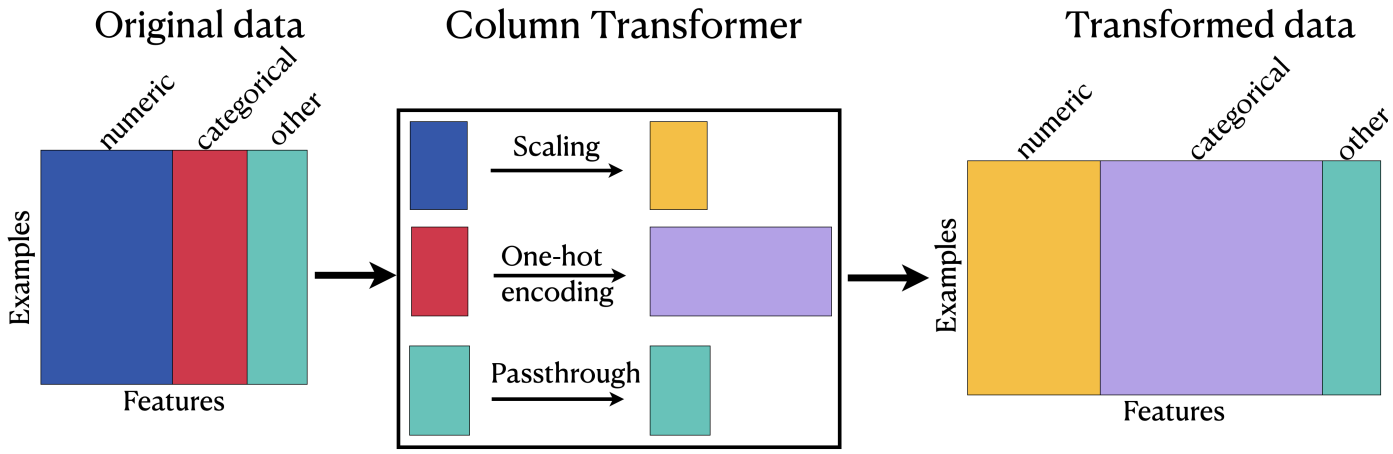
Feature Engineering 101

Day 2

Column Transformer



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('covid_toy.csv')
```

In [3]:
```
df
```

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
...	...	...	...	...	...	...
95	12	Female	104.0	Mild	Bangalore	No
96	51	Female	101.0	Strong	Kolkata	Yes
97	20	Female	101.0	Mild	Bangalore	No
98	5	Female	98.0	Strong	Mumbai	No
99	10	Female	98.0	Strong	Kolkata	Yes

100 rows × 6 columns

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
age           0
gender        0
fever        10
cough         0
city          0
has_covid     0
dtype: int64
```

In [5]:
```
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_co
                                                 test_size=0.3)
```

In [6]:
```
X_train
```

Out[6]:

	age	gender	fever	cough	city
39	50	Female	103.0	Mild	Kolkata
92	82	Female	102.0	Strong	Kolkata
11	65	Female	98.0	Mild	Mumbai
52	47	Female	100.0	Strong	Bangalore
28	16	Male	104.0	Mild	Kolkata
...	...	...	...	...	...
32	34	Female	101.0	Strong	Delhi
15	70	Male	103.0	Strong	Kolkata
19	42	Female	NaN	Strong	Bangalore
12	25	Female	99.0	Strong	Kolkata
33	26	Female	98.0	Mild	Kolkata

70 rows × 5 columns

Long method

In [7]:
```
# 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[7]:
```
(70, 1)
```

In [8]:
```
# 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[8]:
```
(70, 1)
```

In [9]:
```
print('City')
print(df['city'].value_counts())
print('Gender')
print(df['gender'].value_counts())
```

```
City
Kolkata      32
Bangalore    30
Delhi        22
Mumbai       16
Name: city, dtype: int64
Gender
Female    59
Male      41
Name: gender, dtype: int64
```

In [10]:
```
# 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[10]:
```
(70, 1)
```

In [11]:
```
# 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[11]:
```
(70, 4)
```

In [12]:
```
# 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[12]:
```
(70, 1)
```

In [13]:
```
X_train_transformed = np.concatenate((X_train_age,X_train_fever,X_train_gender_city,X_trai
# also the test data
X_test_transformed = np.concatenate((X_test_age,X_test_fever,X_test_gender_city,X_test_cou

X_train_transformed.shape
```

Out[13]:
```
(70, 7)
```

Column TransFromer

```
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.linear_model import LogisticRegression

# Define the column transformer
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), ['numerical_feature_1',
'numerical_feature_2']),
        ('cat', OneHotEncoder(), ['categorical_feature'])
    ])

# Define the pipeline
pipe = Pipeline([
    ('preprocessor', preprocessor),
    ('classifier', LogisticRegression())
])

# Fit the pipeline to the training data
pipe.fit(X_train, y_train)

# Use the pipeline to predict on the test data
y_pred = pipe.predict(X_test)
```

In [14]:
```
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)
```

Out[16]:
```
array([[103.  ,   0.  ,   0.  ,   0.  ,   1.  ,   0.  ,
         50.  ],
       [102.  ,   1.  ,   0.  ,   0.  ,   1.  ,   0.  ,
         82.  ],
       [ 98.  ,   0.  ,   0.  ,   0.  ,   0.  ,   1.  ,
         65.  ],
       [100.  ,   1.  ,   0.  ,   0.  ,   0.  ,   0.  ,
         47.  ],
       [104.  ,   0.  ,   1.  ,   0.  ,   1.  ,   0.  ,
         16.  ],
       [101.  ,   0.  ,   0.  ,   0.  ,   0.  ,   1.  ,
         81.  ],
       [104.  ,   0.  ,   0.  ,   0.  ,   1.  ,   0.  ,
         17.  ],
```

```
[101.     ,   0.     ,   0.     ,   0.     ,   0.     ,   1.     ,
  19.     ],
[103.     ,   0.     ,   1.     ,   0.     ,   1.     ,   0.     ,
  83.     ],
[100.     ,   0.     ,   0.     ,   0.     ,   1.     ,   0.     ,
   5.     ],
[104.     ,   0.     ,   0.     ,   0.     ,   0.     ,   0.     ,
  18.     ],
[100.     ,   0.     ,   1.     ,   1.     ,   0.     ,   0.     ,
  27.     ],
[101.     ,   0.     ,   1.     ,   1.     ,   0.     ,   0.     ,
  19.     ],
[101.     ,   1.     ,   1.     ,   0.     ,   0.     ,   0.     ,
  47.     ],
[102.     ,   0.     ,   1.     ,   0.     ,   0.     ,   1.     ,
  74.     ],
[102.     ,   1.     ,   1.     ,   1.     ,   0.     ,   0.     ,
  20.     ],
[104.     ,   0.     ,   1.     ,   0.     ,   0.     ,   0.     ,
  25.     ],
[100.     ,   0.     ,   1.     ,   0.     ,   0.     ,   0.     ,
  10.     ],
[102.     ,   0.     ,   0.     ,   1.     ,   0.     ,   0.     ,
  49.     ],
[ 99.     ,   1.     ,   0.     ,   1.     ,   0.     ,   0.     ,
  59.     ],
[101.     ,   0.     ,   0.     ,   0.     ,   1.     ,   0.     ,
   8.     ],
[104.     ,   1.     ,   0.     ,   0.     ,   1.     ,   0.     ,
  54.     ],
[ 98.     ,   0.     ,   0.     ,   0.     ,   1.     ,   0.     ,
  31.     ],
[104.     ,   1.     ,   0.     ,   1.     ,   0.     ,   0.     ,
  34.     ],
[ 99.     ,   0.     ,   1.     ,   0.     ,   0.     ,   0.     ,
  72.     ],
[103.     ,   0.     ,   1.     ,   0.     ,   1.     ,   0.     ,
  60.     ],
[100.90625,   0.     ,   1.     ,   0.     ,   0.     ,   1.     ,
  23.     ],
[ 98.     ,   0.     ,   1.     ,   0.     ,   1.     ,   0.     ,
  24.     ],
[104.     ,   1.     ,   0.     ,   0.     ,   0.     ,   0.     ,
  56.     ],
[102.     ,   0.     ,   1.     ,   0.     ,   0.     ,   0.     ,
  64.     ],
[ 98.     ,   1.     ,   0.     ,   0.     ,   0.     ,   1.     ,
   5.     ],
[104.     ,   0.     ,   1.     ,   0.     ,   1.     ,   0.     ,
  51.     ],
[100.     ,   0.     ,   1.     ,   0.     ,   1.     ,   0.     ,
  55.     ],
[ 98.     ,   0.     ,   1.     ,   0.     ,   0.     ,   0.     ,
  73.     ],
[100.     ,   0.     ,   1.     ,   0.     ,   0.     ,   0.     ,
  80.     ],
[100.     ,   1.     ,   0.     ,   0.     ,   1.     ,   0.     ,
  11.     ],
[ 99.     ,   0.     ,   0.     ,   0.     ,   0.     ,   1.     ,
  60.     ],
[102.     ,   0.     ,   1.     ,   0.     ,   1.     ,   0.     ,
   5.     ],
[ 98.     ,   1.     ,   1.     ,   0.     ,   0.     ,   1.     ,
  23.     ],
[100.     ,   1.     ,   0.     ,   0.     ,   0.     ,   0.     ,
  19.     ],
```

```
       [ 99.     ,   0.     ,   1.     ,   1.     ,   0.     ,   0.     ,
         65.     ],
       [101.     ,   0.     ,   0.     ,   0.     ,   1.     ,   0.     ,
         83.     ],
       [100.90625,   1.     ,   0.     ,   0.     ,   0.     ,   1.     ,
         34.     ],
       [103.     ,   0.     ,   0.     ,   1.     ,   0.     ,   0.     ,
         73.     ],
       [100.     ,   1.     ,   0.     ,   0.     ,   1.     ,   0.     ,
         13.     ],
       [100.90625,   1.     ,   1.     ,   0.     ,   1.     ,   0.     ,
         71.     ],
       [101.     ,   0.     ,   0.     ,   0.     ,   0.     ,   0.     ,
         38.     ],
       [ 99.     ,   0.     ,   0.     ,   0.     ,   0.     ,   0.     ,
         22.     ],
       [100.90625,   0.     ,   1.     ,   0.     ,   1.     ,   0.     ,
         82.     ],
       [ 99.     ,   1.     ,   0.     ,   0.     ,   0.     ,   0.     ,
         49.     ],
       [104.     ,   1.     ,   0.     ,   1.     ,   0.     ,   0.     ,
         75.     ],
       [102.     ,   1.     ,   0.     ,   0.     ,   0.     ,   0.     ,
         82.     ],
       [101.     ,   0.     ,   0.     ,   0.     ,   0.     ,   0.     ,
         20.     ],
       [ 98.     ,   1.     ,   1.     ,   0.     ,   1.     ,   0.     ,
         34.     ],
       [ 98.     ,   1.     ,   0.     ,   1.     ,   0.     ,   0.     ,
         40.     ],
       [101.     ,   1.     ,   0.     ,   1.     ,   0.     ,   0.     ,
         68.     ],
       [104.     ,   0.     ,   0.     ,   0.     ,   0.     ,   0.     ,
         12.     ],
       [101.     ,   0.     ,   1.     ,   1.     ,   0.     ,   0.     ,
         15.     ],
       [103.     ,   0.     ,   0.     ,   0.     ,   0.     ,   0.     ,
         16.     ],
       [ 98.     ,   1.     ,   0.     ,   0.     ,   1.     ,   0.     ,
         10.     ],
       [101.     ,   0.     ,   0.     ,   1.     ,   0.     ,   0.     ,
         49.     ],
       [ 99.     ,   0.     ,   1.     ,   0.     ,   0.     ,   0.     ,
         65.     ],
       [100.90625,   0.     ,   1.     ,   1.     ,   0.     ,   0.     ,
         38.     ],
       [104.     ,   0.     ,   1.     ,   0.     ,   0.     ,   0.     ,
         51.     ],
       [ 98.     ,   0.     ,   1.     ,   1.     ,   0.     ,   0.     ,
         83.     ],
       [101.     ,   1.     ,   0.     ,   1.     ,   0.     ,   0.     ,
         34.     ],
       [103.     ,   1.     ,   1.     ,   0.     ,   1.     ,   0.     ,
         70.     ],
       [100.90625,   1.     ,   0.     ,   0.     ,   0.     ,   0.     ,
         42.     ],
       [ 99.     ,   1.     ,   0.     ,   0.     ,   1.     ,   0.     ,
         25.     ],
       [ 98.     ,   0.     ,   0.     ,   0.     ,   1.     ,   0.     ,
         26.     ]])
```

In [17]:

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

Out[17]:

```
array([[100.90625,   0.     ,   0.     ,   1.     ,   0.     ,   0.     ,
         75.     ],
```

```
       [ 99.     ,   0.     ,   0.     ,   0.     ,   0.     ,   1.     ,
         14.     ],
       [101.     ,   1.     ,   0.     ,   0.     ,   1.     ,   0.     ,
         51.     ],
       [101.     ,   1.     ,   1.     ,   0.     ,   0.     ,   0.     ,
         14.     ],
       [ 98.     ,   1.     ,   0.     ,   0.     ,   1.     ,   0.     ,
         71.     ],
       [100.90625,   1.     ,   0.     ,   0.     ,   0.     ,   1.     ,
         20.     ],
       [103.     ,   0.     ,   0.     ,   0.     ,   1.     ,   0.     ,
         48.     ],
       [103.     ,   1.     ,   1.     ,   0.     ,   0.     ,   0.     ,
         46.     ],
       [100.     ,   0.     ,   1.     ,   0.     ,   0.     ,   0.     ,
         11.     ],
       [100.     ,   0.     ,   1.     ,   1.     ,   0.     ,   0.     ,
         27.     ],
       [ 98.     ,   1.     ,   1.     ,   0.     ,   0.     ,   0.     ,
         12.     ],
       [ 98.     ,   1.     ,   0.     ,   0.     ,   0.     ,   1.     ,
         81.     ],
       [ 99.     ,   1.     ,   1.     ,   0.     ,   0.     ,   0.     ,
         66.     ],
       [ 98.     ,   0.     ,   0.     ,   0.     ,   0.     ,   0.     ,
         64.     ],
       [104.     ,   0.     ,   1.     ,   0.     ,   0.     ,   1.     ,
         42.     ],
       [101.     ,   0.     ,   0.     ,   1.     ,   0.     ,   0.     ,
         64.     ],
       [101.     ,   0.     ,   1.     ,   1.     ,   0.     ,   0.     ,
         42.     ],
       [ 98.     ,   1.     ,   0.     ,   0.     ,   0.     ,   1.     ,
         69.     ],
       [102.     ,   1.     ,   0.     ,   1.     ,   0.     ,   0.     ,
         33.     ],
       [100.90625,   0.     ,   0.     ,   0.     ,   0.     ,   0.     ,
         84.     ],
       [104.     ,   0.     ,   0.     ,   0.     ,   1.     ,   0.     ,
          6.     ],
       [100.     ,   0.     ,   1.     ,   0.     ,   1.     ,   0.     ,
         27.     ],
       [102.     ,   0.     ,   0.     ,   0.     ,   0.     ,   0.     ,
         69.     ],
       [ 98.     ,   0.     ,   0.     ,   1.     ,   0.     ,   0.     ,
         80.     ],
       [100.90625,   1.     ,   1.     ,   0.     ,   1.     ,   0.     ,
         79.     ],
       [103.     ,   0.     ,   0.     ,   0.     ,   1.     ,   0.     ,
         69.     ],
       [104.     ,   0.     ,   1.     ,   0.     ,   0.     ,   1.     ,
         44.     ],
       [100.     ,   0.     ,   0.     ,   0.     ,   1.     ,   0.     ,
         19.     ],
       [102.     ,   1.     ,   0.     ,   0.     ,   0.     ,   0.     ,
         24.     ],
       [101.     ,   0.     ,   0.     ,   0.     ,   0.     ,   1.     ,
         65.     ]])
```

In [18]:

```
print(transformer.fit_transform(X_train).shape)
print(transformer.transform(X_test).shape)
```

```
(70, 7)
(30, 7)
```

In [19]:
```
print('X_Train')
print(X_train)

print('X_Test')
print(X_test)
```

```
X_Train
    age  gender  fever   cough       city
39   50  Female  103.0    Mild    Kolkata
92   82  Female  102.0  Strong    Kolkata
11   65  Female   98.0    Mild     Mumbai
52   47  Female  100.0  Strong  Bangalore
28   16    Male  104.0    Mild    Kolkata
..  ...     ...    ...     ...        ...
32   34  Female  101.0  Strong      Delhi
15   70    Male  103.0  Strong    Kolkata
19   42  Female    NaN  Strong  Bangalore
12   25  Female   99.0  Strong    Kolkata
33   26  Female   98.0    Mild    Kolkata

[70 rows x 5 columns]
X_Test
    age  gender  fever   cough       city
10   75  Female    NaN    Mild      Delhi
80   14  Female   99.0    Mild     Mumbai
96   51  Female  101.0  Strong    Kolkata
6    14    Male  101.0  Strong  Bangalore
22   71  Female   98.0  Strong    Kolkata
7    20  Female    NaN  Strong     Mumbai
79   48  Female  103.0    Mild    Kolkata
89   46    Male  103.0  Strong  Bangalore
78   11    Male  100.0    Mild  Bangalore
42   27    Male  100.0    Mild      Delhi
20   12    Male   98.0  Strong  Bangalore
61   81  Female   98.0  Strong     Mumbai
48   66    Male   99.0  Strong  Bangalore
18   64  Female   98.0    Mild  Bangalore
64   42    Male  104.0    Mild     Mumbai
9    64  Female  101.0    Mild      Delhi
2    42    Male  101.0    Mild      Delhi
84   69  Female   98.0  Strong     Mumbai
27   33  Female  102.0  Strong      Delhi
5    84  Female    NaN    Mild  Bangalore
59    6  Female  104.0    Mild    Kolkata
93   27    Male  100.0    Mild    Kolkata
65   69  Female  102.0    Mild  Bangalore
23   80  Female   98.0    Mild      Delhi
94   79    Male    NaN  Strong    Kolkata
16   69  Female  103.0    Mild    Kolkata
49   44    Male  104.0    Mild     Mumbai
26   19  Female  100.0    Mild    Kolkata
60   24  Female  102.0  Strong  Bangalore
4    65  Female  101.0    Mild     Mumbai
```

In [20]:
```
print(X_train.shape)
print(X_test.shape)
```

```
(70, 5)
(30, 5)
```

In [ ]:
```
```