How can NaN values be filled with the mean using Pandas?

Authored by

stats writer

July 1, 2024

NaN (Not a Number) values can be filled with the mean using Pandas by using the "fillna" method. This method allows for filling missing values in a DataFrame with a specified value, in this case, the mean of the column. It is a convenient way to handle missing data and can help maintain the integrity of the data set. By filling NaN values with the mean, it ensures that the data remains consistent and accurate for further analysis. This method is commonly used in data analysis and can be easily implemented using Pandas in Python.

Pandas: Fill NaN Values with Mean (3 Examples)

You can use the fillna() function to replace NaN values in a pandas DataFrame.

Here are three common ways to use this function:

Method 1: Fill NaN Values in One Column with Mean

df = df.fillna(df.mean())

Method 2: Fill NaN Values in Multiple Columns with Mean

df] = df].fillna(df].mean())

Method 3: Fill NaN Values in All Columns with Mean

df = df.fillna(df.mean())

The following examples show how to use each method

in practice with the following pandas DataFrame:

```
import numpy as np
import pandas as pd

#create DataFrame with some NaN values
df = pd.DataFrame({'rating': ,
'points': ,
'assists': ,
'rebounds': })

#view DataFrame
df
```

rating points assists rebounds
0 NaN 25.0 5.0 11
1 85.0 NaN 7.0 8
2 NaN 14.0 7.0 10
3 88.0 16.0 NaN 6
4 94.0 27.0 5.0 6
5 90.0 20.0 7.0 9
6 76.0 12.0 6.0 6
7 75.0 15.0 9.0 10
8 87.0 14.0 9.0 10
9 86.0 19.0 5.0 7

Example 1: Fill NaN Values in One Column with Mean

The following code shows how to fill the NaN values in the rating column with the mean value of the rating column:

#fill NaNs with column mean in 'rating' columndf = df.fillna(df.mean())

#view updated DataFrame
df

rating points assists rebounds
0 85.125 25.0 5.0 11
1 85.000 NaN 7.0 8
2 85.125 14.0 7.0 10
3 88.000 16.0 NaN 6
4 94.000 27.0 5.0 6
5 90.000 20.0 7.0 9
6 76.000 12.0 6.0 6
7 75.000 15.0 9.0 10
8 87.000 14.0 9.0 10
9 86.000 19.0 5.0 7

The mean value in the rating column was 85.125 so

each of the NaN values in the rating column were filled with this value.

Example 2: Fill NaN Values in Multiple Columns with Mean

The following code shows how to fill the NaN values in both the rating and points columns with their respective column means:

```
#fill NaNs with column means in 'rating' and 'points' columns
df] = df].fillna(df].mean())

#view updated DataFrame
df

rating points assists rebounds
0 85.125 25.0 5.0 11
1 85.000 18.0 7.0 8
2 85.125 14.0 7.0 10
3 88.000 16.0 NaN 6
4 94.000 27.0 5.0 6
5 90.000 20.0 7.0 9
6 76.000 12.0 6.0 6
7 75.000 15.0 9.0 10
8 87.000 14.0 9.0 10
```

9 86.000 19.0 5.0 7

Example 3: Fill NaN Values in All Columns with Mean

The following code shows how to fill the NaN values in each column with the column means:

```
#fill NaNs with column means in each column
df = df.fillna(df.mean())

#view updated DataFrame
df
```

```
rating points assists rebounds
0 85.125 25.0 5.000000 11
1 85.000 18.0 7.000000 8
2 85.125 14.0 7.000000 10
3 88.000 16.0 6.666667 6
4 94.000 27.0 5.000000 6
5 90.000 20.0 7.000000 9
6 76.000 12.0 6.000000 6
7 75.000 15.0 9.000000 10
8 87.000 14.0 9.000000 10
9 86.000 19.0 5.000000 7
```

Notice that the NaN values in each column were filled with their column mean.

You can find the complete online documentation for the fillna() function .

Additional Resources

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