Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

Convert pandas dataframe to numpy array, preserving index

I am interested in knowing how to convert a pandas dataframe into a numpy array, including the index, and set the dtypes.

dataframe:

```
label   A    B    C
ID
1   NaN  0.2  NaN
2   NaN  NaN  0.5
3   NaN  0.2  0.5
4   0.1  0.2  NaN
5   0.1  0.2  0.5
6   0.1  NaN  0.5
7   0.1  NaN  NaN
```

convert df to array returns:

```
array([[ nan,  0.2,  nan],
       [ nan,  nan,  0.5],
       [ nan,  0.2,  0.5],
       [ 0.1,  0.2,  nan],
       [ 0.1,  0.2,  0.5],
       [ 0.1,  nan,  0.5],
       [ 0.1,  nan,  nan]])
```

However, I would like:

Join Stack Overflow to learn, share knowledge, and build your career. Email Sign Up OR SIGN IN WITH Google Facebook

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
       [ 7, 0.1,  nan,  nan]],
      dtype=[('ID', '<i4'), ('A', '<f8'), ('B', '<f8'), ('B', '<f8')])
```

(or similar)

Any suggestions on how to accomplish this? (I don't know if I need 1D or 2D array at this point.) I've seen a few posts that touch on this, but nothing dealing specifically with the dataframe.index.

I am writing the dataframe disk using to_csv (and reading it back in to create array) as a workaround, but would prefer something more eloquent than my new-to-pandas kludging.

python arrays numpy pandas type-conversion

edited Jun 16 '15 at 23:06
smci
11.1k 5 55 90

asked Nov 2 '12 at 0:57
mister.nobody.nz
541 2 5 3

9 Answers

To convert a pandas dataframe (df) to a numpy ndarray, use this code:

```
df=df.values
```

df now becomes a numpy ndarray.

answered May 5 '16 at 5:29
User456898
1,330 7 25

5 This doesn't work, the dtype is still erased (you lose the names). – Joseph Garvin Feb 13 at 17:05

3 This does not answers the question. – An economist Aug 8 at 15:39

Answers my question lol – Malachi Bazar Dec 26 at 19:42

Join Stack Overflow to learn, share knowledge, and build your career. Email Sign Up OR SIGN IN WITH Google Facebook ✕

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
numpyMatrix = df.as_matrix()
```

edited Jul 17 '14 at 1:22

answered Jul 17 '14 at 1:13

ZJS

2,349 7 15

14 This does not give a structured array, all columns are of dtype `object` . – sebix Oct 9 '14 at 11:24

I would just chain the DataFrame.reset_index() and DataFrame.values functions to get the Numpy representation of the dataframe, including the index:

```
In [8]: df
Out[8]:
          A         B         C
0 -0.982726  0.150726  0.691625
1  0.617297 -0.471879  0.505547
2  0.417123 -1.356803 -1.013499
3 -0.166363 -0.957758  1.178659
4 -0.164103  0.074516 -0.674325
5 -0.340169 -0.293698  1.231791
6 -1.062825  0.556273  1.508058
7  0.959610  0.247539  0.091333

[8 rows x 3 columns]

In [9]: df.reset_index().values
Out[9]:
array([[ 0.        , -0.98272574,  0.150726  ,  0.69162512],
       [ 1.        ,  0.61729734, -0.47187926,  0.50554728],
       [ 2.        ,  0.4171228 , -1.35680324, -1.01349922],
       [ 3.        , -0.16636303, -0.95775849,  1.17865945],
       [ 4.        , -0.16410334,  0.0745164 , -0.67432474],
       [ 5.        , -0.34016865, -0.29369841,  1.23179064],
       [ 6.        , -1.06282542,  0.55627285,  1.50805754],
       [ 7.        ,  0.95961001,  0.24753911,  0.09133339]])
```

To get the dtypes we'd need to transform this ndarray into a structured array using view:

Join Stack Overflow to learn, share knowledge, and build your career. Email Sign Up OR SIGN IN WITH Google Facebook ✕

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
       ( 1,  0.61729734, -0.47187926,  0.50554728),
       ( 2,  0.4171228 , -1.35680324, -1.01349922),
       ( 3, -0.16636303, -0.95775849,  1.17865945),
       ( 4, -0.16410334,  0.0745164 , -0.67432474),
       ( 5, -0.34016865, -0.29369841,  1.23179064),
       ( 6, -1.06282542,  0.55627285,  1.50805754),
       ( 7,  0.95961001,  0.24753911,  0.09133339),
       dtype=[('index', '<i8'), ('A', '<f8'), ('B', '<f8'), ('C', '<f8')])
```

edited Mar 26 '14 at 7:35

answered Mar 26 '14 at 6:23

MonkeyButter
1,131 1 13 24

1 This should be marked as the complete answer, then... – durbachit Nov 26 '16 at 4:05

1 the only thing missing in this answer is how to construct the dtype from the data frame so that you can write a generic function – Joseph Garvin Feb 13 at 17:07

You can use the `to_records` method, but have to play around a bit with the dtypes if they are not what you want from the get go. In my case, having copied your DF from a string, the index type is string (represented by an `object` dtype in pandas):

```
In [102]: df
Out[102]:
label    A    B    C
ID
1      NaN  0.2  NaN
2      NaN  NaN  0.5
3      NaN  0.2  0.5
4      0.1  0.2  NaN
5      0.1  0.2  0.5
6      0.1  NaN  0.5
7      0.1  NaN  NaN

In [103]: df.index.dtype
Out[103]: dtype('object')
In [104]: df.to_records()
Out[104]:
```

Join Stack Overflow to learn, share knowledge, and build your career. Email Sign Up OR SIGN IN WITH Google Facebook ✕

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
Out[106]: dtype([('index', '|O8'), ('A', '<f8'), ('B', '<f8'), ('C', '<f8')])
```

Converting the recarray dtype does not work for me, but one can do this in Pandas already:

```
In [109]: df.index = df.index.astype('i8')
In [111]: df.to_records().view([('ID', '<i8'), ('A', '<f8'), ('B', '<f8'), ('C',
'<f8')])
Out[111]:
rec.array([(1, nan, 0.2, nan), (2, nan, nan, 0.5), (3, nan, 0.2, 0.5),
       (4, 0.1, 0.2, nan), (5, 0.1, 0.2, 0.5), (6, 0.1, nan, 0.5),
       (7, 0.1, nan, nan)],
      dtype=[('ID', '<i8'), ('A', '<f8'), ('B', '<f8'), ('C', '<f8')])
```

Note that Pandas does not set the name of the index properly (to `ID`) in the exported record array (a bug?), so we profit from the type conversion to also correct for that.

At the moment Pandas has only 8-byte integers, `i8` , and floats, `f8` (see this issue).

answered Nov 2 '12 at 10:16

meteore
1,761 2 13 12

2 To get the sought-after structured array (which has better performance than a recarray) you just pass the recarray to the `np.array` constructor. – meteore Nov 2 '12 at 10:19

Index name bug: github.com/pydata/pandas/issues/2161 – Wes McKinney Nov 2 '12 at 14:39

We just put in a fix for setting the name of the index shown above. – Chang She Nov 2 '12 at 22:23

Here is my approach to making a structure array from a pandas DataFrame.

Create the data frame

```
import pandas as pd
import numpy as np
import six
```

Join Stack Overflow to learn, share knowledge, and build your career. Email Sign Up OR SIGN IN WITH Google Facebook ✕

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
columns = {'A':A, 'B':B, 'C':C}
df = pd.DataFrame(columns, index=ID)
df.index.name = 'ID'
print(df)

      A    B    C
ID
1   NaN  0.2  NaN
2   NaN  NaN  0.5
3   NaN  0.2  0.5
4   0.1  0.2  NaN
5   0.1  0.2  0.5
6   0.1  NaN  0.5
7   0.1  NaN  NaN
```

Define function to make a numpy structure array (not a record array) from a pandas DataFrame.

```
def df_to_sarray(df):
    """
    Convert a pandas DataFrame object to a numpy structured array.
    This is functionally equivalent to but more efficient than
    np.array(df.to_array())

    :param df: the data frame to convert
    :return: a numpy structured array representation of df
    """

    v = df.values
    cols = df.columns

    if six.PY2:  # python 2 needs .encode() but 3 does not
        types = [(cols[i].encode(), df[k].dtype.type) for (i, k) in
enumerate(cols)]
    else:
        types = [(cols[i], df[k].dtype.type) for (i, k) in enumerate(cols)]
    dtype = np.dtype(types)
    z = np.zeros(v.shape[0], dtype)
    for (i, k) in enumerate(z.dtype.names):
        z[k] = v[:, i]
    return z
```

Use `reset_index` to make a new data frame that includes the index as part of its data

Join Stack Overflow to learn, share knowledge, and build your career. Email Sign Up OR SIGN IN WITH Google Facebook ×

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
sa

array([(1L, nan, 0.2, nan), (2L, nan, nan, 0.5), (3L, nan, 0.2, 0.5),
       (4L, 0.1, 0.2, nan), (5L, 0.1, 0.2, 0.5), (6L, 0.1, nan, 0.5),
       (7L, 0.1, nan, nan)],
      dtype=[('ID', '<i8'), ('A', '<f8'), ('B', '<f8'), ('C', '<f8')])
```

EDIT: Updated df_to_sarray to avoid error calling .encode() with python 3. Thanks to Joseph Garvin and halcyon for their comment and solution.

edited Jun 23 at 14:28

answered Jun 11 '15 at 5:38

Phil
2,329 11 31

doesn't work for me, error: TypeError: data type not understood – Joseph Garvin Feb 13 at 17:55

Thanks for your comment and to halcyon for the correction. I updated my answer so I hope it works for you now. – Phil Jun 23 at 14:30

Further to meteore's answer, I found the code

```
df.index = df.index.astype('i8')
```

doesn't work for me. So I put my code here for the convenience of others stuck with this issue.

```
city_cluster_df = pd.read_csv(text_filepath, encoding='utf-8')
# the field 'city_en' is a string, when converted to Numpy array, it will be an
object
city_cluster_arr =
city_cluster_df[['city_en','lat','lon','cluster','cluster_filtered']].to_records()
descr=city_cluster_arr.dtype.descr
# change the field 'city_en' to string type (the index for 'city_en' here is 1
because before the field is the row index of dataframe)
descr[1]=(descr[1][0], "S20")
newArr=city_cluster_arr.astype(np.dtype(descr))
```

Join Stack Overflow to learn, share knowledge, and build your career. Email Sign Up OR SIGN IN WITH Google Facebook ✕

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

thanks for Phil's answer, it's great.

reply for

> doesn't work for me, error: TypeError: data type not understood – Joseph Garvin Feb 13 at 17:55

I use python 3, and get the same Error. and then I delete .encode() , then expression is as following.

```
types = [(cols[i], df[k].dtype.type) for (i, k) in enumerate(cols)]
```

then it works.

answered Jun 10 at 14:00

Renke

53 1 10

Thank you for your correction. I updated my answer above to use the six package to avoid the `.encode()` for python 3. – Phil Jun 23 at 14:31

Just had a similar problem when exporting from dataframe to arcgis table and stumbled on a solution from usgs (https://my.usgs.gov/confluence/display/cdi/pandas.DataFrame+to+ArcGIS+Table). In short your problem has a similar solution:

```
df
Out[109]:
      A    B    C
ID
1   NaN  0.2  NaN
```

Join Stack Overflow to learn, share knowledge, and build your career. Email Sign Up OR SIGN IN WITH Google Facebook ✕

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
7   0.1  NaN  NaN

np_data = np.array(np.rec.fromrecords(df.values))
np_names = df.dtypes.index.tolist()
np_data.dtype.names = tuple([name.encode('UTF8') for name in np_names])

np_data
Out[113]:
array([( nan,  0.2,  nan), ( nan,  nan,  0.5), ( nan,  0.2,  0.5),
       ( 0.1,  0.2,  nan), ( 0.1,  0.2,  0.5), ( 0.1,  nan,  0.5),
       ( 0.1,  nan,  nan)],
      dtype=(numpy.record, [('A', '<f8'), ('B', '<f8'), ('C', '<f8')]))
```

answered Nov 10 at 14:41

lars
1 1

Two ways to convert the data-frame to its Numpy-array representation.

- `mah_np_array = df.as_matrix(columns=None)`
- `mah_np_array = df.values`

Doc: https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.as_matrix.html

answered 2 days ago

Priyanshu Chauhan
1,440 11 23

Join Stack Overflow to learn, share knowledge, and build your career. Email Sign Up OR SIGN IN WITH Google Facebook ✕