Introduction to Machine Learning

In [1]:
```
# Process ==> Data Preparation ===> ML Model ===> Performance Evaluation
```

(1). Data Preparation

In [2]:
```
# Data ===> Independent data (x) + Dependent Data (y)

# x ===>  x_train , x_test

# y ===> y_train , y_test

# Data Prepare ===>  ML Model ===> Performance Evaluation
```

In [3]:
```
import numpy as np
import pandas as pd
```

In [4]:
```
df = pd.read_csv("D:\\Summer Training Video\\ML\\placement.csv")
```

In [5]:
```
df
```

Out[5]:

	cgpa	resume_score	placed
0	8.14	6.52	1
1	6.17	5.17	0
2	8.27	8.86	1
3	6.88	7.27	1
4	7.52	7.30	1
...	...	...	...
95	6.33	6.38	0
96	8.23	7.76	1
97	6.65	7.78	0
98	8.14	5.63	1
99	6.09	6.61	0

100 rows × 3 columns

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

Out[6]:

	cgpa	resume_score	placed
0	8.14	6.52	1
1	6.17	5.17	0
2	8.27	8.86	1
3	6.88	7.27	1
4	7.52	7.30	1

In [7]:
```
df.shape
```

Out[7]:
```
(100, 3)
```

In [8]:
```
df.size
```

Out[8]:
```
300
```

In [9]:
```
x = df.drop(columns = ['placed'] , axis = 1)       # Independent Columns
y = df['placed']     # Target Column
```

In [10]:
```
print(x.shape)
print(y.shape)
```

```
(100, 2)
(100,)
```

In [11]:
```
from sklearn.model_selection import train_test_split
```

In [12]:
```
x_train , x_test , y_train , y_test = train_test_split(x,y,test_size = 0.2 , ra
```

In [13]:
```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(80, 2)
(20, 2)
(80,)
(20,)
```

In [14]:
```
# Standarzation  ===> Data Mean = 0 , Standard Deviation = 1
```

In [15]:
```
np.round(x_train.describe(), 1)
```

Out[15]:

	cgpa	resume_score
count	80.0	80.0
mean	6.9	7.0
std	1.1	1.0
min	5.3	5.0
25%	6.0	6.3
50%	6.6	7.2
75%	8.1	7.7
max	9.3	9.1

In [16]:
```
np.round(x_train.describe(), 3)
```

Out[16]:

	cgpa	resume_score
count	80.000	80.000
mean	6.950	7.000
std	1.126	1.011
min	5.270	4.950
25%	5.998	6.280
50%	6.620	7.155
75%	8.062	7.692
max	9.310	9.060

In [17]:
```
from sklearn.preprocessing import StandardScaler
```

In [18]:
```
sc = StandardScaler()
```

In [19]:
```
sc = sc.fit_transform(x_train)   # fit means learn the parameter and transform
```

In [20]:
```
x_train_new = pd.DataFrame(x_train_sc , columns = x_train.columns)
```

In [21]:
```
x_train_new.head(3)
```

Out[21]:

	cgpa	resume_score
0	-0.008602	-0.128926
1	1.001293	-0.925381
2	-0.607389	-0.527154

In [22]:
```
np.round(x_train_new.describe() , 1)
```

Out[22]:

	cgpa	resume_score
count	80.0	80.0
mean	-0.0	-0.0
std	1.0	1.0
min	-1.5	-2.0
25%	-0.9	-0.7
50%	-0.3	0.2
75%	1.0	0.7
max	2.1	2.1

In [23]:
```
df = pd.read_csv("D:\\Summer Training Video\\ML\\insurance.csv")
```

In [24]:
```
df.head()
```

Out[24]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

In [25]:
```
df = df.drop(columns = ['sex','smoker','region'])
```

In [26]:
```
df.head()
```

Out[26]:

	age	bmi	children	charges
0	19	27.900	0	16884.92400
1	18	33.770	1	1725.55230
2	28	33.000	3	4449.46200
3	33	22.705	0	21984.47061
4	32	28.880	0	3866.85520

In [27]:
```
x = df.drop(columns = ['charges'], axis = 1)        # Independent Data
y = df['charges']                                   # Target Data
```

In [28]:
```
from sklearn.model_selection import train_test_split
```

In [29]:
```
x_train , x_test , y_train , y_test = train_test_split(x,y,test_size = 0.2 , ra
```

In [30]:
```
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)
```

```
(1338, 4)
(1338, 3)
(1070, 3)
(268, 3)
(1338,)
(1070,)
(268,)
```

In [31]:
```
np.round(x_train.describe() , 1)
```

Out[31]:

	age	bmi	children
count	1070.0	1070.0	1070.0
mean	39.4	30.6	1.1
std	14.1	6.0	1.2
min	18.0	16.0	0.0
25%	27.0	26.2	0.0
50%	39.5	30.2	1.0
75%	51.0	34.5	2.0
max	64.0	53.1	5.0

In [32]:
```
from sklearn.preprocessing import StandardScaler
```

In [35]:
```
sc = StandardScaler()
```

In [36]:
```
x_train_sc = sc.fit_transform(x_train)
```

In [37]:
```
x_train_sc
```

Out[37]:
```
array([[ 0.47222651, -1.75652513,  0.73433626],
       [ 0.54331294, -1.03308239, -0.91119211],
       [ 0.8987451 , -0.94368672, -0.91119211],
       ...,
       [ 1.3252637 , -0.89153925, -0.91119211],
       [-0.16755139,  2.82086429,  0.73433626],
       [ 1.1120044 , -0.10932713, -0.91119211]])
```

In [39]:
```
x_train_new = pd.DataFrame(x_train_sc , columns = x_train.columns)
```

In [42]:
```
np.round(x_train_new.describe() , 1)
```

Out[42]:

	age	bmi	children
count	1070.0	1070.0	1070.0
mean	-0.0	-0.0	-0.0
std	1.0	1.0	1.0
min	-1.5	-2.4	-0.9
25%	-0.9	-0.7	-0.9
50%	0.0	-0.1	-0.1
75%	0.8	0.7	0.7
max	1.8	3.7	3.2

Normalization ==> min = 0 , max = 1

In [43]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [44]:
```
mn = MinMaxScaler()
```

In [45]:
```
x_train_mn = mn.fit_transform(x_train)
```

In [46]:
```
x_train_new  = pd.DataFrame(x_train_mn , columns = x_train.columns)
```

In [47]:
```
np.round(x_train.describe() , 1)
```

Out[47]:

	age	bmi	children
count	1070.0	1070.0	1070.0
mean	39.4	30.6	1.1
std	14.1	6.0	1.2
min	18.0	16.0	0.0
25%	27.0	26.2	0.0
50%	39.5	30.2	1.0
75%	51.0	34.5	2.0
max	64.0	53.1	5.0

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

Out[48]:

	age	bmi	children
count	1070.0	1070.0	1070.0
mean	0.5	0.4	0.2
std	0.3	0.2	0.2
min	0.0	0.0	0.0
25%	0.2	0.3	0.0
50%	0.5	0.4	0.2
75%	0.7	0.5	0.4
max	1.0	1.0	1.0

In [ ]:
```

```