AI INVASION

Introduction To Regression Analysis

Regression Analysis

Linear Regression

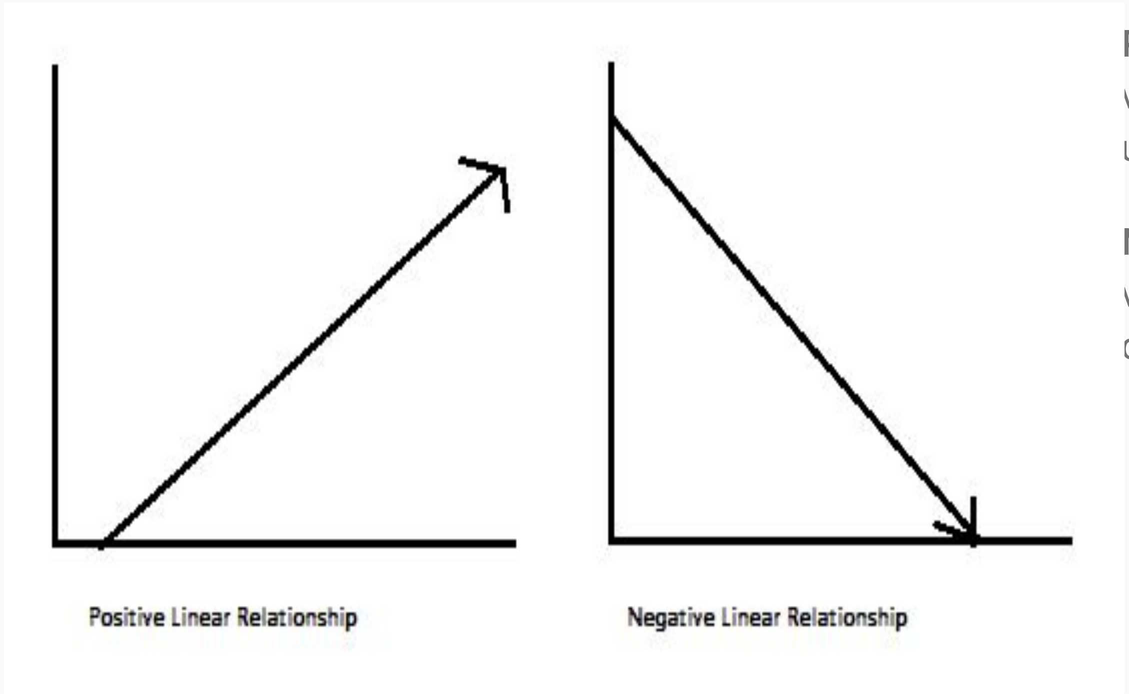
- **Linear Regression:** This is a statistical tool used to model the relationship between a scalar response (or **dependent variable**) and one or more explanatory variables (**independent variables**) that have a linear relationship.
- **Linear Regression** is used to determine the extent to which there is a **linear relationship** between a **dependent variable** and one or more **independent variables**

Concept of Linear Regression

Linear Regression is very suitable for answering certain questions such as:

- Is there a relationship between two variables?
- How strong is the relationship?
- Which variable contributes the most?
- How accurately can we estimate the effect of each variable?
- How accurately can we predict the target?
- Is the relationship linear?

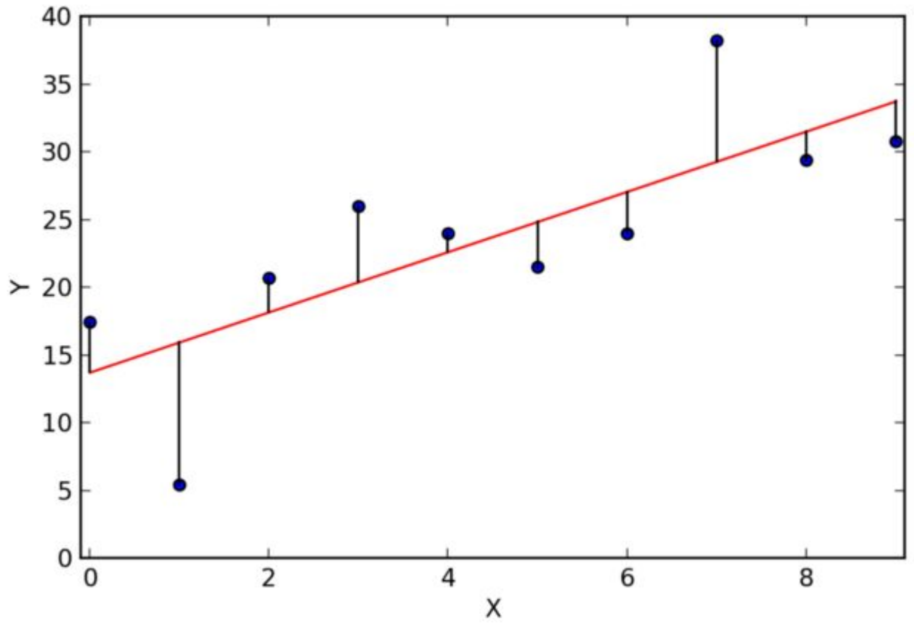
Concept of Linear Regression



Positive Linear Relationship: Independent variable goes up, dependent variable goes up also.

Negative Linear Relationship: Independent variable goes up, dependent variable goes down.

Concept of Linear Regression



Linear Regression models ultimately aims at minimizing the lengths of the black lines (residual errors).

Or reducing the distance of the blue dot from the red line as close to zero as possible.

This is similar to reducing the Mean Squared Error MSE, or the Sum f Squares of Errors SSE, also called the Residual Sum of Squares RSS.

A Little Bit About the Math

Linear Regression model is denoted by a mathematical expression similar as below.

$$y=c+mx$$ (Straight Line Graph)

y: is dependent variable

c: the intercept on the Y-axis of the graph

m: the slope of the graph. (Slope, m is, the rate of change of the dependent variable (y) with respect to a change in the independent variable (x)).

x: is the independent variable

Linear model example in machine learning

$$h_\theta = \theta_0 + \theta_1 x_1$$

The *thetas* in the above equations are the parameters we need to find so we could make more accurate predictions with our model

Training a model (Under the microscope)

The Steps in Training a model are outlined below

- **Hypothesis:** $h_\theta = \theta_0 + \theta_1 x_1$, $h_\theta = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_m x_m$

  This is the process of acquiring a prediction h_θ independent variable) from a linear regression model by substituting the corresponding value of X in the dataset.

  In the initial stage estimation, a random value for *theta* is chosen. Say $\theta_0 = 0.1 \; and \; \theta_1 = 0.002$. The result gotten is the first prediction h_θ

Steps in training a model

Cost Function (Mean Squared Error, MSE):

This is a method used to determine how good the estimation of the coefficient is by minimizing the error. $error\ (e_1) = prediction\ (h_1) - actual(y_1)$

The cost function is denoted by the expression $J = \frac{1}{2m}\sum_{i=1}^{m}((\theta_0 + \theta_1 x_1) - y_1)^2$, $J = \frac{1}{2m}\sum_{i=1}^{m}(h_1 - y_1)^2$

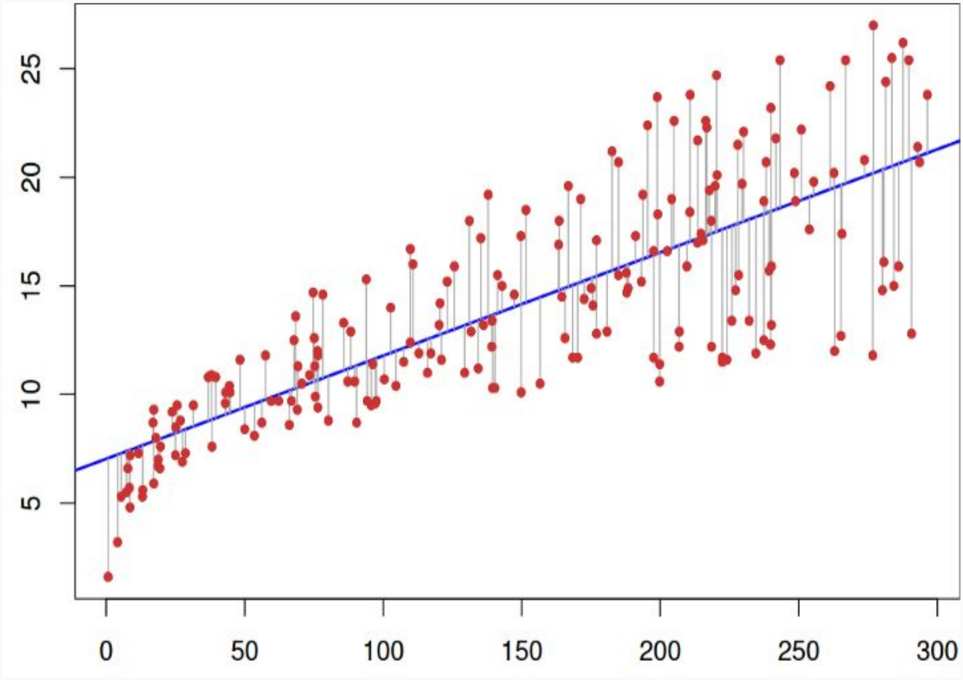
Where:

J: *(is the cost function)*, **m:** *(total number of data in the dataset)*, **h1:** *(prediction)*, **y1:** *(actual value in the dataset)*

Steps in Training a Model

Cost function is used to optimize the values of θ_0 *and* θ_1 parameters over the dataset by trial and error of the theta values from the initial stage of estimation.

Cost function aims at reducing the distance of the **red dots** from the **blue line**



Steps in Training a Model

Gradient Descent:

This is a function used to iterate the estimation of the θ_0 *and* θ_1 parameters used in the hypothesis model for getting the **prediction** values **h1** close to **y1** (i.e minimizing the **cost function**, **J**)

$$\theta_0 := \theta_0 - J\alpha$$

$$\theta_1 := \theta_1 - J\alpha * x_1$$

where: ***J*** *= the value gotten from the cost function*

α *= the learning rate*

θ_0 *= the intercept value of the hypothesis model*

θ_1 *= the slope value for the hypothesis model.*

Repeat_until_convergence {

$$\theta_0 := \theta_0 - \alpha\frac{1}{2m}\sum_{i=1}^{m}(h_1 - y_1)^2$$

$$\theta_1 := \theta_1 - \alpha\frac{1}{2m}\sum_{i=1}^{m}(h_1 - y_1)^2 * x_1$$

}

Steps in Training a model (Multivariate model)

Hypothesis

$$h_0 = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_m x_m$$

where $x_0 = 1$ *and* m *= the total number of data in the dataset.*

$$error\ (e_0) = prediction\ (h_0) - actual(y_0)$$

$$e_0 = h_0 - y_0$$

Cost Function

$$J(\theta_0, \theta_1, \ldots, \theta_m) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x)^{(i)} - y^{(i)})^2$$

Gradient Descent

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x)^{(i)} - y^{(i)})^2 * x_0$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x)^{(i)} - y^{(i)})^2 * x_1$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x)^{(i)} - y^{(i)})^2 * x_2$$

.

.

.

$$\theta_m := \theta_m - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x)^{(i)} - y^{(i)})^2 * x_m$$

Recap

- **Hypothesis model** is used to get a prediction value
- **Cost function** is used to calculate **the Mean Squared Error (MSE)** between the **Prediction** value and the **Actual** value.
- **Gradient descent** function is used to minimize the **Error** accumulation from the cost function by multiplying it with a learning rate denoted by the Alpha symbol
- The theta values from the **Gradient Descent** function is updated in the **Hypothesis** model for the next Epoch (loop)

Activity 1: Predicting Boston Housing Prices

Tools:

- **Scikit learn:** To load datasets and the algorithms needed.
- **Pandas:** To load data as a Panda Data frame and analyse the data
- Numpy:Scikit learn:
- Matplotlib:

Procedures:

Thanks!