How can I reshape a Pandas DataFrame from wide format to long format?

Authored by

stats writer

July 2, 2024

Reshaping a Pandas DataFrame from wide format to long format involves transforming the structure of the data to make it more suitable for analysis. This process involves converting columns into rows and vice versa, resulting in a longer and narrower format. This can be achieved using the melt() function in Pandas, which allows for the selection of specific columns to be converted into a new identifier column and a new value column. This reshaping technique can be useful for data manipulation and visualization purposes, as it allows for easier comparison and analysis of data.

Pandas: Reshape DataFrame from Wide to Long

You can use the following basic syntax to convert a pandas DataFrame from a wide format to a long format:

df = pd.melt(df, id_vars='col1', value_vars=)

In this scenario, col1 is the column we use as an identifier and col2, col3, etc. are the columns we unpivot.

The following example shows how to use this syntax in practice.

Example: Reshape Pandas DataFrame from Wide to Long

Suppose we have the following pandas DataFrame:

import pandas as pd

#create DataFrame

```
df = pd.DataFrame({'team': ,
'points': ,
'assists': ,
'rebounds': })

#view DataFrame
df

team points assists rebounds
0 A 88 12 22
1 B 91 17 28
2 C 99 24 30
3 D 94 28 31
```

We can use the following syntax to reshape this DataFrame from a wide format to a long format:

```
#reshape DataFrame from wide format to long format
df = pd.melt(df, id_vars='team', value_vars=)

#view updated DataFrame
df

team variable value
0 A points 88
```

1 B points 91

2 C points 99

3 D points 94

4 A assists 12

5 B assists 17

6 C assists 24

7 D assists 28

8 A rebounds 22

9 B rebounds 28

10 C rebounds 30

11 D rebounds 31

The DataFrame is now in a long format.

We used the 'team' column as the identifier column and we unpivoted the 'points', 'assists', and 'rebounds' columns.

Note that we can also use the var_name and value_name arguments to specify the names of the columns in the new long DataFrame:

#reshape DataFrame from wide format to long format

df = pd.melt(df, id_vars='team', value_vars=,

```
var_name='metric', value_name='amount')

#view updated DataFrame
df

team metric amount
0 A points 88
1 B points 91
2 C points 99
3 D points 94
4 A assists 12
5 B assists 17
6 C assists 24
7 D assists 28
8 A rebounds 22
9 B rebounds 28
10 C rebounds 30
11 D rebounds 31
```


Note: You can find the complete documentation for the pandas melt() function .

Additional Resources

The following tutorials explain how to perform other

common operations in Python: