```
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

```
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

```
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

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

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

```
df.dtypes #even if o/p set is object,needn'nt to be convert it to numerical data.
```

```
Id                int64
SepalLengthCm    float64
SepalWidthCm     float64
PetalLengthCm    float64
PetalWidthCm     float64
Species           object
dtype: object
```

```
df.drop(['Id'],axis=1,inplace=True)
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=df.iloc[:,:-1].values
x
```

```
y=df.iloc[:,-1].values
y
```

```
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,random_state=1)
x_train
```

```
x_test
```

```
y_train
```

```
y_test
```

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

```
x_train
```

```
         2.69637740e-01],
        [-1.05669938e-15, -8.06571522e-01,  7.65053975e-01,
         9.10179882e-01],
        [-4.75895093e-01,  8.87228675e-01, -1.14624252e+00,
        -1.26766340e+00],
        [-9.51790185e-01, -8.06571522e-02, -1.20245712e+00,
        -1.26766340e+00],
        [ 3.56921319e-01, -1.04854298e+00,  1.04612699e+00,
         2.69637740e-01],
        [ 3.56921319e-01, -5.64600066e-01,  1.46693344e-01,
         1.41529311e-01],
        [ 1.66563282e+00, -8.06571522e-02,  1.15855619e+00,
         5.25854597e-01],
        [-1.18973773e-01, -1.04854298e+00, -1.34379669e-01,
        -2.42795974e-01],
        [ 5.94868866e-01, -5.64600066e-01,  7.65053975e-01,
         3.97746168e-01],
        [ 7.13842639e-01,  1.61314304e-01,  9.89912386e-01,
         7.82071454e-01],
        [ 5.94868866e-01, -1.29051444e+00,  6.52624769e-01,
         3.97746168e-01],
        [ 1.07076396e+00,  1.61314304e-01,  1.04612699e+00,
         1.55072202e+00],
        [-1.07076396e+00,  1.61314304e-01, -1.25867172e+00,
        -1.39577183e+00]])
```

```
x_test
```

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=7)
knn.fit(x_train,y_train)
y_pred=knn.predict(x_test)
y_pred
```

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
       'Iris-setosa', 'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
       'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',
       'Iris-setosa', 'Iris-versicolor', 'Iris-virginica',
       'Iris-versicolor'], dtype=object)
```

```
from sklearn.metrics import confusion_matrix,accuracy_score
cm=confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[14  0  0]
 [ 0 18  0]
 [ 0  1 12]]
```

```
score=accuracy_score(y_test,y_pred)
score
```

```
0.9777777777777777
```

```
#To display confusion matrix
from sklearn.metrics import ConfusionMatrixDisplay
labels=['Iris-virginica','Iris-setosa','Iris-versicolor'] #variable set as list and provide
                                                          #the prediction value in the same order as in y_train
cmd=ConfusionMatrixDisplay(cm,display_labels=labels)
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7c4e665f3c70>
```