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

```
df=pd.read_csv("/content/Iris.csv",sep=",")
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.size
```

```
900
```

```
df.shape
```

```
(150, 6)
```

```
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.columns
```

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

```
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

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

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

to find count of unique values in a column

```
df['Species'].value_counts
```

```
<bound method IndexOpsMixin.value_counts of 0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
```

```
4         Iris-setosa
            ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object>
```

```
df['SepalLengthCm'].value_counts
```

```
<bound method IndexOpsMixin.value_counts of 0      5.1
1      4.9
2      4.7
3      4.6
4      5.0
      ...
145    6.7
146    6.3
147    6.5
148    6.2
149    5.9
Name: SepalLengthCm, Length: 150, dtype: float64>
```

```
df['PetalWidthCm'].value_counts
```

```
<bound method IndexOpsMixin.value_counts of 0      0.2
1      0.2
2      0.2
3      0.2
4      0.2
      ...
145    2.3
146    1.9
147    2.0
148    2.3
149    1.8
Name: PetalWidthCm, Length: 150, dtype: float64>
```

SPLITTING INPUT AND OUTPUT VALUES

```
x=df.iloc[:,:-1]
x
```

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

150 rows × 4 columns

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

```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
            ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object
```

TRAIN TEST SPLIT

```
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
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
79	5.7	2.6	3.5	1.0
130	7.4	2.8	6.1	1.9
43	5.0	3.5	1.6	0.6
42	4.4	3.2	1.3	0.2
61	5.9	3.0	4.2	1.5
...	...	...	...	...
121	5.6	2.8	4.9	2.0
131	7.9	3.8	6.4	2.0
113	5.7	2.5	5.0	2.0
41	4.5	2.3	1.3	0.3
100	6.3	3.3	6.0	2.5

105 rows × 4 columns

```
x_test
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
77	6.7	3.0	5.0	1.7
119	6.0	2.2	5.0	1.5
104	6.5	3.0	5.8	2.2
70	5.9	3.2	4.8	1.8
138	6.0	3.0	4.8	1.8
117	7.7	3.8	6.7	2.2
122	7.7	2.8	6.7	2.0
11	4.8	3.4	1.6	0.2
4	5.0	3.6	1.4	0.2
12	4.8	3.0	1.4	0.1
67	5.8	2.7	4.1	1.0
44	5.1	3.8	1.9	0.4
55	5.7	2.8	4.5	1.3
133	6.3	2.8	5.1	1.5
120	6.9	3.2	5.7	2.3
53	5.5	2.3	4.0	1.3
63	6.1	2.9	4.7	1.4
54	6.5	2.8	4.6	1.5
1	4.9	3.0	1.4	0.2
91	6.1	3.0	4.6	1.4
102	7.1	3.0	5.9	2.1
6	4.6	3.4	1.4	0.3
141	6.9	3.1	5.1	2.3
2	4.7	3.2	1.3	0.2

22	4.6	3.6	1.0	0.2
116	6.5	3.0	5.5	1.8
93	5.0	2.3	3.3	1.0
9	4.9	3.1	1.5	0.1
65	6.7	3.1	4.4	1.4
27	5.2	3.5	1.5	0.2
95	5.7	3.0	4.2	1.2

```
y_train
```

```
79     Iris-versicolor
130     Iris-virginica
43         Iris-setosa
42         Iris-setosa
61     Iris-versicolor
           ...
121     Iris-virginica
131     Iris-virginica
113     Iris-virginica
41         Iris-setosa
100     Iris-virginica
Name: Species, Length: 105, dtype: object
```

```
y_test
```

```
77     Iris-versicolor
119     Iris-virginica
104     Iris-virginica
70     Iris-versicolor
138     Iris-virginica
117     Iris-virginica
122     Iris-virginica
11         Iris-setosa
4          Iris-setosa
12         Iris-setosa
67     Iris-versicolor
44         Iris-setosa
55     Iris-versicolor
```

```
133     Iris-virginica
120     Iris-virginica
53     Iris-versicolor
63     Iris-versicolor
54     Iris-versicolor
1          Iris-setosa
91     Iris-versicolor
102     Iris-virginica
6          Iris-setosa
141     Iris-virginica
2          Iris-setosa
22         Iris-setosa
116     Iris-virginica
93     Iris-versicolor
9          Iris-setosa
65     Iris-versicolor
27         Iris-setosa
95     Iris-versicolor
136     Iris-virginica
114     Iris-virginica
0          Iris-setosa
147     Iris-virginica
148     Iris-virginica
86     Iris-versicolor
92     Iris-versicolor
10         Iris-setosa
32         Iris-setosa
89     Iris-versicolor
72     Iris-versicolor
47         Iris-setosa
45         Iris-setosa
17         Iris-setosa
Name: Species, dtype: object
```

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

▾ StandardScaler

StandardScaler()

```
x_train=scaler.transform(x_train)
x_test=scaler.transform(x_test)
```

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

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=7)
```

```
y_pred=knn.predict(x_test)
y_pred
```

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

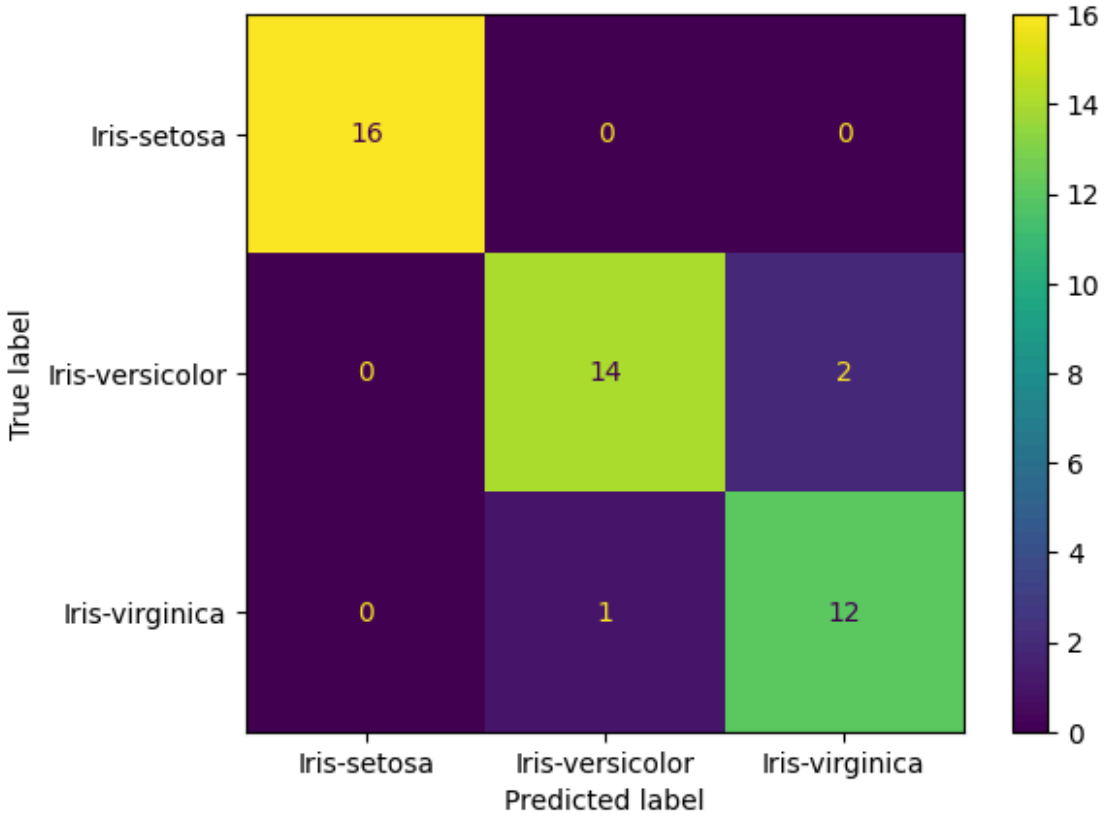
```
from sklearn.metrics import confusion_matrix,accuracy_score,ConfusionMatrixDisplay
```

```
import matplotlib.pyplot as plt
```

```
label=['Iris-setosa','Iris-versicolor','Iris-virginica']
```

```
mat=confusion_matrix(y_pred,y_test)
cmd=ConfusionMatrixDisplay(mat,display_labels=label)
cmd.plot()
print(mat)
```

```
[[16  0  0]
 [ 0 14  2]
 [ 0  1 12]]
```




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

```
0.9333333333333333
```