How do you calculate Mean Absolute Error in Python?

Authored by

stats writer

April 24, 2024

Mean Absolute Error (MAE) is a commonly used metric in data analysis to measure the average difference between the predicted and actual values of a dataset. In Python, MAE can be calculated by first determining the absolute difference between each predicted value and its corresponding actual value. These differences are then averaged to find the overall MAE. This can be achieved by using a built-in function, such as "mean_absolute_error" from the scikit-learn library, or by manually computing the sum of absolute differences and dividing it by the number of data points. The resulting value represents the average magnitude of error in the predicted values, making it a useful tool for evaluating the accuracy of a predictive model.

Calculate Mean Absolute Error in Python

In statistics, the mean absolute error (MAE) is a way to measure the accuracy of a given model. It is calculated as:

MAE = (1/n) * Σ|yi - xi|

where:

Σ: A Greek symbol that means "sum"yi: The observed value for the ith observationxi: The predicted value for the ith observationn: The total number of observations

We can easily calculate the mean absolute error in Python by using the function from Scikit-learn.

This tutorial provides an example of how to use this function in practice.

Example: Calculating Mean Absolute Error in Python

Suppose we have the following arrays of actual values and predicted values in Python:

```
actual =
pred =
```

The following code shows how to calculate the mean absolute error for this model:

```
from sklearn.metricsimport mean_absolute_error as mae

#calculate MAE
mae(actual, pred)
```

2.4285714285714284

The mean absolute error (MAE) turns out to be 2.42857.

This tells us that the average difference between the actual data value and the value predicted by the model is 2.42857.

We can compare this MAE to the MAE obtained by other

forecast models to see which models perform best.

The lower the MAE for a given model, the more closely the model is able to predict the actual values.

Note: The array of actual values and the array of predicted values should both be of equal length in order for this function to work correctly.