ML - Day 2

Linear Regression Algorithm

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

→ **Convergence Algorithm**

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

$h_\theta(x)^{(i)}$ → Predicted Pts. $\quad y^{(i)}$ → Truth Points

→ **Cost Function Vs Loss Function** → Every observation

The Loss function is to capture the difference b/w the actual and predicted values for a single record.

Cost function aggregates the difference b/w actual and predicted points for the entire training dataset.

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

$\Rightarrow \left(\hat{y}(i) - y(i) \right)^2$

$\hat{y}(i)$ → Predicted pts. $\quad y(i)$ → Actual pts.

→ Calculating the slope

$$\theta_j : \theta_j - \alpha \left[\frac{d}{d\theta_j} J(\theta_j) \right]$$

derivative

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

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

for $j = 0$

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

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

for $j = 1$

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

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

Repeat until Convergence

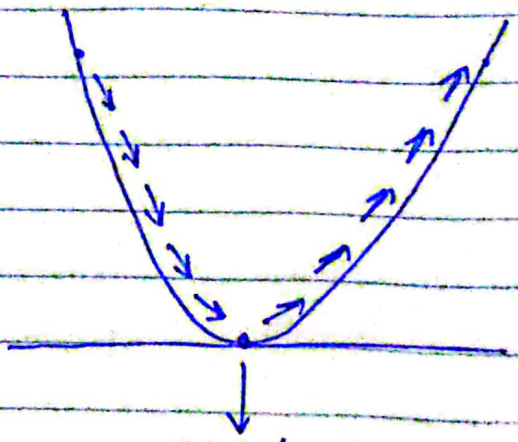
{

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

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

}

α = Speed of Convergence



Cost Functions

1. Mean Squared Error (MSE)
2. Mean Absolute Error (MAE)
3. Root Mean Squared Error (RMSE)

① MSE (Mean squared Error)

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

↳ Quadratic Eqn → $(a-b)^2$ ↓ $a^2 - 2ab + b^2$ ↓ $ax^2 + by + c = 0$

Here $\hat{y} = \theta_0 + \theta_1 x$

↳ predicted values.

~~when we~~

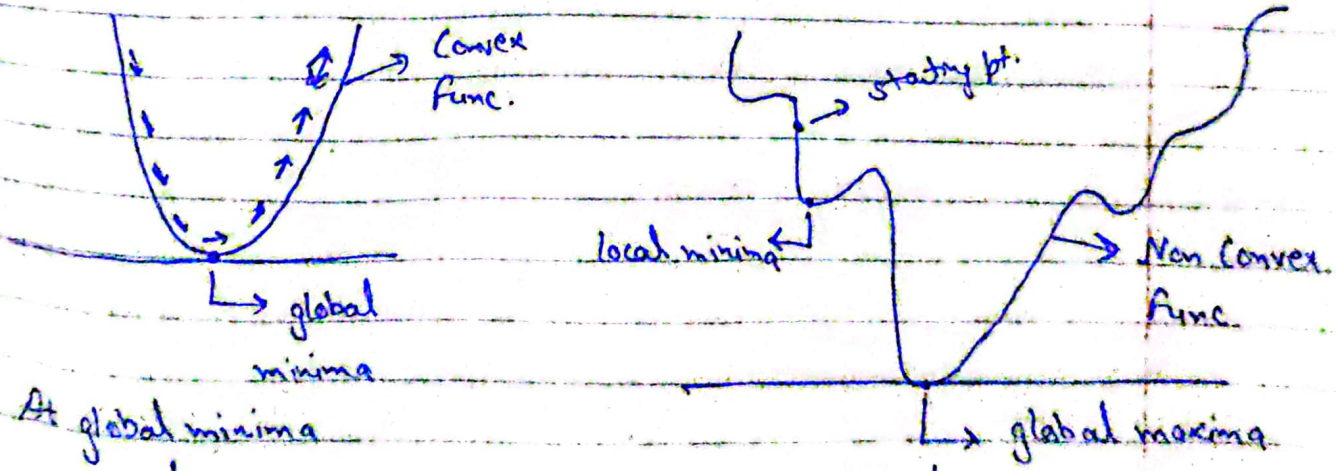
→ Advantages of MSE

① This equation is differentiable.

② The equation has only one global minima.

⇓

When we plot quadratic eqn we get a parabola



At global minima

↓

slope = 0

In this case at local ~~minima~~ minima

↓

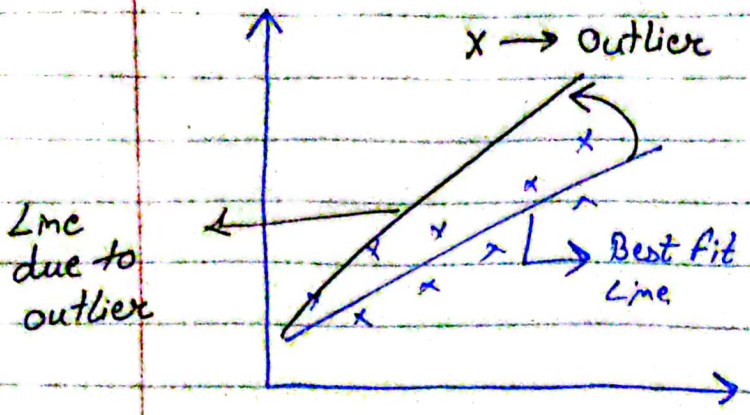
slope = 0, so convergence will not go upto global minima

⇓

we won't be able to reduce Cost function.

→ Disadvantages of MSE

① This is not robust to outliers.



Because of this outlier the cost function will increase by a huge amt

⇓

Best fit line will get updated by a big margin

In order to solve this problem we need to remove the outliers.

② Penalizing the error → Changing the Unit

Let say we have a data

Indepen. ← Exp.	Salary (Lakhs inr)

$(y - \hat{y})^2$

↓

$(\text{salary} - \text{pred. salary})^2$

$(\text{Error})^2$ ←

↓

penalized

↓

increasing the order

⇓

As the units for salary is Lakhs so when we square the difference we are also squaring Lakhs

⇓

$(\text{Lakhs})^2$ → Unit changes

② Mean Absolute Error (MAE)

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

→ Advantages of MAE

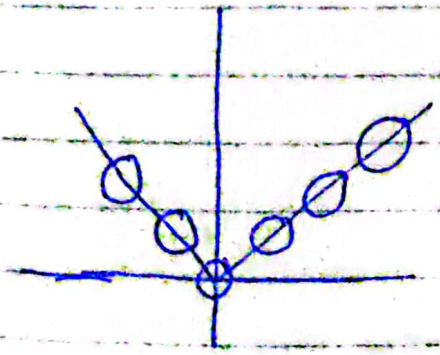
① Robust to Outliers

② It will also be in the same unit.

→ Disadvantages of MAE

① Convergence usually takes more time. Optimization is a complex task.

for MAE → we cannot find derivative at 0.
↓
To find derivatives we use Sub-Gradient concept.



② Time consuming.

③ Huber Loss :-

→ Huber Loss is a combination of MSE and MAE.

→ We use Huber loss anytime we need a balance b/w giving outliers some weight, but not too much.

⇓

for cases, where outliers are very important or there are not outliers we use MSE.
for cases, where we don't care at all about the outliers, we use MAE.

$$\text{Huber loss} = \begin{cases} \frac{1}{2}(y-\hat{y})^2, & |y-\hat{y}_p| \le \delta \\ \delta|y-\hat{y}| - \frac{1}{2}\delta^2, & |y-\hat{y}| > \delta \end{cases}$$

Hyperparamter (delta), δ is used to switch two error functions.

→ Advantages of Huber Loss

ⓐ Outliers are handled properly.
ⓑ Local minima situation is handled here.

→ Disadvantages of Huber Loss

ⓐ It is complex.
ⓑ In order to maximize model accuracy, the hyperparameter δ will also need to be optimized which increases training requirements.

④ Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{MSE}$$

→ Advantages of RMSE

It will also be in the same unit.

→ Disadvantages of RMSE

Not robust to outliers.

Performance Metrices

⇓

Helps to check the performance of the model.

① R-Squared :-

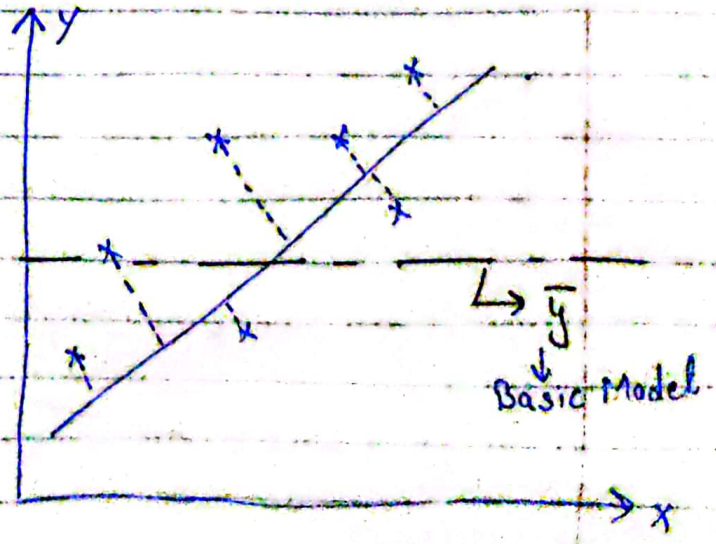
$$R\text{-Squared} = 1 - \frac{SS_{Res}}{SS_{Total}}$$

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

SS_{Res} = Sum of Square Residuals

SS_{Total} = Sum of Square Average

$\bar{y}$ = Average of y



→ If model is fitted well → SS_{Res} → Low & SS_{Tot} → High

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

$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ → Low value

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

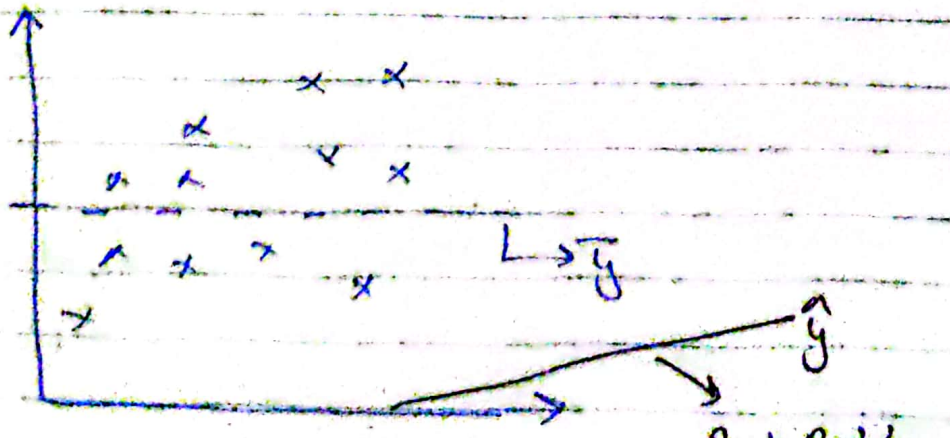
$= 1 - \frac{\text{Small Num}}{\text{Bigger Num}}$ → Small Number

$\Rightarrow$ **R-Squared < 1**

Let R-Squared = 0.85 → 85% accurate

R-Squared measures the performance of model

→ * If R-Squared value is -ve, then the model is very bad.

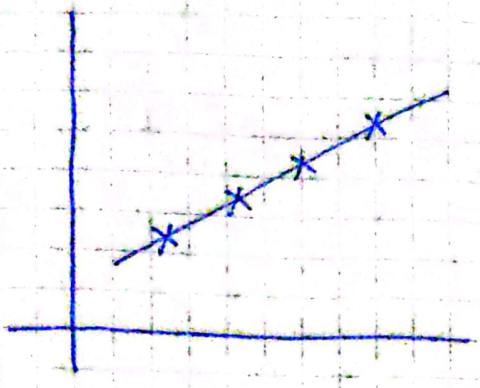


Best Fit Line

⇓

R-Squared is -ve

→ $R^2 = 1$



② Adjusted R-Squared

Let say we have a dataset

Size of House	City Location	No. of Bedrooms	Gender	Price

Here, size of house, city location and num of bedrooms are highly correlated with the Price

Gender is <u>not</u> highly correlated with Price

Now

ⓐ 2 features only Size & Price
$R^2 = 65\%$

ⓑ Add a feature to dataset, say city location
$R^2 = 75\%$

ⓒ Add another feature, no. of bedrooms
$R^2 = 88\%$

} when features are highly correlated, the accuracy will increase rapidly

ⓓ Add gender column
$R^2 = 90\%$

} Accuracy will be very less increased.

⇓

Because there is no direct correlation

Here the accuracy should not have increased. To solve this problem we use adjusted R^2.

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

N = No. of Data Points

P = No. of Independent Features.

~~when P~~

when,

$P=1$, $R^2 = 65\%$ $\longrightarrow$ Adj $R^2 = 63\%$

$P=2$, $R^2 = 75\%$ $\longrightarrow$ Adj $R^2 = 73\%$

$P=3$, $R^2 = 88\%$ $\longrightarrow$ Adj $R^2 = 87\%$

$P=4$, $R^2 = 90\%$ $\longrightarrow$ Adj $R^2 = 83\%$

↓

when we are adding Gender to our dataset, which is not correlated to Price column

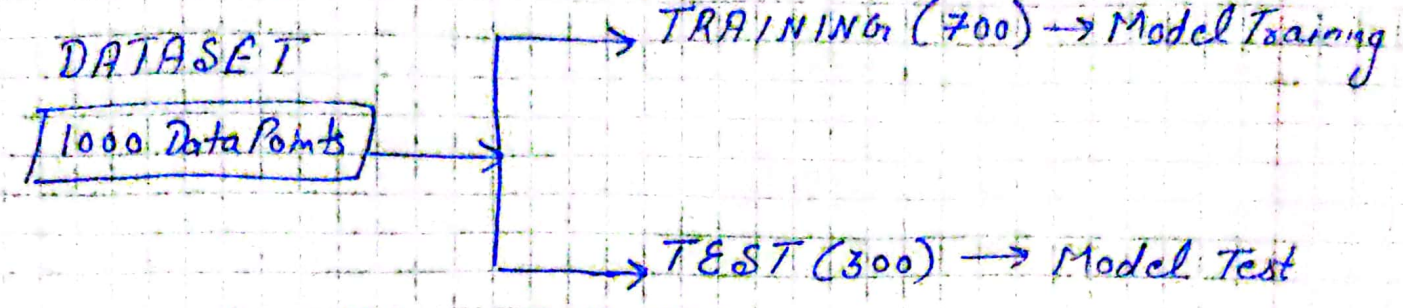
↓

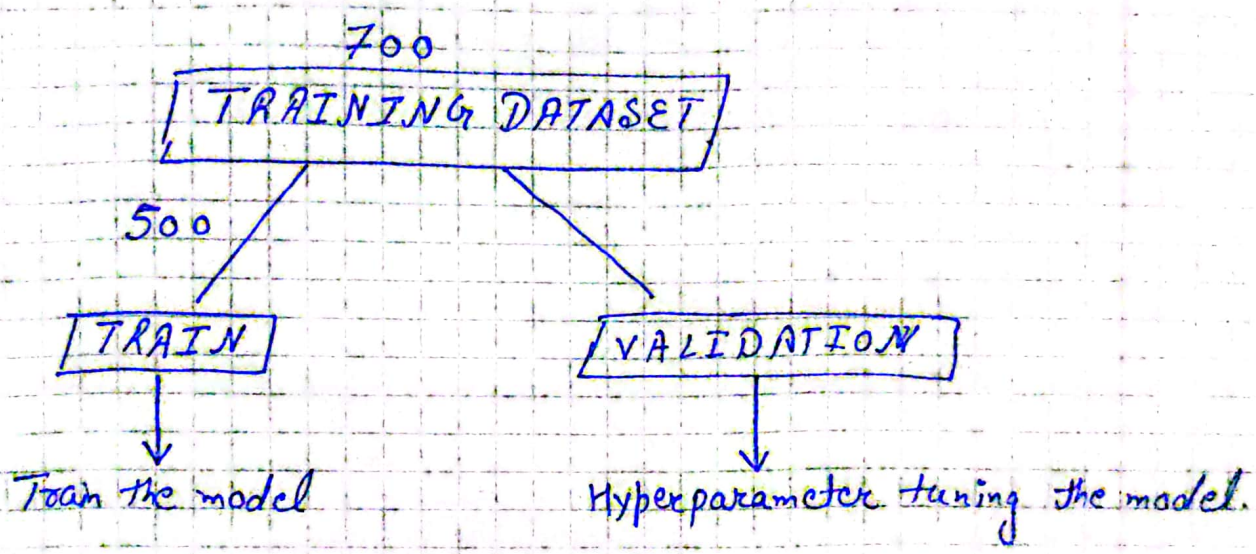
R^2 is increasing but adj R^2 is decreasing

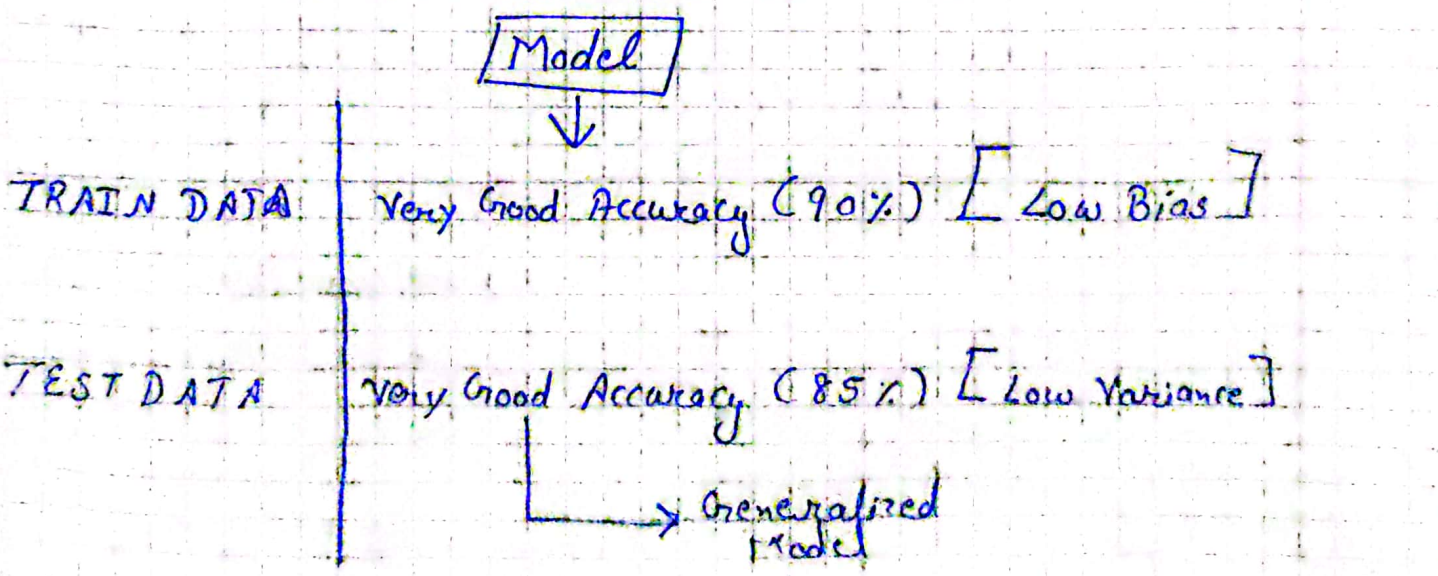
→ Adjusted R^2 will measure the performance of model based on very important features.

Overfitting & Underfitting

(Bias and Variance)







* Aim - Our model should be able to get good training and good test accuracy.

→ | TRAIN | Very Good Accuracy (90%) [Low Bias]
| TEST | Bad Accuracy (50%) [High Variance]

⇓

[Overfitting]

→ To solve overfitting problem, we need to perform Hyperparameter Tunning

or

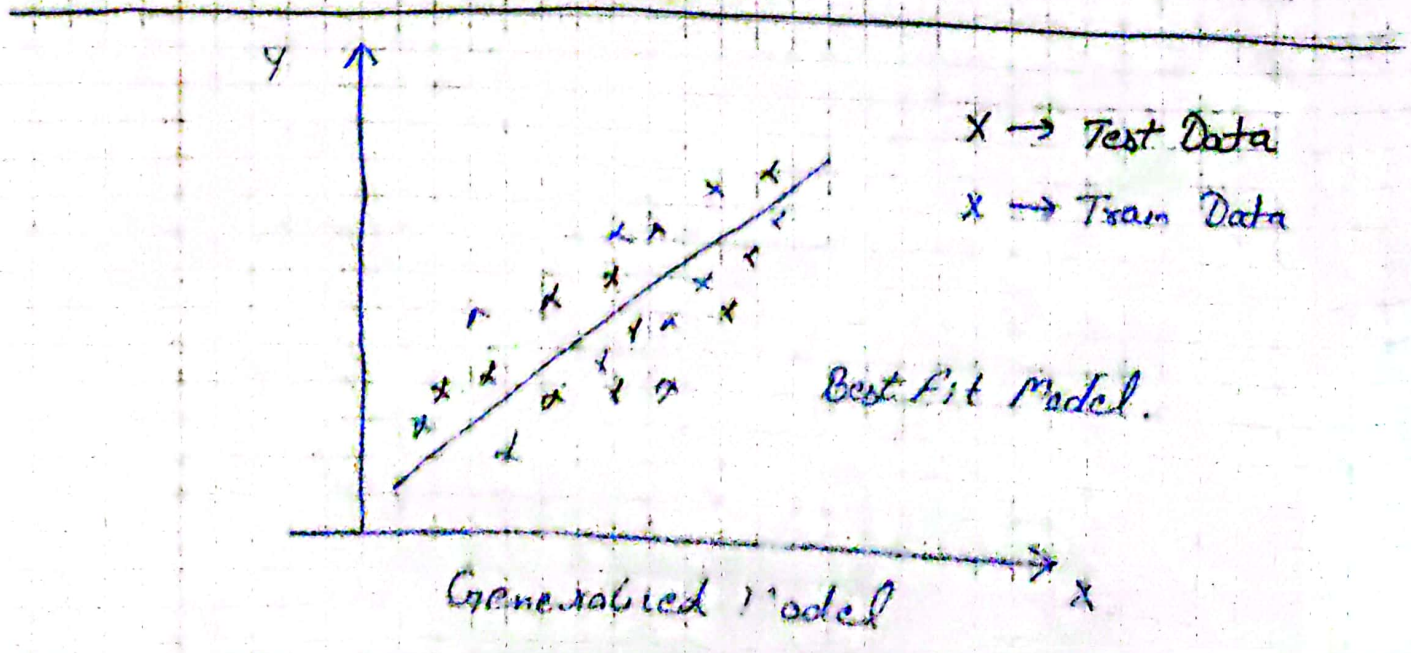
we can increase the data.

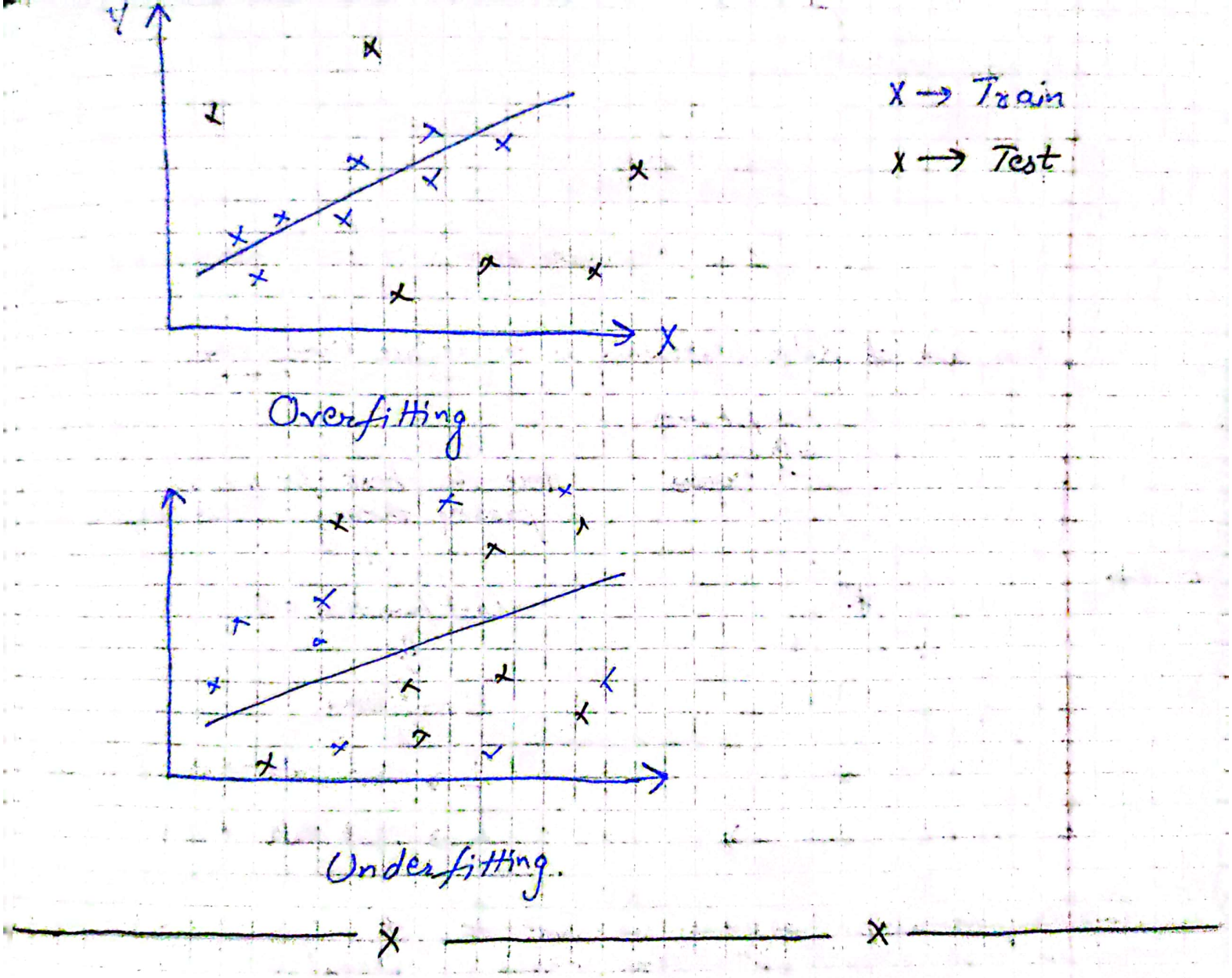
→ Now,

| TRAIN | Model accuracy is low [High Bias]
| TEST | Accuracy is Low/High [Low or High Variance]

⇓

[Underfitting]




X → Train
X → Test
Overfitting
Underfitting.