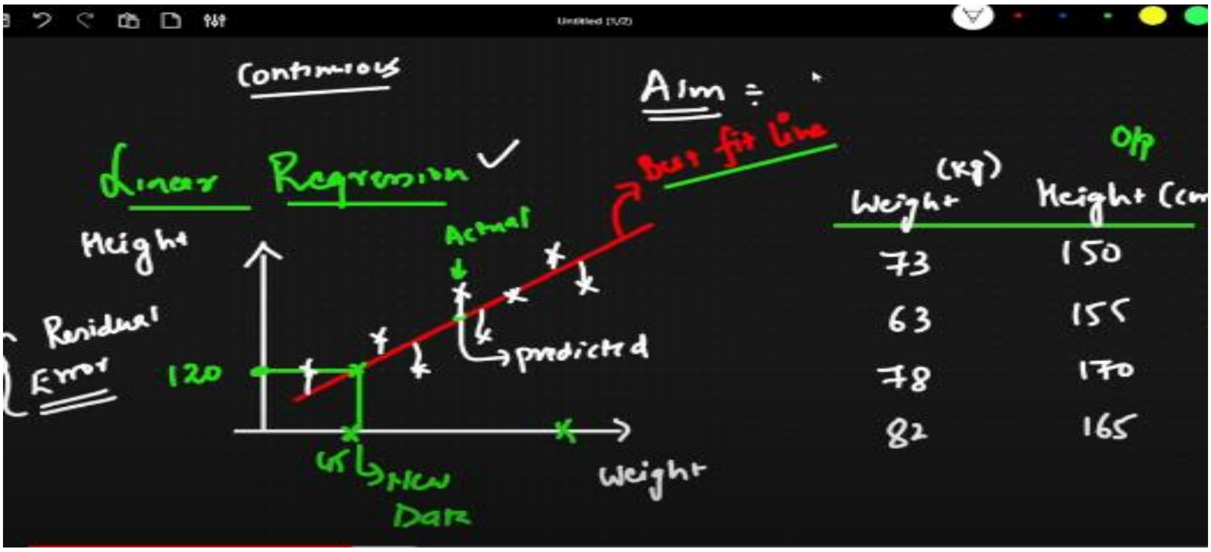
Introduction to ML

1. Supervised ML – Independent & Dependent features will be known in the dataset. We solve two types of problems in supervised ML
    a. Classification problem statement – When the dependent or target variable will be a categorical variable with a fixed number. Ex: Pass / Fail, Fraud / Non-Fraud (since there are only 2 possible values, we call it binary classification problem). If there are more than 2 possible outcome values, we call it multi class classification problem.
    b. Regression problem statement - When the dependent or target variable will be a continuous feature. Ex: House price
2. Unsupervised ML – Used when we categorize features into segments or buckets. Ex: Salary, Age – In this case, we segment groups of people whose salary is high, medium and low. This is very helpful in customer segmentation. Types of unsupervised ML algorithms are as follows –
    a. Hierarchal clustering
    b. K means clustering
    c. DB scan clustering

ML Algorithms

REGRESSION ALGORITHMS	CLASSIFICATION ALGORITHMS
Linear Regression	Logistic Regression
Lasso & Ridge Regression	Decision Tree Classification
Decision Tree Regression	Random Forest Classification
Random Forest Regression	Xgboost Classification
Xgboost Regression	ANN
ANN	CNN
RNN	

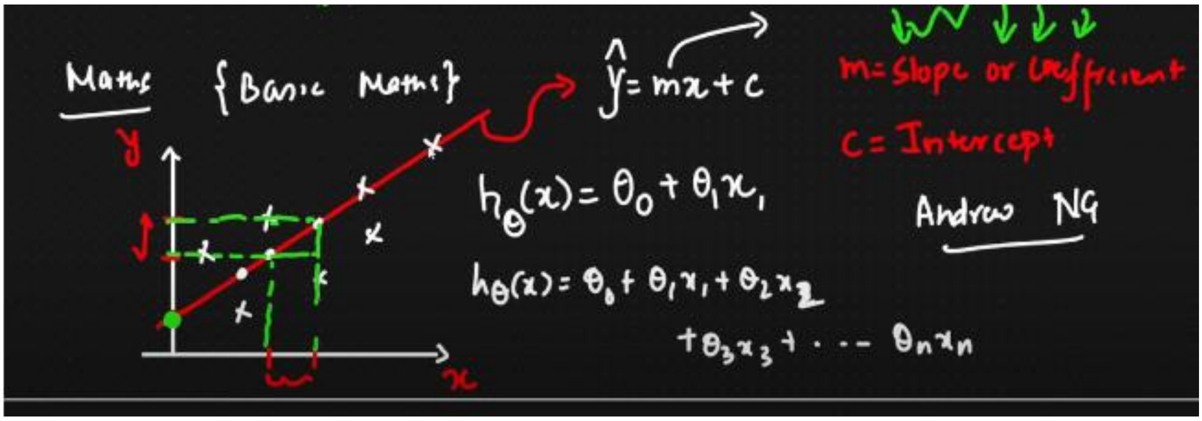
Simple Linear Regression



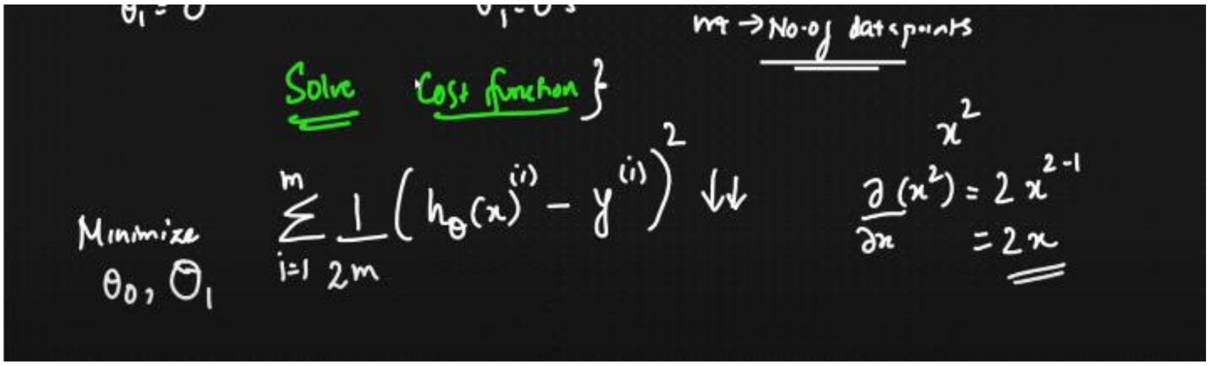
Weight (Kg)	O/p Height (cm
73	150
63	155
78	170
82	165

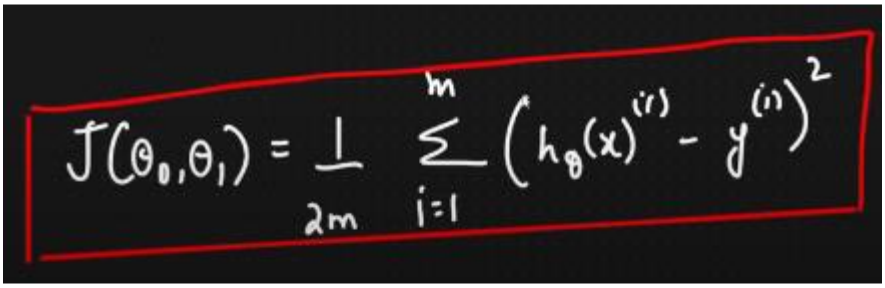
Linear Regression follows the approach of computing a best fit line aiming to pass through most of the data points within the training data set as shown in the above figure (red line). The points denoted by x mark are actual values and corresponding green dot on the best fit line will be the predicted value. A summation of differences between actual value & predicted value across all the points computes the residual error. Our main aim as data scientist will be to minimize the residual error in order to improve the model performance.

In real world scenario, if the residual error is 0, then it is a case of overfitting.

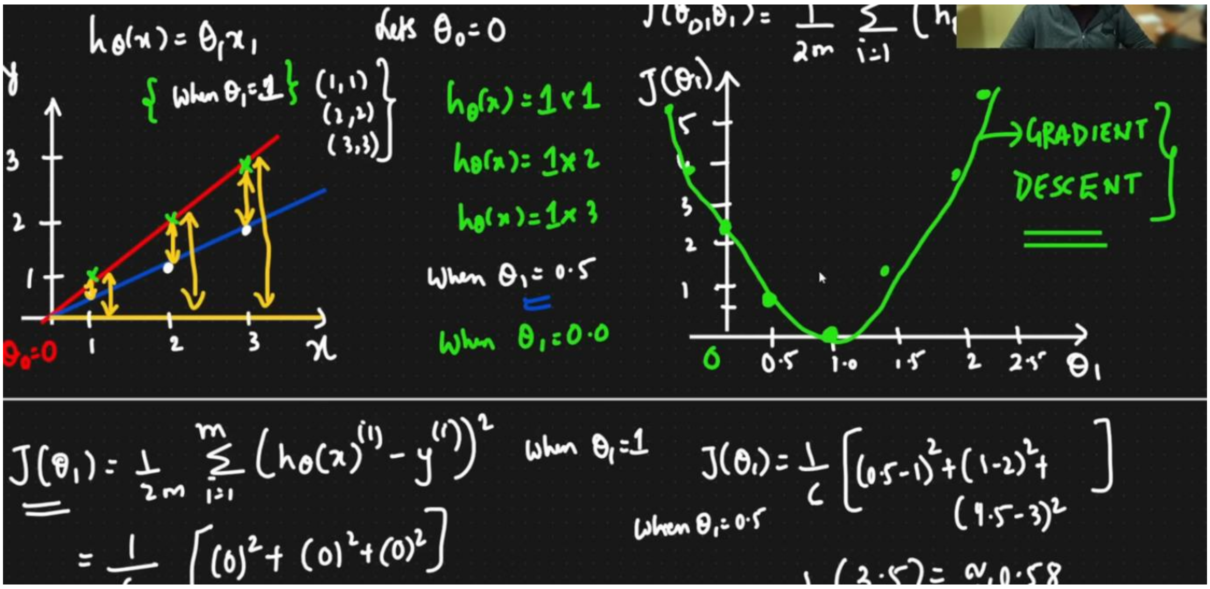

Maths
{Basic Maths}
y
x
$\hat{y} = mx + c$
m= Slope or coefficient
c= Intercept
$h_\theta(x) = \theta_0 + \theta_1 x_1$
$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots \theta_n x_n$
Andrew NG

The best fit line can be denoted by y = mx + c where m = slope or coefficient which denotes what is the change in y-axis when there is a 1 unit change in x-axis; c = intercept which determines what is the value of y when x = 0; y = predicted output;

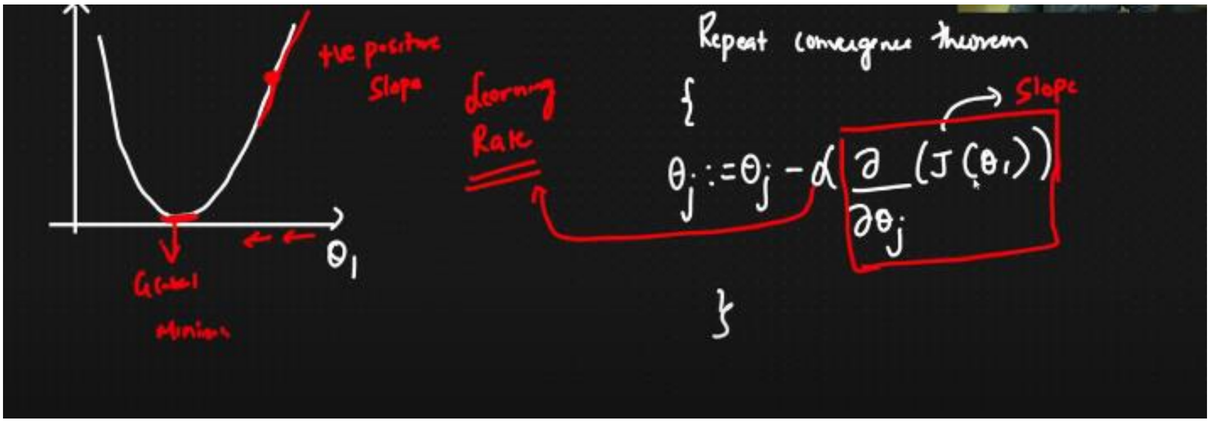

$\theta_1 = 0$
m → No. of datapoints
Solve
Cost function }
Minimize
θ_0, θ_1
$\sum_{i=1}^{m} \frac{1}{2m} \left(h_\theta(x)^{(i)} - y^{(i)}\right)^2$
x^2
$\frac{\partial (x^2)}{\partial x} = 2x^{2-1} = 2x$


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

The above formula depicts the cost function which should be minimum to compute the best fit line. Here h(x) indicates the predicted value; y indicates the actual value obtained from the training dataset; **we square these values to ensure a positive value in the output. A summation is used from 1 to m to cover m number of data points; we divide by m to take the average of all error values; Introduction of 2 is to find out slope (differentiation). Ultimately the aim is to reduce cost function.**



The least point in the gradient descent line is called global minima and this point gives us the minimum cost function which intern helps achieve the best fit line.

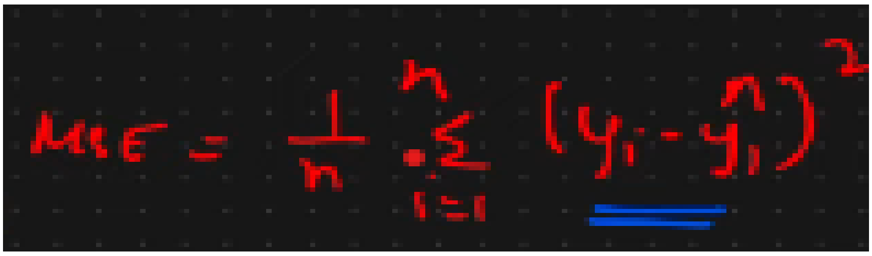


Now once we initialize Θ_1 value, we need to achieve global minima in an automatic way. This can be done by using repeat convergence theorem. Learning rate decides number of iterations to achieve the global minima point. It is advised to use a small learning rate value.

Steps to achieve global minima:

1. Start with Θ_0 and Θ_1
2. Keep changing Θ_0 ,Θ_1 to reduce J(Θ_0 ,Θ_1) until we reach somewhere near global minima

Mean Squared Error (MSE)



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

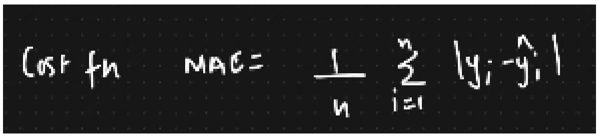
Advantages –

1. It is easily differentiable. In other words, we can easily find slope of any point on the gradient descent curve
2. It has only one local minima and global minima

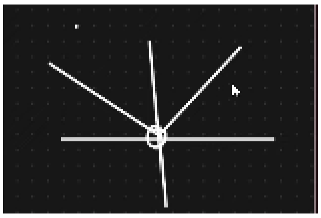
Disadvantages –

1. It is not robust to outliers.
2. It changes its units with respect to the output variable. Ex: When we consider salary (unit in INR), since in the MSE formula we are calculating difference between actual and predicted value square, the output will be salary square.

Mean Absolute Error (MAE)



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



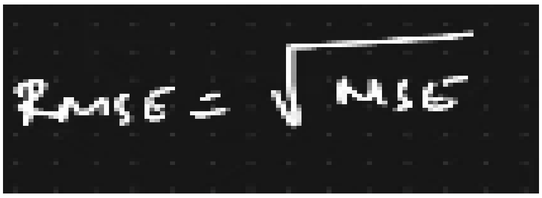
Advantages –

1. This is robust to outliers since we are squaring the difference between actual and predicted value
2. The unit will also not change

Disadvantages -

1. Convergence usually takes more time. In other words, the gradient descent line looks like how it is shown in the figure. In this case, we can calculate derivative of 0. Hence it sub-gradient concept to calculate derivatives
2. Optimization process is complex

Root Mean Square Error (RMSE) –



$\text{RMSE} = \sqrt{\text{MSE}}$

Advantages –

1. When assessing how well a model fits a dataset, we use the RMSE more often because it is measured in the same units as the response variable.

Disadvantages –

1. One major drawback of RMSE is its sensitivity to outliers and the outliers have to be removed for it to function properly.
2. RMSE increases with an increase in the size of the test sample.

Ridge & Lasso Regression

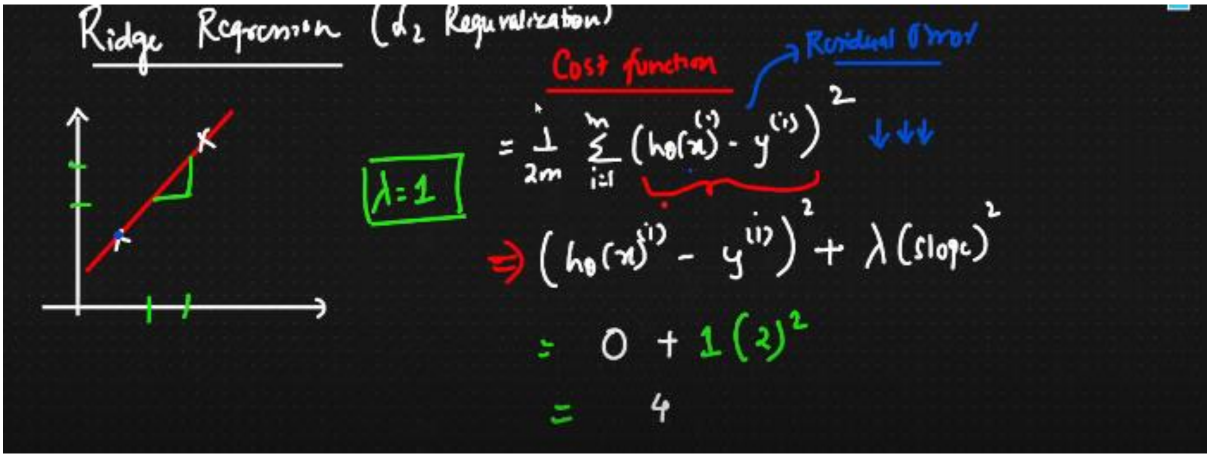
Case when train accuracy is very high (~90%), test accuracy is decreased to around 70% where model is biased towards training data is called **overfitting**. Case when train accuracy is very low around 60%, test accuracy is also around 62% is called **under-fitting**.

High training accuracy & Reduced test accuracy is called **low bias & high variance**.

Low training accuracy & low test accuracy is called **high bias & high variance**.

Our target model should have **low bias & low variance**.

To prevent the case of overfitting, we use Ridge Regression (L2 Regularisation).



Ridge Regression (L_2 Regularization)

Cost function

Residual Error

$$\boxed{\lambda = 1} \qquad = \frac{1}{2m}\sum_{i=1}^{m}\underbrace{(h_\theta(x)^{(i)} - y^{(i)})}^{2} \quad \downarrow\downarrow\downarrow$$

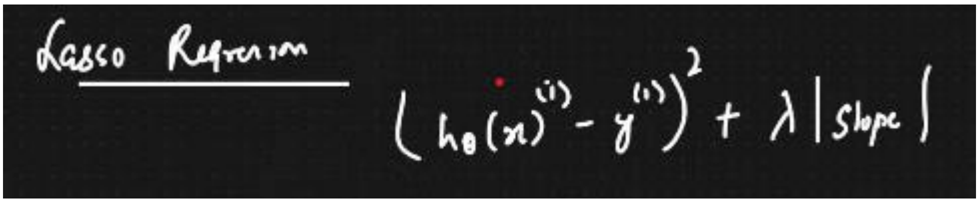
$$\Rightarrow (h_\theta(x)^{(i)} - y^{(i)})^2 + \lambda(\text{slope})^2$$

$$= 0 + 1(2)^2$$

$$= 4$$

Ridge regression (L2 regularization) introduces lambda & slope into our cost function equation. Ridge regression only reduces the coefficients close to zero but not zero. The slope penalizes the residual error in a way to avoid overfitting.

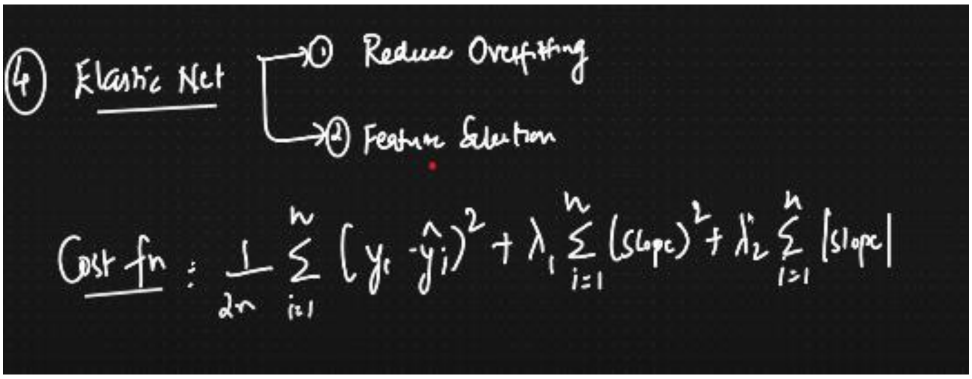
Lasso regression (L1 regularization) also has introduction of lambda & slope. But the only difference is in the formula where we have modulus of slope. This avoids overfitting and also helps in feature selection. Lasso regression can reduce coefficients of some features to zero, thus resulting in better feature selection.



Lasso Regression

$$(h_\theta(x)^{(i)} - y^{(i)})^2 + \lambda|\text{Slope}|$$

Elastic Net Regression



④ Elastic Net

① Reduce Overfitting

② Feature Selection

Cost fn : $\frac{1}{2n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{i=1}^{n} (slope)^2 + \lambda_2 \sum_{i=1}^{n} |slope|$

This is a combination of Ridge and Lasso regression where we aim to reduce overfitting and improvise feature selection too.