How can I check if two DataFrames are equal in Pandas?

Authored by

stats writer

June 26, 2024

To check if two DataFrames are equal in Pandas, you can use the built-in function "equals()". This function compares the two DataFrames and returns a boolean value, True if they are equal and False if they are not. It compares the index, column labels, and values of the DataFrames to determine equality. This method is useful for verifying data integrity and ensuring that the data in both DataFrames is consistent. Additionally, you can use the "compare()" function to identify the differences between the two DataFrames.

Pandas: Check if Two DataFrames Are Equal

You can use the following basic syntax to check if two pandas DataFrames are equal:

df1.equals(df2)

This will return a value of True or False.

If two DataFrames are not equal, then you can use the following syntax to find the rows in the second DataFrame that do not exist in the first DataFrame:

```
#perform outer join on two DataFrames
all_df = df1.merge(df2, indicator=True, how='outer')

#find which rows only exist in second DataFrame
only_df2 = all_df == 'right_only']
only_df2 = only_df2.drop('_merge', axis=1)
```

The following example shows how to use this syntax in

practice.

Example: Check if Two pandas DataFrames Are Equal

Suppose we have the following two pandas DataFrames:

```
import pandas as pd

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

print(df1)

team points
0 A 12
1 B 15
2 C 22
3 D 29
4 E 24

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

print(df2)
```

team points
0 A 12
1 D 29
2 F 15
3 G 19
4 H 10

We can use the following syntax to check if the two DataFrames are equal:

#check if two DataFrames are equaldf1.equals(df2)

False

The output returns False, which means the two DataFrames are not equal.

We can then use the following syntax to find which rows exist in the second DataFrame but not in the first:

#perform outer join on two DataFrames
all_df = df1.merge(df2, indicator=True, how='outer')

#find which rows only exist in second DataFrame
only_df2 = all_df == 'right_only']

```
only_df2 = only_df2.drop('_merge', axis=1)

#view results
print(only_df2)

team points
5 F 15
6 G 19
7 H 10
```

From the output we can see that there are three rows in the second DataFrame that do not exist in the first DataFrame.

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