Check dtype for All Columns in DataFrame

Authored by

stats writer

November 29, 2025

Checking the dtype for all columns in a DataFrame is a way to verify that the data stored in each column is of the correct type. This helps to ensure that the data is well formatted and can be used properly in analysis and other operations. It also helps to identify potential issues that may arise when working with the data. For example, if a column contains numeric values but is stored as a string, it may cause errors when performing calculations.

You can use the following methods to check the data type () for columns in a pandas DataFrame:

Method 1: Check dtype of One Column

df.column_name.dtype

Method 2: Check dtype of All Columns

df.dtypes

Method 3: Check which Columns have Specific dtype

df.dtypes

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

import pandas as pd

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

#view DataFrame
print(df)

team points assists all_star
0 A 18 5 True
1 B 22 7 False
2 C 19 7 False
3 D 14 9 True
4 E 14 12 True
5 F 11 9 True

Example 1: Check dtype of One Column

We can use the following syntax to check the data type of just the **points** column in the DataFrame:

```
#check dtype of points column
df.points.dtype
```

```
dtype('int64')
```

From the output we can see that the **points** column has a data type of integer.

Example 2: Check dtype of All Columns

We can use the following syntax to check the data type of all columns in the DataFrame:

```
#check dtype of all columns
df.dtypes
```

```
team object
points int64
assists int64
all_star bool
dtype: object
```

From the output we can see:

team column: object (this is the same as a string)
points column: integer
assists column: integer
all_star column: boolean

By using this one line of code, we can see the data type of each column in the DataFrame.

Example 3: Check which Columns have Specific dtype

We can use the following syntax to check which columns in the DataFrame have a data type of int64:

```
#show all columns that have a class of int64
df.dtypes
```

points int64
assists int64
dtype: object

From the output we can see that the **points** and **assists** columns both have a data type of int64.

We can use similar syntax to check which columns have other data types.

For example, we can use the following syntax to check which columns in the DataFrame have a data type of object:

#show all columns that have a class of object (i.e. string)
df.dtypes

team object
dtype: object

We can see that only the **team** column has a data type of 'O', which stands for object.

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