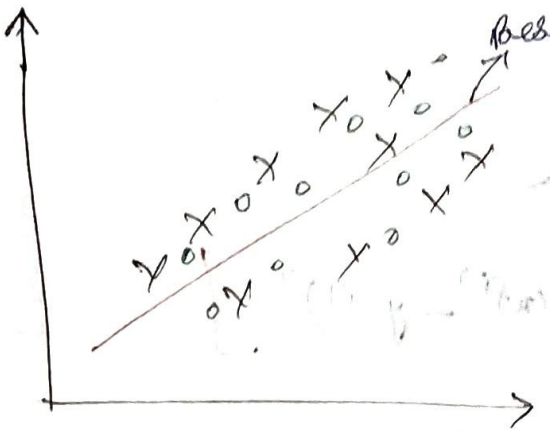
Underfitting

Train Data → Low Accuracy → [HIGH BIAS]

Test Data → High Accuracy / Low Accuracy → [HIGH / LOW VARIANCE]

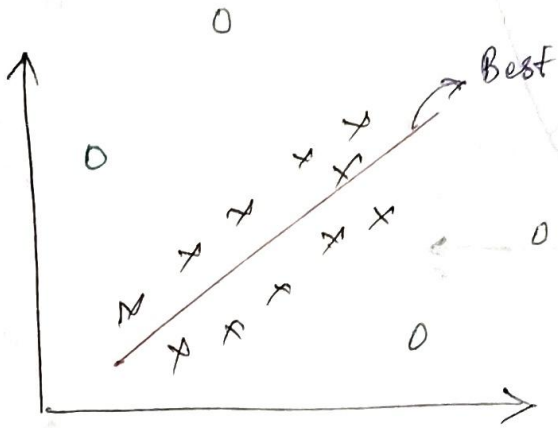
Generalize Model



X → Training Data
O → Test Data

Prediction will be good.

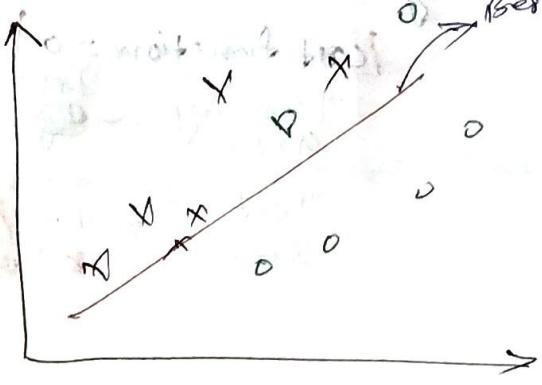
Overfitting Model



X → Training Data
O → Test Data

Prediction will not be good.

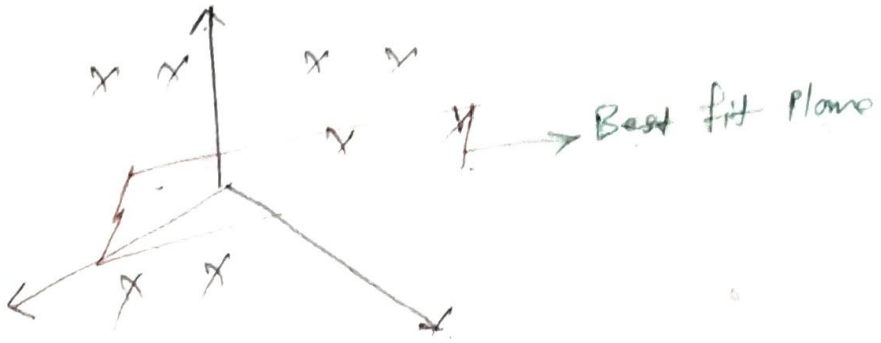
Underfitting Model



X → Training Data
O → Test Data

Prediction will not be good.

3-D



More than 3-D → Hyperplane.

09-OCT-2022 M2 - Day-2

Convergence Algorithm

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

Difference b/w cost function and loss function

Cost function

- We find error for all the points and do average

Loss function

- We find error for observed points and if we find error for all the points, then it is cost function.

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

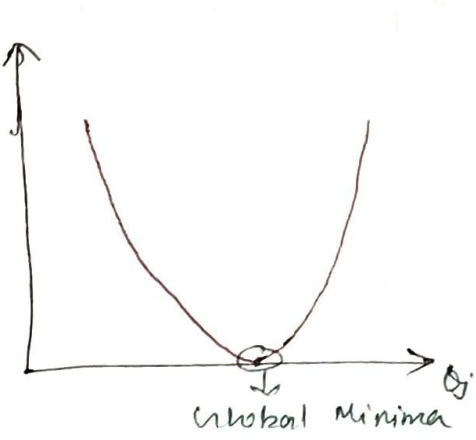
↓ Predicted value — Actual value

Repeat until convergence,

{

$$\theta_j = \theta_j - \alpha \boxed{\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_j}}$$

}



→ To achieve global minima.

Derivative w.r.t θ_0, $\boxed{j=0}$

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

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

↓ Just for derivative purpose.

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

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

$$\left\{\frac{\partial \theta_0}{\partial \theta_0} = 1\right\} \quad \left\{\frac{\partial \theta_1 x}{\partial \theta_0} = 0\right\} \quad \left\{\frac{\partial (y)^{(i)}}{\partial \theta_0} = 0\right\}$$

Derivative w.r.t θ_1, $\boxed{j=1}$

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

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

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

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

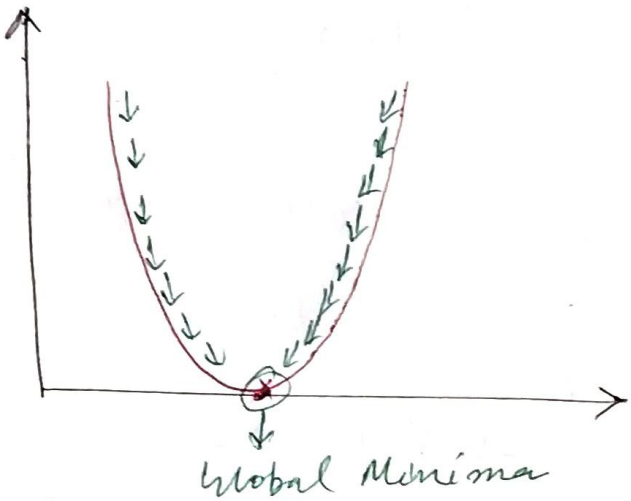
Repeat until convergance,

{

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

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

}



Learning Rate: Speed of convergance.

Types of lost function

① MSE — Mean Squared Error

② MAE — Mean Absolute Error

③ RMSE — Root Mean Squared Error.

① MSE

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

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

$\hat{y}$ → Predicted

Advantages

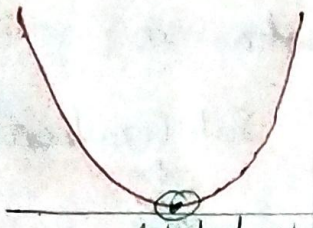
① This equation is differentiable, we can calculate θ_0 and θ_1.

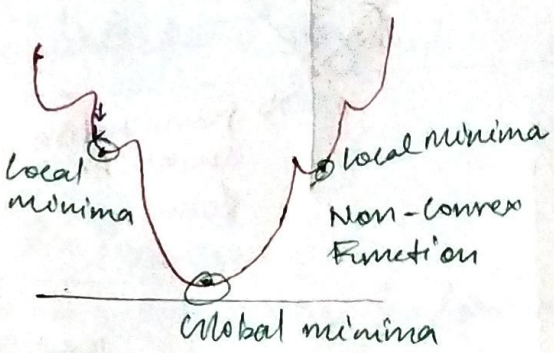
② This equation also has only one global minima.

Disadvantages

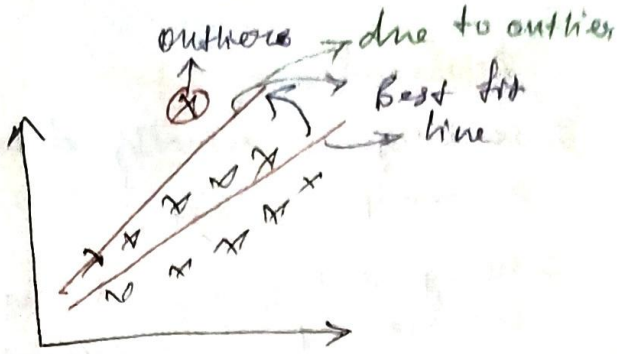
① This equation is not robust to outliers.

convex Function





③ This equation has convex function.



→ Best fit line gets updated with huge margin in the presence of outliers.

Independent dependent

② Exp Salary (Lakhs INR) $(y-\hat{y})^2$ $(lakhs)^2$

Time complexities need increased ← unit changing

we don't do scaling for dependent feature.

$(y-\hat{y})^2$ → Squared → Error → Penalized.

↓ Increased

② MAE

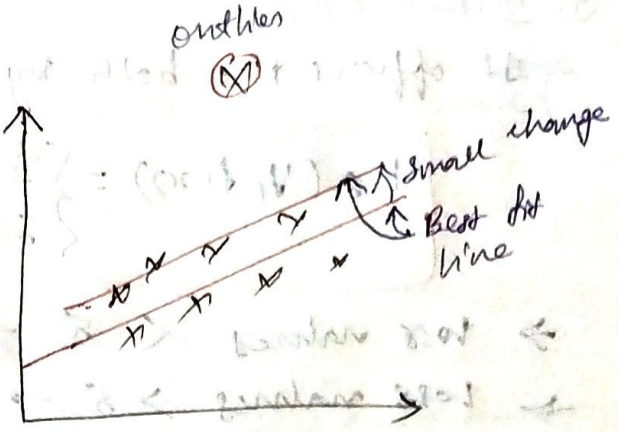
$$MAE = \frac{1}{m}\sum_{i=1}^{m} |y-\hat{y}|$$

$\hat{y}$ = Do-I DIR

Predicted.

Advantages

① Robust to outliers.

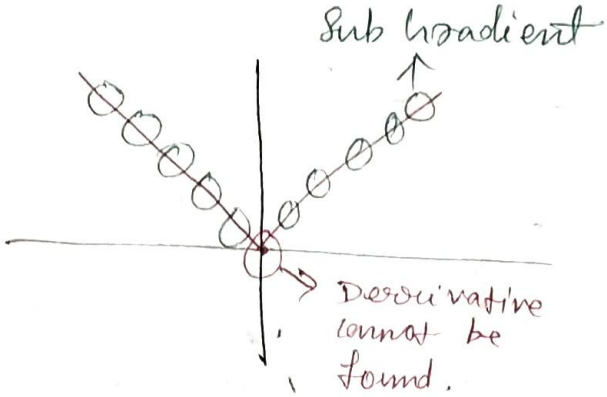


Disadvantages

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

- Using sub-gradient concept, we can find derivative.

② Time Consuming..



Huber Loss

③ RMSE

$$RMSE = \sqrt{MSE}$$

Advantages

① It is differentiable.
② Unit will be same.

Disadvantages

① Not robust to outliers

④ Huber Loss

- It offers the both by balancing MSE and MAE. together.

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{, for } |y - f(x)| \le \delta, \\ \delta |y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}$$

→ loss values $< \delta$ → Use MSE

→ loss values $> \delta$ → Use MAE

NOTE: Use the Huber loss any time you feel that you need a balance b/w giving outliers some weights but not too much.

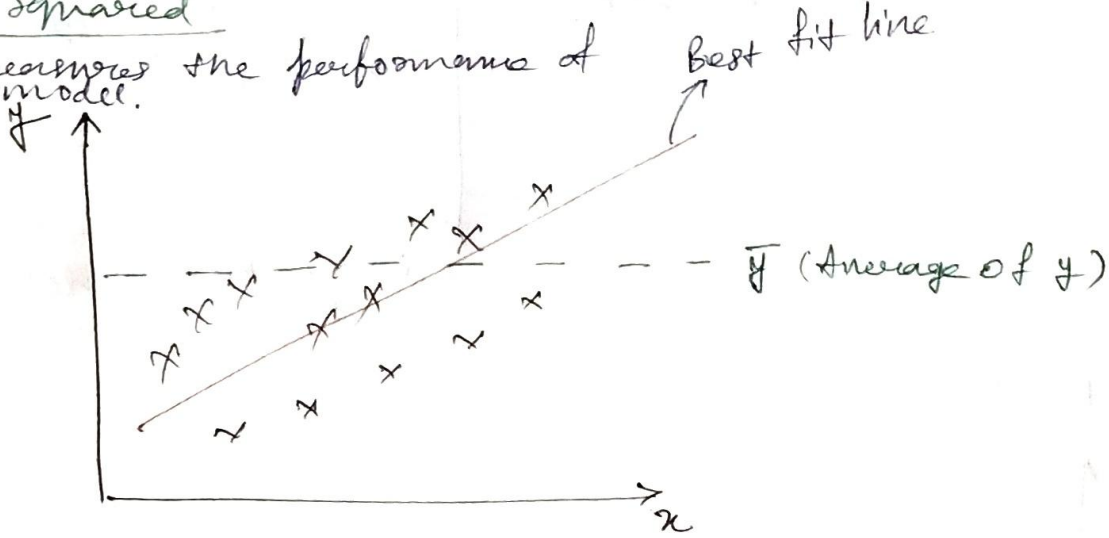
How can we check if a model is good or not?
- Using performance metrics

Performance Metrics

1. R-Squared
2. Adjusted - R Squared

1. R-Squared

→ It measures the performance of the model.



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

SS_{Res} = Sum of Square Residuals

SS_{Total} = Sum of Square Average

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

→ Low value

→ High value

} for good model

$\bar{y}$ = Average of $\bar{y}$

$$= 1 - \left\{ \frac{\text{Small number}}{\text{Bigger number}} \right\}_{\text{Small number}}$$

$$\text{R-Squared} \leq 1$$

If R-Squared = 0.85 → 85% Accurate

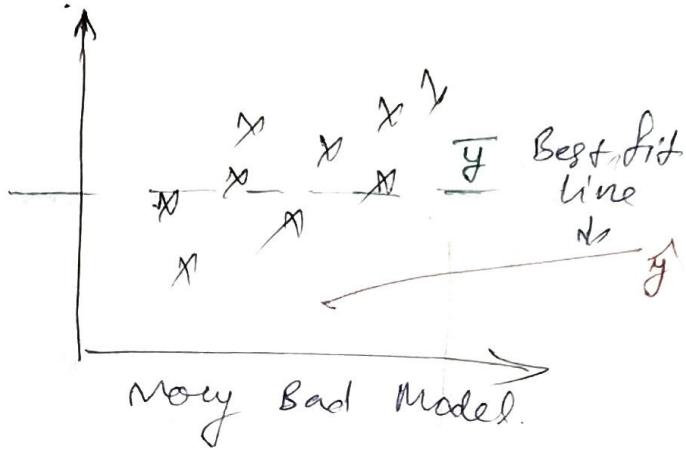
R-Squared = 0.75 → 75% Accurate

Que Can R-squared value be negative?

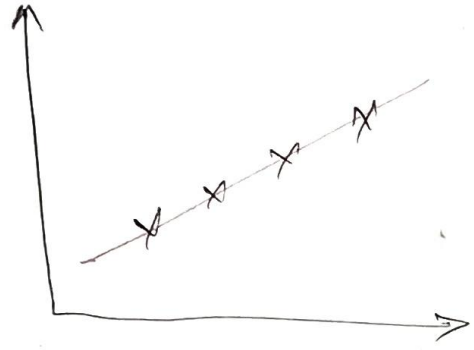
Ans if it is '-ve', then our model will be very bad.

Here $(y_i - \hat{y}) > (y_i - \bar{y})$

then, R-squared will be '-ve'.



If $R^2 = 1$



② Adjusted R-Squared

Size of House | City location | Price | No. of Bedroom | Salary of the owner

No direct correlation between Price and Bedroom

→ Earlier only size of house was present as an independent feature to get the price and has some R-Squared value. But with the inclusion of city location, R-Squared value got increased. After that again with the inclusion of city location & no. of bedroom and hence the value got increased. But as there is no direct correlation b/w bedroom and price the value shouldn't increased.

R^2	features inclusion	Adjusted R^2	Independent features
65%	Size of House	63%	P=1
75%	City location	73%	P=2
88%	No. of Bedrooms	85%	P=3
90%	Gender. → No direct correlation w.r.t price.	85% (decreased)	P=4

now slight increase, but this shouldn't happen. To solve this problem, we use Adjusted R-Squared.

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

N = No. of data points
P = No. of independent features.

→ Adjusted R^2 is the better metrics to evaluate the model.

Overfitting and Underfitting (Bias and Variance)

Dataset

10000 datapoints

Training (7000) → Model Training to be done with training dataset.

Test (new dataset) (3000) → Model Test

Exp	Year

* Training Dataset (1000)

↓ TRAIN (500) → used for training of the model

↓ VALIDATION (500) → used for hyperparameter tuning of the model

MODEL

Generalize Model

TRAIN DATA → Very Good Accuracy (90%)

TEST DATA → Very Good Accuracy (85%)

} our aim is to get good train and test accuracy.

Train Data → very good accuracy (90%) → [LOW BIAS]

Test Data → very good accuracy (85%) → [LOW VARIANCE]

Train Data → Low accuracy → [High BIAS]

Test Data → Low accuracy → [High Variance]

Overfitting { Ratta maarna }

Very Good

Train Data → Very Good Accuracy (90%) → [LOW BIAS]

Test Data → Bad Accuracy (50%) → [HIGH VARIANCE]

→ We can solve this by using by performing hyperparameter tuning or by increasing the no. of dataset.