Linear regression is a supervised machine learning algorithm which uses one independent variable to explain or predict the outcome of another variable. There are two types of linear regression - simple and multivariable.

Simple regression is used when only one variable is accounted for change in another variable.

$$y = mx+b$$

In real life scenarios one variable generally depends more than one independent variable so simple linear regression is not usually used to solve real-life tasks.

Multivariable regression, unlike simple linear regression,takes in account different variables to predict the dependent outcome of a dependent variable.

A multiple linear regression model looks like this: where *w* represents the coefficients, or weights, our model will try to learn.

$f(x,y,z) = w1*x + w2*y + w3*z$ where x,y,z represent the independent variables.

Simple Regression Calculation

Our algorithm will try to *learn* the correct values for Weight and Bias. By the end, our equation will approximate the *line of best fit*.

After this we calculate the cost function which is used to optimize our weights.We use MSE as our cost function. MSE measures the average squared difference between an observation's actual and predicted values

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

- N is the total number of observations (data points)
- $\frac{1}{N}\sum_{i=1}^{n}$ is the mean
- y_i is the actual value of an observation and $mx_i + b$ is our prediction

To minimize MSE we use Gradient Descent to calculate the gradient of our cost function.Gradient descent observes the error that our weight currently gives and using the derivative of the cost function to find the gradient and then changing our weight to move in the direction opposite of the gradient.

We iterate through our data points using our new weight and bias values and keep updating until we reach a minimum. The size of our update is controlled by the learning rate.

The value of weight and bias which gives minimum is the optimal line of fit for our data.

Logistic Regression is a classification algorithm. It returns a probability value which can then be mapped to two or more discrete classes.There are two types of logistic regression:
i) Binary logistic regression :- prediction is mapped to only two class
ii) Multiclass logistic regression :- prediction can be mapped to more than two classes

Binary logistic regression calculation

In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

$$S(z) = \frac{1}{1 + e^{-z}}$$

- $s(z)$ = output between 0 and 1 (probability estimate)
- z = input to the function (your algorithm's prediction e.g. mx + b)
- e = base of natural log

Set decision boundaries like : $p \geq 0.5, class=1$

$p < 0.5, class=0$

Stating relation between prediction and input like in multivariable regression:

$z = w0 + w1*x + w2*y$

We use cross entropy for cost function.Cross-entropy loss can be divided into two separate cost functions: one for $y=1$ and one for $y=0$.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\text{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \qquad \text{if } y = 1$$
$$\text{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \qquad \text{if } y = 0$$

The combined cost function is

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))]$$

Gradient descent for error optimization is calculated as

$s'(z) = s(z)(1-s(z))$

$C' = x(s(z)-y)$

We update weight and biases as required and finally classify the result