How can I export a Pandas DataFrame to a text file?

Authored by

stats writer

June 25, 2024

Exporting a Pandas DataFrame to a text file is a simple process that allows you to save your data in a format that is easily readable and shareable. To do so, you can use the `to_csv()` function in Pandas, which converts the DataFrame into a comma-separated values (CSV) file. This file can then be opened in any text editor or spreadsheet program. This method is particularly useful when working with large datasets as it reduces the file size and can be easily manipulated for further analysis. Overall, exporting a Pandas DataFrame to a text file provides a convenient and efficient way to store and share your data.

Export Pandas DataFrame to Text File

You can use the following syntax to export a pandas DataFrame to a text file:

```
#specify path for export
path = r'c:data_foldermy_data.txt'

#export DataFrame to text file
with open(path, 'a') as f:
df_string = df.to_string(header=False, index=False)
f.write(df_string)
```

The argument header=False tells pandas not to include the header row in the text file and index=False tells pandas not to include the index column in the text file.

Feel free to leave out these arguments if you'd like to include the header row or the index column in the text file.

The following example shows how to use this syntax to export a pandas DataFrame to a text file in practice.

Example: Export Pandas DataFrame to Text File

Suppose we have the following pandas DataFrame that contains information about various basketball players:

```
import pandas as pd

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

#view DataFrame
print(df)
```

team points assists rebounds
0 A 18 5 11
1 B 22 7 8
2 C 19 7 10
3 D 14 9 6
4 E 14 12 6
5 F 11 9 5

6 G 20 9 9

7 H 28 4 12

We can use the following syntax to export this DataFrame to a text file called basketball_data.txt:

```
#specify path for export
path = r'c:data_folderbasketball_data.txt'

#export DataFrame to text file
with open(path, 'a') as f:
df_string = df.to_string(header=False, index=False)
f.write(df_string)
```

If I navigate to the folder where I exported this file, I can view the text file:

```
A 18  5 11
B 22  7  8
C 19  7 10
D 14  9  6
E 14 12  6
F 11  9  5
G 20  9  9
H 28  4 12
```


The values in the text file match the values in the pandas DataFrame.

Notice that the header row and the index column have both been removed from the DataFrame, just as we specified.

If you'd like to keep the header row and index column in the text file, you can instead use the following syntax:

#specify path for export
path = r'c:data_folderbasketball_data.txt'

#export DataFrame to text file (keep header row and

index column)
with open(path, 'a') as f:
df_string = df.to_string()
f.write(df_string)

If I navigate to the folder where I exported this file, I can view the text file:

```
  team  points  assists  rebounds
0    A      18        5        11
1    B      22        7         8
2    C      19        7        10
3    D      14        9         6
4    E      14       12         6
5    F      11        9         5
6    G      20        9         9
7    H      28        4        12
```


Notice that the header row and index column are both included in the text file.

The following tutorials explain how to perform other

common tasks in pandas: