```
import numpy as np
import pandas as pd
df=pd.read_csv("/content/Iris.csv")
df
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

```
# columns
print(df.columns)
```

```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

```
#·head
df.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
#·tail
df.tail()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

```
# shape
df.shape
```

```
(150, 6)
```

```
#·total·no.·of·missing·values·
```

```
print(df.isna().sum())
```

```
Id              0
SepalLengthCm   0
```

```
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

```
# drop the id column
df=df.drop(['Id'],axis=1)
df
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

```
#·x·variable·>>·set·input
x=df.iloc[:,:-1].values

#·y·variable·>>·set·output
y=df.iloc[:,-1].values·
```

```
x
y
```

```
array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
```

```
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica'], dtype=object)
```

```
# **training and testing data**

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30)

x_train
```

```
       [4.8, 3. , 1.4, 0.1],
       [5.7, 4.4, 1.5, 0.4],
       [5.5, 2.4, 3.8, 1.1],
       [6.4, 2.9, 4.3, 1.3],
       [7.7, 2.6, 6.9, 2.3],
       [6.3, 3.4, 5.6, 2.4],
       [5.8, 2.7, 5.1, 1.9],
       [5.7, 2.6, 3.5, 1. ],
       [6.2, 2.2, 4.5, 1.5],
       [7.9, 3.8, 6.4, 2. ],
       [5.4, 3.9, 1.7, 0.4],
       [4.7, 3.2, 1.6, 0.2],
       [5.7, 2.5, 5. , 2. ],
       [5.2, 2.7, 3.9, 1.4],
       [6.6, 2.9, 4.6, 1.3],
       [6.5, 3.2, 5.1, 2. ],
       [7.7, 3. , 6.1, 2.3],
       [6.1, 2.9, 4.7, 1.4],
       [6.3, 2.3, 4.4, 1.3],
       [4.9, 3.1, 1.5, 0.1],
       [5.9, 3. , 4.2, 1.5],
       [5.2, 4.1, 1.5, 0.1],
       [6.7, 3. , 5.2, 2.3],
       [5.1, 3.8, 1.6, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [6.3, 3.3, 6. , 2.5],
       [6. , 2.2, 5. , 1.5],
       [5.5, 2.4, 3.7, 1. ],
       [6.2, 2.8, 4.8, 1.8],
       [5.6, 2.9, 3.6, 1.3],
       [6.7, 3.3, 5.7, 2.5],
       [4.9, 2.4, 3.3, 1. ],
       [5.4, 3.4, 1.7, 0.2],
       [4.4, 3.2, 1.3, 0.2],
       [4.8, 3.1, 1.6, 0.2],
       [6.9, 3.2, 5.7, 2.3],
       [5.9, 3. , 5.1, 1.8],
       [5. , 3. , 1.6, 0.2],
       [4.9, 2.5, 4.5, 1.7],
       [6.9, 3.1, 5.4, 2.1],
       [7.4, 2.8, 6.1, 1.9],
       [6. , 2.7, 5.1, 1.6],
       [4.6, 3.6, 1. , 0.2],
       [5. , 3.2, 1.2, 0.2],
       [6.2, 3.4, 5.4, 2.3],
       [4.8, 3.4, 1.9, 0.2],
       [5.8, 4. , 1.2, 0.2],
       [6.4, 2.8, 5.6, 2.1],
       [4.7, 3.2, 1.3, 0.2],
       [6.3, 2.5, 5. , 1.9],
       [6. , 3. , 4.8, 1.8],
       [5.7, 2.8, 4.5, 1.3],
       [5. , 3.4, 1.6, 0.4],
       [6. , 2.9, 4.5, 1.5],
       [5.4, 3.4, 1.5, 0.4],
       [7.6, 3. , 6.6, 2.1],
       [5.8, 2.7, 5.1, 1.9],
       [5.1, 3.5, 1.4, 0.2]])
```

```
# Standard scalar :

from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
scaler.fit(x_train)
x_train=scaler.transform(x_train)
x_test=scaler.transform(x_test)
x_test
```

```
array([[-0.32108455, -0.58392329,  0.57978262,  0.93818063],
       [ 0.61217159,  0.12352223,  0.91987857,  0.68231319],
       [-0.67105561,  1.53841328, -1.34742776, -1.36462637],
       [-1.13768368,  0.12352223, -1.34742776, -1.49256009],
       [-0.90436964,  1.06678293, -1.40411042, -1.23669265],
       [-0.55439859, -0.11229294,  0.35305199,  0.29851202],
       [ 0.72882861, -0.11229294,  0.74983059,  0.93818063],
       [-1.48765473,  0.83096776, -1.40411042, -1.23669265],
       [ 0.26220053, -0.11229294,  0.40973464,  0.1705783 ],
       [ 1.07879966, -0.11229294,  0.91987857,  1.06611435],
       [ 0.49551457, -0.58392329,  0.69314793,  0.29851202],
       [-1.02102666,  1.06678293, -1.2907451 , -0.85289148],
       [-0.43774157, -1.05555364,  0.29636933, -0.08528915],
       [-0.32108455, -0.11229294,  0.35305199,  0.29851202],
       [ 1.07879966,  0.35933741,  1.1466092 ,  1.3219818 ],
       [-0.20442754,  1.77422846, -1.23406244, -1.23669265],
       [-0.08777052, -0.81973846,  0.12632135, -0.34115659],
       [ 1.31211369,  0.35933741,  0.4664173 ,  0.1705783 ],
       [-0.90436964,  1.77422846, -1.12069713, -1.10875893],
       [ 1.19545668,  0.12352223,  0.57978262,  0.29851202],
       [-0.08777052, -0.81973846,  0.01295604, -0.08528915],
       [ 0.61217159, -0.81973846,  0.80651325,  0.81024691],
       [-0.43774157,  2.71748915, -1.40411042, -1.36462637],
       [ 0.14554352, -1.99881434,  0.06963869, -0.34115659],
       [-0.43774157, -1.29136881,  0.06963869,  0.04264457],
       [-1.25434069, -0.11229294, -1.40411042, -1.23669265],
       [ 1.54542773, -0.11229294,  1.08992654,  0.42644574],
       [ 0.26220053, -0.58392329,  0.4664173 , -0.08528915],
       [-0.55439859,  1.53841328, -1.34742776, -1.36462637],
       [ 0.96214264,  0.12352223,  0.4664173 ,  0.29851202],
       [ 0.49551457, -0.34810811,  0.97656122,  0.68231319],
       [-0.20442754, -0.11229294,  0.18300401, -0.08528915],
       [-1.60431175, -1.76299916, -1.46079308, -1.23669265],
       [ 1.07879966, -0.58392329,  0.52309996,  0.1705783 ],
       [-0.43774157,  1.06678293, -1.46079308, -1.36462637],
       [ 0.96214264, -1.29136881,  1.08992654,  0.68231319],
       [-1.72096877, -0.34810811, -1.40411042, -1.36462637],
       [-0.78771262,  1.06678293, -1.34742776, -1.36462637],
       [-1.02102666, -2.47044469, -0.2137746 , -0.34115659],
       [ 0.61217159, -0.58392329,  0.97656122,  1.19404807],
       [-0.55439859,  2.01004363, -1.46079308, -1.10875893],
       [-1.13768368, -0.11229294, -1.40411042, -1.36462637],
       [-0.78771262,  0.83096776, -1.40411042, -1.36462637],
       [ 0.96214264,  0.12352223,  0.29636933,  0.1705783 ],
       [-0.90436964,  0.59515258, -1.23406244, -0.9808252 ]])
```

```
from sklearn.neighbors import KNeighborsClassifier
classifier=KNeighborsClassifier(n_neighbors=5)
classifier.fit(x_train,y_train)
y_pred=classifier.predict(x_test)   # x_test actual value in y_test
print(y_pred)
```

```
['Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica'
 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa']
```

```
# performance evaluation  - Confusion Matrix
# TP | FP
#---------
# FN | TN
```

```
from sklearn.metrics._plot.confusion_matrix import confusion_matrix
from sklearn.metrics import classification_report,accuracy_score,ConfusionMatrixDisplay
labels=['Iris-versicolor','Iris-setosa','Iris-virginica']
# cm=ConfusionMatrixDisplay()
result=confusion_matrix(y_test,y_pred)
print(result)

cm=ConfusionMatrixDisplay(result,display_labels=labels)
cm.plot()

# Accuracy score
score = accuracy_score(y_test,y_pred)
print(score)
```

```
[[17  1  0]
 [ 0 16  0]
 [ 0  1 10]]
0.9555555555555556
```