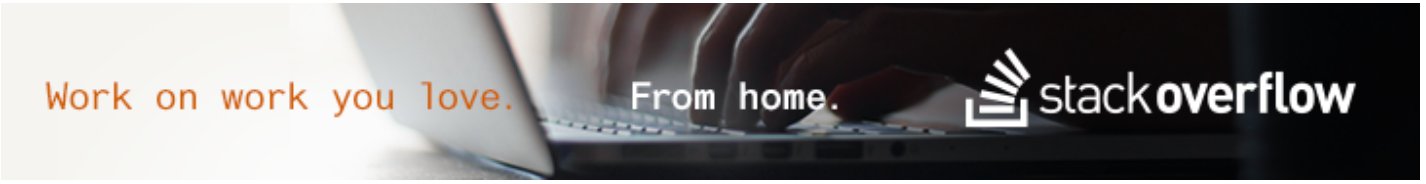
# Encoding column labels in Pandas for machine learning




I am working on car evaulation dataset for machine learning and the dataset is like this

```
buying,maint,doors,persons,lug_boot,safety,class
vhigh,vhigh,2,2,small,low,unacc
vhigh,vhigh,2,2,small,med,unacc
vhigh,vhigh,2,2,small,high,unacc
vhigh,vhigh,2,2,med,low,unacc
vhigh,vhigh,2,2,med,med,unacc
vhigh,vhigh,2,2,med,high,unacc
```

i want to convert these strings to unique enumerated integers columnwise. i see that pandas.factorize() is the way to go, but it only works on one column. how do i factorize the dataframe in one go with one command.

i tried lambda function and it is not working.

df.apply(lambda c:pd.factorize(c),axis=1)

Output:

```
0     ([0, 0, 1, 1, 2, 3, 4], [vhigh, 2, small, low,...
1     ([0, 0, 1, 1, 2, 3, 4], [vhigh, 2, small, med,...
2     ([0, 0, 1, 1, 2, 3, 4], [vhigh, 2, small, high...
3     ([0, 0, 1, 1, 2, 3, 4], [vhigh, 2, med, low, u...
4       ([0, 0, 1, 1, 2, 2, 3], [vhigh, 2, med, unacc])
5     ([0, 0, 1, 1, 2, 3, 4], [vhigh, 2, med, high, ...
```

i see the encoded values but cant pull that out from above array

python pandas machine-learning scikit-learn

edited Aug 27 '14 at 15:28
Fred Foo
223k 33 417 595

asked Aug 27 '14 at 14:56
pbu
591 8 30

Don't you want to do `df.apply(pd.factorize)` instead? – EdChum Aug 27 '14 at 15:17

## 1 Answer

Factorize returns a tuple of (values, labels). You'll just want the values in the DataFrame.

```
In [26]: cols = ['buying', 'maint', 'lug_boot', 'safety', 'class']

In [27]: df[cols].apply(lambda x: pd.factorize(x)[0])
Out[27]:
   buying  maint  lug_boot  safety  class
0       0      0         0       0      0
1       0      0         0       1      0
2       0      0         0       2      0
3       0      0         1       0      0
4       0      0         1       1      0
5       0      0         1       2      0
```

Then concat that to the numeric data.

A word of warning though: this implies that "low" safety and "high" safety are the same distance from "med" safety. You might be better off using `pd.get_dummies` :

```
In [37]: dummies = []

In [38]: for col in cols:
   ....:     dummies.append(pd.get_dummies(df[col]))
   ....:

In [39]: pd.concat(dummies, axis=1)
Out[39]:
   vhigh  vhigh  med  small  high  low  med  unacc
0      1      1    0      1     0    1    0      1
1      1      1    0      1     0    0    1      1
2      1      1    0      1     1    0    0      1
3      1      1    1      0     0    1    0      1
4      1      1    1      0     0    0    1      1
5      1      1    1      0     1    0    0      1
```

`get_dummies` has some optional parameters to control the naming, which you'll probably want.

answered Aug 27 '14 at 15:20

TomAugspurger
**8,332** 1 19 27

Very useful :) Thank you so much. – pbu Aug 28 '14 at 17:34