Python

Pandas Convert Column to Int

1 year ago • by Cherry Mae Barolo

Pandas is a free and open-source Python library that provides fast, flexible, and expressive data structures that make working with scientific data easy.

Pandas is one of Python's most valuable data analysis and manipulation packages.

It offers features such as custom data structures that are built on top of Python.

This article will discuss converting a column from one data type to an int type within a Pandas DataFrame.

Setting Up Pandas

Before diving into how to perform the conversion operation, we need to setup Pandas in our Python environment.

If you are using the base environment in the Anaconda interpreter, chances are you have Pandas installed.

However, on a native Python install, you will need to install it manually.

You can do that by running the command:

```
$ pip install pandas
```

On Linux, run

```
$ sudo pip3 install pandas
```

```
(base) debian@master:~$ sudo pip3 install pandas
[sudo] password for debian:
Collecting pandas
  Downloading pandas-1.4.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.7 MB)
     |████████████████████████████████| 11.7 MB 2.8 MB/s
Collecting numpy>=1.18.5
  Downloading numpy-1.22.4-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.8 MB)
     |████████████████████████████████| 16.8 MB 695 kB/s
Collecting python-dateutil>=2.8.1
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
     |████████████████████████████████| 247 kB 2.7 MB/s
Collecting pytz>=2020.1
  Downloading pytz-2022.1-py2.py3-none-any.whl (503 kB)
     |████████████████████████████████| 503 kB 3.1 MB/s
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.8.1->pandas) (1.16.
0)
Installing collected packages: pytz, python-dateutil, numpy, pandas
Successfully installed numpy-1.22.4 pandas-1.4.2 python-dateutil-2.8.2 pytz-2022.1
```

In Anaconda or Miniconda environments, install pandas with conda.

```
$ conda install pandas
$ sudo conda install pandas
```

Pandas Create Sample DataFrame

Let us set up a sample DataFrame for illustration purpo
the code below or use your DataFrame.

```
import pandas as pd
df = pd.DataFrame({'id': ['1', '2', '3', '4', '5'],
                   'name': ['Marja Jérôme', 'Alexios Shiva', 'Mohan Famke', 'Lovrenco
Ilar', 'Steffen Angus'],
                   'points': ['50000', '70899', '70000', '81000', '110000']})
```

Once the DataFrame is created, we can check the data.

```
df
```

	id	name	points
0	1	Marja Jérôme	50000
1	2	Alexios Shiva	70899
2	3	Mohan Famke	70000
3	4	Lovrenco Ilar	81000
4	5	Steffen Angus	110000

Pandas Show Column Type

It is good to know if the existing type can be cast to an int before converting a column from one type to an int.

For example, attempting to convert a column containing names cannot be converted to an int.

We can view the type of a DataFrame using the dtypes property

Use the syntax:

```
DataFrame.dtypes
```

In our sample DataFrame, we can get the column types as:

```
df.dtypes
id        object
name      object
points    object
dtype: object
```

We can see from the output above that none of the columns hold an int type.

Pandas Convert Column From String to Int.

To convert a single column to an int, we use the astype() function and pass the target data type as the parameter.

The function syntax:

```
DataFrame.astype(dtype, copy=True, errors='raise')
```

1. dtype – specifies the Python type or a NumPy dtyp
2. copy – allows you to return a copy of the object ins
3. errors – specifies the action in case of error. By def

In our sample DataFrame, we can convert the id column to int type using the astype() function as shown in the code below:

```
df['id'] = df['id'].astype(int)
```

The code above specifies the ‘id’ column as the target object. We then pass an int as the type to the astype() function.

We can check the new data type for each column in the DataFrame:

```
df.dtypes
id          int32
name       object
points     object
dtype: object
```

The id column has been converted to an int while the rest remains unchanged.

Pandas Convert Multiple Columns to Int

The astype() function allows us to convert more than one column and convert them to a specific type.

For example, we can run the following code to convert the id and points columns to int type.

```
df[['id', 'points']] = df[['id', 'points']].astype(i
```

Here, we are specifying multiple columns using the squ
us to convert the columns to the data type specified in

```
name      object
points     int32
dtype: object
```

We can now see that the id and points column has been converted to int32 type.

Pandas Convert Multiple Columns to Multiple Types

The astype() function allows us to specify a column and target type as a dictionary.

Assume that we want to convert the id column to int32 and the points column to float64.

We can run the following code:

```
convert_to = {"id": int, "points": float}
df = df.astype(convert_to)
```

In the code above, we start by defining a dictionary holding the target column as the key and the target type as the value.

We then use the astype() function to convert the columns in the dictionary to the set types.

Checking the column types should return:

```
df.dtypes
id          int32
name       object
points    float64
dtype: object
```

Note that the id column is int32 and the points column

Pandas Convert Column to Int – to_numeric()

Pandas also provides us with the to_numeric() function. This function allows us to convert a column to a numeric type.

The function syntax is as shown:

```
pandas.to_numeric(arg, errors='raise', downcast=None)
```

For example, to convert the id column to numeric in our sample DataFrame, we can run:

```
df['id'] = pd.to_numeric(df['id'])
```

The code should take the id column and convert it into an int type.

Pandas Convert DataFrame to Best Possible Data Type

The convert_dtypes() function in Pandas allows us to convert an entire DataFrame to the nearest possible type.

The function syntax is as shown:

```
DataFrame.convert_dtypes(infer_objects=True, convert_string=True,
                         convert_integer=True, convert_boolean=True,
convert_floating=True)
```

You can check the docs in the resource below:

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.convert_dtypes.html

For example, to convert our sample DataFrame to the nearest possible type, we can run:

```
df = df.convert_dtypes()
```

If we check the type:

```
df.dtypes
id          Int32
name       string
points      Int64
dtype: object
```

You will notice that each column has been converted to the nearest appropriate type. For example, the function converts small ints to int32 type.

Likewise, the names column is converted to string type as it holds string values.

Finally, since the points column holds larger integers, it is converted to an int64 type.

Conclusion

In this article, we gave detailed methods and examples
DataFrame from one type to another.

Explore More

search text

ABOUT THE AUTHOR

Cherry Mae Barolo

View all posts

Python IO Module

Pandas Floor

Python File readable() Method

Python Shutil Move()

Pandas Get_Dummies()

Pandas Standard Deviation

Python Math Isclose() Method