What is the difference between NumPy mean() and average()?

Authored by

stats writer

June 28, 2024

RECOMMENDED CITATION

stats writer (2024). *What is the difference between NumPy mean() and average()?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=156003

ARABPSYCHOLOGY.COM

NumPy mean() and average() are both functions used for calculating the average value of a given array. However, there is a slight difference between the two. The mean() function calculates the average by taking the sum of all the elements in the array and dividing it by the total number of elements. On the other hand, the average() function calculates the average by taking the sum of all the elements in the array and dividing it by the number of non-zero elements. This means that the average() function considers only the non-zero elements in the array, while the mean() function takes into account all the elements. Therefore, the result of mean() and average() may differ if the array contains any zero values.

NumPy mean() vs. average(): What's the Difference?

You can use the np.mean() or np.average() functions to calculate the average value of an array in Python.

Here is the subtle difference between the two functions:

np.mean always calculates the arithmetic mean.np.average has an optional weights parameter that can be used to calculate a weighted average.

The following examples show how to use each function in practice.

Example 1: Use np.mean() and np.average() without Weights

Suppose we have the following array in Python that contains seven values:

#create array of values
data =

ARABPSYCHOLOGY.COM

We can use np.mean() and np.average() to calculate the average value of this array:

import numpy as np

#calculate average value of array
np.mean(data)

6.142857142857143

#calcualte average value of array
np.average(data)6.142857142857143

Both functions return the exact same value.

Both functions used the following formula to calculate the average:

Average = (1 + 4 + 5 + 7 + 8 + 8 + 10) / 7 = 6.142857...

Example 2: Use np.average() with Weights

Once again suppose we have the following array in Python that contains seven values:

#create array of values
data =

ARABPSYCHOLOGY.COM

We can use np.average() to calculate a weighted average for this array by supplying a list of values to the weights parameters:

```
import numpy as np

#calculate weighted average of array
np.average(data, weights=(.1, .2, .4, .05, .05, .1, .1))5.45
```

Here is the formula that np.average() used to calculate this value:

Weighted Average = 1*.1 + 4*.2 + 5*.4 + 7*.05 + 8*.05 + 8*.1 + 10*.1 = 5.45.

Note that we could not use np.mean() to perform this calculation since that function doesn't have a weights parameter.

Refer to the NumPy documentation for a complete explanation of the and functions.

Additional Resources

The following tutorials explain how to calculate other average values in Python:

ARABPSYCHOLOGY.COM