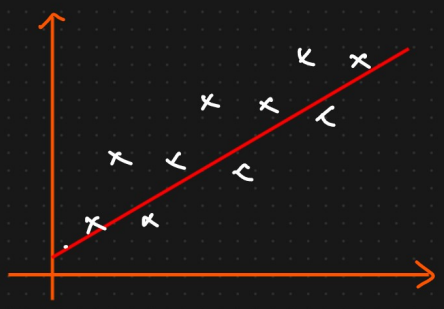
Regression

1. Mean Square Error (MSE), MAE, RMSE ✓
2. Ridge Regression ✓
3. Lasso Regression ✓
4. ElasticNet ✓
5. Practicals. [Simple Implementation]

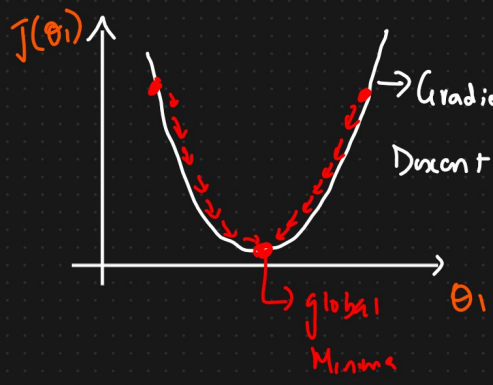
Many regression models rely on distance metrics to determine the convergence to the best result. Even the definition of a "best" result needs to be explained quantitatively by some metric(COST FUNCTION).

Usually the metrics used are the Mean Average Error (MAE), the Mean Squared Error (MSE) or the Root Mean Squared Error (RMSE).

Distance calculated by taking difference of predicted and actual value should always be +ve.
Distance obtained by difference can be +ve either:
1. taking sqaure=MSE
2. taking absolute difference=MAE
3. taking sqare root of squared quantity=RMSE

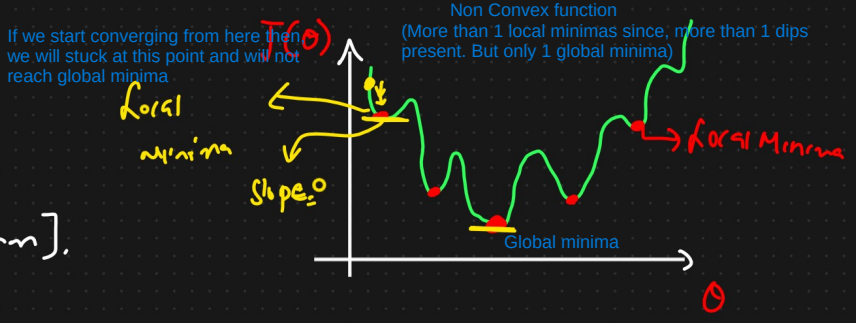


1. Mean Squared Error (MSE)
2. MAE
3. RMSE



MSE

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



MSE used while determining cost function

1. Mean Square Error [Cost function].

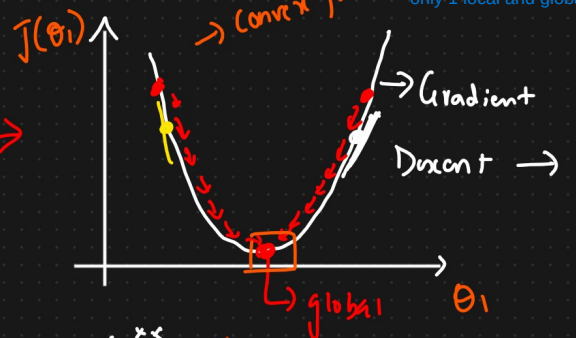
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \Rightarrow \text{Quadratic function.}$$

Parabolic function → Convex function

Convex fun^: only 1 local and global minima

MSE is a quadratic function(also, called convex function). Below is the general representation of this equation

$$(a+b)^2 = a^2 + 2ab + b^2$$



$ax + by + c = 0$

$by = -ax - c$

$y = \boxed{\frac{-a}{b}}\,x + \boxed{\frac{-c}{b}}$

$\downarrow \quad\quad \downarrow$

$mx + c$

Below is representation of a straight line

$$ax + by + c = 0$$

$$\Downarrow$$

$$y = mx + c$$

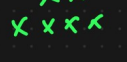
$$(y_i - \hat{y}_i)^2$$

INR $(INR)^2$

US $(US)^2$ ERROR↑↑

MSE

MSE $(y_i - \hat{y}_i)^2$ ↑↑↑



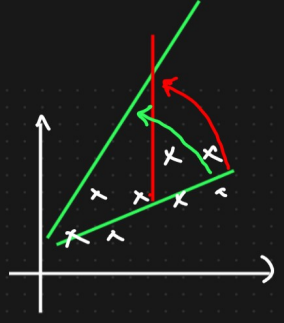
Advantages

1. It is differentiable.
2. It has one local and one global minima.

Not robust to outliers means best fit line gets affected by outlier. Shift of best fit line is affected in huge way since we are squaring the difference. Ideally best fit line should move in small increments with varying data points but due to presence of outliers this shift increments drastically. Best fit line should shift in order to accommodate maximum data points.

Disadvantages

1. Not Robust to outliers {Affected by outliers}



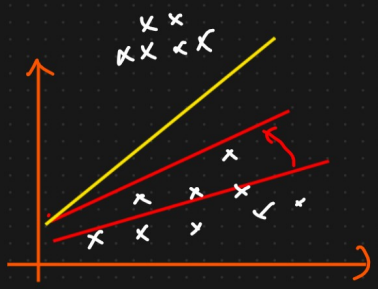
2. MSE changes the unit

Since, in this we perform squaring operation. So, if unit is in cm for a feature it gets converted into cm square.

2. Mean Absolute Error (MAE)

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

Price	↓Model Predicted price	$(INR)^2$ $(y_i-\hat{y}_i)^2$
100	120	400



ERROR ↑

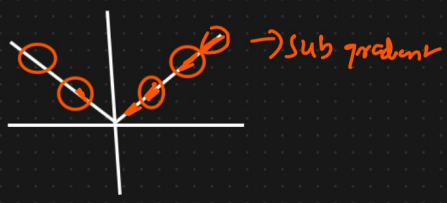
Advantage

Since, here we are not squaring the differences, there will be shift in movement of best fit line in case outliers but this movement will be less as compared to MSE.

1. Robust to outliers
2. It will be in same unit

Disadvantage

Time Complexity is more for optimizer.



Due to above 2 advantages MAE is always preferred over MSE only in case there exists an outlier.

If there is no outlier then prefer MSE since it converges faster(since differentiable due to parabolic curve). On other hand MAE which will be straight line will converge at lower rate, thereby increasing the time complexity.

3. Root Mean Square Error

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \Rightarrow$$ Advantages And Disadvantages.

Advantage:
1.Unit does not changes

Read this: https://towardsdatascience.com/comparing-robustness-of-mae-mse-and-rmse-6d69da870828

We can conclude that MSE and RMSE are more sensitive to outliers are compared to MAE.

Performance Metric = R^2 and Adjusted R^2

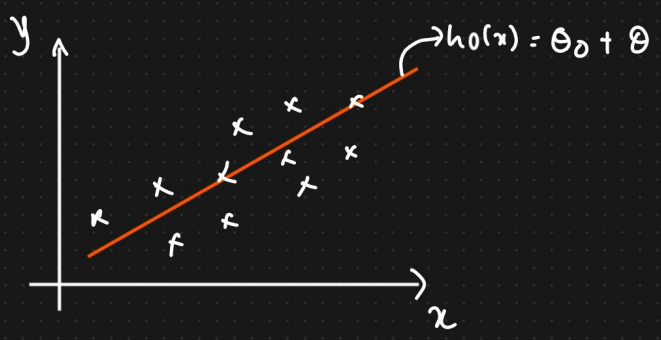
Cost function = MSE, MAE, RMSE, Huber loss ← DL.

Huber loss cost fun^ will be discussed in deep learning.

Ridge, lasso and ElasticNet

Linear Regression

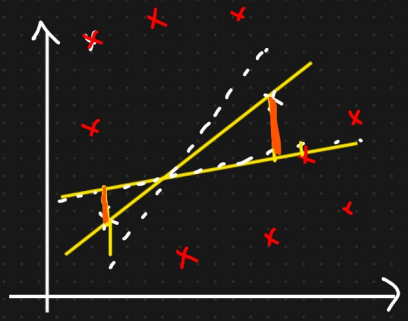
Simple regression already explained earlier



$$\text{Cost function} = \frac{1}{n}\sum_{i=1}^{n}(y_i - h_\theta(x))^2$$

Ridge Reg follows L2 regularization used for reducing over fitting of data.

① Ridge Regression (L2 Regularization) → Reducing Overfitting



TRAINING
Acc => 100%

⇓
Overfitting

Test Data Acc ↓↓

TRAINING DATA 100 ↑↑

Over fitting occurs when accuracy of Train data increases whereas accuracy of test data decreases(Reason self explanatory).

DATASET

⇓ DATA Leakage

Book ⇓ => TRAINING ⇑ TRAIN Model

Exam ⇓ TEST ⇑ Test the Accuracy or performance of the model

Model should not have any info of Test data beforehand. Model should only know about Train data based on which model training is done. If model knows about test data as well then it would be a big blunder which is called data leakage.
Eg: You know about questions coming in exams.

Feature scaling should be done after train/test split. If done before then data leakage. Will be discussed in implementation.

In case of multiple regression this parameter will be squared sum of theta1,theta2....so on.

Ridge Regression ⇓

$$\text{Cost fn} = \frac{1}{n}\sum_{i=1}^{n}(y_i - h_\theta(x)_i)^2 + \lambda\sum_{i=1}^{n}(\text{slope})^2$$

(slope) → Coefficients (Coefficient of x ie; theta1, theta2....so on)

Hyperparameter

$\lambda \geq 0$

In general lambda is greater than zero. This value will decide the degree to which penalization happens which again will depend on problem statement.

$\lambda = 1$

So, through ridge regression we are adding a extra parameter to cost function which is penalizing cost fun to avoid over fitting. When we add extra parameter then error increases making sure that best fit line does not entirely covers training dataset.

⇓
0 Zero

Important in case best fit line covers all training data points(overfitting) then also extra parameter is added that will make entire cost fun^ non zero

$$+ 1 * \left[(\theta_1)^2 + (\theta_2)^2 + (\theta_3)^2\right] \Rightarrow \text{+ve value}$$

Cost fn = +ve value. ↓↓↓

So to conclude after introduction of extra parameter accuracy of train data will decrease but model will become more capable to perform predictions based on any new varied data. That is test accuracy will increase. This is how we are reducing overfitting.

Coefficients ↓↓

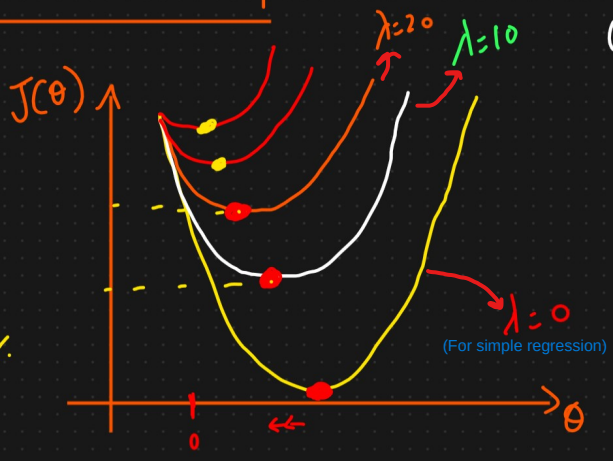
Cost fun^ here will always be +ve since in formula we are squaring the errors.

$\lambda = 1, 2, 3, 4, 10, 20, 30, 40, 50$

Relationship between λ and slope

Observe that as we increase lambda, global minima is shifting left hand side and theta(on x axis) which is slope is getting shifted left or decreases.

Also on increasing lambda, error is also increasing which is represented by cost fun^ on y axis.



(For simple regression)

$$\text{Cost} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \cdot \sum_{i=1}^{n}(\text{slope})^2$$

$\lambda \uparrow\uparrow$ slope $\downarrow\downarrow$

$\lambda \uparrow\uparrow\uparrow$ slope $\downarrow\downarrow\downarrow$

$\lambda \uparrow\uparrow\uparrow\uparrow$ ERROR $\uparrow\uparrow\uparrow$

COST $\uparrow\uparrow\uparrow$

TRAINING

Accuracy = 100%

TRAINING ERROR

② Lasso Regression (L_1 Regularization) → Feature Selection

(L1 regularization)

multiplying a certain lambda that makes least correlated feature's coefficient zero and while doing so slopes of other features will decrease but will not become zero

$$\text{Cost fn}: \frac{1}{n}\sum_{i=1}^{n}(y_i - h_\theta(x)_i)^2 + \left|\lambda \sum_{i=1}^{n} |\text{slope}|\right| \leftarrow L_1 \text{ Regularization}$$

Eg:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_3 + \theta_3 x_3$$

$$h_\theta(x) = 0.52 + |0.65 x_1| + 0.72 x_2 + |0.12 x_3|$$

Unit change in x_1

0.65 change in output

unit change in x_3

0.12 change in output

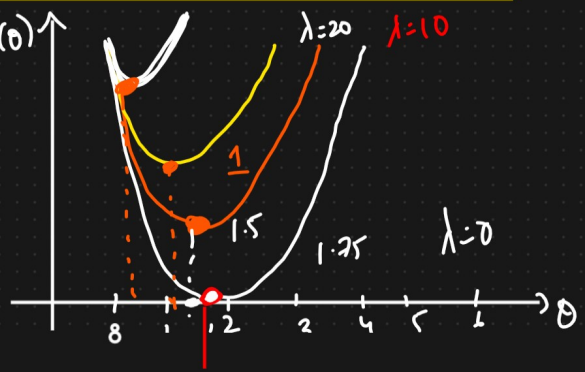
↓

Lasso Regression

Since x3 is least correlated feature it will get removed get removed by applying lasso regression. This is how lasso does feature selection by eliminating the least correlated feature.

Relationship between λ and |slope|

with increasing lamda the feature with least slope(theta value) will get eliminated as will tend to zero. In this way feature reduction is done here. With this other learning parameters will also decrease but will not become zero. The least important one will become zero and get eliminated from the list of features and in the process other important features will decrease but will not become zero.



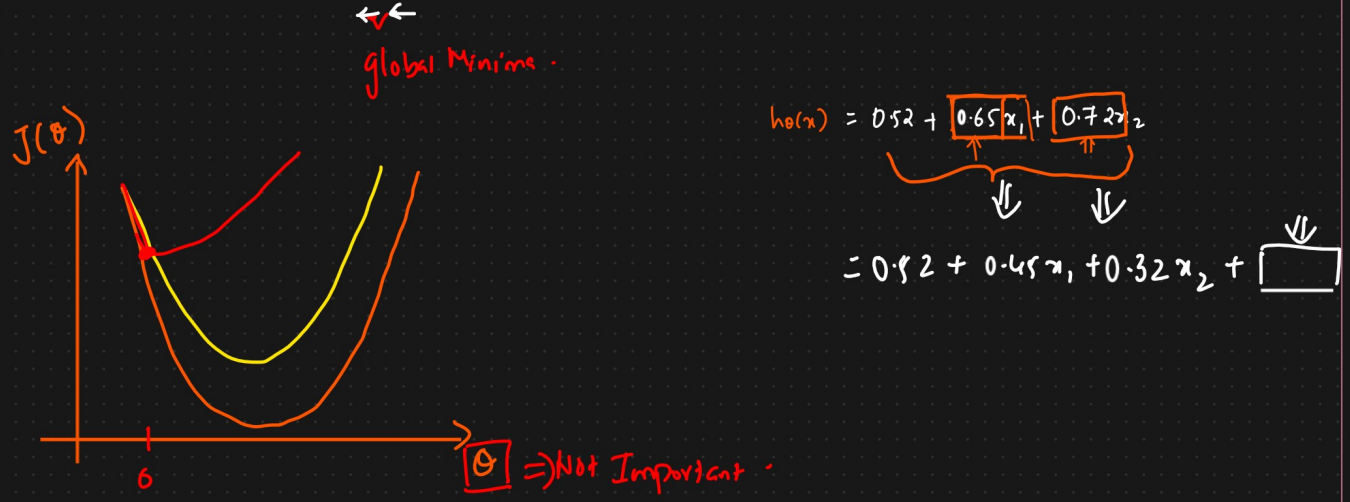
$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_3 + \theta_3 x_3$

$h_\theta(x) = 0.52 + |0.65 x_1| + |0.72 x_2| + |0.12 x_3|$

0

λ - - - -

0.12, 0.20, 0.30



③ ElasticNet

- ① Reducing overfitting
- ② Feature Selection

} Ridge + Lasso

$$\text{Cost fn} = \frac{1}{n}\sum_{i=1}^{n}(y_i - h_\theta(x_i))^2 + \lambda_1 \sum_{i=1}^{n}(\text{slope})^2 + \lambda_2 \sum_{i=1}^{n}|\text{slope}|$$

λ_1 → Reducing Overfitting

λ_2 → Feature selection.

λ_1, λ_2 [Hyperparameters].

Tuning of these hyperparameters is done is such a way so that we built a learning model that reduces overfitting and eliminates less correlated features.

Hyper tuning, gradient etc. these all are done just to built the best ml model.