Chapter 4
Linear Regression

Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent and independent variable. Linear regression is perhaps one of the most well-known and well understood algorithms in statistics and machine learning. Linear Regression is a *supervised machine learning* algorithm where the predicted output is *continuous* and has a constant slope. It's used to predict values within a continuous range, (e.g. sales, price).

4.1 Type of Variables: Simple linear regression is a regression technique in which the independent variable has a linear relationship with the dependent variable.

- *Dependent Variable:* A Dependent Variable is the variable to be predicted or explained in a regression model. This variable is assumed to be functionally related to the independent variable.
- *Independent Variable:* An Independent Variable is the variable related to the dependent variable in a regression equation. The independent variable is used in a regression model to estimate the value of the dependent variable.

For example, we take two kinds of variables such as amount of rainfall and wheat production. The wheat production variable is the dependent variable and the amount of rainfall is the independent variable. There can also be more than one independent variable. When there is just one independent variable it is called Simple Linear Regression. If there is more than one variable it is called Multiple or Multivariate Linear Regression.

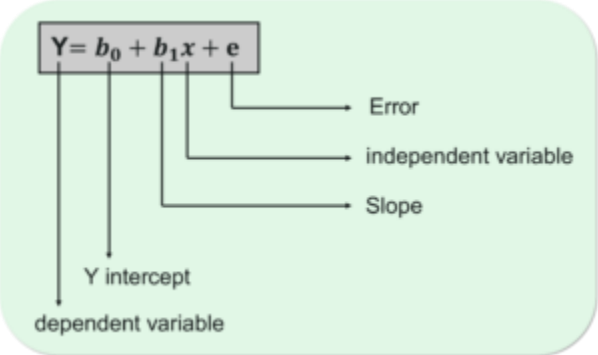
We always put dependent variable on Y-axis and independent variable on X-axis.

4.2 The Types of Linear Relationships

- *Positive Linear Relationship:* it shows the positive relationship between two variables.
- *Relationship NOT Linear:* There is a relationship where we can put the points in a mathematical equation.
- *Negative Linear Relationship:* It shows the negative relationship between two variables.
- *No Relationship:* It shows that there are no relationships between two variables.

4.3 Linear Regression Model

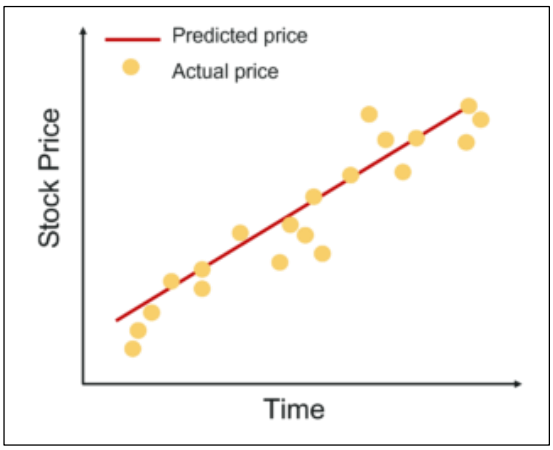
The following equation is used to represent a linear regression model:

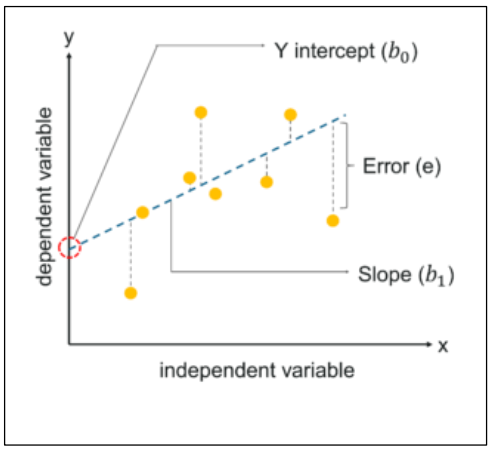


- ✓ Y stands for the dependent variable that needs to be predicted.
- ✓ b0 is the Y-intercept, which is basically the point on the line which touches the y-axis.
- ✓ b1 is the slope of the line (the slope can be negative or positive depending on the relationship between the dependent variable and the independent variable.)
- ✓ x here represents the independent variable that is used to predict our resultant dependent value.
- ✓ e denotes the error in the computation

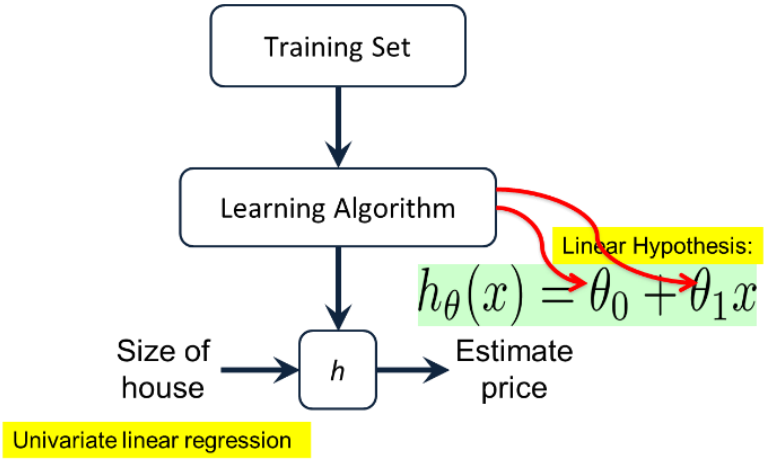
Let's assume that you want to predict the price of a stock over a period of time. For such problems, you can make use of linear regression by studying the relationship between the dependent variable which is the stock price and the independent variable which is the time. In this case, the stock price is the dependent variable, since the price of a stock depends and varies over time. And take note that the value of a stock is always a continuous quantity. On the other hand, time is the independent variable that can be either continuous or discrete. And this independent variable is used to decide the value of the dependent variable.

The first step in linear regression is to draw out a relationship between our dependent variable and independent variable by using a best fitting linear line.





Let the Regression Line Hypothesis is $h_\theta(x) = \theta_0 + \theta_1 x$. SO Learning algorithm for hypothesis function h will be



Choose θ_0 ,θ_1 so that $h_\theta(x)$ is close to y for our training examples

	Simplified ($\theta_0 = 0$)
Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$	$h_\theta(x) = \theta_1 x$
Parameters: θ_0, θ_1	θ_1
Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$	$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$
Goal: $\underset{\theta_0,\theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$	$\underset{\theta_1}{\text{minimize}}\ J(\theta_1)$

4.4. Linear Regression: Cost Model

The cost function helps us to figure out the best possible values for θ_0 and θ_1 which would provide the best fit line for the data points. Since we want the best values for θ_0 and θ_1, we convert this search problem into a minimization problem where we would like to minimize the error between the predicted value and the actual value.

4.4.1 Least Square Method – Finding the best fit line

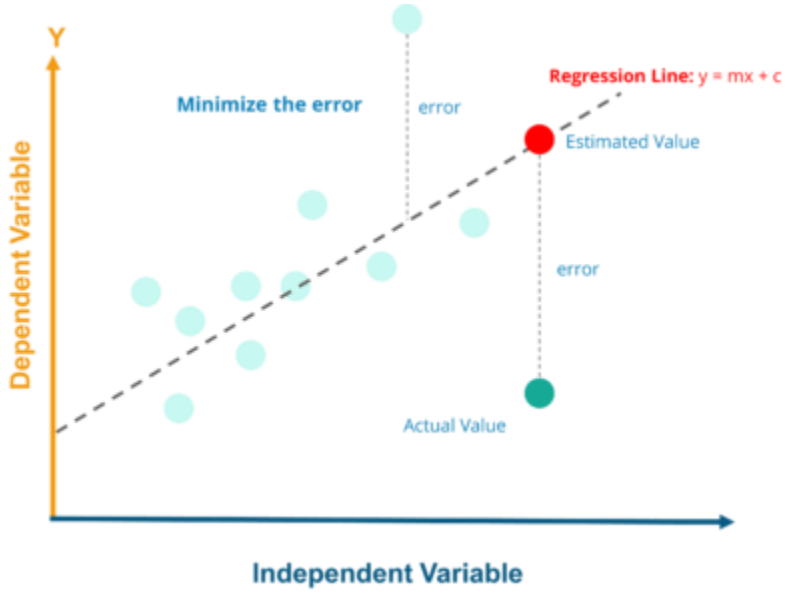
Least squares is a statistical method used to determine the best fit line or the regression line by minimizing the sum of squares created by a mathematical function. The "square" here refers to squaring the distance

between a data point and the regression line. The line with the minimum value of the sum of square is the best-fit regression line.

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

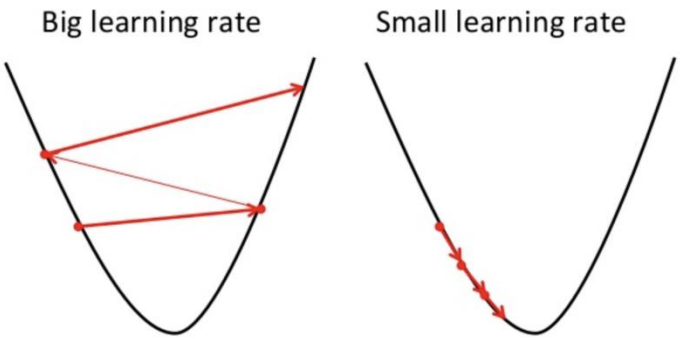
$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

We choose the above function to minimize. The difference between the predicted values and ground truth measures the error difference. We square the error difference and sum over all data points and divide that value by the total number of data points. This provides the average squared error over all the data points. Therefore, this cost function is also known as the Mean Squared Error(MSE) function. Now, using this MSE function we are going to change the values of θ_0 and θ_1 such that the MSE value settles at the minima.



4.4.2 Gradient Descent - Finding the best fit line

Gradient descent is a method of updating θ_0 and θ_1 to reduce the cost function(MSE). The idea is that we start with some values for θ_0 and θ_1, then we change these values iteratively to reduce the cost. Gradient descent helps us on how to change the values.

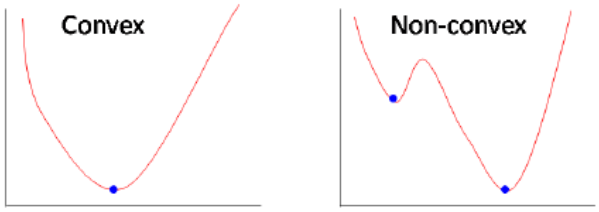


To draw an analogy, imagine a pit in the shape of U and you are standing at the topmost point in the pit and your objective is to reach the bottom of the pit. There is a catch, you can only take a discrete number of steps to reach the bottom. If you decide to take one step at a time you would eventually reach the bottom of the pit but this would take a longer time. If you choose to take longer steps each time, you would reach sooner but, there is a chance that you could overshoot the bottom of the pit and not exactly at the bottom. In the gradient descent algorithm, the number of steps you take is the *learning rate (α)*. This decides on how fast the algorithm converges to the minima.

If α is too small, gradient descent can be slow. If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.

Sometimes the cost function can be a non-convex function where you could settle at local minima but for linear regression, it is always a convex function.

You may be wondering how to use gradient descent to update θ_0 and θ_1. To update θ_0 and θ_1, we take gradients from the cost function. To find these gradients, we take partial derivatives with respect to θ_0 and θ_1. Now, to understand how the partial derivatives are found below you would require some calculus but if you don't, it is alright. You can take it as it is.



The partial derivates are the gradients and they are used to update the values of θ_0 and θ_1 Alpha is the learning rate which is a hyperparameter that you must specify. A smaller learning rate could get you closer to the minima but takes more time to reach the minima, a larger learning rate converges sooner but there is a chance that you could overshoot the minima.

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

$$J = \frac{1}{n}\sum_{i=1}^{n}(a_0 + a_1 \cdot x_i - y_i)^2$$

$$\frac{\partial J}{\partial a_0} = \frac{2}{n}\sum_{i=1}^{n}(a_0 + a_1 \cdot x_i - y_i) \implies \frac{\partial J}{\partial a_0} = \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i)$$

$$\frac{\partial J}{\partial a_1} = \frac{2}{n}\sum_{i=1}^{n}(a_0 + a_1 \cdot x_i - y_i) \cdot x_i \implies \frac{\partial J}{\partial a_1} = \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i) \cdot x_i$$

$$a_0 = a_0 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i)$$

$$a_1 = a_1 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i) \cdot x_i$$

where α is the *learning parameter*. The values of θ_0 and θ_1 are updated at each iteration to get the optimal solution

Repeat until convergence {

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

We repeat the steps until the cost function converges to the minimum value. If the value of α is too small, the cost function takes larger time to converge. If α is too large, gradient descent may overshoot the minimum and may finally fail to converge.

4.5 Goodness-of-fit Measure

There are two popular methods for evaluating the performance of a linear regression model

- Root mean squared error(RMSE)
- Coefficient of Determination (R^2 score)

Root mean squared error(RMSE): RMSE is the square root of the average of the sum of the squares of residuals. RMSE is defined by:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(h(x^i) - y^i)^2}$$

Coefficient of Determination (R^2 score): R^2 score or the coefficient of determination explains how much the total variance of the dependent variable can be reduced by using the least square regression. R^2 is determined by:

$$R^2 = 1 - \frac{SS_r}{SS_t}$$

Where, SS_t is the total sum of errors if we take the mean of the observed values as the predicted value.

$$SS_t = \sum_{i=1}^{m}(y^i - \bar{y})^2$$

SS_r is the sum of the square of residuals

$$SS_r = \sum_{i=1}^{m}(h(x^i) - y^i)^2$$