

Mean_absolute_percentage_error(MAPE)

The **mean absolute percentage error (MAPE)**, also known as **mean absolute percentage deviation (MAPD)**, is a measure of the accuracy of a predictive method of calculating in mathematics, for example in a measure of trend, which is also used as a function of loss of machine learning problems. It usually shows accuracy as a measure defined by the formula:

$$M = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|,$$

where A_t is the actual value and F_t is the forecast value.

The Mean Absolute Percentage Error (MAPE) is a measure of quality for regression models.

Parameters

y_truearray-like of shape (n_samples,) or (n_samples, n_outputs)
Ground truth (correct) target values.

y_predarray-like of shape (n_samples,) or (n_samples, n_outputs)
Estimated target values.

sample_weightarray-like of shape (n_samples,), default=None
Sample weights.

multioutput{'raw_values', 'uniform_average'} or array-like
Defines aggregating of multiple output values. Array-like value defines weights used to average errors. If input is list then the shape must be (n_outputs,).

'raw_values' :
Returns a full set of errors in case of multioutput input.

'uniform_average' :
Errors of all outputs are averaged with uniform weight.

Returns

Loss float or ndarray of floats in the range [0, 1/eps]

If multioutput is 'raw_values', then mean absolute percentage error is returned for each output separately. If multioutput is 'uniform_average' or an ndarray of weights, then the weighted average of all output errors is returned.

MAPE output is non-negative floating point. The best value is 0.0. But note the fact that bad predictions can lead to arbitrarily large MAPE values, especially if some y_true values are very close to zero. Note that we return a large value instead of inf when y_true is zero.

#Example: 1

```
from sklearn.metrics import mean_absolute_percentage_error
y_true = [3, -0.5, 2, 7]
y_pred = [3, -0.5, 2, 7]
mean_absolute_percentage_error(y_true, y_pred)
```

Output:

0.0

Example: 2

```
from sklearn.metrics import mean_absolute_percentage_error
y_true = [3, -0.5, 2, 7]
y_pred = [2, 0.0, 2, 9]
mean_absolute_percentage_error(y_true, y_pred)
```

Output:

0.40476