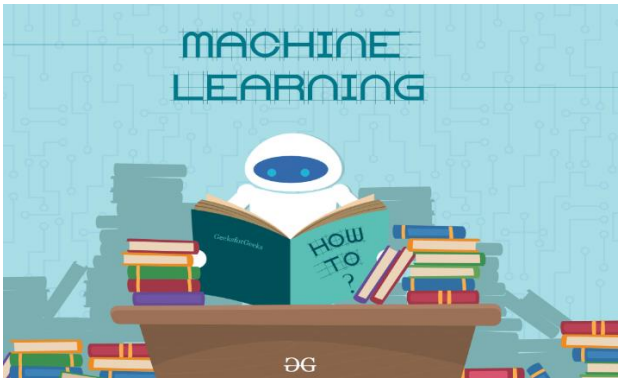
Machine Learning

Learning is finding out the pattern

Algorithm ≈ Equation

First we look into the data and find-out the pattern or outcome we are expecting, based on this we decide the algorithm.



Sai Subahsish Rout

Connect with me :

Linked in - https://www.linkedin.com/in/sai-subhasish-rout-655707151/

Github - https://github.com/saisubhasish/Concepts

Machine Learning :

Finding out the relationship between the data using some mathematical equation so that the system will be able to learn and adopt the instructions is called Machine Learning.

Machine tries to find out the optimal value of it's parameter.

Machine Learning is of 3 types :

Supervised ML → Regression (When we look for exact output)

→ Classification (When output is in label)

Machine tries to find-out the relationship between input & output.

Unsupervised ML Clustering → Grouping

Semi-supervised ML Regression + Classification + Clustering

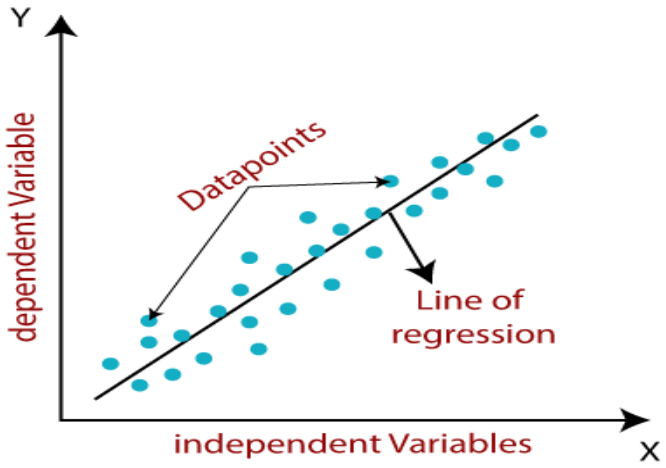
System tries to find out the pattern/relationship between dataset and based on that it'll form group

Regression

In Regression model we plot graph between the variables to find-out best fit line using which we will make prediction on the data.

Linear Regression :

It is a supervised Machine Learning Algorithm that tries to plot the best fit line between the feature (independent variable >> X) and label (dependent variable >> y) based on the dataset Where error should be minimal.



It is of two types :

- Simple Linear Regression

  y = mX + c

- Multiple Linear Regression

  y = mX1 + mX2 + mX3 + c

1. Simple Linear Regression

Simple Linear Regression is a regression model which defines the relationship and estimates the value of a dependent variable based on an independent variable.

$$\hat{y}_i = b_0 + b_1 x_i$$

2. Multiple Linear Regression

Multiple Linear Regression is a regression model which defines the relationship between one dependent variable and two or more independent variable. Here we predict the value of dependent variable based on the input features.

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_m x_m + \epsilon$$

In every algorithm we try to find-out the point where error = 0. This is called machine learning.

Our primary objective is to find-out relationship between feature and label.

If the dataset has some relationship we try to draw a line to find out mathematical equation. These equations show tendency of the dataset.

We need to define the accuracy of assumption score. (Confidence score). To find-out accuracy we use residual.

Slope :

The change in the value in y-axis due to change in the x -axis.

Intercept :

Intercept means a line crosses an axis.

Intercept is of two types

i. X-intercept
ii. Y-intercept

i. X-intercept

   X-intercept is the line where a line crosses X axis. At this point Y coordinate will be zero.

ii. Y-intercept

Y-intercept is the line where a line crosses Y axis. At this point X coordinate will be zero.

Residual :

The difference between original value and expected value.

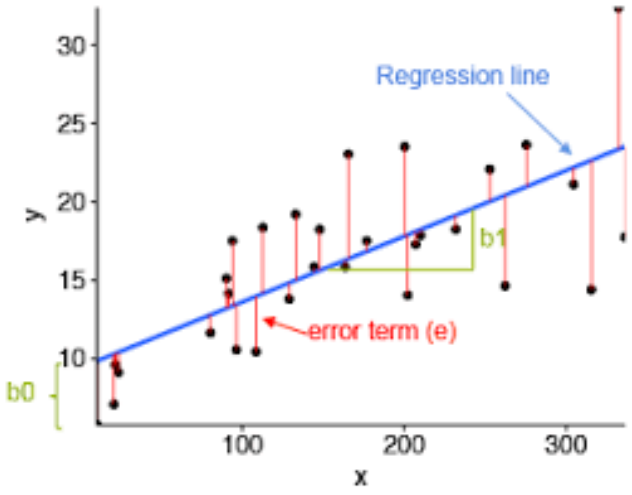
Residual= actual y value−predicted y value,

ri=yi−^yi.

Having a negative residual means the predicted value is too high, if you have a positive residual that means the predicted value is too low. The aim of best fit line(regression line) is to minimize the residual(error).

If we are having hundreds of data, we will have multiple residuals.

Our objective is find-out a line from hundreds of dataset where error term will be minimum or equal to zero.



Loss Function :

Calculating error (difference between actual value and predicted value) with respect to a single value.

ri=yi−^yi.

Cost Function

Calculating the aggregate error(deviation between actual value and expected value) .Cost function helps you to find out the point when your Machine Learning Model is most accurate by finding the relationship between input & output and how badly your model is predicting.

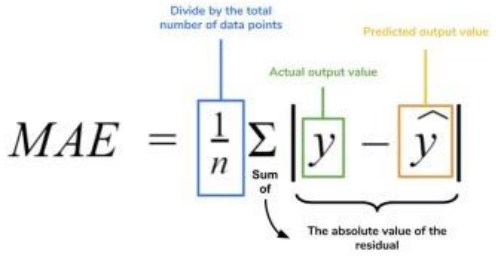
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

We calculate error after plotting of best fit line to check how much fit is the line is.

In Linear Regression we create a cost function and keeps on decreasing it to develop an accurate model with minimum error.

Mean Absolute Error

Mean Absolute Error is the measure of average absolute value of difference between actual value and predicted value (residuals) in the dataset.



Advantages	Disadvantage
i. Robust to outliers.	i. Convergence usually takes more time. Optimization is a complex task.
ii. It will be in same unit.	ii. Time consuming.

MSE :

It is the summation average of quadratic equation of difference between actual value and predicted value. It is used to calculate how close the predicted value to actual. Lower value of MSE indicates a better fit.

It also refers to variance of residuals.

Because of outliers in the dataset distribution will get changed and as we are finding the quadratic value the MSE value will differ a lot more. So in that case we will go for MAE.

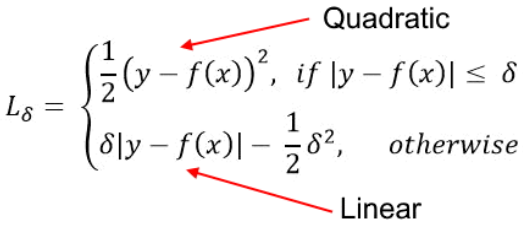
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Advantages	Disadvantage
i. This equation is differentiable.	i. This equation is not robust to outliers.
ii. This equation also has one global minima.	ii. Unit is changing

Huber Loss

Huber function is a loss function that is less sensitive to outliers in the dataset compared to MSE. It handles the data which are not handled by MAE(with outliers) and MSE(without outliers).

Huber loss provide function by balancing MAE & MSE



This function states that for the loss values less than **δ** use quadratic equation and for the loss values greater than **δ** use linear equation.

RMSE

RMSE (Root Mean Squared Error) is a model to get the difference between predicted values and actual values(residuals). It gets calculated by square root of Mean Square Error(MSE).

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

RMSE is the square root of variance, so called as standard deviation.

The lower the RSME the better the model is able to fit a dataset. The range of the dataset you are working with is important in determining whether your RSME value is low or not.

Performance Matrics

To measure the performance/accuracy of a regression model we use two metrics

1. R-squared
2. Adjusted R-squared

R Squared :

R-squared score defines the performance of your model, not the absolute loss.

MAE and MSE depend on the context, where as R-squared score is independent of context.

So in R-square value we have a baseline model to compare. The same we have in classification problems where threshold is fixed at 0.5.

Basically R-squared value calculates how much regression line is better than the mean line.

R-squared value ranges between 0 to 1. Which means the accuracy of your model.

If R-squared value is negative then our model is considered as very bad model.

If R-squared value is 1, then all the points fit in the line and we get the best fit line.

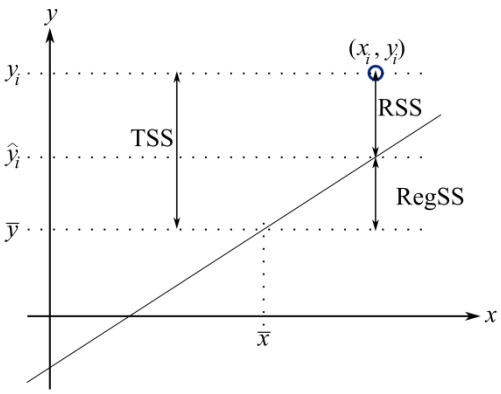
Formula

$$R^2 = 1 - \frac{RSS}{TSS}$$

R^2 = coefficient of determination

RSS = sum of squares of residuals

TSS = total sum of squares



TSS is a constant depends on data.

TSS is the distance between original data and average line.

Sum of Squares

Formula in Statistics

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

WallStreetMojo

Drawbacks :

When there are multiple features in the model and features are highly correlated, the increase in accuracy will be very high, later if there will be irrelevant feature while calculating the accuracy there will be a small increase but there is no direct correlation.

Adjusted R-squared :

Adjusted R-square came into picture to overcome the drawbacks of R-squared. It determines the accuracy of your model based on important features. (No biasness)

In adjusted R-square if there is any feature which is not highly correlated then it'll decrease the accuracy of your model.

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

Where

R^2 Sample R-Squared

N Total Sample Size

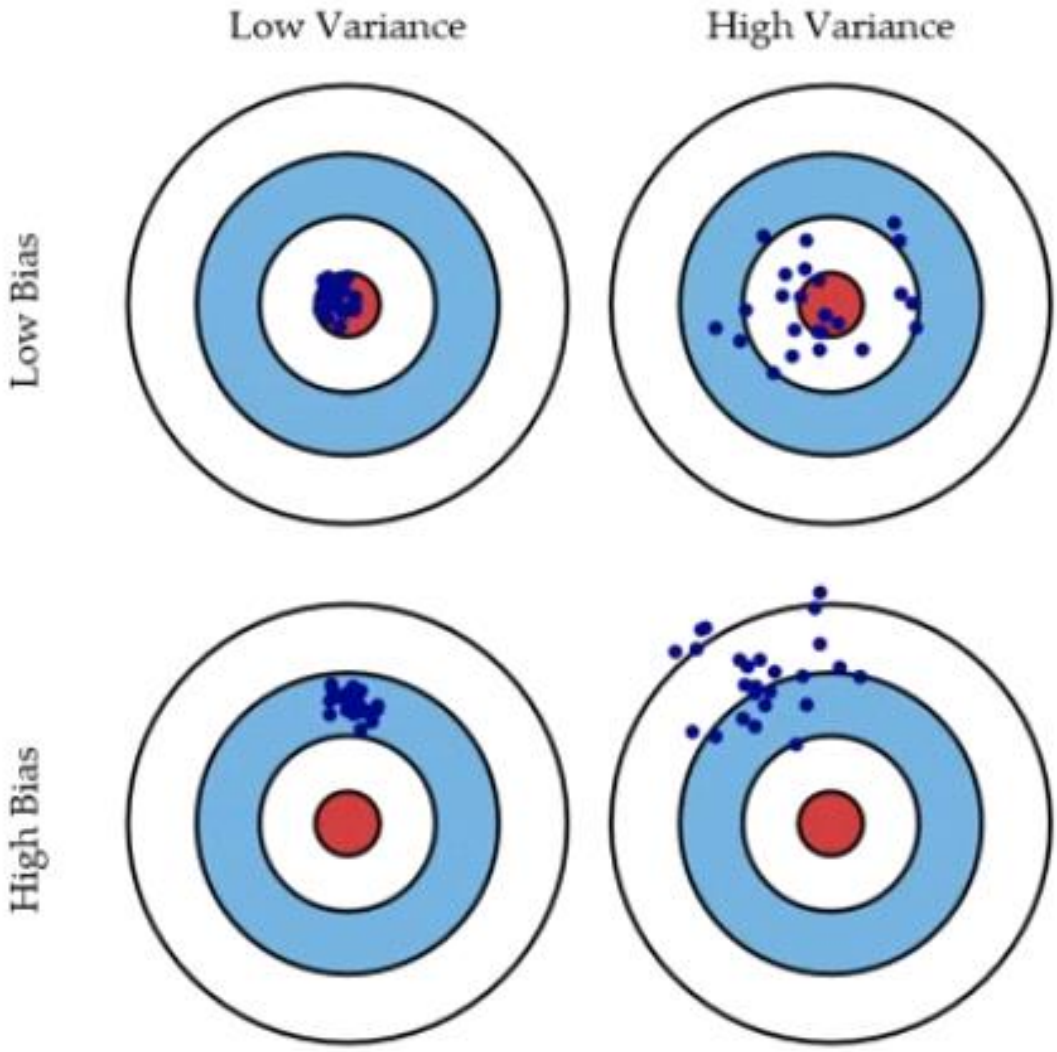
p Number of independent variable

As per R-squared value all the independent variables affect the result of the model, whereas the adjusted-R squared value defines only the independent variables which actually have an effect on the performance of the model.

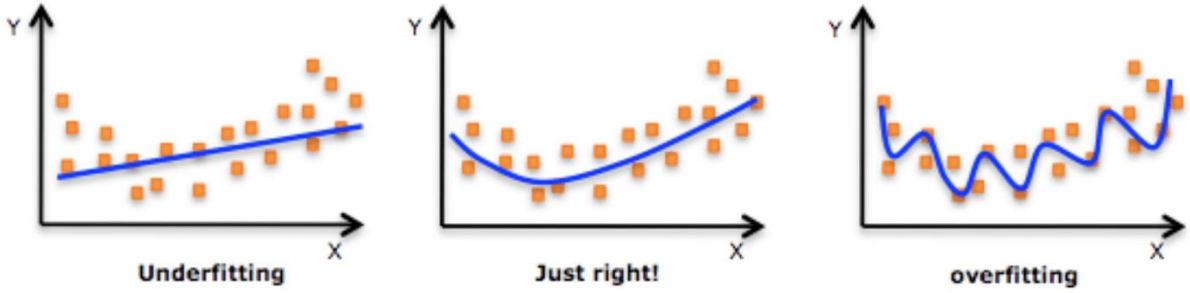
Bias: Bias is defined as the error rate of training data. When error value is high it is called as high bias and when the error value is low it is called as low bias.

Bias occurs when the model makes assumptions during training but these assumptions are not be correct when the model applies to test data.

Variance: The difference between the error rate of training data and test data is called variance. If the difference is high then it is called high variance and if the difference is low it is called low variance



Over-fitting and Under-fitting :



Over fitting occurs when your model fits exactly against your training data by becoming closely aligned to a limited set of data points. This leads the algorithm not to perform well against test data. . (Low bias and high variance)

Under fitting means your model won't perform well in training dataset and it perform well in test dataset. (High bias and low variance)

Training data information – Bias

Test data information – Variance

Training data – Very Good accuracy – Low bias

- Bad Accuracy – High bias

Test data – Very Good Accuracy – Low variance

- Bad Accuracy – High variance

High bias and high variance – Under-fitting

High bias and low variance – Under-fitting

Low bias and high variance – Over-fitting

Low bias and low variance – good model

When we apply Linear Regression ?

i. There must be some relationship you must find between X & Y.

(Linear Tendency)

ii. Mean of residuals should be zero. (residual should cancel each other)

iii. Error term are not supposed to be correlated with Y. (You shouldn't be able to predict y from e)

iv. Independent variable X and residuals must be uncorrelated. (exogenity)

v. Error term must show case constant variance. (homo sedacity) (constantly it should change)

vi. There should not be any multi-colinearity.(If there is any related feature we must drop it)

vii. Error term are supposed to be normally distributed.

Exogenity

If there is no relation between residual and independent variable X & Y.

Multicolinearity

It is phenomena where there is relation between X & X.

Correlation between independent variables leads to data redundancy.

If there is multicollinearity we may see some biasness in the model.

We will take non-multicollinear data always.

To identify multicollinearity we can use VIF, corr(), Feature Transformation

We can use Ridge and Lasso Regression to eliminate multicollinear variables.

If there is any feature which is more than 95% correlated then we need to go with any one.

Homosedacity

Homoscedasticity describes a situation where variance of residuals or error term is same across all the independent variables. Error term should showcase constant variance.

To handle homoscedasticity in the data set we can use

- Transformation
- Feature Engineering

Gradient Descent

In linear regression the model tries to get the best fit line to predict the value of y. We calculate the cost function(error) while training the model. In the model our aim is to minimize the cost function and get global minima where error will be close to 0.

Our objective is to find out the best value minimum of m & c or θ_1 and θ_2. If error is huge we cannot take huge jump, it may create problem. Initially model selects random value of θ_1 and θ_2

And then iteratively we do reduce the value until we get the best value of θ_1 and θ_2.

Here we are converging where the value tending towards zero. Every time we try to find-out a new value of m & c and calculate whether it is tending towards zero or not.

Learning rate (α) can be range in between (0.0001-10)

Mean Square Error is the reason why we get gradient descent.

We use MSE in gradient descent as it is differentiable.

Gradient descent algorithm

repeat until convergence {

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

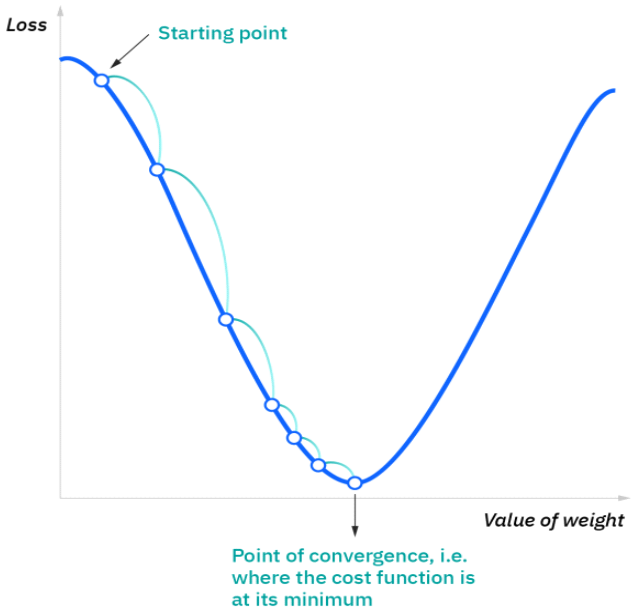
$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)}$$

}

update θ_0 and θ_1 simultaneously

$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$



What is convergence of an algorithm?

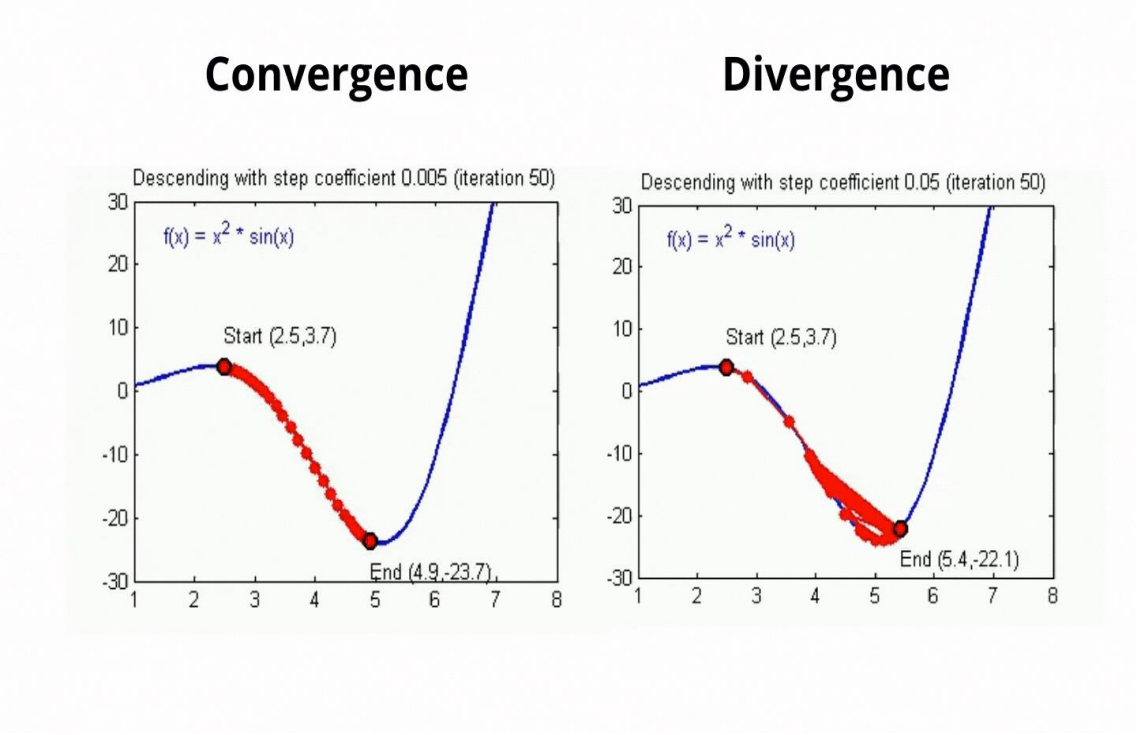
It is an iterative algorithm where we iterate multiple times to get a specific value. Here each step size is getting reduced as we get closer to the value.

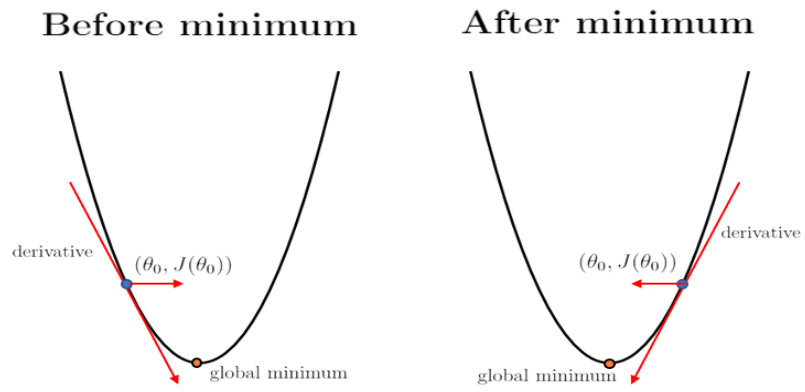
Repeat until convergence

{

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

}





Since the derivative is negative, if we subtract the derivative from θ_0, it will increase and go closer the minimum.

Since the derivative is positive, if we subtract the derivative from θ_0, it will decrease and go closer the minimum.

If the right side of the line is down then it is a negative slope. So θJ will get increased.

If the right side of the line is up then it is a positive slope. So θJ will get decreased.

Once we will be in global minima we will stop learning.

If our learning rate will be larger one then coordinate value will go out of gradient descent.

Learning rate (**α**) can be range in between (0.0001-10)

Parameter :

Something we pass as argument and which is learnt during machine learning process.

Hyper parameter :

Hyper parameters are the parameters whose value is used to control the learning process. (**λ**, nth iteration)

Hyper parameter affects the speed and quality of learning process.

Hyper parameter tuning :

Hyper parameter tuning is the process of finding the optimized value of hyper parameters for a learning algorithm by applying which the accuracy of your model will get maximized.

Why we try to find out/build our model with error ≈ 0 ?

If error will reduce, accuracy of our model will increase. But it is almost impossible to build a ML model that will make decision accurate 100%.

Unless and until we will reach a point where (r-squared) residual square tend to zero we won’t stop learning.

Write the assumptions of linear regression?

The linear regression has five key assumptions

i. Linear relationship
(Linear relationship is a correlation between two variables which describes how much one variable changes as related to change in the other variable)

ii. Multivariate Normality
(Multiple regression assumes that the residuals are normally distributed)

iii. No or little multicollinearity (Multicollinearity is a statistical concept which describes the relationship between two features)

iv. No auto-correlation
(Auto-correlation refers to correlation between the values of independent variables.)

v. Homosedacity
(Homosedacity describes the situation where the error term is same or constant across all the values of independent variable)

How hypothesis testing is used in linear regression?

In Linear regression model when we try to fit the straight line we get the slope and intercept for the line. Whenever we re-run the model by reducing slope and intercept and we test if the line is significant or not by checking if the coefficient is s
ignificant.

Steps to perform hypothesis test :

i. Formulate a hypothesis

ii. Determine the significance level

iii. Determine the type of test

iv. Calculate the test statistic value and p-values

v. Make decision.

Rigde Regression : (L2 Regularization)

Ridge regression reduces over fitting of the model. It works in the same way as the linear regression but it just adds an extra term (**λ**) which helps in the reduction of overfitting. The goal of any machine learning model is to generalize the pattern which it needs to be predicted; i.e. the model should work best on both training as well as test data.

Regularization : Regularization is the concept that is used to avoid overfitting and underfitting of data, when there is a variance in train and test data.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}) + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Here, λ Hyper-parameter

Hyper-parameter is used to make sure the line is not over fit, by changing it's value continuously.

Relation between λ and slope :

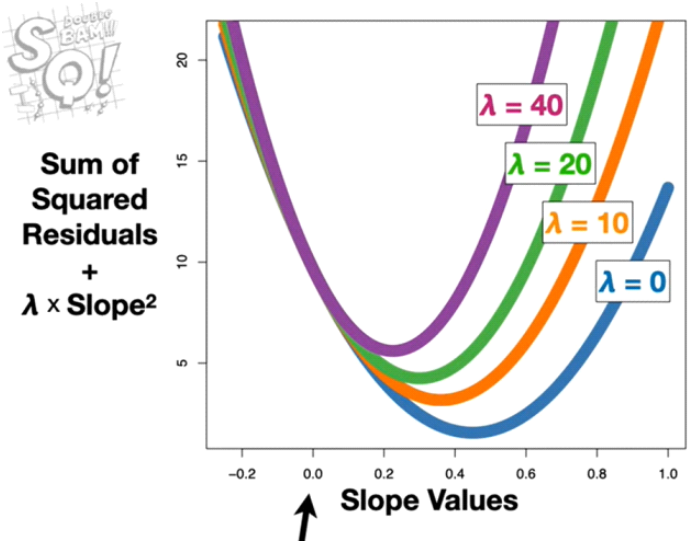
The influence of the penalty term is control by the λ parameter. When λ =0 it is same to cost function of Linear Regression.

In the training data if we get best fit line we add penalty to it.

With change in λ value (increases) the global minima shifts and slope value decreases. Increase in λ value decrease the cost function. And the value will never be negative as we are doing square.

So θ value will go towards 0.

In L2 normalization we never get the θ value as 0 and we will get a parabola shaped curve.

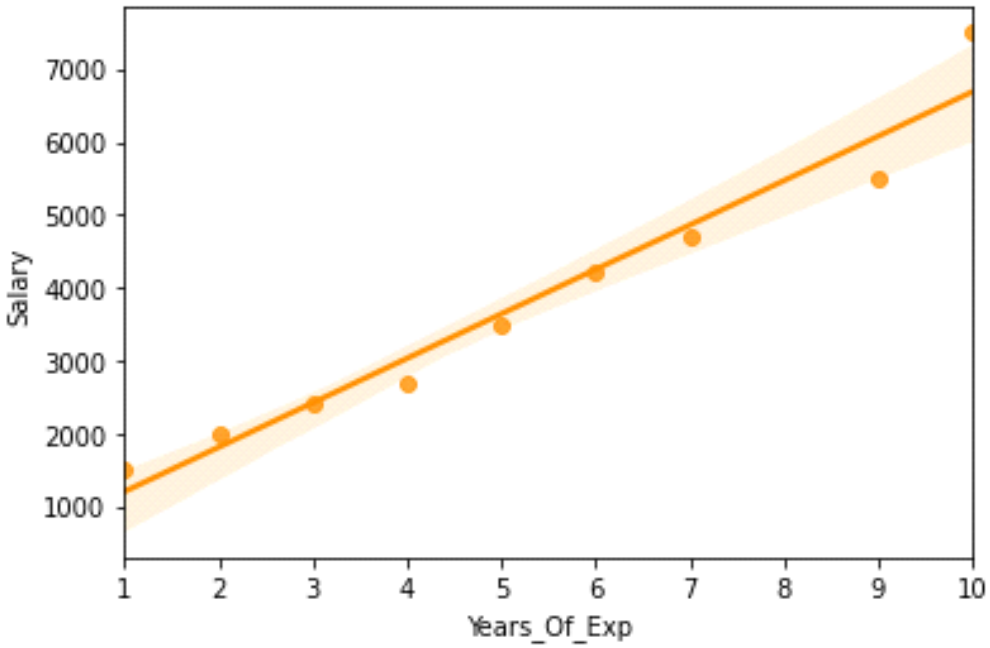


If our model don't perform well even after adding "λ*(slope)2" , we keep on changing the λ value.

In ridge regression λ value never be zero otherwise the feature will get deleted completely.

In the above mode in training if we get best fit line then cost function will become 0 and model will be over fitted. To avoid this situation we add "λ*(slope)2 " to the cost function and it will be positive.

As cost function will be positive, our model will try to reduce the cost function and line will switch it's position(rotate) in small amount as features will get reduced and line tries to be fit.



So λ is inversely proportional to the slope.

Lasso Regression : (L1 Normalization)

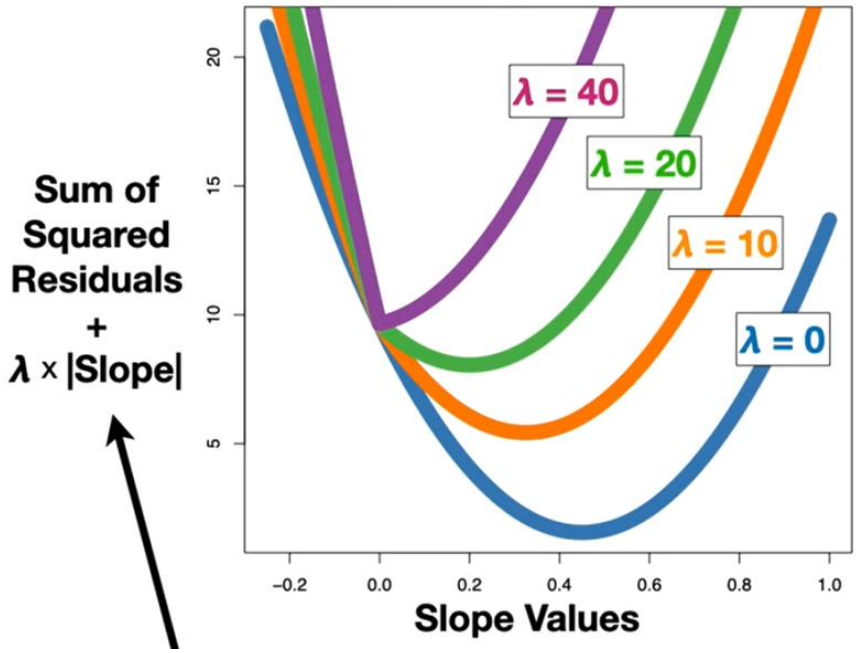
Least Absolute Shrinkage and Selection Operator (Lasso) is a technique used in regression methods for more accurate prediction. It is similar to ridge regression but there is a change in the penalty parameter.

L1 Normalization technique reduces the features which are less correlated.

With the increase in λ, global minima will shift and θ value changes, θ will go till 0 and the features which are not highly correlated or nearer to 0 will get deleted(as magnitude of the coefficients will reduce).

Only the features that are important will get considered for model prediction.

In L1 Normalization the optimal value becomes zero since we have a sharp curve.



L1 Normalization is used when we have more features and it performs feature selection automatically.

In Normalization we keep same number of features but reduce the magnitude of the coefficients using different types of techniques. The features which are not important will get reduced.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}) + \frac{\lambda}{m}\sum_{j=1}^{n} |\theta_j|$$

If data has outliers we will use Lasso Algorithm.

Elastic Net :

It is the combination Ridge and Lasso (of L1 Normalization and L2 Regularization.)

Elastic net always improves the limitation of ridge and lasso. Where lasso takes few sample of data, elastic net takes n number of variables. We can use Ridge and Lasso when we have limited data or we know

about the features but it is difficult to use them when we have a large dataset. In those cases we can use elastic-net regression which is good at estimating the relation between features.

That's why elastic net is preferred over both lasso & ridge.

$$ElasticNet = \sum_{i=1}^{n} (y_i - y(x_i))^2 + \alpha \sum_{j=1}^{p} |w_j| + \alpha \sum_{j=1}^{p} (w_i)^2$$

It takes care of over fitting and feature selection. The most optimum combination of λ parameters can be determined using cross-validation.

Box-cox Transformation :

Box-cox transformation is the transformation of a non-normal dependent variable to a normal shape. Normality is an important assumption for many statistical analysis. If your data is not normal and you are applying box-cox transformation then you are able to run a broader number of tests.

Your independent variables can be a data frame.

But the dependent feature can be a series or single dimensional array.

The formula is $y^l = y^{Lambda}$.

Thank you

Refer my GitHub for more information