



```
In [2]: import numpy as np
import pandas as pd
```

```
In [3]: from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import OrdinalEncoder
```

Column Transformer

Column Transformer is a scikit-learn class used to create and apply separate transformers for numerical and categorical data. To create transformers we need to specify the transformer object and pass the list of transformations inside a tuple along with the column on which you want to apply the transformation.

```
In [4]: df=pd.read_csv('covid_toy.csv')
```

```
In [6]: df.head()
```

```
Out[6]:
```

	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

```
In [8]: df['cough'].value_counts()
```

```
Out[8]: Mild      62
Strong    38
Name: cough, dtype: int64
```

```
In [9]: df['city'].value_counts()
```

```
Out[9]: Kolkata      32
Bangalore    30
Delhi        22
Mumbai       16
Name: city, dtype: int64
```

```
In [10]:
```

```
Out[10]: age          0
gender      0
fever       10
cough       0
city        0
has_covid   0
dtype: int64
```

```
In [7]: from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(df.drop(columns=['has_covid']),df['has_covid'])
```

Without Column Transformer

```
In [15]: #Using simpleImputer for fever column
si=SimpleImputer()
X_train_fever=si.fit_transform(X_train[['fever']])
X_test_fever=si.fit_transform(X_test[['fever']])
X_train_fever.shape
```

```
Out[15]: (80, 1)
```

```
In [19]: #Using OrdinalEncoding for cough column
oe=OrdinalEncoder(categories=[['Mild','Strong']])
X_train_gender_city=oe.fit_transform(X_train[['cough']])
X_test_gender_city=oe.fit_transform(X_test[['cough']])
X_train_gender_city.shape
```

```
Out[19]: (80, 1)
```

```
In [22]: #Using OrdinalEncoding for cough column
ohe=OneHotEncoder(drop='first',sparse=False)
X_train_cough=ohe.fit_transform(X_train[['gender','city']])
X_test_cough=ohe.fit_transform(X_test[['gender','city']])
X_train_cough.shape
```

```
Out[22]: (80, 4)
```

Extracting Age

```
In [23]: X_train_age=X_train.drop(columns=['gender','fever','cough','city']).values
X_test_age=X_test.drop(columns=['gender','fever','cough','city']).values
```

Concatination of all columns

```
In [25]: X_train_transformed=np.concatenate((X_train_age,X_train_fever,X_train_gender_city,X_train_cough))
X_test_transformed=np.concatenate((X_test_age,X_test_fever,X_test_gender_city,X_test_cough))
X_train_transformed.shape
```

```
Out[25]: (80, 7)
```

Using Column Transformer

```
from sklearn.compose import ColumnTransformer
```

```
In [30]: transformer = ColumnTransformer(transformers=[  
    ('tnf1', SimpleImputer(), ['fever']),  
    ('tnf2', OrdinalEncoder(categories=[['Mild', 'Strong']]), ['cough']),  
    ('tnf3', OneHotEncoder(sparse=False, drop='first'), ['gender', 'city'])  
], remainder='passthrough')
```

```
In [31]: transformer.fit_transform(X_train).shape
```

```
Out[31]: (80, 7)
```

```
In [32]: transformer.transform(X_test).shape
```

```
Out[32]: (20, 7)
```

Thank you

Author

[Muhammad Zaman Ali](#)

© [Mt Learners](#) 2022. All rights reserved.