

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import OrdinalEncoder
from sklearn.preprocessing import MinMaxScaler
from google.colab import drive
```

```
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
df = pd.read_csv('/content/drive/MyDrive/python-Saylani/covid_toy.csv')
```

```
df.head()
```

	age	gender	fever	cough	city	has_covid
0	60	Male	103.0	Mild	Kolkata	No
1	27	Male	100.0	Mild	Delhi	Yes
2	42	Male	101.0	Mild	Delhi	No
3	31	Female	98.0	Mild	Kolkata	No
4	65	Female	101.0	Mild	Mumbai	No

```
df.isnull().sum()
```

	0
age	0
gender	0
fever	10
cough	0
city	0
has_covid	0

```
dtype: int64
```

```
x=df.drop(columns=['has_covid'])
y=df['has_covid']
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
```

```
x_train
```

	age	gender	fever	cough	city
94	79	Male	NaN	Strong	Kolkata
48	66	Male	99.0	Strong	Bangalore
14	51	Male	104.0	Mild	Bangalore
16	69	Female	103.0	Mild	Kolkata
22	71	Female	98.0	Strong	Kolkata
...
51	11	Female	100.0	Strong	Kolkata
67	65	Male	99.0	Mild	Bangalore
68	54	Female	104.0	Strong	Kolkata
70	68	Female	101.0	Strong	Delhi
34	74	Male	102.0	Mild	Mumbai

80 rows × 5 columns

x_test

	age	gender	fever	cough	city
33	26	Female	98.0	Mild	Kolkata
64	42	Male	104.0	Mild	Mumbai
63	10	Male	100.0	Mild	Bangalore
41	82	Male	NaN	Mild	Kolkata
38	49	Female	101.0	Mild	Delhi
66	51	Male	104.0	Mild	Kolkata
28	16	Male	104.0	Mild	Kolkata
72	83	Female	101.0	Mild	Kolkata
0	60	Male	103.0	Mild	Kolkata
54	60	Female	99.0	Mild	Mumbai
8	19	Female	100.0	Strong	Bangalore
73	34	Male	98.0	Strong	Kolkata
36	38	Female	101.0	Mild	Bangalore
86	25	Male	104.0	Mild	Bangalore
79	48	Female	103.0	Mild	Kolkata
12	25	Female	99.0	Strong	Kolkata
87	47	Male	101.0	Strong	Bangalore
60	24	Female	102.0	Strong	Bangalore
65	69	Female	102.0	Mild	Bangalore
10	75	Female	NaN	Mild	Delhi

▼ Column Transformer

```
from sklearn.compose import ColumnTransformer

transformer = ColumnTransformer(transformers=[
    ('tnf1',MinMaxScaler(),['fever']),
    ('tnf2',OrdinalEncoder(categories=[['Mild','Strong']]),['cough']),
    ('tnf3',OneHotEncoder(sparse_output=False,drop='first'),['gender','city'])
],remainder='passthrough')

transformer.fit_transform(x_train)
```

```
array([[ nan,  1.      ,  1.      ,  0.      ,  1.      ,  1.      ,
       0.      ,  79.     ],
       [ 0.16666667,  1.      ,  1.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  66.     ],
       [ 1.      ,  0.      ,  1.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  51.     ],
       [ 0.83333333,  0.      ,  0.      ,  0.      ,  1.      ,  0.      ,
        0.      ,  69.     ],
       [ 0.      ,  1.      ,  0.      ,  0.      ,  1.      ,  0.      ,
        0.      ,  71.     ],
       [ 0.66666667,  1.      ,  0.      ,  0.      ,  1.      ,  0.      ,
        0.      ,  82.     ],
       [ 0.16666667,  0.      ,  1.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  72.     ],
       [ 0.5      ,  0.      ,  0.      ,  0.      ,  0.      ,  0.      ,
        1.      ,  81.     ],
       [ 1.      ,  0.      ,  0.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  18.     ],
       [ 0.      ,  0.      ,  0.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  64.     ],
       [ 0.5      ,  0.      ,  1.      ,  1.      ,  0.      ,  0.      ,
        0.      ,  15.     ],
       [ 0.5      ,  1.      ,  1.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  14.     ],
       [ 0.66666667,  1.      ,  0.      ,  1.      ,  0.      ,  0.      ,
        0.      ,  33.     ],
       [ 1.      ,  1.      ,  0.      ,  1.      ,  0.      ,  0.      ,
        0.      ,  34.     ],
       [ 0.16666667,  1.      ,  0.      ,  1.      ,  0.      ,  0.      ,
        0.      ,  59.     ],
       [ 0.33333333,  1.      ,  0.      ,  0.      ,  1.      ,  0.      ,
        0.      ,  13.     ],
       [ 0.5      ,  0.      ,  0.      ,  0.      ,  0.      ,  0.      ,
        1.      ,  19.     ],
       [ 0.      ,  1.      ,  0.      ,  0.      ,  0.      ,  0.      ,
        1.      ,  81.     ],
       [ 0.33333333,  1.      ,  0.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  47.     ],
       [ 0.83333333,  0.      ,  0.      ,  0.      ,  1.      ,  0.      ,
        0.      ,  50.     ],
       [ nan,  0.      ,  1.      ,  0.      ,  0.      ,  0.      ,
        1.      ,  23.     ],
       [ 0.      ,  1.      ,  0.      ,  0.      ,  0.      ,  0.      ,
        1.      ,  5.     ],
       [ 0.33333333,  0.      ,  1.      ,  1.      ,  0.      ,  0.      ,
        0.      ,  27.     ],
       [ 0.5      ,  0.      ,  1.      ,  1.      ,  0.      ,  0.      ,
        0.      ,  19.     ],
       [ 0.83333333,  1.      ,  1.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  46.     ],
       [ 0.16666667,  0.      ,  0.      ,  0.      ,  0.      ,  0.      ,
        1.      ,  14.     ],
       [ 0.      ,  1.      ,  1.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  12.     ],
       [ 0.83333333,  0.      ,  1.      ,  0.      ,  1.      ,  0.      ,
        0.      ,  83.     ],
       [ 1.      ,  0.      ,  0.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  12.     ],
       [ 0.      ,  1.      ,  1.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  1.     ]])
```

```
transformer.fit_transform(x_test)
```

```
array([[ 0.      ,  0.      ,  0.      ,  0.      ,  1.      ,  0.      ,
       0.      ,  26.     ],
       [ 1.      ,  0.      ,  1.      ,  0.      ,  0.      ,  0.      ,
        1.      ,  42.     ],
       [ 0.33333333,  0.      ,  1.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  10.     ],
       [ nan,  0.      ,  1.      ,  0.      ,  1.      ,  0.      ,
        0.      ,  82.     ],
       [ 0.5      ,  0.      ,  0.      ,  1.      ,  0.      ,  0.      ,
        0.      ,  49.     ],
       [ 1.      ,  0.      ,  1.      ,  0.      ,  1.      ,  0.      ,
        0.      ,  51.     ],
       [ 1.      ,  0.      ,  1.      ,  0.      ,  0.      ,  1.      ,
        0.      ,  16.     ],
       [ 0.5      ,  0.      ,  0.      ,  0.      ,  1.      ,  0.      ,
        0.      ,  83.     ],
       [ 0.83333333,  0.      ,  1.      ,  0.      ,  1.      ,  0.      ,
        0.      ,  60.     ],
       [ 0.16666667,  0.      ,  0.      ,  0.      ,  0.      ,  0.      ,
        1.      ,  60.     ],
       [ 0.33333333,  1.      ,  0.      ,  0.      ,  0.      ,  0.      ,
        0.      ,  19.     ],
       [ 0.      ,  1.      ,  1.      ,  0.      ,  1.      ,  0.      ,
        0.      ,  1.     ]])
```

```
0.      , 34.      ],
[ 0.5   , 0.      ],
0.      , 38.      ],
[ 1.    , 0.      ],
0.      , 25.      ],
[ 0.83333333, 0.    ,
0.      , 48.      ],
[ 0.16666667, 1.    ,
0.      , 25.      ],
[ 0.5    , 1.      ],
0.      , 47.      ],
[ 0.66666667, 1.    ,
0.      , 24.      ],
[ 0.66666667, 0.    ,
0.      , 69.      ],
[       nan, 0.    ,
0.      , 75.      ]])
```

Start coding or [generate](#) with AI.