* Evaluation of Regression model

- We can evaluate our models based on different Regression cost functions.
  - → L2 Loss, Mean squared error
  - → L1 Loss, Mean absolute error
  - → Root mean squared Error

$$\boxed{\text{Cost function} \propto 1/\text{Accuracy of model}}$$

- we also have two other metrics which help us evaluate accuracy of our model, but instead of measuring the absolute error, it measures the performance.

→ R squared (R^2)

- It yields a base line model.
- Its independent of context.
- It basically tells how much better our regression line is compared to base or mean regression line.
- Also called 'Coefficent of Determination' or 'Goodness of fit'

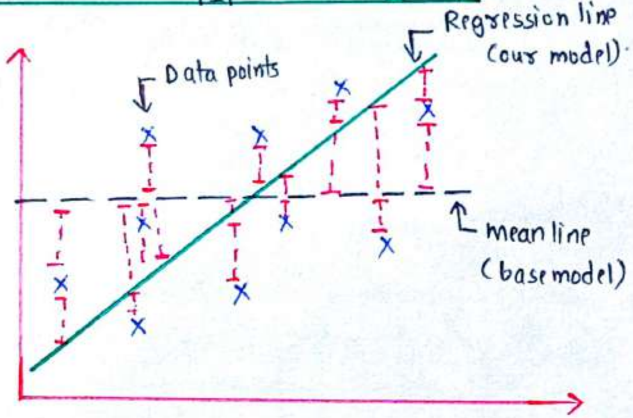
$$\boxed{R^2 = 1 - \frac{SS_{residual}}{SS_{mean}}}$$

$SS_{residual}$ = Squared sum of residual or error/ squared sum of regression line error

SS_{mean} = Squared sum of mean line error

i.e,

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



- R^2 is always less than 1.
- If the error of regression line ($SS_{residual}$) is smaller, more the accuracy.
- If the error of mean line (SS_{mean}) is bigger, less the accuracy.
- If R^2 is negative that means our model is even worse than the base model.
- If R^2 is 1, means baseline and regression line overlap.
- Only demerit is that it doesnt tests the contribution of individual features (columns) to model accuracy, so it may happen we increase feature with little to no impact on model accuracy.

→ Adjusted R squared (R^2_{adj})

- The demerit of R^2 is solved by introducing few parameters.
- Features like,

  n = number of observations / data points

  p = number of independent features.

- when we introduce n, p to R^2, its value dont increase just because of addition of feature, it increases only when feature contribute significantly to the performance.

$$\boxed{R^2_{adj} = 1 - \frac{(1-R^2)(n-1)}{n-p-1}}$$

* Cost function

- It is average of loss functions over the entire dataset.
- It help us reach the optimal solution.
- It is techique to evaluate the performance of our algorithm.
- Our strategy would be to minimize the cost.

$$\text{cost} \propto 1/\text{Accuracy of model.}$$

- Their are various cost functions.

[A] Regression Cost function ($J(\theta_0, \theta_n)$)
- Used for regression models.

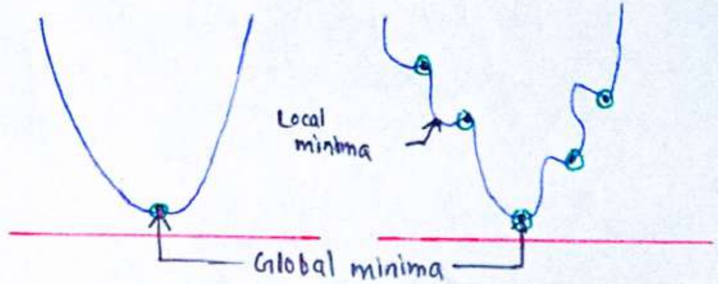
[A·1] Mean Squared Error (L2 Loss)

$$\boxed{J(\theta_0,\theta_n) = \frac{1}{n}\sum_{i=1}^{n}[h_\theta(x)^i - y^i]^2}$$

- Measured sum of squared differences of predicted and actual values
- Basically

$$\boxed{J(\theta_0,\theta_n) = \text{sum squared error}/n}$$

- As it is squared, it penlizes even small deviations in predictions, which means this cost function has only one global minima. i.e a convex function.



Due to local minima regressor could think it is best fit because neighbours are higher although a better option global mima is available.

- Not robust to outliers, as if outliers it will square the error leading to less accuracy.
- As it penalizes the error it squares the unit.

[A·2] Mean Absolute Error (L1 Loss)

$$\boxed{J(\theta_0,\theta_n) = \frac{1}{n}\sum_{i=1}^{n}|h_\theta(x)^i - y^i|}$$

- Measured sum of modulo of differences between predicted and actual value.
- Basically,

$$\boxed{J(\theta_0,\theta_n) = \text{sum modulo error}/n}$$

- As it dont square the errors its robust to outliers and units also dont get squared.
- But convergence usually take more time optimization. (Time consuming)

[A·3] Root mean squared error

$$\boxed{J(\theta_0,\theta_n) = \left[\frac{1}{n}\sum_{i=1}^{n}[h_\theta(x)^i - y^i]^2\right]^{1/2}}$$

- Measured root of square mean error
- Basically,

$$\boxed{J(\theta_0,\theta_n) = (L_2\,Loss)^{1/2}}$$

- Doesnt penalize the error as much as L2Loss
- It is time optimized.