How can I compare two DataFrames row by row in Pandas?

Authored by

stats writer

June 25, 2024

The process of comparing two DataFrames row by row in Pandas involves iterating through each row of the DataFrames and comparing the values in each column. This can be done using built-in functions such as "equals()" or by using conditional statements and logical operators. This method allows for a detailed and comprehensive comparison of the data in the DataFrames, providing insights into any similarities or differences between the two. By comparing the data at a granular level, this approach can aid in identifying patterns, trends, and discrepancies in the data.

Pandas: Compare Two DataFrames Row by Row

You can use the following methods to compare two pandas DataFrames row by row:

Method 1: Compare DataFrames and Only Keep Rows with Differences

df_diff = df1.compare(df2, keep_equal=True, align_axis=0)

Method 2: Compare DataFrames and Keep All Rows

df_diff = df1.compare(df2, keep_equal=True, keep_shape=True, align_axis=0)

The following examples show how to use each method with the following pandas DataFrames:

import pandas as pd

```
#create first DataFrame
df1 = pd.DataFrame({'team': ,
'points': ,
'assists': })

print(df1)

team points assists
0 A 18 5
1 B 22 7
2 C 19 7
3 D 14 9

#create second DataFrame
df2 = pd.DataFrame({'team': ,
'points': ,
'assists': })

print(df2)

team points assists
0 A 18 5
1 B 30 7
2 C 19 7
3 E 20 9
```

Example 1: Compare DataFrames and Only Keep Rows with Differences

The following code shows how to compare the two DataFrames row by row and only keep the rows that have differences in at least one column:

```
#compare DataFrames and only keep rows with differences
df_diff = df1.compare(df2, keep_equal=True, align_axis=0)

#view results
print(df_diff)
```

```
  team points
1 self B 22
  other B 30
3 self D 14
  other E 20
```

We can see that the DataFrames have two rows that are different.

In particular, we can see that the rows in index positions 1 and 3 of each DataFrame have different

values in at least one column.

The values in the self row show the values from the first DataFrame while the values in the other row show the values from the second DataFrame.

For example, we can see:

The row in index position 1 of the first DataFrame contains B in the team column and 22 in the points column.
The row in index position 1 of the second DataFrame contains B in the team column and 30 in the points column.

Note: The argument keep_equal=True tells pandas to keep values that are equal. Otherwise, equal values are shown as NaNs.

Example 2: Compare DataFrames and Keep All Rows

```
#compare DataFrames and keep all rows
df_diff = df1.compare(df2, keep_equal=True, keep_shape=True, align_axis=0)

#view results
```

```
print(df_diff)

team points assists
0 self A 18 5
other A 18 5
1 self B 22 7
other B 30 7
2 self C 19 7
other C 19 7
3 self D 14 9
other E 20 9
```

The resulting DataFrame contains all of the rows and columns from the original DataFrames.

Note #1: The compare() function assumes that both DataFrames have the same dimensions.

Note #2: You can find the complete documentation for the pandas compare() function .

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