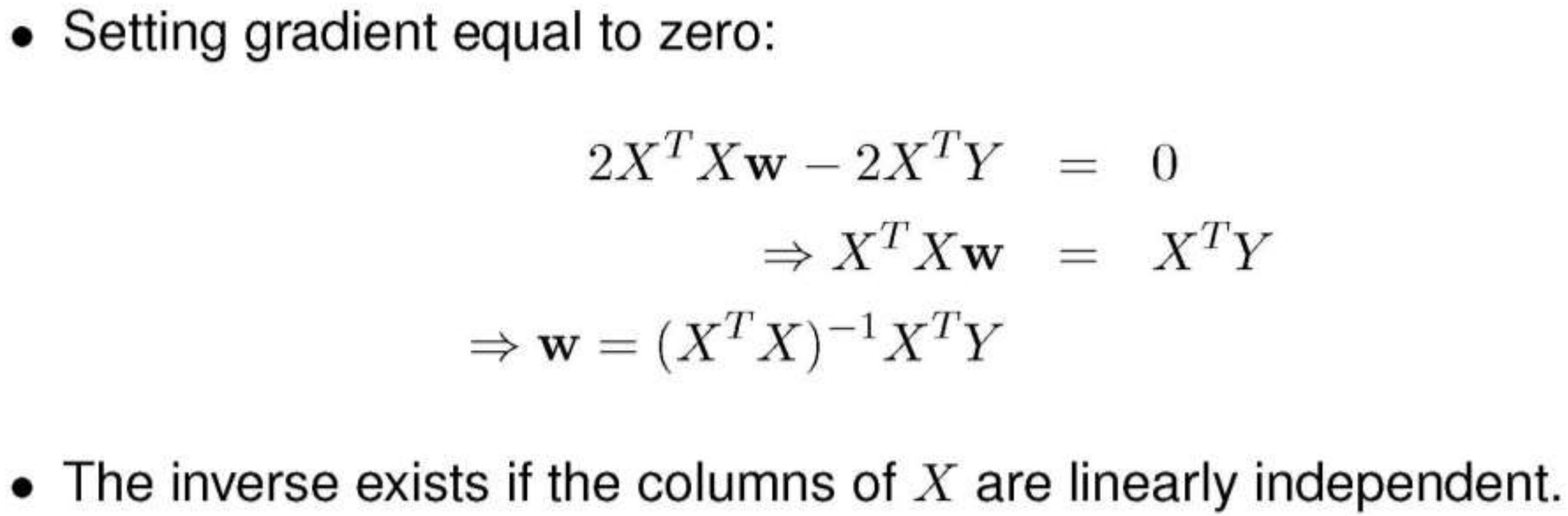
Linear Regression

Example with code

Python Notebook

https://colab.research.google.com/drive/1Q0XGUHv5tpbl5IixjuJAEGQIPhNDaIjc?usp=sharing

Some Linear Algebra - The Solution!



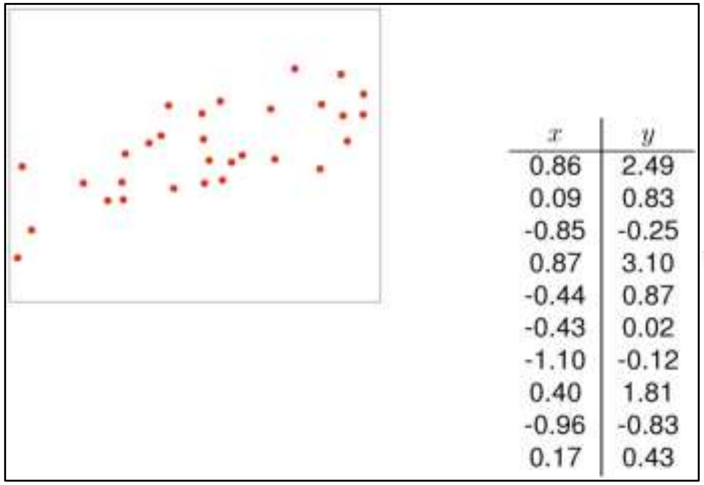
- Setting gradient equal to zero:

$$\begin{aligned} 2X^T X\mathbf{w} - 2X^T Y &= 0 \\ \Rightarrow X^T X\mathbf{w} &= X^T Y \end{aligned}$$

$$\Rightarrow \mathbf{w} = (X^T X)^{-1} X^T Y$$

- The inverse exists if the columns of X are linearly independent.

Example of Linear Regression - Data Matrices



x	y
0.86	2.49
0.09	0.83
-0.85	-0.25
0.87	3.10
-0.44	0.87
-0.43	0.02
-1.10	-0.12
0.40	1.81
-0.96	-0.83
0.17	0.43

$$X = \begin{bmatrix} 0.86 & 1 \\ 0.09 & 1 \\ -0.85 & 1 \\ 0.87 & 1 \\ -0.44 & 1 \\ -0.43 & 1 \\ -1.10 & 1 \\ 0.40 & 1 \\ -0.96 & 1 \\ 0.17 & 1 \end{bmatrix} \qquad Y = \begin{bmatrix} 2.49 \\ 0.83 \\ -0.25 \\ 3.10 \\ 0.87 \\ 0.02 \\ -0.12 \\ 1.81 \\ -0.83 \\ 0.43 \end{bmatrix}$$

X^TX

$$X^TX =$$

$$\begin{bmatrix} 0.86 & 0.09 & -0.85 & 0.87 & -0.44 & -0.43 & -1.10 & 0.40 & -0.96 & 0.17 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0.86 & 1 \\ 0.09 & 1 \\ -0.85 & 1 \\ 0.87 & 1 \\ -0.44 & 1 \\ -0.43 & 1 \\ -1.10 & 1 \\ 0.40 & 1 \\ -0.96 & 1 \\ 0.17 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 4.95 & -1.39 \\ -1.39 & 10 \end{bmatrix}$$

X^TY

$$X^TY =$$

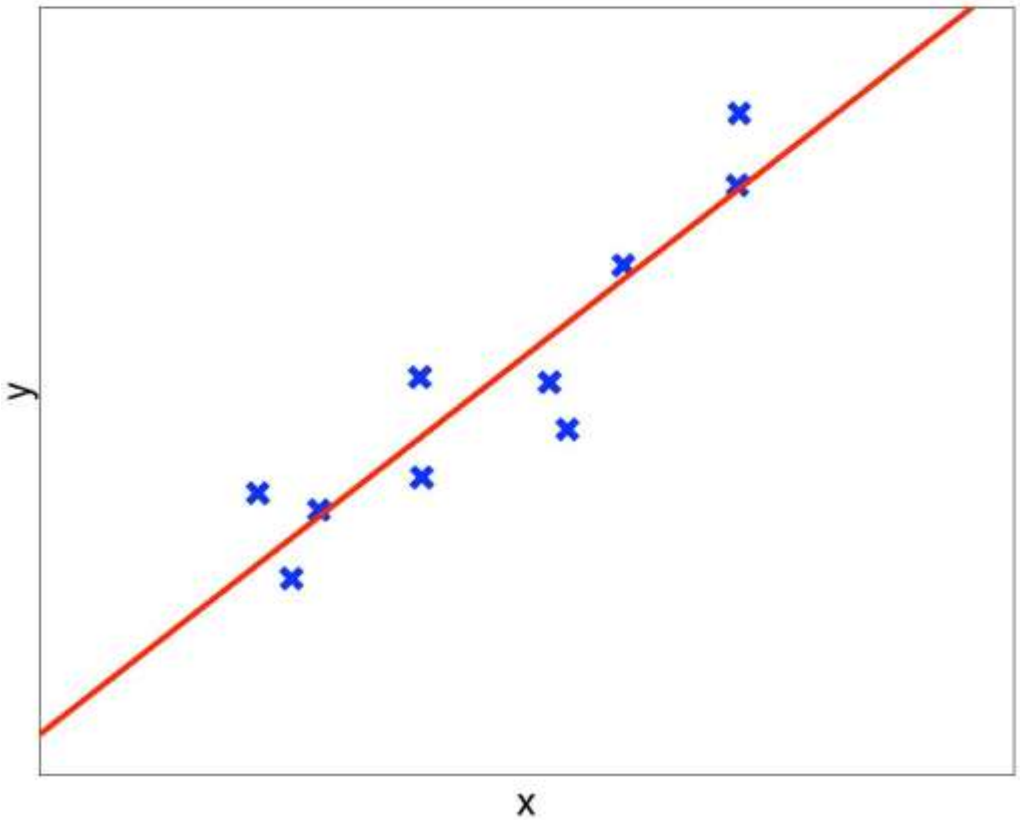
$$\begin{bmatrix} 0.86 & 0.09 & -0.85 & 0.87 & -0.44 & -0.43 & -1.10 & 0.40 & -0.96 & 0.17 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 2.49 \\ 0.83 \\ -0.25 \\ 3.10 \\ 0.87 \\ 0.02 \\ -0.12 \\ 1.81 \\ -0.83 \\ 0.43 \end{bmatrix}$$

$$= \begin{bmatrix} 6.49 \\ 8.34 \end{bmatrix}$$

Solving for *w* – Regression Curve

$$\mathbf{w} = (X^T X)^{-1} X^T Y = \begin{bmatrix} 4.95 & -1.39 \\ -1.39 & 10 \end{bmatrix}^{-1} \begin{bmatrix} 6.49 \\ 8.34 \end{bmatrix} = \begin{bmatrix} 1.60 \\ 1.05 \end{bmatrix}$$

So the best fit line is $y = 1.60x + 1.05$.



Logistic Regression

Logistic regression (also called logit regression) is commonly used to estimate the probability that an instance belongs to a particular class (e.g., what is the probability that this email is spam?).

If the estimated probability is greater than a given threshold (typically 50%), then the model predicts that the instance belongs to that class (called the positive class, labeled “1”), and otherwise it predicts that it does not (i.e., it belongs to the negative class, labeled “0”). This makes it a binary classifier.

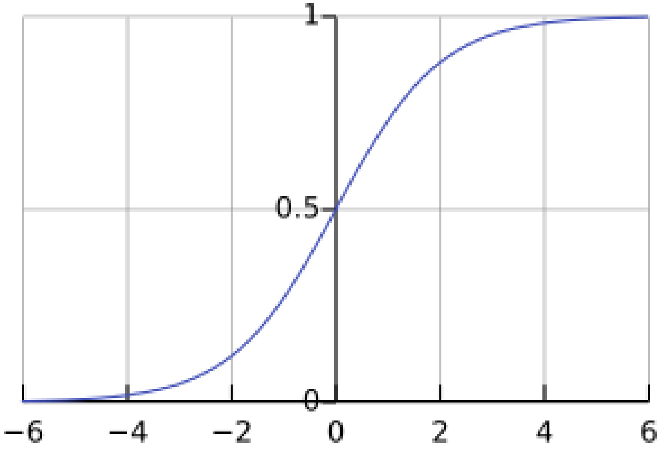
Just like a **linear regression** model, a **logistic regression** model computes a weighted sum of the input features (plus a bias term), but instead of outputting the result directly like the linear regression model does, it outputs the logistic of this result.

$$\boldsymbol{\sigma(t) = \frac{1}{1 + e^{-t}}}$$

Logistic Regression

$\boldsymbol{\hat{y}_i = \theta_1 + \theta_2 x_i}$ becomes $\boldsymbol{\hat{y}_i = \sigma(\theta_1 + \theta_2 x_i)}$

where $\boldsymbol{\sigma(t)} = \frac{\mathbf{1}}{\mathbf{1+e^{-t}}}$



The function is also know as sigmoid function.

Notice that $\boldsymbol{\sigma(t) < 0.5}$ when $\boldsymbol{t < 0}$, and $\boldsymbol{\sigma(t) \geq 0.5}$ when $\boldsymbol{t \geq 0}$, so a logistic regression model using the default threshold of **50%** probability predicts **1** if $\boldsymbol{\theta_1 + \theta_2 x_i}$ is positive and **0** if it $\boldsymbol{\theta_1 + \theta_2 x_i}$ is negative.

In this way Logistic Regression can be used as a binary classifier.

https://en.wikipedia.org/wiki/Sigmoid_function