Encoding ==> This is the method to convert our categorical data into numerical data.

(1). LabelEncoding ==> Using this method , we can convert our target or one dimensional data.

In [3]:

```python
import pandas as pd
import numpy as np
```

In [4]:

```python
df = pd.read_csv("D:\\Summer Training Video\\ML\\covid_toy.csv")
```

In [5]:

```python
df.head(2)
```

Out[5]:

	age	gender	fever	cough	city	has_covid
0	60	Male	103.0	Mild	Kolkata	No
1	27	Male	100.0	Mild	Delhi	Yes

In [6]:

```python
df = df.dropna()
```

In [7]:

```python
from sklearn.preprocessing import LabelEncoder
```

In [8]:

```python
lb = LabelEncoder()
```

In [9]:

```python
df['gender'] = lb.fit_transform(df['gender'])
df['cough'] = lb.fit_transform(df['cough'])
df['city'] = lb.fit_transform(df['city'])
df['has_covid'] = lb.fit_transform(df['has_covid'])
```

In [10]:

```python
df.sample(5)
```

Out[10]:

	age	gender	fever	cough	city	has_covid
46	19	0	101.0	0	3	0
9	64	0	101.0	0	1	0
72	83	0	101.0	0	2	0
92	82	0	102.0	1	2	0
64	42	1	104.0	0	3	0

In [11]:
```
from sklearn.preprocessing import StandardScaler
```

In [12]:
```
sc = StandardScaler()
```

In [13]:
```
df_sc = sc.fit_transform(df)
```

In [14]:
```
# df_sc
```

In [15]:
```
df_new = pd.DataFrame(df_sc , columns = df.columns)
```

In [16]:
```
np.round(df.describe() , 1)
```

Out[16]:

	age	gender	fever	cough	city	has_covid
count	90.0	90.0	90.0	90.0	90.0	90.0
mean	43.0	0.4	100.8	0.4	1.3	0.4
std	24.7	0.5	2.1	0.5	1.1	0.5
min	5.0	0.0	98.0	0.0	0.0	0.0
25%	19.2	0.0	99.0	0.0	0.0	0.0
50%	45.0	0.0	101.0	0.0	1.0	0.0
75%	65.0	1.0	102.8	1.0	2.0	1.0
max	83.0	1.0	104.0	1.0	3.0	1.0

In [17]:
```
df.head()
```

Out[17]:

	age	gender	fever	cough	city	has_covid
0	60	1	103.0	0	2	0
1	27	1	100.0	0	1	1
2	42	1	101.0	0	1	0
3	31	0	98.0	0	2	0
4	65	0	101.0	0	3	0

In [18]:
```
x = df.drop(columns = ['has_covid'] , axis = 1)
y = df['has_covid']
```

In [20]:
```
from sklearn.model_selection import train_test_split
```

In [21]:

```
x_train , x_test , y_train , y_test , = train_test_split(x,y,test_size = 0.2 ,
                                                         random_state = 40)
```

In [22]:

```
print(df.shape)
print(x.shape)
print(y.shape)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(90, 6)
(90, 5)
(90,)
(72, 5)
(18, 5)
(72,)
(18,)
```

In [23]:

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

In [24]:

```
mn = MinMaxScaler()
```

In [25]:

```
x_train_mn = mn.fit_transform(x_train)
```

In [26]:

```
x_test_mn = mn.fit_transform(x_test)
```

In [27]:

```
x_train_new = pd.DataFrame(x_train_mn , columns = x_train.columns)
```

In [28]:

```
np.round(x_train_new.describe() , 1)
```

Out[28]:

	age	gender	fever	cough	city
count	72.0	72.0	72.0	72.0	72.0
mean	0.5	0.4	0.5	0.4	0.4
std	0.3	0.5	0.3	0.5	0.4
min	0.0	0.0	0.0	0.0	0.0
25%	0.2	0.0	0.2	0.0	0.0
50%	0.5	0.0	0.5	0.0	0.3
75%	0.8	1.0	0.7	1.0	0.7
max	1.0	1.0	1.0	1.0	1.0

(2). OrdinalEncoder

In [29]:
```
df = pd.read_csv("D:\\Summer Training Video\\ML\\covid_toy.csv")
```

In [30]:
```
df.head()
```

Out[30]:

	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 [31]:
```
df = df.drop(columns = ['age', 'fever'])
```

In [32]:
```
df.head()
```

Out[32]:

	gender	cough	city	has_covid
0	Male	Mild	Kolkata	No
1	Male	Mild	Delhi	Yes
2	Male	Mild	Delhi	No
3	Female	Mild	Kolkata	No
4	Female	Mild	Mumbai	No

In [33]:
```
df['city'].value_counts()
```

Out[33]:
```
city
Kolkata      32
Bangalore    30
Delhi        22
Mumbai       16
Name: count, dtype: int64
```

In [34]:
```
df['cough'].value_counts()
```

Out[34]:
```
cough
Mild      62
Strong    38
Name: count, dtype: int64
```

In [35]:
```
from sklearn.preprocessing import OrdinalEncoder
```

In [36]:
```
oe = OrdinalEncoder(categories=[['Male','female'],['Mild','Strong'],['Kolkata'
                                ,'Bangalore','Delhi','Mumbai'],['Yes','No']])
```

In [37]:
```
oe
```

Out[37]:
```
▾                            OrdinalEncoder

OrdinalEncoder(categories=[['Male', 'female'], ['Mild', 'Strong'],
                           ['Kolkata', 'Bangalore', 'Delhi', 'Mumbai'],
                           ['Yes', 'No']])
```

In [40]:
```
oe = OrdinalEncoder(handle_unknown='use_encoded_value', unknown_value=-1)
```

In [41]:
```
oe.fit(df)
```

Out[41]:
```
▾                            OrdinalEncoder

OrdinalEncoder(handle_unknown='use_encoded_value', unknown_value=-1)
```

In [42]:
```
df_new = oe.transform(df)
```

In [43]:
```
oe.categories_
```

Out[43]:
```
[array(['Female', 'Male'], dtype=object),
 array(['Mild', 'Strong'], dtype=object),
 array(['Bangalore', 'Delhi', 'Kolkata', 'Mumbai'], dtype=object),
 array(['No', 'Yes'], dtype=object)]
```

In [44]:
```
df = pd.DataFrame(df_new , columns = df.columns)
```

In [48]:
```
df.sample(7)
```

Out[48]:

	gender	cough	city	has_covid
50	1.0	0.0	1.0	1.0
68	0.0	1.0	2.0	0.0
86	1.0	0.0	0.0	1.0
19	0.0	1.0	0.0	1.0
16	0.0	0.0	2.0	1.0
22	0.0	1.0	2.0	1.0
44	1.0	1.0	1.0	0.0

In [49]:
```
df = pd.read_csv("D:\Summer Training Video\ML\Attrition.csv")
```

In [50]:
```
df.head()
```

Out[50]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educa
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life
4	27	No	Travel_Rarely	591	Research & Development	2	1	

5 rows × 35 columns

In [51]:
```
df.columns
```

Out[51]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
       'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

In [52]:
```
df.shape
```

Out[52]:
```
(1470, 35)
```

In [53]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Age                       1470 non-null   int64 
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64 
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64 
 6   Education                 1470 non-null   int64 
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64 
 9   EmployeeNumber            1470 non-null   int64 
 10  EnvironmentSatisfaction   1470 non-null   int64 
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64 
 13  JobInvolvement            1470 non-null   int64 
 14  JobLevel                  1470 non-null   int64 
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64 
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64 
 19  MonthlyRate               1470 non-null   int64 
 20  NumCompaniesWorked        1470 non-null   int64 
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64 
 24  PerformanceRating         1470 non-null   int64 
 25  RelationshipSatisfaction  1470 non-null   int64 
 26  StandardHours             1470 non-null   int64 
 27  StockOptionLevel          1470 non-null   int64 
 28  TotalWorkingYears         1470 non-null   int64 
 29  TrainingTimesLastYear     1470 non-null   int64 
 30  WorkLifeBalance           1470 non-null   int64 
 31  YearsAtCompany            1470 non-null   int64 
 32  YearsInCurrentRole        1470 non-null   int64 
 33  YearsSinceLastPromotion   1470 non-null   int64 
 34  YearsWithCurrManager      1470 non-null   int64 
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

In [ ]: