

How can the issue of 'numpy.float64' object not being interpreted as an integer be fixed?

Authored by
stats writer

May 12, 2024

RECOMMENDED CITATION

stats writer (2024). *How can the issue of 'numpy.float64' object not being interpreted as an integer be fixed?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=144017>

The issue of 'numpy.float64' object not being interpreted as an integer can be fixed by using the "astype" function to convert the float64 object into an integer. This will ensure that the object is properly interpreted as an integer and any errors related to data type mismatch can be avoided. Additionally, checking for any missing or incorrect values in the data can also help resolve this issue. It is important to carefully handle data types and ensure proper conversion in order to avoid any potential errors in data analysis or computations.

Fix: 'numpy.float64' object cannot be interpreted as an integer

One error you may encounter when using NumPy is:

TypeError: 'numpy.float64' object cannot be interpreted as an integer

This error occurs when you supply a float to some function that expects an integer.

The following example shows how to fix this error in practice.

How to Reproduce the Error

Suppose we attempt to use the following for loop to print out various numbers in a NumPy array:

```
import numpy as np
```

```
#define array of values
```

```
data = np.array()
```

```
#use for loop to print out range of values at each index
```

```
for i in range(len(data)):
```

```
print(range(data))
```

TypeError: 'numpy.float64' object cannot be interpreted as an integer

We receive an error because the range() function expects an integer, but the values in the NumPy array are floats.

How to Fix the Error

There are two ways to quickly fix this error:

Method 1: Use the int() Function

One way to fix this error is to simply wrap the call with int() as follows:

```
import numpy as np
```

```
#define array of values
```

```
data = np.array()
```

```
#use for loop to print out range of values at each index  
for i in range(len(data)):  
print(range(int(data)))
```

```
range(0, 3)  
range(0, 4)  
range(0, 5)  
range(0, 7)  
range(0, 10)  
range(0, 11)
```

By using the `int()` function, we convert each float value in the NumPy array to an integer so we avoid the `TypeError` we encountered earlier.

Method 2: Use the `.astype(int)` Function

Another way to fix this error is to first convert the values in the NumPy array to integers:

```
import numpy as np
```

```
#define array of values
```

```
data = np.array()
```

```
#convert array of floats to array of integers
```

```
data_int = data.astype(int)
```

```
#use for loop to print out range of values at each index
```

```
for i in range(len(data)):
```

```
print(range(data))
```

```
range(0, 3)
```

```
range(0, 4)
```

```
range(0, 5)
```

```
range(0, 7)
```

```
range(0, 10)
```

```
range(0, 11)
```

The following tutorials explain how to fix other common errors in Python: