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

```
In [3]: df = pd.read_csv("D:\\Summer Training Video\\ML\\tips.csv")
```

```
In [4]: df
```

Out[4]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

```
In [8]: df.head()
```

Out[8]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

```
In [9]: df.shape
```

Out[9]: (244, 7)

```
In [10]: df.size
```

Out[10]: 1708

```
In [14]: x = df.drop(columns=['sex', 'smoker', 'time' , 'day'] , axis = 1)  # Independer
         y = df[['sex', 'smoker', 'time' , 'day']]  # Target columns
```

```
In [15]: print(x.shape)
         print(y.shape)
```

```
(244, 3)
(244, 4)
```

```
In [16]: from sklearn.model_selection import train_test_split
```

```
In [17]: x_train , x_test , y_train , y_test = train_test_split(x,y,test_size = 0.2 , ra
```

```
In [18]: print(x_train.shape)
         print(x_test.shape)
         print(y_train.shape)
         print(y_test.shape)
```

```
(195, 3)
(49, 3)
(195, 4)
(49, 4)
```

```
In [19]: # Standarzation   ===> Data Mean = 0 , Standard Deviation = 1
```

```
In [20]: np.round(x_train.describe(), 1)
```

Out[20]:

	total_bill	tip	size
count	195.0	195.0	195.0
mean	20.2	3.1	2.6
std	8.8	1.4	0.9
min	5.8	1.0	1.0
25%	13.7	2.0	2.0
50%	17.9	3.0	2.0
75%	24.9	3.7	3.0
max	50.8	10.0	6.0

```
In [21]: np.round(x_train.describe(), 3)
```

Out[21]:

	total_bill	tip	size
count	195.000	195.000	195.000
mean	20.218	3.088	2.574
std	8.771	1.429	0.941
min	5.750	1.000	1.000
25%	13.660	2.000	2.000
50%	17.920	3.000	2.000
75%	24.855	3.695	3.000
max	50.810	10.000	6.000

```
In [22]: from sklearn.preprocessing import StandardScaler
```

```
In [23]: sc = StandardScaler()
```

```
In [24]: x_train_sc = sc.fit_transform(x_train)    # fit means Learn the parameter and tr
```

```
In [25]: x_train_new = pd.DataFrame(x_train_sc , columns = x_train.columns)
```

```
In [26]: x_train_new.head(3)
```

Out[26]:

	total_bill	tip	size
0	-0.793062	-0.258033	-0.612141
1	0.463227	-0.742114	-0.612141
2	0.807307	0.639973	-0.612141

```
In [27]: np.round(x_train_new.describe() , 1)
```

Out[27]:

	total_bill	tip	size
count	195.0	195.0	195.0
mean	0.0	-0.0	-0.0
std	1.0	1.0	1.0
min	-1.7	-1.5	-1.7
25%	-0.7	-0.8	-0.6
50%	-0.3	-0.1	-0.6
75%	0.5	0.4	0.5
max	3.5	4.8	3.7

```
In [28]: df = pd.read_csv("D:\\Summer Training Video\\ML\\tips.csv")
```

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

Out[30]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

```
In [31]: df = df.drop(columns = ['sex','smoker','day', 'time'])
```

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

Out[32]:

	total_bill	tip	size
0	16.99	1.01	2
1	10.34	1.66	3
2	21.01	3.50	3
3	23.68	3.31	2
4	24.59	3.61	4

```
In [33]: x = df.drop(columns = ['total_bill'], axis = 1)        # Independent Data
         y = df['total_bill']              # Target Data
```

```python
In [34]: from sklearn.model_selection import train_test_split
```

```python
In [35]: x_train , x_test , y_train , y_test = train_test_split(x,y,test_size = 0.2 , ra
```

```python
In [36]: print(df.shape)
         print(x.shape)
         print(x_train.shape)
         print(x_test.shape)
         print(y.shape)
         print(y_train.shape)
         print(y_test.shape)
```

```
(244, 3)
(244, 2)
(195, 2)
(49, 2)
(244,)
(195,)
(49,)
```

```python
In [37]: np.round(x_train.describe() , 1)
```

Out[37]:

	tip	size
count	195.0	195.0
mean	3.1	2.6
std	1.4	0.9
min	1.0	1.0
25%	2.0	2.0
50%	3.0	2.0
75%	3.7	3.0
max	10.0	6.0

```python
In [38]: from sklearn.preprocessing import StandardScaler
```

```python
In [39]: sc = StandardScaler()
```

```python
In [40]: x_train_sc = sc.fit_transform(x_train)
```

```
In [41]: x_train_sc
```

```
Out[41]: array([[-2.58032900e-01, -6.12140682e-01],
               [-7.42114421e-01, -6.12140682e-01],
               [ 6.39973400e-01, -6.12140682e-01],
               [ 9.90757111e-01,  1.51942062e+00],
               [-7.42114421e-01, -6.12140682e-01],
               [ 2.96205363e-01, -6.12140682e-01],
               [-7.63161444e-01, -6.12140682e-01],
               [ 1.44677594e+00, -6.12140682e-01],
               [-9.38553299e-01, -6.12140682e-01],
               [-2.44001552e-01,  4.53639970e-01],
               [-6.15940219e-02,  4.53639970e-01],
               [ 1.43976026e+00,  2.58520127e+00],
               [ 2.89189689e-01, -6.12140682e-01],
               [ 1.79755965e+00, -6.12140682e-01],
               [-4.12377733e-01, -6.12140682e-01],
               [-7.56145770e-01, -6.12140682e-01],
               [ 6.96098794e-01, -6.12140682e-01],
               [-7.00020376e-01, -6.12140682e-01],
               [-1.46472887e+00, -1.67792133e+00],
```

```
In [42]: x_train_new = pd.DataFrame(x_train_sc , columns = x_train.columns)
```

```
In [43]: np.round(x_train_new.describe() , 1)
```

Out[43]:

	tip	size
count	195.0	195.0
mean	-0.0	-0.0
std	1.0	1.0
min	-1.5	-1.7
25%	-0.8	-0.6
50%	-0.1	-0.6
75%	0.4	0.5
max	4.8	3.7

Normalization ==> min = 0 , max = 1

```
In [44]: from sklearn.preprocessing import MinMaxScaler
```

```
In [45]: mn = MinMaxScaler()
```

```
In [46]: x_train_mn = mn.fit_transform(x_train)
```

```
In [47]: x_train_new  = pd.DataFrame(x_train_mn , columns = x_train.columns)
```

```
In [48]: np.round(x_train.describe() , 1)
```

Out[48]:

	tip	size
count	195.0	195.0
mean	3.1	2.6
std	1.4	0.9
min	1.0	1.0
25%	2.0	2.0
50%	3.0	2.0
75%	3.7	3.0
max	10.0	6.0

```
In [49]: np.round(x_train_new.describe() ,1)
```

Out[49]:

	tip	size
count	195.0	195.0
mean	0.2	0.3
std	0.2	0.2
min	0.0	0.0
25%	0.1	0.2
50%	0.2	0.2
75%	0.3	0.4
max	1.0	1.0

```
In [ ]:
```

```
In [ ]:
```