How can I achieve the equivalent of np.where() in Pandas?

Authored by

stats writer

June 28, 2024

The equivalent of np.where() in Pandas is achieved by using the .where() method. This method allows users to specify a condition and return values based on that condition. It works similarly to np.where() by taking in a condition and returning a new series or dataframe with the values that meet that condition. This can be useful for filtering, replacing values, or creating new columns based on specific criteria. By utilizing the .where() method, users can effectively manipulate and analyze data in a Pandas dataframe.

Pandas: Use Equivalent of np.where()

You can use the NumPy function to quickly update the values in a NumPy array using if-else logic.

For example, the following code shows how to update the values in a NumPy array that meet a certain condition:

```
import numpy as np

#create NumPy array of values
x = np.array()

#update valuesin array based on condition
x = np.where((x < 5) | (x > 8), x/2, x)

#view updated array
x

array()
```

If a given value in the array was less than 5 or greater than 8, we divided the value by 2.

Else, we left the value unchanged.

We can perform a similar operation in a pandas DataFrame by using the pandas function, but the syntax is slightly different.

Here's the basic syntax using the NumPy where() function:

x = np.where(condition, value_if_true, value_if_false)

And here's the basic syntax using the pandas where() function:

df = (value_if_false).where(condition, value_if_true)

The following example shows how to use the pandas where() function in practice.

Example: The Equivalent of np.where() in Pandas

Suppose we have the following pandas DataFrame:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'A': ,
'B': })

#view DataFrame
print(df)
```

A B
0 18 5
1 22 7
2 19 7
3 14 9
4 14 12
5 11 9
6 20 9
7 28 4

We can use the following pandas where() function to update the values in column A based on a specific condition:

```
#update values in column A based on condition
df = (df / 2).where(df < 20, df * 2)
```

#view updated DataFrame
print(df)

A B
0 9.0 5
1 44.0 7
2 9.5 7
3 7.0 9
4 7.0 12
5 5.5 9
6 40.0 9
7 56.0 4

If a given value in column A was less than 20, we multiplied the value by 2.

Additional Resources

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