FORMERLY Industries Helping Hands

THE iSCALE

Data Science | 30 Days of Machine Learning | Day - 25

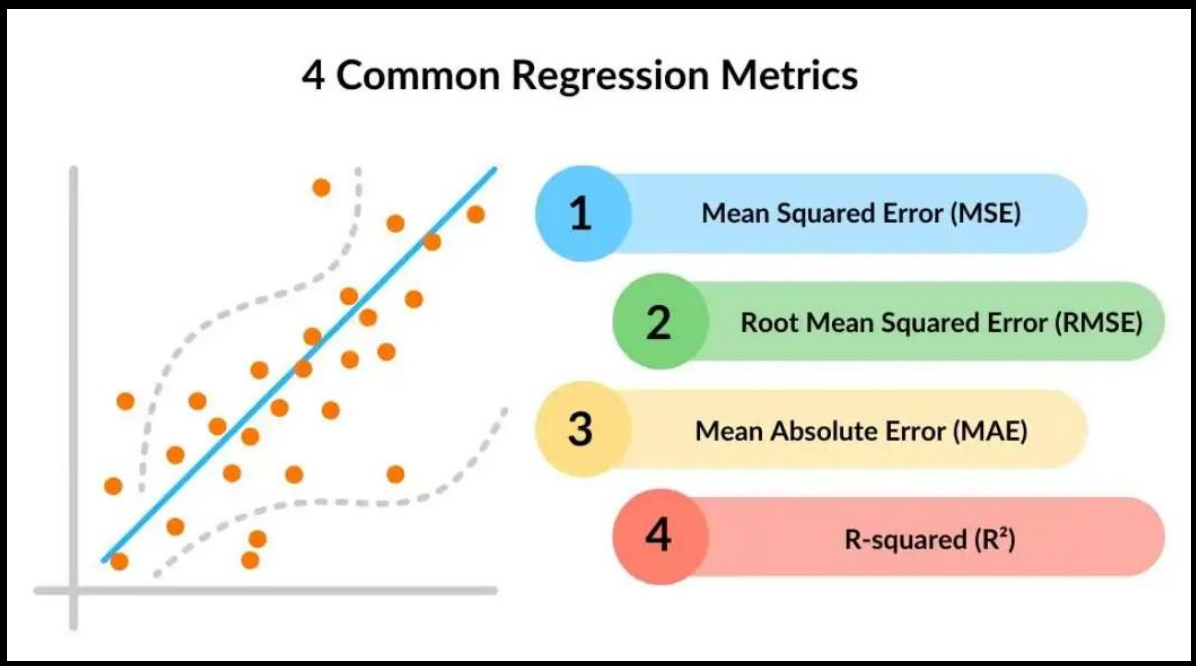
Educator Name: Nishant Dhote
Support Team: **+91-7880-113-112**

----Today Topics | Day 25----

Regression Metrics

- MAE: Mean Absolute Error
- MSE: Mean Squared Error
- RMSE: Root Mean Squared Error
- R Squared Score R2 (R^2)
- Adjusted R Squared (R^2) Score

Dataset Link GitHub: https://github.com/TheiScale/30_Days_Machine_Learning/

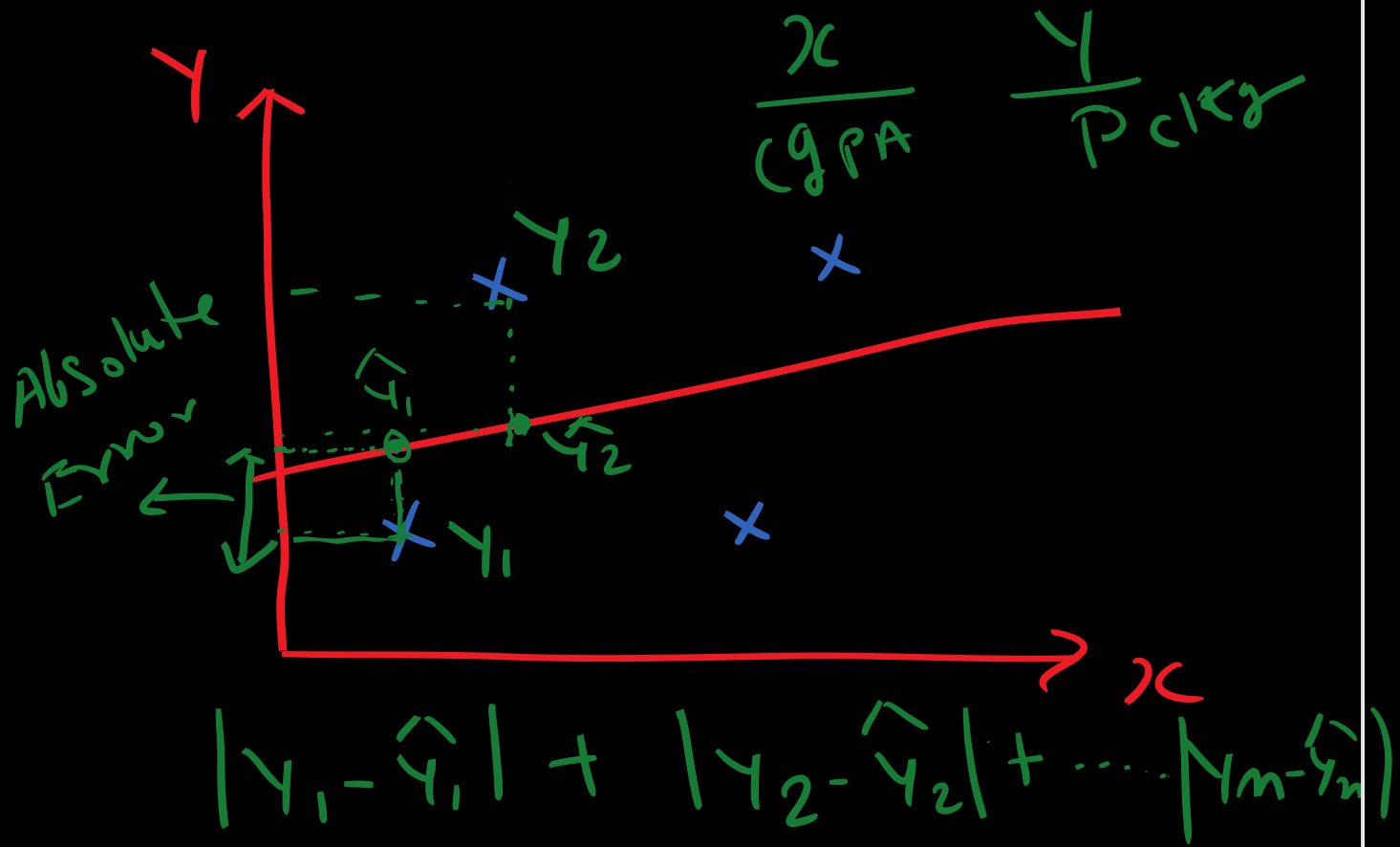


- MAE: Mean Absolute Error

The Mean Absolute Error (MAE) is only slightly different in definition from the MSE, but interestingly provides almost exactly opposite properties! To calculate the MAE, you take the difference between your model's predictions and the ground truth, apply the absolute value to that difference, and then average it out across the whole dataset.

The MAE, like the MSE, will never be negative since in this case we are always taking the absolute value of the errors. The MAE is formally defined by the following equation:

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$



Total Absolute Error

$$|y_1 - \hat{y}_1| + |y_2 - \hat{y}_2| + \dots |y_n - \hat{y}_n|$$

Divide by (n) → Mean Absolute Error

$$= \frac{|y_1 - \hat{y}_1| + |y_2 - \hat{y}_2| + \dots |y_n - \hat{y}_n|}{n}$$

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

Advantage →

(1) Same unit

(2) Robust for outliers

Disadvantage → Non Differentiable

MAE Advantage and Disadvantage:

Advantage: The beauty of the MAE is that its advantage directly covers the MSE disadvantage. Since we are taking the absolute value, all of the errors will be weighted on the same linear scale. Thus, unlike the MSE, we won't be putting too much weight on our outliers and our loss function provides a generic and even measure of how well our model is performing.

Disadvantage: If we do in fact care about the outlier predictions of our model, then the MAE won't be as effective. The large errors coming from the outliers end up being weighted the exact same as lower errors. This might result in our model being great most of the time, but making a few very poor predictions every so-often.

Metric	Advantages	Disadvantages
MAE	Easy to interpret and understand. Less sensitive to outliers.	Does not take into account the direction of errors.
MSE	Penalizes larger errors more heavily, giving it more sensitivity to outliers.	Harder to interpret than MAE, as it is not in the same unit as the original data.
RMSE	Has the same units as the original data, making it easier to interpret.	Sensitive to outliers.
R-squared	Easy to interpret, as it measures the proportion of the variance in the dependent variable that is explained by the independent variables.	Cannot distinguish between linear and nonlinear relationships. Can be misleading if used without considering the underlying assumptions.
Adjusted R-squared	Takes into account the number of independent variables in the model, making it more reliable than R-squared when comparing models with different numbers of variables.	Can be misleading if used without considering the underlying assumptions.

- MSE: Mean Squared Error

The Mean Squared Error (MSE) is perhaps the simplest and most common loss function, often taught in introductory Machine Learning courses. To calculate the MSE, you take the difference between your model's predictions and the ground truth, square it, and average it out across the whole dataset.

The MSE will never be negative, since we are always squaring the errors. The MSE is formally defined by the following equation:

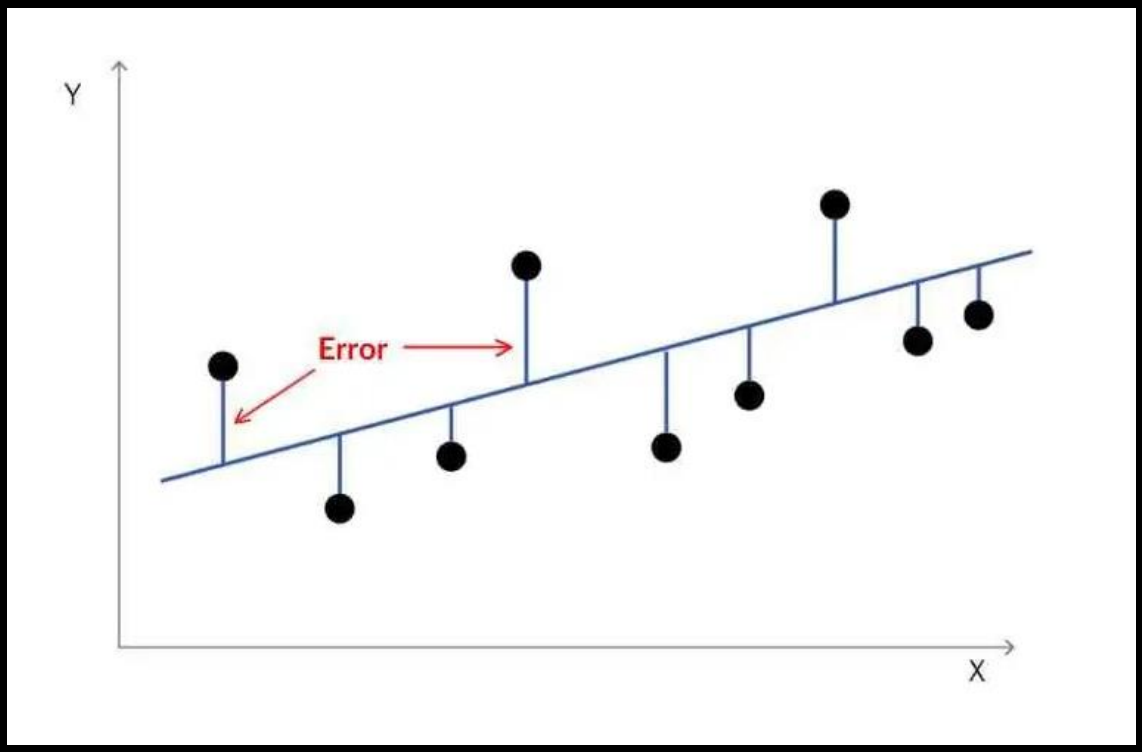
$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

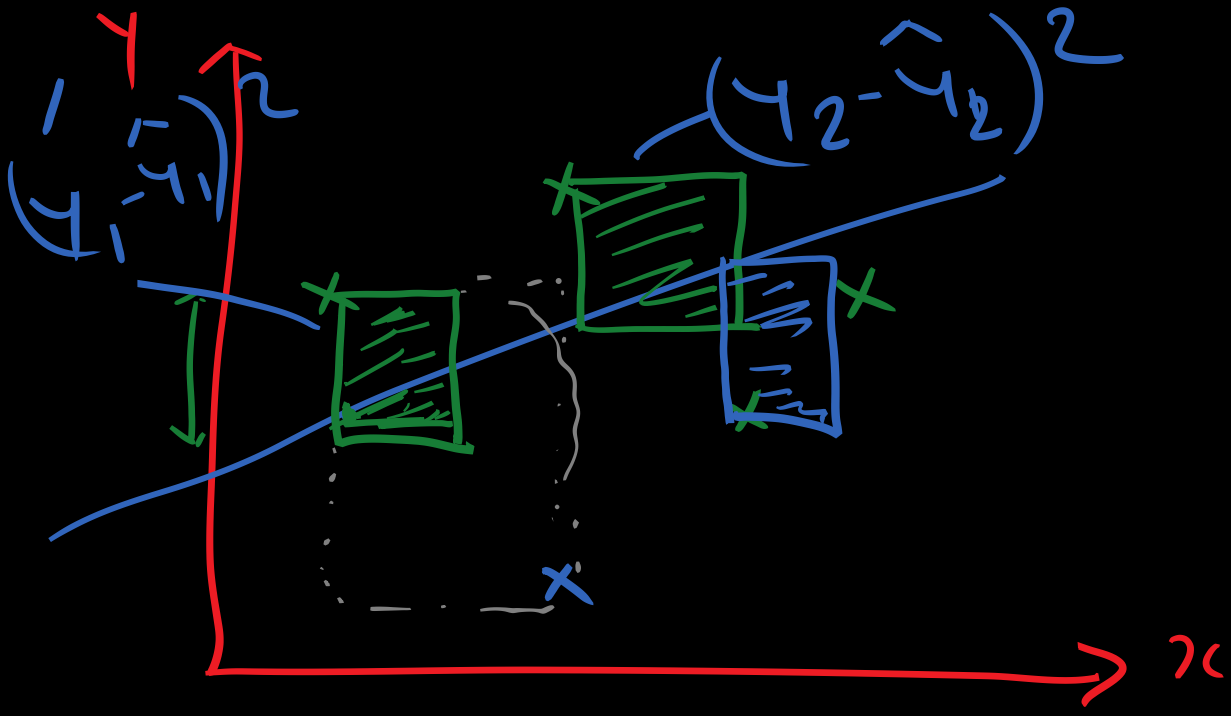
MSE calculates the average squared difference between predicted and actual values.

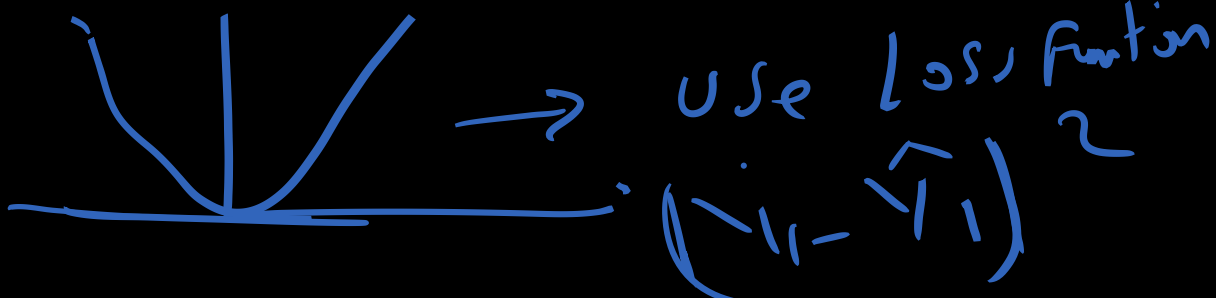
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\text{Actual}_i - \text{Predicted}_i)^2$$

where yi represents the actual value, y^i represents the predicted value, and n is the number of observations.

MSE measures the average squared error, with higher values indicating more significant discrepancies between predicted and actual values.


Y
Error
X


Y
X
X
X


use loss function

MSE Advantage and Disadvantage:

Advantage: The MSE is great for ensuring that our trained model has no outlier predictions with huge errors, since the MSE puts larger weight on these errors due to the squaring part of the function.

Disadvantage: If our model makes a single very bad prediction, the squaring part of the function magnifies the error. Yet in many practical cases we don't care much about these outliers and are aiming for more of a well-rounded model that performs good enough on the majority.

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

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

Advantage

① Use as a loss function

Disadvantage

① Unit ? 2C | Y
CGPA | LPA

$Y = (LPA)^2$

② Not Robust to outliers

- RMSE: Root Mean Squared Error

A metric that tells us the square root of the average squared difference between the predicted values and the actual values in a dataset. The lower the RMSE, the better a model fits a dataset.

Root Mean Squared Error is the square root of Mean Squared error. It measures the standard deviation of residuals.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

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

Also => Same Unit value

RMSE vs. MSE: Which Metric Should You Use?

When assessing how well a model fits a dataset, we use the RMSE more often because it is measured in the same units as the response variable.

Conversely, the MSE is measured in squared units of the response variable.

To illustrate this, suppose we use a regression model to predict the number of points that 10 players will score in a basketball game.

The following table shows the predicted points from the model vs. the actual points the players scored:

Predicted Points ($\hat{y}_i$)	Actual points (y_i)
14	12
15	15
18	20
19	16
25	20
18	19
12	16
12	20
15	16
22	16

We would calculate the mean squared error (MSE) as:

MSE = Σ(ŷi – yi)2 / n

MSE = ((14-12)2+(15-15)2+(18-20)2+(19-16)2+(25-20)2+(18-19)2+(12-16)2+(12-20)2+(15-16)2+(22-16)2) / 10

MSE = 16

The mean squared error is 16. This tells us that the average squared difference between the predicted values made by the model and the actual values is 16.

========

The root mean squared error (RMSE) would simply be the square root of the MSE:

RMSE = √MSE
RMSE = √16
RMSE = 4

The root mean squared error is 4. This tells us that the average deviation between the predicted points scored and the actual points scored is 4.

- R Squared Score (R^2)

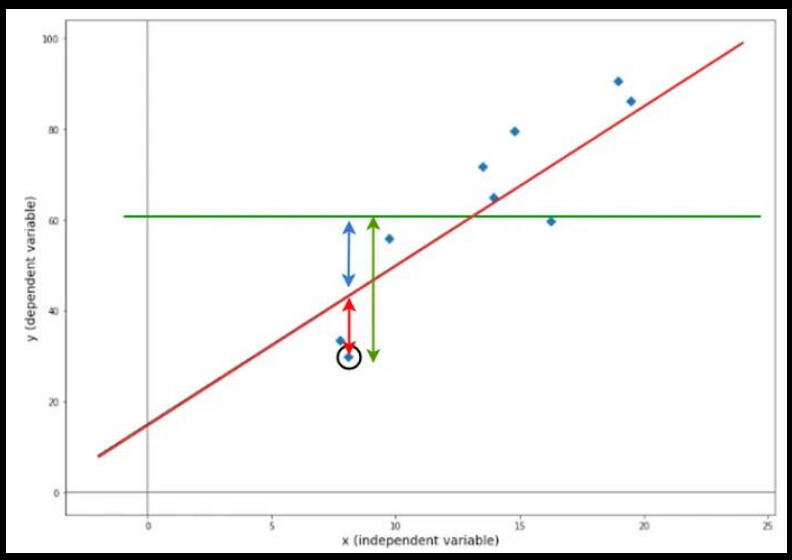
R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.

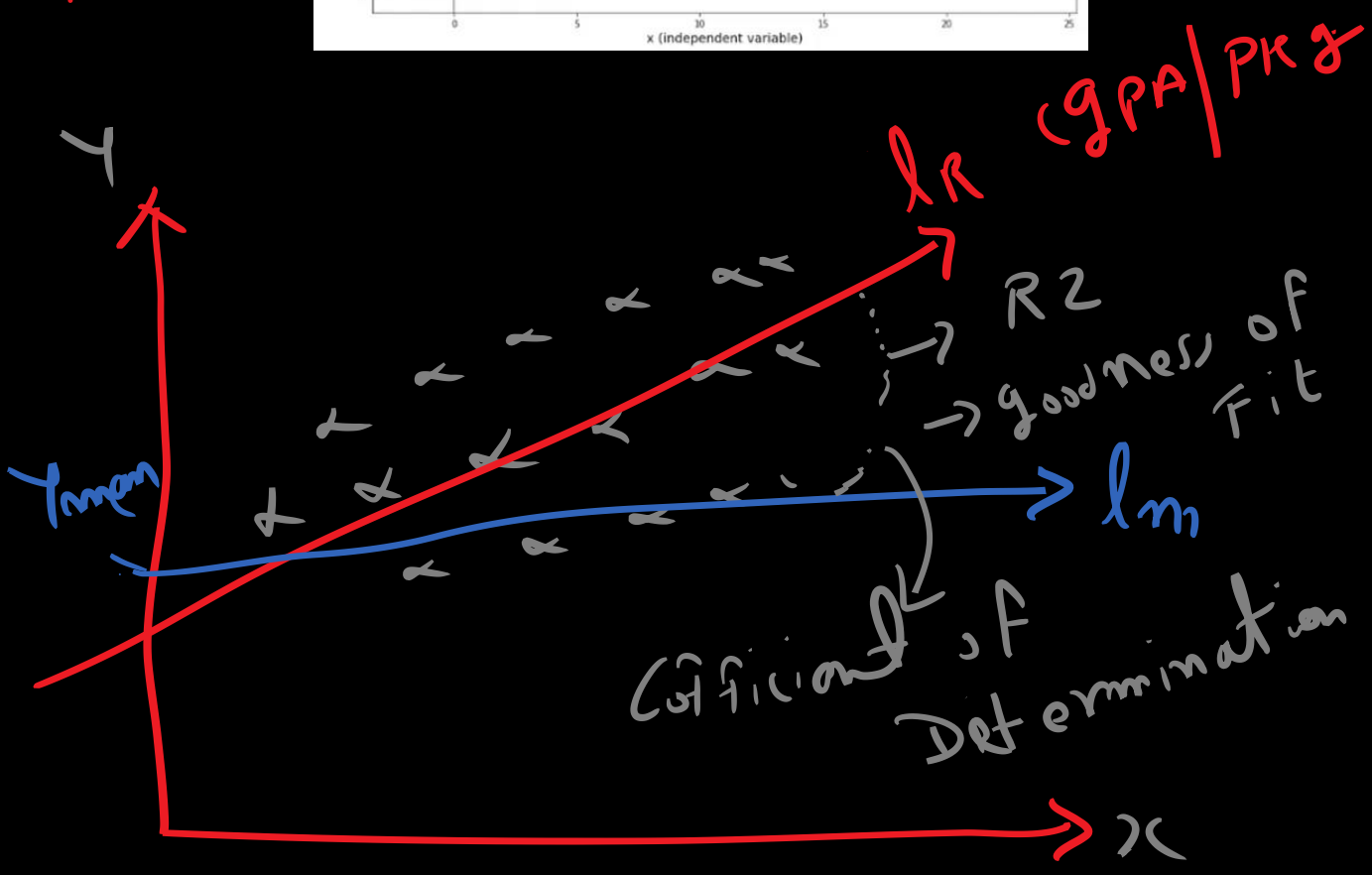
How to Calculate R-Squared

The R-Squared formula compares our fitted regression line to a baseline model. This baseline model is considered the "worst" model. The baseline model is a flat-line that predicts every value of y will be the mean value of y. R-Squared checks to see if our fitted regression line will predict y better than the mean will.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

The bottom of our formula is the total sum of squared errors (SStot). This is comparing the actual y values to our baseline model the mean. So we square the difference between the all the actual y values and the mean and add them together.





$$R2 = 1 - \frac{SS_R}{SS_m}$$

$SS_R \rightarrow$ Sum of Squared Error in Regression line

$SS_m \Rightarrow$ Sum of Squared Error in mean line

$$R2 = 1 - \frac{\left[\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right]_{Reg}}{\left[\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right]_{mean}}$$

$R2 = 0$	$R2 = 1$
$R2 = 1 - 1$	$R2 = 1 - \frac{SS_R}{SS_m} = 1$
$R2 = 0$	$R2 = 1$

$SSR > SSm$

- Adjusted R Squared Score (R^2)

This measures the variation for a multiple regression model, and helps you determine goodness of fit. Unlike R-squared, adjusted R-squared only adds new predictors to its model if it improves the model's predicting power.

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

Where

R^2 Sample R-Squared

N Total Sample Size

n Number of independent variable

K

(n - K - 1)

K=3 K=4

Adjusted R2 Score

CgPA | Exp (Yrs) | Pe%ile | Fees

All Input col (R2)

n = No of Row

$$R^2_{ADJ} = 1 - \left[\frac{(1 - R_2)(n-1)}{(n - 1 - K)}\right]$$

K = Total no of input col.

<Start Coding>

```
#Import Library

import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

#Import Dataset

df = pd.read_csv('placement.csv')
----

df.head()
df.shape

#Data Plot in Graph

plt.scatter(df['cgpa'],df['package'])
plt.xlabel('CGPA')
plt.ylabel('Package(in lpa)')


#Define X and y as a Input and Output Column

X = df.iloc[:,0:1]
y = df.iloc[:,-1]

----
X
----
y
```

```
#Train Test Split

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.2,random_state=2)


#Import linear regression


from sklearn.linear_model import LinearRegression


#Define LR

lr = LinearRegression()


#Use "fit" for train the model

lr.fit(X_train,y_train)

#Draw line "Linear Regression model"

plt.scatter(df['cgpa'],df['package'])
plt.plot(X_train,lr.predict(X_train),color='red')
plt.xlabel('CGPA')
plt.ylabel('Package(in lpa)')

#Import SK learn

from sklearn.metrics import
mean_absolute_error,mean_squared_error,r2_score
```

```
#Predict X test values


lr.predict(X_test)
----
y_pred = lr.predict(X_test)
----
y_test
----
y_test.values

#Calculate "Mean Absolute Error"
print("MAE",mean_absolute_error(y_test,y_pred))



#Calculate "Mean Squared Error"
print("MSE",mean_squared_error(y_test,y_pred))

#Calculate "Root Mean Squared Error"
print("RMSE",np.sqrt(mean_squared_error(y_test,y_pre
 d)))

#Calculate "R2 Score"
print("R2S",r2_score(y_test,y_pred))
----
```

```
r2 = r2_score(y_test,y_pred)
#Calculate "Adjusted R2 Score"
 X_test.shape
 ----
 1 - ((1-r2)*(40-1)/(40-1-1))
```

Day 25 | Data Curious Minds

Suggest topic – Next class