Date:- 9th October, 2022

14

MACHINE LEARNING (Day-2)

Agenda :-

1) Revision of Day-1
2) Cost Function
3) Loss Function
4) Performance Metrics
5) Overfitting and Underfitting

1) Revision of Day-1 :-

*) Linear Regression Algorithm :-

a) Simple Regression

$h_\theta(x) = \theta_0 + \theta_1 x$

(b) Multiple Regression

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$

*) Convergence Algorithm :-

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

Predicted Value

Actual Value

• Loss Function = $\left(h_\theta(x)^i - y^{(i)}\right)^2$

$\boxed{= (\hat{y}_i - y_i)^2}$

Predicted Value

Actual Value

Scanned with CamScanner

*) For Convergence Algorithm :-

Repeat Until Convergence

$$\{$$

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

$$\}$$

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

$$h_\theta(x) = \theta_0 + \theta_1 x \quad \text{—— Eqn-no } (1)$$

Let,
$J = 0$;

$$= \frac{\partial}{\partial \theta_0} \left[\frac{1}{m} \sum_{i=1}^{m} \left((\theta_0 + \theta_1 x)^{(i)} - y^{(i)} \right)^2 \right] \rightarrow \text{Using Eqn. no. } (1)$$

$$= \frac{2}{m} \sum_{i=1}^{m} \left[(\theta_0 + \theta_1 x)^i - y^{(i)} \right] \times 1$$

Let,
$J = 1$;

$$= \frac{\partial}{\partial \theta_1} \left[\frac{1}{m} \sum_{i=1}^{m} \left((\theta_0 + \theta_1 x)^i - y^{(i)} \right)^2 \right]$$

$$\Rightarrow \frac{2}{m} \sum_{i=1}^{m} \left[(\theta_0 + \theta_1 x)^i - y^i \right] x$$

Repeat Until Convergence

{

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

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

}

2) Cost Function :–

⇒ A Cost Function is an Important Parameter that Determines how well a ML Model performs for a given Dataset.

It's ~~Calculated~~ ~~the~~ measures the Error between Predicted and their Actual Values Across the Whole Dataset.

*) Different Types of Cost Function are :–

1) MSE
2) MAE
3) RMSE

(A) MSE (Mean Squared Error) :-

⇒ MSE (Mean Squared Error) represents the Average of the Squared Different between the Original and Predicted Values in the dataset. Its Measures the Variance of the residuals.

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

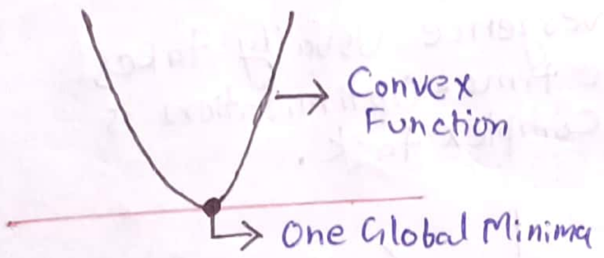
(Quadratic Equation)

Here, $\hat{y} = \theta_0 + \theta_1 x$ (Predicted Value)

Advantages :-

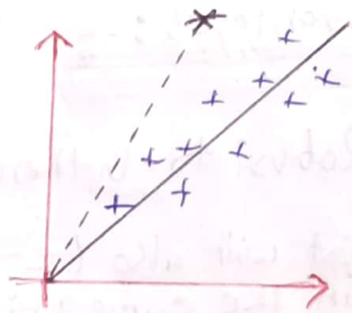
i) This Equation is differentiable.

ii) This Equation also has only one - Global Minima.



Disadvantages :-

① This Equation is not robust to outliers.



Here,
Because of outlier the line will switch more towards outliers.
So, we can Remove Outliers.

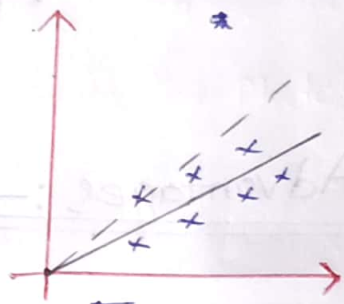
② Penalizing the error or Changing the Unit.

Note :- Our Work is to work with Convex-function because there is not any local minima Present.

2) MAE (Mean Absolute Error) :-

⇒ MAE (Mean Absolute Error) represents the Average of the absolute different between the Actual and Predicted Values in the dataset.
It measures the Average of the residuals in the Dataset.

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



It will not more attracted to outlier Because here $(y-\hat{y})$ is not Squaring.

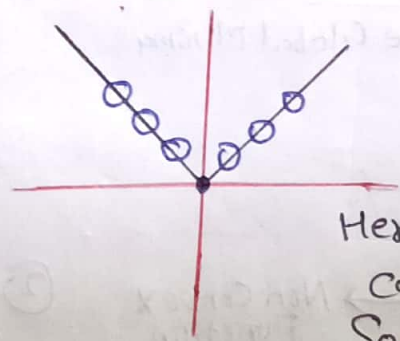
Advantages :-

① Robust to Outliers.

② It will also be in the same unit.

DisAdvantages :-

① Convergence Usually takes more time - Optimization is a complex task.



Here, At origin we can't do deriviative So, we will divide Part by Part and do deriviative, this is called Subgradient Concept.

② Time Consuming.

3) RMSE (Root Mean Squared Error):-

⇒ RMSE (Root Mean Squared Error) is the Square Root of Mean Squared Error. Its measures the Standard-Deviation of residuals.

$$RMSE = \sqrt{MSE}$$

Advantages:-

i) It is not Robust to Outliers.
ii) It is differentiable.

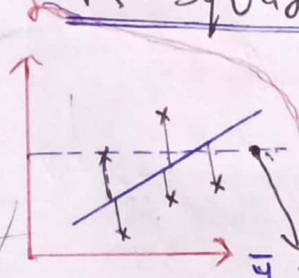
⊛ **Loss Function:-** MAE, MSE, RMSE, Huber Loss.

⊛ R-Squared [Performance Matrix]:-

⇒ R-Squared (or The Coefficient of determination) Represents the proportion of the variance in the dependent Variable which is explained by the Linear Regression Model. It is a Scale-Free Score i.e irrespective of the values being small or large, the value of R-Square will be less than one.

$$R\text{-Squared} = 1 - \frac{SS_{Res}}{SS_{Total}},$$

Where,
SS_{Res} = Sum of Square residual
SS_{Total} = Sum of Square Average



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

$\bar{y}$ = Average (Basic Model)

If the model is fitted, then $= 1 - \frac{\text{Small Number}}{\text{Bigger Number}}$

↓ outcome

If; R-Squared = 0.85 mean 85% accurate n
If, R-Squared = 0.75 mean model is 75 % accurate.

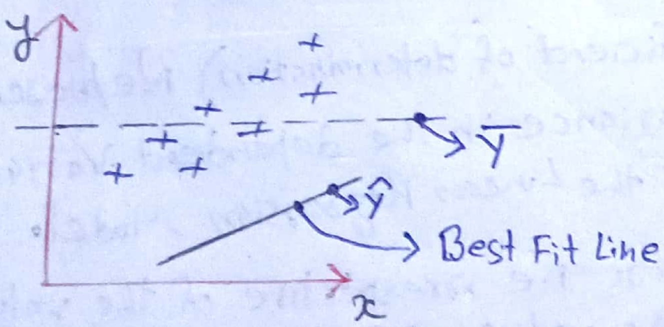
∴ R-Squared measures the Performance of the Model.

Q) Can R-Squared value -ve ?

⇒ If it is happening then the model is very bad.

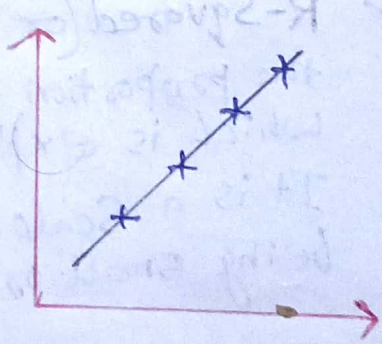
Scenario-1 :-

If, $R^2 = -ve$



Scenario-2 :-

If, $R^2 = 1$



In this case $R^2 = -ve$
Worth Model
here, $\bar{y}$ is better than best Fit Line.

2) Adjusted R-Squared :-

⇒ Adjusted R-Squared is a modified Version of R-Square, and it is adjusted for the number of Independent Variables in the model, and it will always be less than or Equal to R^2.

Dataset :-

Size of House	City Location	No. of Bedrooms	Gender	Price
—	—	—	—	—
—	—	—	—	—
—	—	—	—	—
—	—	—	—	—

65 %
75 %
85 %

By Including Features Accuracy is jumping, But Gender Feature is not Correlating to Price Prediction.

So, here we have to handle with these accuracy.

$$\boxed{\text{Adjusted } R^2 = 1 - \frac{(1-R^2)(N-1)}{N-P-1}}$$

where,
N = No. of Data Points
P = No. of Independent Features.

~~Here~~ By this formula:
if the Feature is not correlated with target Feature then it will not increase the accuracy, and we will able to adjust the Value.

5) Overfitting And Underfitting [Bias and Variance]:

Dataset ⟹

Dataset

Exp	Year

1000 Datapoints

↓ 1st we divide the dataset.

Training (700 Datapoint) [Model Training]

Test (300 Datapoint) [Model Test]

★) (700 Records)

Training Dataset

500 → Train ↓ (Train the Model)

200 → Validation ↓ (Hyper-Parameter Tunning the model)

Model :-

Train Data | Very Good Accuracy (90%) → [Low Bias]
Test Data | Very Good Accuracy (85%) → [Low Variance]

∴ This Kind of accuracy creates **Generalized Models.**

Scenario – 1 :-

a) For Training Data :- We have Very Good Accuracy (90%).

⇒ In this Case, it will come Low Bias.

b) For Test Data :- We have Bad Accuracy (50%).

⇒ In this case, it will come High Variance.

Hence, For Such Type of Model where Training Data is very Good and has Bad Test Data then, this Condition is called **Overfitting.**

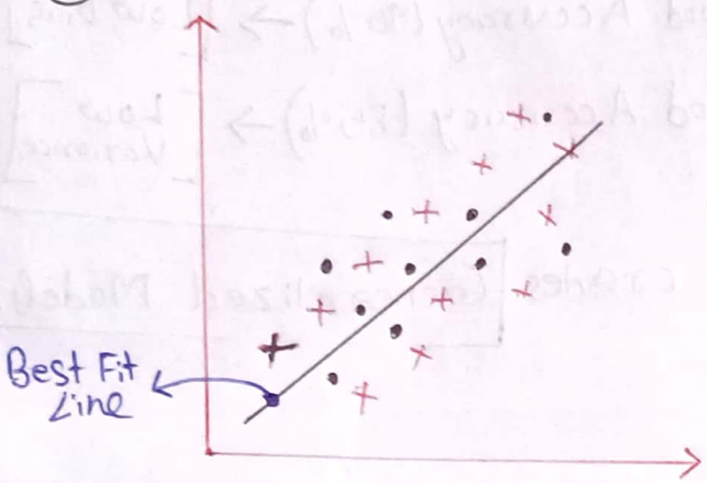
Scenario – 2 :-

Train Data | Model Accuracy is Low → [High Bias]
Test Data | Model Accuracy is Low/High → [Low or High Variance]

∴ This Type of Accuracy Model is **Underfitting.**

*) Graphical Representation of Model :-

(1)



⇒ Here,

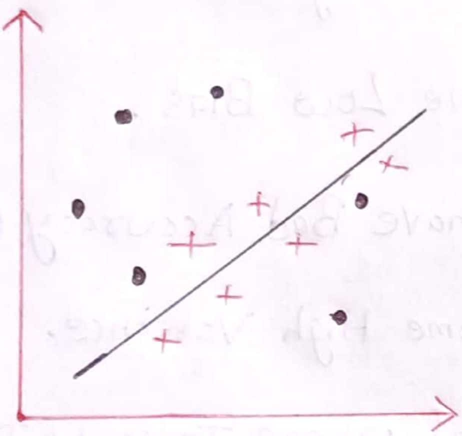
× ● → Training Data

● ■ → Test Data

This is Generalized Model

(Best Fitting)

(2)



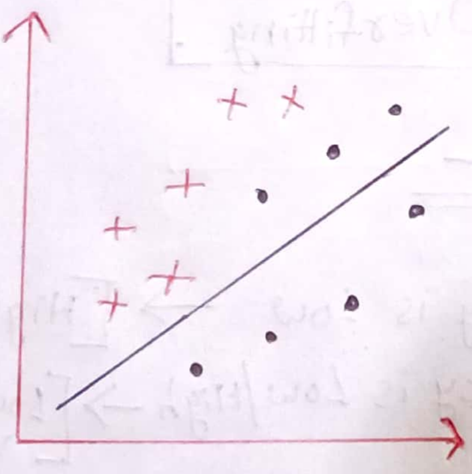
⇒ Here,

× → Training Data

• → Test Data

∴ This is Overfitting Model.

(3)



⇒ Here,

× → Training Data

• → Test Data

∴ This is Underfitting Model.