# Linear Regression in Machine learning

Machine Learning is a branch of Artificial intelligence that focuses on the development of algorithms and statistical models that can learn from and make predictions on data. Linear regression is also a type of machine-learning algorithm more specifically a supervised machine-learning algorithm that learns from the labeled datasets and maps the data points to the most optimized linear functions. which can be used for prediction on new datasets.

First of we should know what is supervised machine learning algorithms. It is a type of machine learning where the algorithm learns from labeled data.  Labeled data means the dataset whose respective target value is already known. Supervised learning has two types:

- •Classification: It predicts the class of the dataset based on the independent input variable. Class is the categorical or discrete values. like the image of an animal is a cat or dog?
- •Regression: It predicts the continuous output variables based on the independent input variable. like the prediction of house prices based on different parameters like house age, distance from the main road, location, area, etc.

Here, we will discuss one of the simplest types of regression i.e Linear Regression.

## Linear Regression

Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between a dependent variable and one or more independent features. When the number of the independent feature, is 1 then it is known as Univariate Linear regression, and in the case of more than one feature, it is known as multivariate linear regression. The goal of the algorithm is to find the best linear equation that can predict the value of the dependent variable based on the independent variables. The equation provides a straight line that represents the relationship between the dependent and independent variables. The slope of the line indicates

how much the dependent variable changes for a unit change in the independent variable(s).

Linear regression is used in many different fields, including finance, economics, and psychology, to understand and predict the behavior of a particular variable. For example, in finance, linear regression might be used to understand the relationship between a company's stock price and its earnings or to predict the future value of a currency based on its past performance.

One of the most important supervised learning tasks is regression. In regression set of records are present with X and Y values and these values are used to learn a function so if you want to predict Y from an unknown X this learned function can be used. In regression we have to find the value of Y, So, a function is required that predicts continuous Y in the case of regression given X as independent features.

Here Y is called a dependent or target variable and X is called an independent variable also known as the predictor of Y. There are many types of functions or modules that can be used for regression. A linear function is the simplest type of function. Here, X may be a single feature or multiple features representing the problem.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x)). Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best-fit line for our model.

## Assumption for Linear Regression Model

Linear regression is a powerful tool for understanding and predicting the behavior of a variable, however, it needs to meet a few conditions in order to be accurate and dependable solutions.

1. Linearity: The independent and dependent variables have a linear relationship with one another. This implies that changes in the dependent variable follow those in the independent variable(s) in a linear fashion.

2. Independence: The observations in the dataset are independent of each other. This means that the value of the dependent variable for one

observation does not depend on the value of the dependent variable for another observation.

3.Homoscedasticity: Across all levels of the independent variable(s), the variance of the errors is constant. This indicates that the amount of the independent variable(s) has no impact on the variance of the errors.

4.Normality: The errors in the model are normally distributed.

5.No multicollinearity: There is no high correlation between the independent variables. This indicates that there is little or no correlation between the independent variables.

Hypothesis function for Linear Regression :

As we have assumed earlier that our independent feature is the experience i.e X and the respective salary Y is the dependent variable. Let's assume there is a linear relationship between X and Y then the salary can be predicted using:

$$\hat{Y} = \theta_1 + \theta_2 X$$

OR

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

Here,

$\cdot\, y_i \epsilon Y \ (i = 1, 2, \cdots, n)$ are labels to data (Supervised learning)

$\cdot\, x_i \epsilon X \ (i = 1, 2, \cdots, n)$ are the input independent training data (univariate – one input variable(parameter))

$\cdot\, \hat{y}_i \epsilon \hat{Y} \ (i = 1, 2, \cdots, n)$ are the predicted values.

The model gets the best regression fit line by finding the best $\theta_1$ and $\theta_2$ values.

•$\theta_1$: intercept

•$\theta_2$: coefficient of x

Once we find the best $\theta_1$ and $\theta_2$ values, we get the best-fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

Cost function

The cost function or the loss function is nothing but the error or difference between the predicted value $\hat{Y}$ and the true value Y. It is the Mean Squared Error (MSE) between the predicted value and the true value. The cost function (J) can be written as:

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

How to update θ1 and θ2 values to get the best-fit line?

To achieve the best-fit regression line, the model aims to predict the target value $\hat{Y}$ such that the error difference between the predicted value $\hat{Y}$ and the true value Y is minimum. So, it is very important to update the θ1 and θ2 values, to reach the best value that minimizes the error between the predicted y value (pred) and the true y value (y).

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

**Gradient Descent**:

A linear regression model can be trained using the optimization algorithm gradient descent by iteratively modifying the model's parameters to reduce the mean squared error (MSE) of the model on a training dataset. To update θ1 and θ2 values in order to reduce the Cost function (minimizing RMSE value) and achieve the best-fit line the model uses Gradient Descent. The idea is to start with random θ1 and θ2 values and then iteratively update the values, reaching minimum cost.
A gradient is nothing but a derivative that defines the effects on outputs of the function with a little bit of variation in inputs.

Let's differentiate the cost function(J) with respect to $\theta_1$

$$J'_{\theta_1} = \frac{\partial J(\theta_1, \theta_2)}{\partial \theta_1}$$

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

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

$$= \frac{1}{n} \left[ \sum_{i=1}^{n} 2(\hat{y}_i - y_i) \left( \frac{\partial}{\partial \theta_1} (\theta_1 + \theta_2 x_i - y_i) \right) \right]$$

$$= \frac{1}{n} \left[ \sum_{i=1}^{n} 2(\hat{y}_i - y_i) \left( 1 + 0 - 0 \right) \right]$$

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

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

Let's differentiate the cost function(J) with respect to $\theta_2$

$$J'_{\theta_2} = \frac{\partial J(\theta_1, \theta_2)}{\partial \theta_2}$$

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

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

$$= \frac{1}{n} \left[ \sum_{i=1}^{n} 2(\hat{y}_i - y_i) \left( \frac{\partial}{\partial \theta_2} (\theta_1 + \theta_2 x_i - y_i) \right) \right]$$

$$= \frac{1}{n} \left[ \sum_{i=1}^{n} 2(\hat{y}_i - y_i) (0 + x_i - 0) \right]$$

$$= \frac{1}{n} \left[ \sum_{i=1}^{n} (\hat{y}_i - y_i) (2x_i) \right]$$

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

Finding the coefficients of a linear equation that best fits the training data is the objective of linear regression. By moving in the direction of the Mean Squared Error negative gradient with respect to the coefficients, the coefficients can be changed. And the respective intercept and coefficient of X will be if $\alpha$ is the learning rate.