How can I remove duplicate rows in a Pandas DataFrame while keeping the row with the maximum value?

Authored by

stats writer

June 27, 2024

This process involves using Pandas, a popular data analysis library in Python, to remove duplicate rows in a DataFrame while retaining the row with the highest value. This can be achieved by using the drop_duplicates() function, which identifies and removes duplicate rows based on specified columns. By specifying the "keep" parameter as "first", the function will keep the first occurrence of a duplicate row and remove all subsequent duplicates. Alternatively, by specifying "keep" as "last", the function will keep the last occurrence of a duplicate row. This method is useful for cleaning and organizing data in a DataFrame, ensuring the retention of the most relevant information.

Pandas: Remove Duplicates but Keep Row with Max Value

You can use the following methods to remove duplicates in a pandas DataFrame but keep the row that contains the max value in a particular column:

Method 1: Remove Duplicates in One Column and Keep Row with Max

```
df.sort_values('var2',
ascending=False).drop_duplicates('var1').sort_index()
```

Method 2: Remove Duplicates in Multiple Columns and Keep Row with Max

```
df.sort_values('var3',
ascending=False).drop_duplicates().sort_index()
```

The following examples show how to use each method

in practice.

Example 1: Remove Duplicates in One Column and Keep Row with Max

Suppose we have the following pandas DataFrame that contains information about points scored by basketball players on various teams:

```
import pandas as pd

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

#view DataFrame
print(df)
```

team points
0 A 20
1 A 24
2 A 28
3 B 30
4 B 14
5 B 19
6 C 29

7 C 40
8 C 22

We can use the following syntax to drop rows with duplicate team names but keep the rows with the max values for points:

```
#drop duplicate teams but keeps row with max points
df_new = df.sort_values('points', ascending=False).drop_duplicates('team').sort_index()

#view DataFrame
print(df_new)
```

team points
2 A 28
3 B 30
7 C 40

Each row with a duplicate team name has been dropped, but the rows with the max value for points have been kept for each team.

Example 2: Remove Duplicates in Multiple Columns and Keep Row

with Max

Suppose we have the following pandas DataFrame:

import pandas as pd

#create DataFrame
df = pd.DataFrame({'team': ,
'position': ,
'points': })

#view DataFrame
print(df)

team position points
0 A G 20
1 A G 24
2 A F 28
3 B G 30
4 B F 14
5 B F 19
6 C G 29
7 C G 40
8 C F 22

We can use the following syntax to drop rows with duplicate team*and*position names but keep the rows with the max values for points:

```
#drop rows with duplicate team and positions but keeps row with max points
df_new = df.sort_values('points', ascending=False).drop_duplicates().sort_index()

#view DataFrame
print(df_new)

team position points
1 A G 24
2 A F 28
3 B G 30
5 B F 19
7 C G 40
8 C F 22
```

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