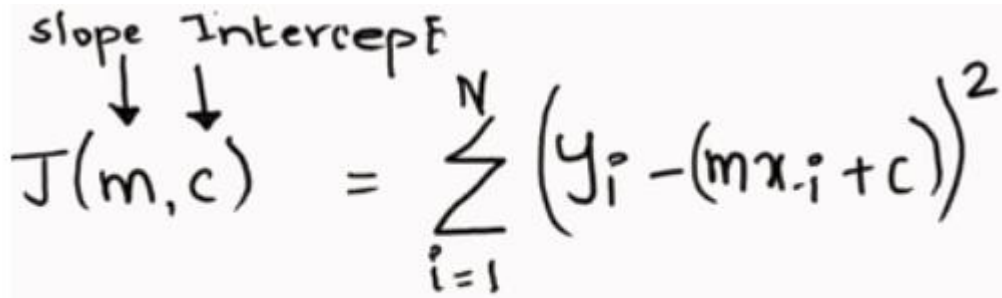


## Linear Regression – Gradient Descent

**Gradient descent** is an optimization algorithm used to find the values of the parameters (coefficients) of a function (f) that minimizes a given cost function (cost).

Let's start implementing the **cost function** for linear regression, which is also known as **MSE** (Mean Squared Error). MSE computes the 'goodness of fit' for a fitted line. The cost function takes two values, i.e. (m,c), where m is the coefficient and c is the intercept. The cost function iterates through each point in the given dataset, and then computes the sum of the square distance between each point and the line.


$$\begin{array}{c} \text{slope} \quad \text{Intercept} \\ \downarrow \quad \downarrow \\ J(m, c) = \sum_{i=1}^N (y_i - (mx_i + c))^2 \end{array}$$

Cost Function

## Linear Regression – Gradient Descent

---

- Our goal is to minimise the cost function, which will result in lower error values. If we minimise the cost function, we will get the best fit line to our data.
- Gradient descent is an optimisation algorithm used to find the values of the parameters (coefficients) of a function ( $f$ ) that minimises a cost function. To run gradient descent on the error function above, you first need to compute its gradient. To calculate it, you will need to differentiate your error function. Since the function is defined by two parameters ( $m$  and  $c$ ), you need to compute a partial derivative for each.

## Linear Regression – Gradient Descent

$$\frac{\partial J}{\partial m} = 2 \sum_{i=1}^N (y_i - (mx_i + c)) (-x_i)$$

$$\frac{\partial J}{\partial c} = 2 \sum_{i=1}^N (y_i - (mx_i + c)) (-1)$$

Derivatives

Now, we will iterate and update **m** and **c** to get the lower error value than the value from the previous iteration.

## Linear Regression – Gradient Descent

- The 'learning\_rate' variable controls the steps we take in downward direction in each iteration. If 'learning\_rate' is too low, the algorithm may take longer to reach the minimum value. On the other hand, if it is high, the algorithm may overstep the minimum value.
- The 'm\_gradient' variable corresponds to  $\partial J / \partial m$ , and 'c\_gradient' corresponds to  $\partial J / \partial c$ . Moreover, m\_current and c\_current correspond to the steps in which we are updating the values.

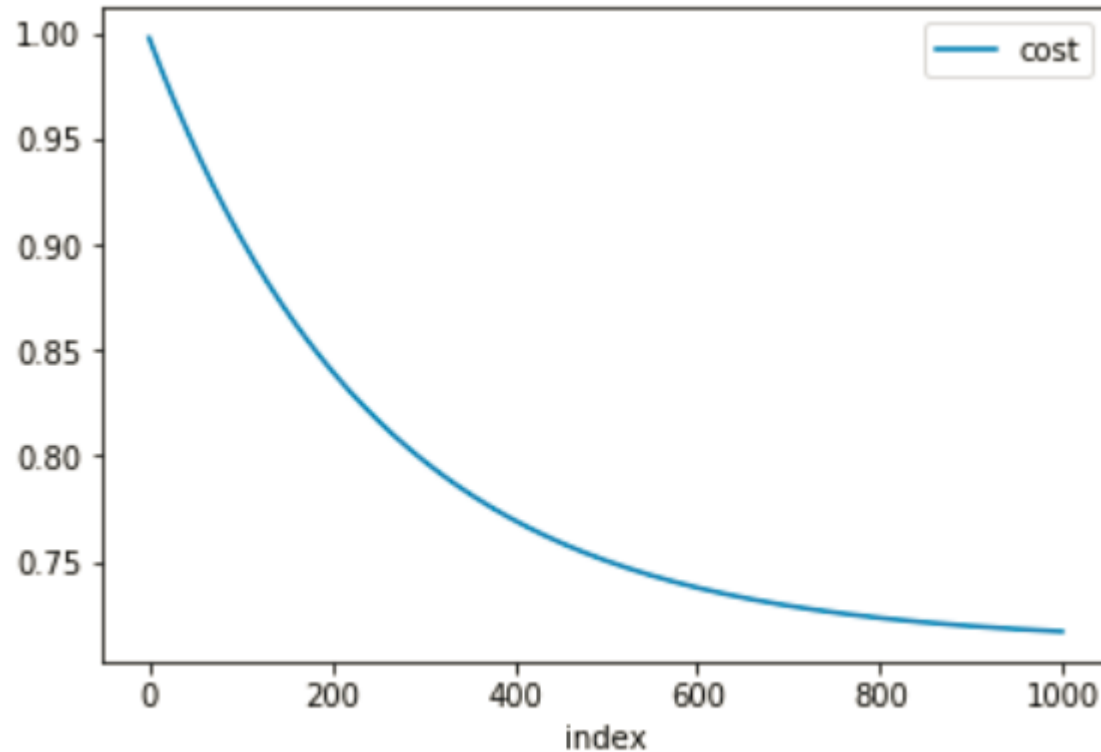
## Linear Regression – Gradient Descent

- We start with a default value  $m=1$ ,  $c=1$  and iterate multiple times till  $m, c$  values comes to stagnant point.
- Equation for calculating new  $m, c$  values from the initial values are:  
 $n$  = training rate. The less training rate the more iterations.

$$m_{new} = m_{old} - n \frac{dj}{dm}$$

$$c_{new} = c_{old} - n \frac{dj}{dc}$$

## Linear Regression – Gradient Descent



Cost Function Graph

The graph above shows how the cost decreases as we increase the number of iterations. The graph above explains the concept of gradient descent, which is to find out the values of the coefficient by minimising the cost function.