How can the Pandas explode() function be used?

Authored by

stats writer

July 1, 2024

The Pandas explode() function is a useful tool for manipulating data in a tabular format. This function allows for the expansion of a column containing lists or arrays into multiple rows, thereby creating a more structured and organized dataset. By using the explode() function, the user can easily analyze and compare individual elements within a list, enabling more in-depth analysis of the data. This function is particularly helpful when dealing with nested data structures, such as JSON or XML files. It allows for easier data manipulation and transformation, leading to more efficient and accurate data analysis.

Use the Pandas explode() Function (With Examples)

You can use the pandas function to transform each element in a list to a row in a DataFrame.

This function uses the following basic syntax:

df.explode('variable_to_explode')

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

Example: Use explode() Function with Pandas DataFrame

Suppose we have the following pandas DataFrame:

import pandas as pd

#create DataFrame
df = pd.DataFrame({'team': , ,],
'position':,

'points': })

#view DataFrame
df

team position points
0 Guard 7
1 Forward 14
2 Center 19

Notice that the team column contains lists of team names.

We can use the explode() function to explode each element in each list into a row:

#explode team column
df.explode('team')

team position points
0 A Guard 7
0 B Guard 7
0 C Guard 7
1 D Forward 14
1 E Forward 14

1 F Forward 14

2 G Center 19

2 H Center 19

2 I Center 19

Notice that the team column no longer contains lists. We "exploded" each element of each list into a row.

Also notice that some rows now have the same index value.

We can use the reset_index() function to reset the index when exploding the team column:

#explode team column and reset index of resulting dataFrame

df.explode('team').reset_index(drop=True)

team position points

0 A Guard 7

1 B Guard 7

2 C Guard 7

3 D Forward 14

4 E Forward 14

5 F Forward 14

6 G Center 19

7 H Center 19

8 I Center 19

Notice that each row now has a unique index value.

Additional Resources

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