How can we compare two DataFrames in Pandas?

Authored by

stats writer

April 17, 2024

Comparing two DataFrames in Pandas involves checking for similarities and differences between the two datasets. This can be done by using various methods such as the equals() function, which checks for equality between the two DataFrames, or the compare() function, which highlights the differences between the two DataFrames. Additionally, users can also use logical operators and conditional statements to compare specific columns or rows within the DataFrames. Overall, comparing two DataFrames in Pandas allows for a comprehensive analysis of the data and assists in identifying any discrepancies or patterns between the two datasets.

Compare Two DataFrames in Pandas

Often you might be interested in comparing the values between two pandas DataFrames to spot their similarities and differences.

This tutorial explains how to do so.

Example: Comparing Two DataFrames in Pandas

Suppose we have the following two pandas DataFrames that each contain data about four basketball players:

```
import pandas as pd

#define DataFrame 1
df1 = pd.DataFrame({'player': ,
'points': ,
'assists': })
df1
```

player points assists
0 A 12 4
1 B 15 6
2 C 17 7
3 D 24 88

#define DataFrame 2
df2 = pd.DataFrame({'player': ,
'points': ,
'assists': })
df2

player points assists
0 A 12 7
1 B 24 8
2 C 26 10
3 D 29 13

Example 1: Find out if the two DataFrames are identical.

We can first find out if the two DataFrames are identical by using the function:

#see if two DataFrames are identical
df1.equals(df2)

False

The two DataFrames do not contain the exact same values, so this function correctly returns False.

Example 2: Find the differences in player stats between the two DataFrames.

We can find the differences between the assists and points for each player by using the pandas function:

```
#subtract df1 from df2
df2.set_index('player').subtract(df1.set_index('player'))
```

```
points assists
player
A 0 3
B 9 2
C 9 3
D 5 5
```


The way to interpret this is as follows:

Player A had the same amount of points in both DataFrames, but they had 3 more assists in DataFrame

2.Player B had 9 more points and 2 more assists in DataFrame 2 compared to DataFrame 1.Player C had 9 more points and 3 more assists in DataFrame 2 compared to DataFrame 1.Player D had 5 more points and 5 more assists in DataFrame 2 compared to DataFrame 1.

Example 3: Find all rows that only exist in one DataFrame.

We can use the following code to obtain a complete list of rows that only appear in one DataFrame:

```
#outer merge the two DataFrames, adding an indicator column called 'Exist'
diff_df = pd.merge(df1, df2, how='outer', indicator='Exist')

#find which rows don't exist in both DataFrames
diff_df = diff_df.loc != 'both']
diff_df
```

player points assists Exist
0 A 12 4 left_only
1 B 15 6 left_only

2 C 17 7 left_only

3 D 24 8 left_only

4 A 12 7 right_only

5 B 24 8 right_only

6 C 26 10 right_only

7 D 29 13 right_only

The column titled "Exist" conveniently tells us which DataFrame each row uniquely appears in.