How can I rank items in a NumPy array?

Authored by

stats writer

June 26, 2024

Ranking items in a NumPy array refers to the process of arranging the elements in the array in ascending or descending order based on their values. This can be achieved by using the built-in function "np.argsort()" which returns the indices of the array elements in sorted order. By using this function, the original array can be sorted or ranked based on the desired criteria. This can be useful in various applications such as data analysis, sorting algorithms, and mathematical operations. Overall, ranking items in a NumPy array allows for efficient manipulation and organization of data in a structured manner.

Rank Items in NumPy Array (With Examples)

You can use one of the following methods to calculate the rank of items in a NumPy array:

Method 1: Use argsort() from NumPy

import numpy as np

ranks = np.array(my_array).argsort().argsort()

Method 2: Use rankdata() from SciPy

from scipy.statsimport rankdata

ranks = rankdata(my_array)

The following examples show how to use each method in practice with the following NumPy array:

```
import numpy as np

#define array of values
my_array = np.array()

#view array
print(my_array)
```

Example 1: Rank Items in NumPy Array Using argsort()

The following code shows how to use the argsort() function from NumPy to rank the items in the array:

```
#calculate rank of each item in array
ranks = np.array(my_array).argsort().argsort()

#view ranks
print(ranks)
```


The results show the rank of each item in the original array, with 0 representing the smallest value.

The benefit of this approach is that you don't have to load any extra modules, but the drawback is that

argsort() only has one method for handling ties.

By default, argsort() uses an ordinal method for handling ties which means the tied value that occurs first is automatically given the lower rank.

Example 2: Rank Items in NumPy Array Using rankdata()

The following code shows how to use the rankdata() function from SciPy to rank the items in the array:

```
from scipy.stats import rankdata

#calculate rank of each item in array
ranks = rankdata(my_array)#view ranks
print(ranks)

array()
```

The results show the rank of each item in the original array, with 1 representing the smallest value.

```
from scipy.statsimport rankdata
#calculate rank of each item in array
ranks = rankdata(my_array) - 1
```

```
#view ranks
print(ranks)
```

By default, the rankdata() function assigns average ranks to any values that have ties.

However, you can use the method argument to handle ties in a different way.

For example, the following code shows how to use ordinal as the method for handling ties:

```
from scipy.statsimport rankdata

#calculate rank of each item in array
ranks = rankdata(my_array, method='ordinal') - 1

#view ranks
print(ranks)
```


This produces the same results as the argsort() method from NumPy.

Other methods for handling ties include min, max, and

dense.

Read about each method in the .

The following tutorials explain how to perform other common tasks in NumPy: