**Regression :**

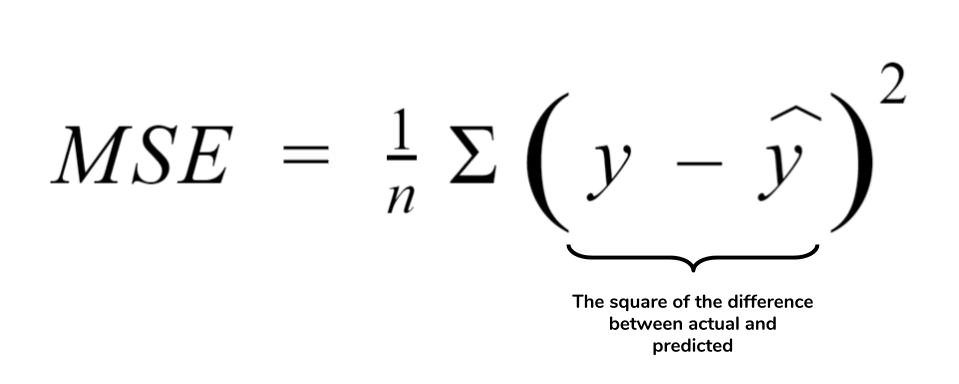
* Regression task is the prediction of the state of an outcome variable at a particular time point with the help of other correlated independent variables.
* The regression task, unlike the classification task, outputs continuous value within a given range.

The various metrics used to evaluate the results of the prediction are :

1. Mean Squared Error(MSE)
2. Root-Mean-Squared-Error(RMSE).
3. Mean-Absolute-Error(MAE).
4. R² or Coefficient of Determination.
5. Adjusted R²
6. Mean Absolute Percentage Error
7. Weighted Mean Absolute Percentage Error (WMAPE)

**Mean Squared Error:**

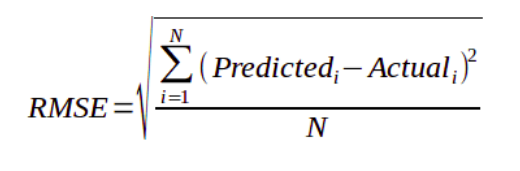
* MSE or Mean Squared Error is one of the most preferred metrics for regression tasks. **It is simply the average of the squared difference between the target value and the value predicted by the regression model.**
* As it squares the differences, it penalizes even a small error which leads to over-estimation of how bad the model is.
* It is preferred more than other metrics because it is differentiable and hence can be optimized better.



**>>> from** **sklearn.metrics** **import** mean\_squared\_error  
**>>>** y\_true = [3, -0.5, 2, 7]  
**>>>** y\_pred = [2.5, 0.0, 2, 8]  
**>>>** mean\_squared\_error(y\_true, y\_pred)  
0.375  
**>>>** y\_true = [[0.5, 1],[-1, 1],[7, -6]]  
**>>>** y\_pred = [[0, 2],[-1, 2],[8, -5]]  
**>>>** mean\_squared\_error(y\_true, y\_pred)   
0.708...  
**>>>** mean\_squared\_error(y\_true, y\_pred, multioutput='raw\_values')  
**...**   
array([0.41666667, 1. ])  
**>>>** mean\_squared\_error(y\_true, y\_pred, multioutput=[0.3, 0.7])  
**...**   
0.825..

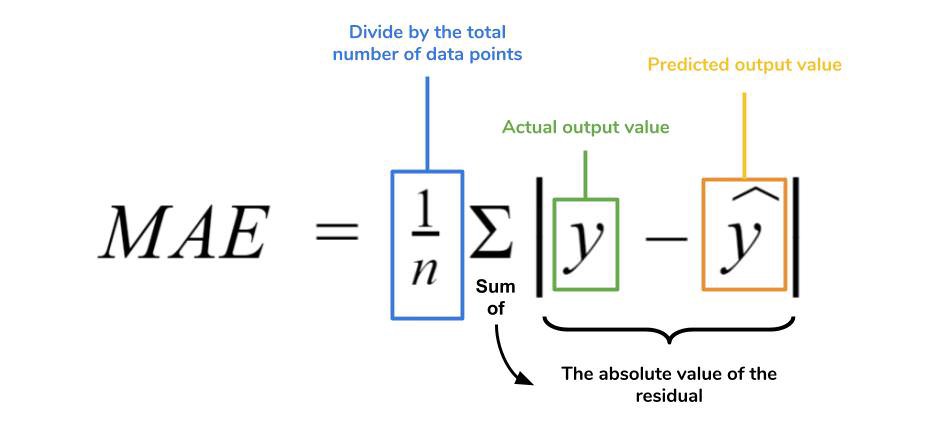
**Root Mean Squared Error:**

* RMSE is the most widely used metric for regression tasks and is the square root of the averaged squared difference between the target value and the value predicted by the model.
* It is preferred more in some cases **because the errors are first squared before averaging which poses a high penalty on large errors.**
* This implies that RMSE is useful when large errors are undesired.



**Mean Absolute Error:**

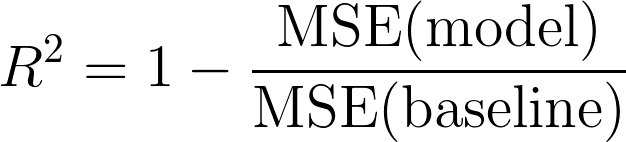
* MAE is the absolute difference between the target value and the value predicted by the model.
* The **MAE is more robust to outliers and does not penalize the errors as extremely as mse**. MAE is a linear score which means all the individual differences are weighted equally.
* It is not suitable for applications where you want to pay more attention to the outliers.



**>>> from** **sklearn.metrics** **import** mean\_absolute\_error  
**>>>** y\_true = [3, -0.5, 2, 7]  
**>>>** y\_pred = [2.5, 0.0, 2, 8]  
**>>>** mean\_absolute\_error(y\_true, y\_pred)  
0.5  
**>>>** y\_true = [[0.5, 1], [-1, 1], [7, -6]]  
**>>>** y\_pred = [[0, 2], [-1, 2], [8, -5]]  
**>>>** mean\_absolute\_error(y\_true, y\_pred)  
0.75  
**>>>** mean\_absolute\_error(y\_true, y\_pred, multioutput='raw\_values')  
array([0.5, 1. ])  
**>>>** mean\_absolute\_error(y\_true, y\_pred, multioutput=[0.3, 0.7])  
**...**   
0.85...

**R² Error:**

* Coefficient of Determination or R² is another metric used for evaluating the performance of a regression model.
* The metric helps us to compare our current model with a constant baseline and tells us how much our model is better.
* The constant baseline is chosen by taking the mean of the data and drawing a line at the mean.
* R² is a scale-free score that implies it doesn't matter whether the values are too large or too small, the R² will always be less than or equal to 1.



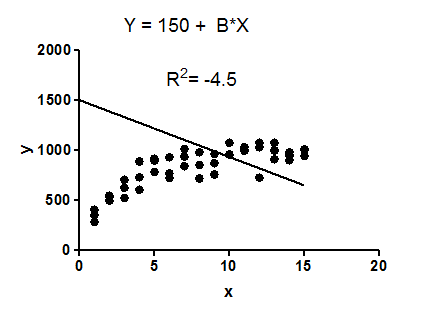
**>>> from** **sklearn.metrics** **import** r2\_score  
**>>>** y\_true = [3, -0.5, 2, 7]  
**>>>** y\_pred = [2.5, 0.0, 2, 8]  
**>>>** r2\_score(y\_true, y\_pred)   
0.948...  
**>>>** y\_true = [[0.5, 1], [-1, 1], [7, -6]]  
**>>>** y\_pred = [[0, 2], [-1, 2], [8, -5]]  
**>>>** r2\_score(y\_true, y\_pred, multioutput='variance\_weighted')  
**...**   
0.938...  
**>>>** y\_true = [1,2,3]  
**>>>** y\_pred = [1,2,3]  
**>>>** r2\_score(y\_true, y\_pred)  
1.0  
**>>>** y\_true = [1,2,3]  
**>>>** y\_pred = [2,2,2]  
**>>>** r2\_score(y\_true, y\_pred)  
0.0  
**>>>** y\_true = [1,2,3]  
**>>>** y\_pred = [3,2,1]  
**>>>** r2\_score(y\_true, y\_pred)  
-3.0

**Why is R² Negative?**

* R² score ranges ranges from -∞ to 1.

The main reasons for R² to be negative are the following:

1. One of the main reason for R² to be negative is that the chosen model does not follow the trend of the data causing the R² to be negative. This causes the mse of the chosen model(numerator) to be more than the mse for constant baseline(denominator) resulting in negative R².

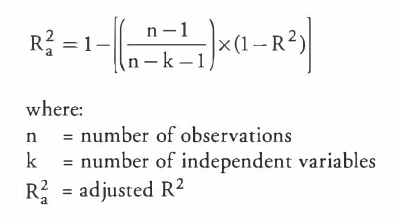


2. Maybe their area a large number of outliers in the data that causes the mse of the model to be more than mse of the baseline causing the R² to be negative(i.e the numerator is greater than the denominator).

3. Sometimes while coding the regression algorithm, the researcher might forget to add the intercept to the regressor which will also lead to R² being negative. This is because, without the benefit of an intercept, the regression could do worse than the sample mean(baseline) in terms of tracking the dependent variable (i.e., the numerator could be greater than the denominator). However, most of the standard machine learning library like scikit-learn include the intercept by default but if you are using stats-model library then you have to add the intercept manually.

**Adjusted R²:**

* Adjusted R² depicts the same meaning as R² but is an improvement of it.
* R² suffers from the problem that the scores improve on increasing terms even though the model is not improving which may misguide the researcher.
* Adjusted R² is always lower than R² as it adjusts for the increasing predictors and only shows improvement if there is a real improvement.t



## #https://www.listendata.com/2014/08/adjusted-r-squared.html?m=1#:~:text=Squared%20Equation%202-,Difference%20between%20R%2Dsquare%20and%20Adjusted%20R%2Dsquare,significant%20and%20affects%20dependent%20variable

**Difference between R-square and Adjusted R-square**

1. Every time you add a independent variable to a model, the **R-squared** **increases**, even if the independent variable is insignificant. It never declines. Whereas **Adjusted R-squared** increases only when independent variable is significant and affects dependent variable.

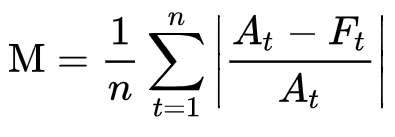
In the table below, adjusted r-squared is maximum when we included two variables. It declines when third variable is added. Whereas r-squared increases when we included third variable. It means third variable is insignificant to the model.

|  |
| --- |
|  |
| R-Squared vs. Adjusted R-Squared |

1. Adjusted r-squared can be negative when r-squared is close to zero.
2. Adjusted r-squared value always be less than or equal to r-squared value.

## Mean Absolute Percentage Error (MAPE)

Instead of using actual value, MAPE uses the percentage error to present the result. Let actual value are [5,10] and the predicted value are [10, 2]. The MAPE is 90% = ((|5–10|/5) + (|10–2|)/10) / 2 \* 100%.



## Weighted Mean Absolute Percentage Error (WMAPE)

WMAPE put more attention on the large value. It overcomes some MAPE limitations. MAPE both small value and large value equally, but it may introduce misleading. Also, it robust to value if it is zero.

The formula shows as below. A is actual, while F is forecast value (predicted value). You may notice that we can eliminate A and simplify it as sum(A-F)\*100/(sum(A)). Since it does not need to divide the individual value, WMAPE can handle zero cases (expect sum(A) is 0).

