FINGERTIPS

Module 9

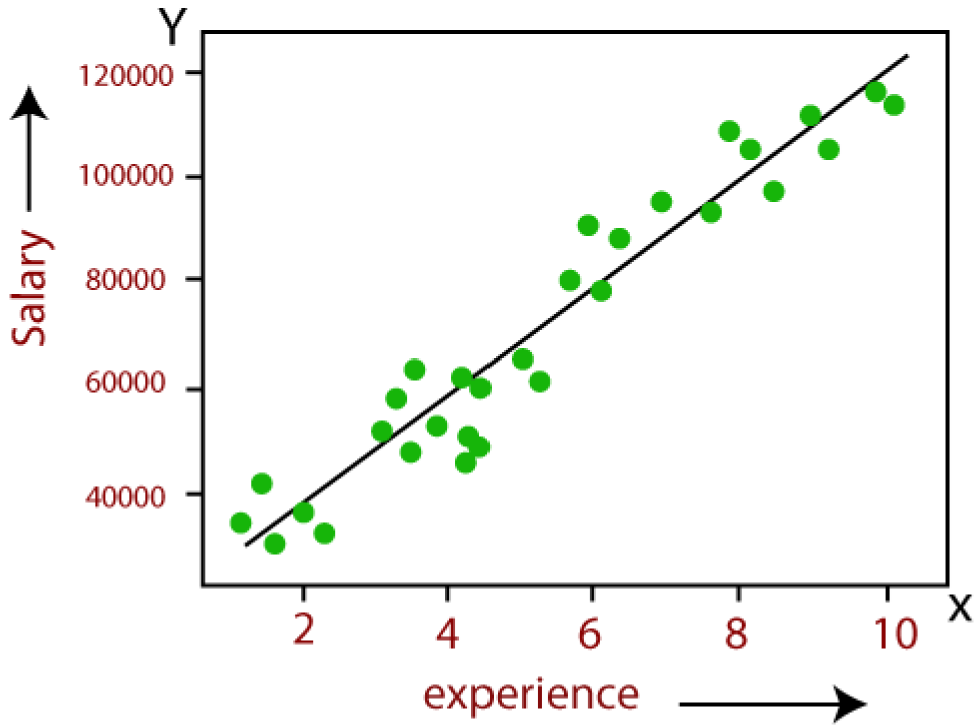
Regression

Regression in Machine Learning

- Regression analysis is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables.
- One of these variables is called predictor variable (independent variable) whose value is gathered through experiments. The other variable is called Target variables (dependent variable) whose value is derived from the predictor variable.
- Regression Analysis is widely used for Prediction and forecasting.
- Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed.
- It predicts continuous/real values such as **temperature, age, salary, price,** etc.

Linear Regression

- Linear regression attempts to find linear relationship between target and one or more predictors.
- It is a statistical regression method which is used for predictive analysis.
- It is one of the very simple and easy algorithms which works on regression and shows the relationship between the continuous variables.
- It is used for solving the regression problem in machine learning.
- Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis), hence called linear regression.
- The relationship between variables in the linear regression model can be explained using the below image. Here we are predicting the salary of an employee on the basis of the year of experience.



Basic Terminology

Target variable

Usually denoted by Y is the variable being predicted and is also called dependent variable, output variable, response variable or outcome variable.

Example-Sales, Salary

Predictor

Usually denoted by X sometimes called an independent or explanatory variable, is a variable that is being used to predict the target variable.

Example-Profit, Experience

Types of linear regression:

- Simple Linear regression
- Multiple Linear Regression

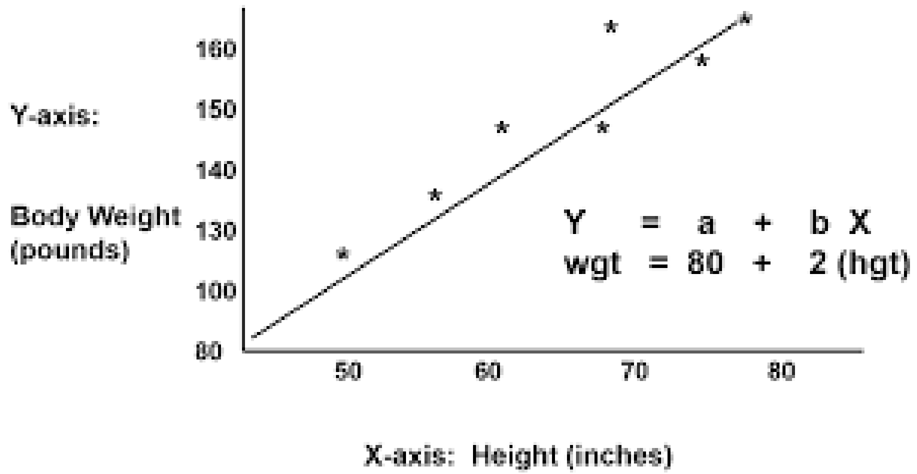
1) Simple Linear Regression

- If there is only one input variable (x), then such linear regression is called simple linear regression. And if there is more than one input variable, then such linear regression is called multiple linear regression.

- The primary idea is to obtain a line that best fits the data. The best fit line is the one for which total prediction error (all data points) are as small as possible. Error is the distance between the point to the regression line.

Simple Linear Equation

- **y= b0 + b1*x1**
- **y= Dependent variable**
- **b0 = Constant**
- **b1 = Coefficient**
- **x1 = Independent variable .**



2.) Multiple Linear Regression

- For estimating the relationship between two or more independent variables and one dependent variable, multiple linear regression is used.

Multiple linear Formula

- Following is the formula for multiple linear regression

$$y = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n + \varepsilon$$

- Here, y = predicted value of the dependent variable
- B0 = the y-intercept (value of y when all other parameters are set to 0)
- B1X1- the regression coefficient (B1) of the first independent variable (X1) (a.k.a. the effect that increasing the value of the independent variable has on the predicted y value)
- BnXn = the regression coefficient of the last independent variable
- e = model error

Loss Function

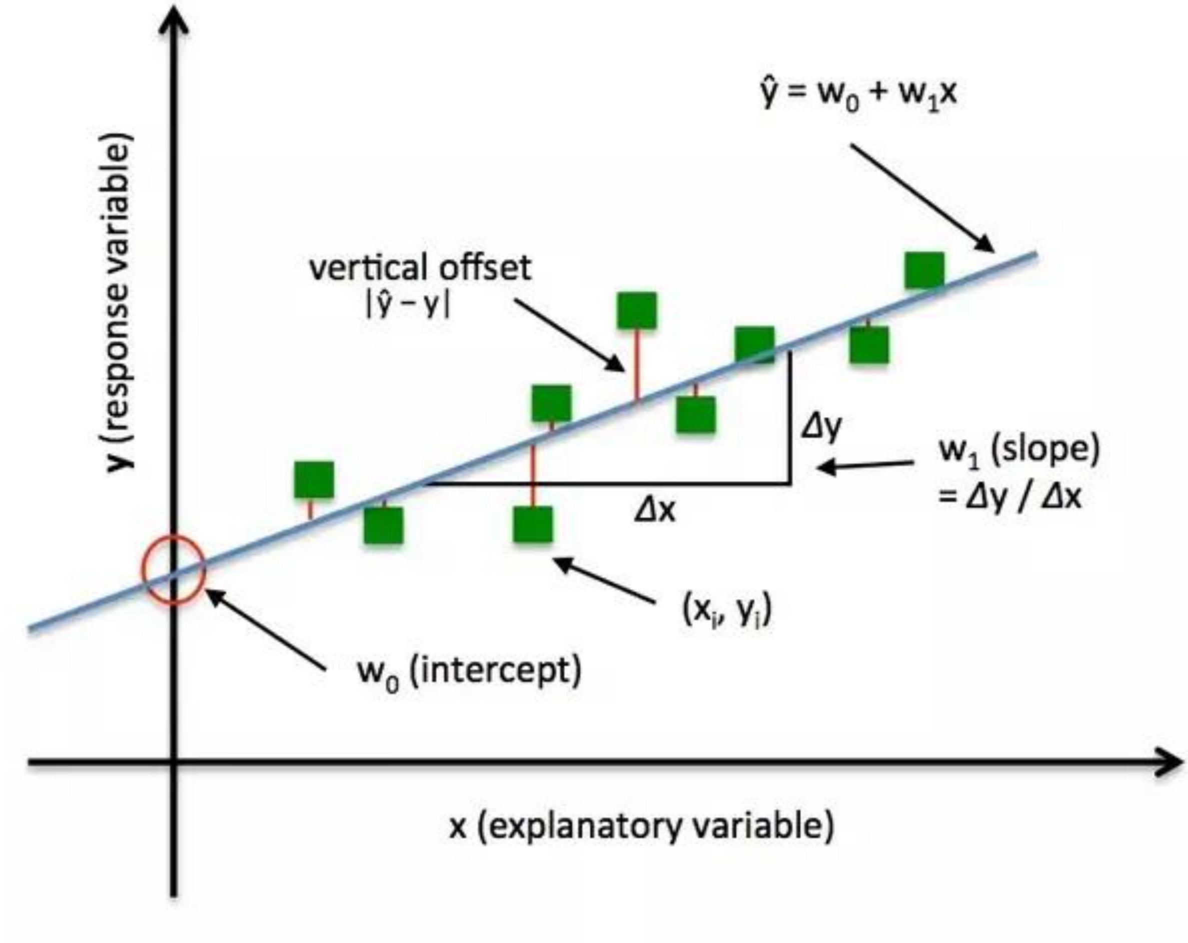
- It's a method of evaluating how well your algorithm models your dataset. If your predictions are totally off, your loss function will output a higher number.
- The loss function is a way of measuring "how well your algorithm models your dataset."
- If they're pretty good, it'll output a lower number. As you change pieces of your algorithm to try and improve your model, your loss function will tell you if you're getting anywhere.
- If your predictions are entirely off, a higher number will be produced by your loss function. If they're pretty good, they're going to yield a lower amount.
- Your loss function will inform you whether or not you're changing when you tune your algorithm to try to refine your model. 'Loss' let's one understand how distinct the expected value is from the real value.

- Following Is the formula for a Loss function-

- $distance = |\hat{y} - y|$

- Here, ŷ = predicted value
- Y = actual value

<table>
<tr><th>Our Predictions</th><th>Actual Values</th><th>Our Total Loss</th></tr>
<tr><td>Harlem: $1,000
SoHo: $2,000
West Village: $3,000</td><td rowspan="3">Harlem: $1,000
SoHo: $2,000
West Village: $3,000</td><td>0 (we got them all right)</td></tr>
<tr><td>Harlem: $500
SoHo: $2,000
West Village: $3,000</td><td>500 (we were off by $500 in Harlem)</td></tr>
<tr><td>Harlem: $500
SoHo: $1,500
West Village: $4,000</td><td>2000 (we were off by $500 in SoHo, and $1,000 in the west village)</td></tr>
</table>

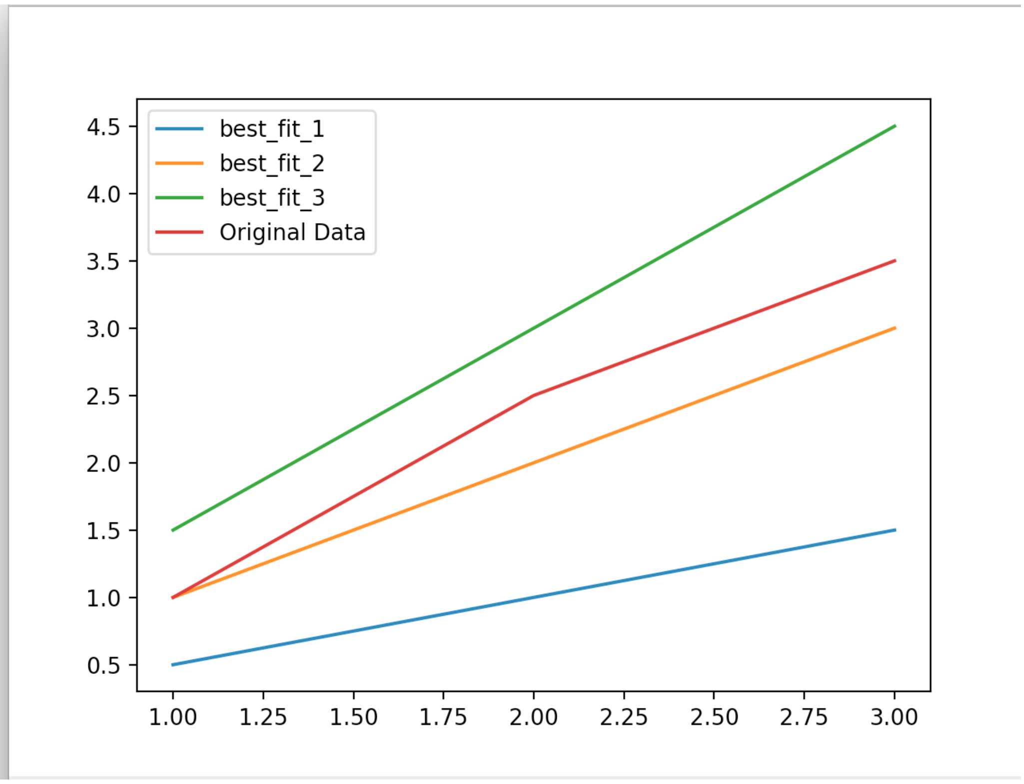


Cost Function

- The cost function is a function that calculates the performance for given data of a Machine Learning model.
- The Cost Function quantifies the error and expresses it in the form of a single real number between the estimated values and the expected values.
- The Cost Function can be generated in several different ways depending on the problem.

$$J = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y} - y)^2$$

- m= no of samples
- ŷ = predicted value
- y = actual value



best_fit_1: 1.083
best_fit_2: 0.083
best_fit_3: 0.25

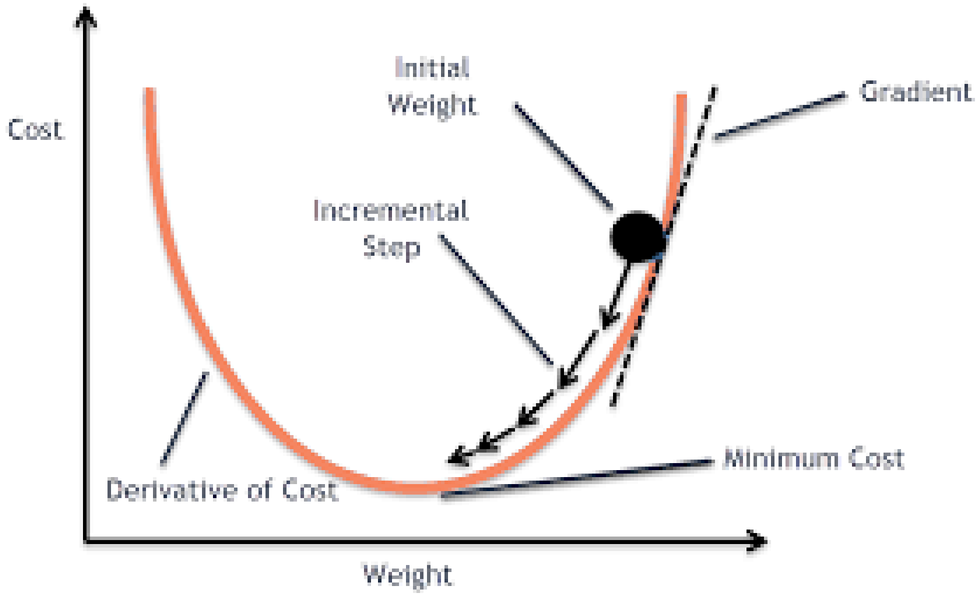
A lowest cost is desirable. A low cost represents a smaller difference. By minimizing the cost, we are finding the best fit. Out of the three-hypothesis presented, best_fit_2 has the lowest cost.

Difference between a Loss Function and a Cost Function

- A loss function is for a single training example. It is also sometimes called an error function.
- A cost function, on the other hand, is the average loss over the entire training dataset. The optimization strategies aim at minimizing the cost function.

Gradient Descent

- Gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function (f) that minimizes a cost function (cost).
- The goal of Gradient Descent is to minimize the objective convex function f(x) using iteration.
- If they're pretty good, they're going to yield a lower amount. Your loss function will inform you whether or not you're changing when you tune your algorithm to try to refine your model.
- 'Loss' let's one understand how distinct the expected value is from the real value.



Steps to implement Gradient Descent

- Randomly initialize values
- Update values (weight) via.

  $$weight^{(new)} = weight^{(old)} - constant\frac{\partial J(\Theta)}{\partial weight}$$

  Here, constant is learning rate

- Repeat until slope = 0
- The slope is described by drawing a tangent line to the graph at the point. So, if we are able to compute this tangent line, we might be able to compute the desired direction to reach the minima.

Model Evaluation

Mean Squared Error

- It is the average of the squared difference between the predicted and actual value. Since it is differentiable and has a convex shape, it is easier to optimize.

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

Mean Absolute Error

- This is simply the average of the absolute difference between the target value and the value predicted by the model. Not preferred in cases where outliers are prominent.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

Root Mean Squared Error (RMSE)

- RMSE is just the root of the average of squared residuals. We know that residuals are a measure of how distant the points are from the regression line. Thus, RMSE measures the scatter of these residuals.

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

Regularization

Ridge Regression

- It adds "squared magnitude" of coefficient as penalty term to the loss function. Here the highlighted part represents L2 regularization element.

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- Here if the lambda is zero, you can imagine that we're having OLS back. However, it will add too much weight if the lambda is very large and it will lead to under-fitting. That being said the way lambda is chosen is essential. To prevent over-fitting problems, this method works very well.

Lasso Regression

- Least Absolute Shrinkage and Selection Operator (LASSO) adds "absolute value of magnitude" of coefficient as penalty term to the loss function.

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- However, if lambda is zero then we will get back OLS whereas very large value will make coefficients zero hence it will under-fit.

- The biggest distinction in both approaches is that Lasso decreases the coefficient of the least relevant function to zero, thereby reducing those features entirely. So, in the event that we have a large range of features, this fits perfectly for feature selection.

Regression: Applications and Uses

- Regression is a prominent machine learning technique that is used in various fields from stock markets to scientific research.
- Analysing engine performance from test data in automobiles.
- Linear regression is used to model causal relationships between parameters in biological systems
- Linear regression can be used in weather data analysis
- Linear regression is often used in customer survey result analysis and market research studies.
- Linear regression is even used in observational astronomy for astronomical data analysis.