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

MAE Mean absolute error

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

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 modules function which is not differentiable at 0

MSE Mean Square error

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

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

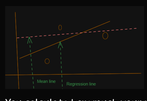
RMSE Root mean square error

Root of MSE

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
- Dis
  - This is Not robust to outlier

R2 score Coefficient of determination goodness of fit



You calculate how much your regression model are better then mean

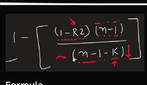
logic

- Formula
  - 1-((sum of squared error in the regression line)/(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)
- div
  - 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

Adjusted R2 score

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

logic



Formula

k=number of columns in train set
n= size of x_test

Working

- imports
  - import matplotlib.pyplot as plt
    import pandas as pd
    import numpy as np
- data

	size	packaging
0	6.89	3.76
1	5.12	1.98
2	7.82	3.25
3	7.42	3.67
4	6.94	3.57

- test train split
  - X = df.iloc[:,0:1]
    y = df.iloc[:,-1]
  - 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)
- 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-r2)*(40-1)/(40-1-1))