In [1]:

```
import numpy as np
import pandas as pd
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import OrdinalEncoder
```

In [2]:

```
df = pd.read_csv("/Users/sudhanshubiswal/Project/covid_toy.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	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 [5]:

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

Out[5]:

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

In [6]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(df.drop(columns=['has_covid']),
                                                test_size=0.2)
```

In [7]:

```
X_train
```

Out[7]:

	age	gender	fever	cough	city
29	34	Female	NaN	Strong	Mumbai
48	66	Male	99.0	Strong	Bangalore
73	34	Male	98.0	Strong	Kolkata
43	22	Female	99.0	Mild	Bangalore
78	11	Male	100.0	Mild	Bangalore
...	...	...	...	...	...
84	69	Female	98.0	Strong	Mumbai
15	70	Male	103.0	Strong	Kolkata
4	65	Female	101.0	Mild	Mumbai
42	27	Male	100.0	Mild	Delhi
20	12	Male	98.0	Strong	Bangalore

80 rows × 5 columns

1. Traditional method

In [8]:

```
# adding simple imputer to fever col
si = SimpleImputer()
X_train_fever = si.fit_transform(X_train[['fever']])

# also the test data
X_test_fever = si.fit_transform(X_test[['fever']])

X_train_fever.shape
```

Out[8]:

```
(80, 1)
```

In [9]:

```
# Ordinalencoding -> cough
oe = OrdinalEncoder(categories=[['Mild','Strong']])
X_train_cough = oe.fit_transform(X_train[['cough']])

# also the test data
X_test_cough = oe.fit_transform(X_test[['cough']])

X_train_cough.shape
```

Out[9]:

```
(80, 1)
```

In [10]:

```
# OneHotEncoding -> gender,city
ohe = OneHotEncoder(drop='first',sparse=False)
X_train_gender_city = ohe.fit_transform(X_train[['gender','city']])

# also the test data
X_test_gender_city = ohe.fit_transform(X_test[['gender','city']])

X_train_gender_city.shape
```

Out[10]:

```
(80, 4)
```

In [11]:

```
# Extracting Age
X_train_age = X_train.drop(columns=['gender','fever','cough','city']).values

# also the test data
X_test_age = X_test.drop(columns=['gender','fever','cough','city']).values

X_train_age.shape
```

Out[11]:

```
(80, 1)
```

In [12]:

```
X_train_transformed = np.concatenate((X_train_age,X_train_fever,X_train_gender_c
# also the test data
X_test_transformed = np.concatenate((X_test_age,X_test_fever,X_test_gender_city,

X_train_transformed.shape
```

Out[12]:

```
(80, 7)
```

2. CT Method

In [13]:

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

In [15]:

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

In [16]:

```
transformer.fit_transform(X_train).shape
```

Out[16]:

```
(80, 7)
```

In [17]:

```
transformer.transform(X_test).shape
```

Out[17]:

```
(20, 7)
```

Easy peasy

In [ ]:

```

```