

Regression Metrics

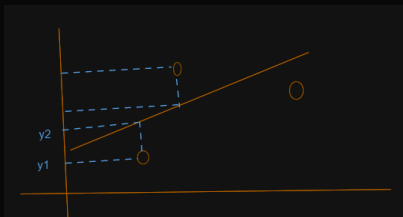
github

video

How to check the Regression you have applied is accurate or not by using Regression Metrics

Types

- Loss function/error function
  - MSE
  - MAE
  - RMSE
- R2 Score
- Adjusted R2 Score



Here in the above example the For the first data point the correct output value is y1 but according to our model its correct value is y2

so we calculate the MAE like this  $((|Y1-Y2|)+(|Y3-Y4|)+.....(|yN-yN|))/N$

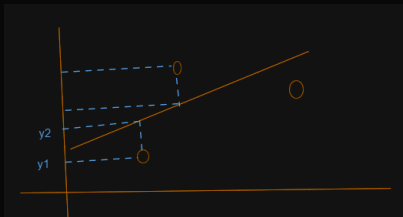
MAE  
Mean absolute error

logic

- the output of this method is the loss, and we try our best to minimize those loose
- the unit of the value/loose comes from this method is same as unit of output column of the regression problem

This is robust to outlier

Dis Here we use modulus function which is not differentiable at 0



Here in the above example the For the first data point the correct output value is y1 but according to our model its correct value is y2

so we calculate the MAE like this  $((Y1-Y2)*(Y1-Y2)+(Y3-Y4)*(Y3-Y4)+..(YN-YN)*(YN-YN))/N$

here instant of taking mod (| |) of values we are taking square of them

MSE  
Mean Square error

logic

- the output of this method is the loss, and we try our best to minimize those loose
- the unit of the value/loose comes from this method is same as unit of output column of the regression problem but the value is squared

Dis This is Not robust to outlier

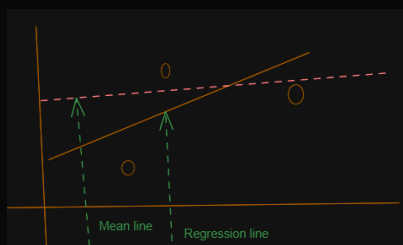
RMSE  
Root mean square error

logic

Root of MSE

- the output of this method is the loss, and we try our best to minimize those loose
- the unit of the value/loose comes from this method is same as unit of output column of the regression problem

Dis This is Not robust to outlier



You calculate how much your regression model are better then mean

R2 score  
Coefficient of determination  
goodness of fit

logic

Formula  $1 - ((\text{sum of squared error in the regression line})/(\text{sum of squared error in the mean line}))$

value always in between 0 - 1

we have to make sure the value always towards 1 (if you get 1 = perfect model)

if you add more and more input feature , even if they are irrelevant your R2 score will start increasing. that's why we use **Adjusted R2 score**

div

Adjusted R2 score

logic

$$1 - \left[ \frac{(1-R^2) \cdot (n-1)}{(n-1-K)} \right]$$

Formula

k=number of columns in train set  
n= size of x\_test

if you add more and more input feature , even if they are irrelevant your R2 score will start increasing. that's why we use **Adjusted R2 score**

Working

imports

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

data

time	package
0	6.89 3.26
1	5.12 1.98
2	7.82 3.25
3	7.42 3.67
4	6.94 3.37

test train split

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

Applying linear regression

from sklearn.linear\_model import LinearRegression  
  
lr = LinearRegression()  
  
lr.fit(X\_train,y\_train)

Applying mean\_absolute\_error, mean\_squared\_error, r2\_score

from sklearn.metrics import  
mean\_absolute\_error,mean\_squared\_error,r2\_score  
  
y\_pred = lr.predict(X\_test)  
  
print("MAE",mean\_absolute\_error(y\_test,y\_pred))  
  
print("MSE",mean\_squared\_error(y\_test,y\_pred))  
  
print("RMSE",np.sqrt(mean\_squared\_error(y\_test,y\_pred)))  
  
print("R2",r2\_score(y\_test,y\_pred))

For Adjusted R2 score -> there is no function we need to get it manually using formula

Find k = number of columns (1)

Find n = number of size of test (x\_test.shape())

$1 - ((1-r^2)*(40-1)/(40-1-1))$