How can I count the number of unique values in a NumPy array?

Authored by

stats writer

June 27, 2024

Counting the number of unique values in a NumPy array can be achieved by using the NumPy function "unique". This function takes in an array as input and returns an array of the unique values present in the original array. The length of the returned array can then be used to determine the number of unique values in the original array. This method is useful for data analysis and processing tasks that require identifying and handling unique values in a dataset.

Count Unique Values in NumPy Array (3 Examples)

You can use the following methods to count unique values in a NumPy array:

Method 1: Display Unique Values

np.unique(my_array)

Method 2: Count Number of Unique Values

len(np.unique(my_array))

Method 3: Count Occurrences of Each Unique Value

np.unique(my_array, return_counts=True)

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

import numpy as np

```
#create NumPy array
my_array = np.array()
```

Example 1: Display Unique Values

The following code shows how to display the unique values in the NumPy array:

```
#display unique values
np.unique(my_array)

array()
```

From the output we can see each of the unique values in the NumPy array: 1, 3, 4, 7, 8.

Example 2: Count Number of Unique Values

The following code shows how to count the total number of unique values in the NumPy array:

```
#display total number of unique values
len(np.unique(my_array))

5
```

From the output we can see there are 5 unique values in the NumPy array.

Example 3: Count Occurrences of Each Unique Value

The following code shows how to count the number of occurrences of each unique value in the NumPy array:

```
#count occurrences of each unique value
np.unique(my_array, return_counts=True)

(array(), array())
```

The first array in the output shows the unique values and the second array shows the count of each unique value.

We can use the following code to print this output in a format that is easier to read:

```
#get unique values and counts of each value
unique, counts = np.unique(my_array, return_counts=True)

#display unique values and counts side by side
print(np.asarray((unique, counts)).T)
```

]

From the output we can see:

The value 1 occurs 1 time.The value 3 occurs 2 times.The value 4 occurs 2 times.The value 7 occurs 1 time.The value 8 occurs 2 times.

The following tutorials explain how to perform other common operations in Python: