

```
import numpy as np
import pandas as pd

Data = {
    'name' : ["abhi","ram","raj","rohan","abhi","raj","rohan"],
    'nae' : [10,11,12,13,14,14,15]
}
```

```
a = pd.DataFrame(Data)
```

a

	name	nae
0	abhi	10
1	ram	11
2	raj	12
3	rohan	13
4	abhi	14
5	raj	14
6	rohan	15

```
!pip install category_encoders
```

```
import category_encoders as ce
```

```
encoder = ce.OneHotEncoder(cols = 'name',handle_unknown = 'return_nan',return_df=True,use_cat_names
```

```
data_encoded = encoder.fit_transform(a)
```

data_encoded

	name_abhi	name_ram	name_raj	name_rohan	nae
0	1.0	0.0	0.0	0.0	10
1	0.0	1.0	0.0	0.0	11
2	0.0	0.0	1.0	0.0	12
3	0.0	0.0	0.0	1.0	13
4	1.0	0.0	0.0	0.0	14
5	0.0	0.0	1.0	0.0	14
6	0.0	0.0	0.0	1.0	15

```
m = pd.concat([a,data_encoded],axis = "columns")
```

m

	name	nae	name_abhi	name_ram	name_raj	name_rohan	nae
0	abhi	10	1.0	0.0	0.0	0.0	10
1	ram	11	0.0	1.0	0.0	0.0	11
2	raj	12	0.0	0.0	1.0	0.0	12
3	rohan	13	0.0	0.0	0.0	1.0	13
4	abhi	14	1.0	0.0	0.0	0.0	14
5	raj	14	0.0	0.0	1.0	0.0	14
6	rohan	15	0.0	0.0	0.0	1.0	15

```
data_encoded=pd.get_dummies(data=a,drop_first
=True)
data_encoded
```

	nae	name_raj	name_ram	name_rohan
0	10	0	0	0
1	11	0	1	0
2	12	1	0	0
3	13	0	0	1
4	14	0	0	0
5	14	1	0	0
6	15	0	0	1

▼ revision

```
import pandas as pd
```

```
import numpy as np
```

```
a = {
    "name":['abhijeet','raj','ram'],
    "age":[21,3,4]
}
```

```
a1 = pd.DataFrame(a)
```

```
a1.to_csv("data.csv")
```

```
data1 = pd.read_csv("data.csv")
```

```
a1
```

	name	age
0	abhijeet	21
1	raj	3
2	ram	4

```
data1
```

	Unnamed: 0	name	age
0	0	abhijeet	21
1	1	raj	3
2	2	ram	4

```
import sklearn
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
scaler = MinMaxScaler()
```

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-46-4d012fd6134d> in <cell line: 1>()  
----> 1 a2 = pd.DataFrame(scaler.fit_transform(data1))
```

```
----- 7 frames -----  
/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py in  
__array__(self, dtype)  
    2068  
    2069     def __array__(self, dtype: npt.DTypeLike | None = None) ->  
np.ndarray:  
-> 2070         return np.asarray(self._values, dtype=dtype)  
    2071  
    2072     def __array_wrap__()
```

```
ValueError: could not convert string to float: 'abhijeet'
```

SEARCH STACK OVERFLOW