In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:

```
df=pd.read_csv('/home/ubuntu/Downloads/IRIS.csv')
```

In [3]:

```
df
```

Out[3]:

	sepal_length	sepal_width	petal_length	petal_width	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

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [6]:

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df['species'] =label_encoder.fit_transform(df['species'])
df
```

Out[6]:

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

150 rows × 5 columns

In [8]:

```
x=df.drop("species",axis=1)
x
```

Out[8]:

	sepal_length	sepal_width	petal_length	petal_width
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

In [9]:

```
y=df.species
y
```

Out[9]:

```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: species, Length: 150, dtype: int64
```

In [10]:

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest =train_test_split(x, y, test_size =0.2,random_state =
```

In [11]:

```
from sklearn.naive_bayes import GaussianNB
gaussian = GaussianNB()
gaussian.fit(xtrain,ytrain)
```

Out[11]:

```
▾ GaussianNB

GaussianNB()
```

In [12]:

```
Y_pred = gaussian.predict(xtest)
```

In [13]:

```
from sklearn.metrics import classification_report,accuracy_score,confusion_matrix
cm=confusion_matrix(ytest,Y_pred)
print(cm)
```

```
[[11  0  0]
 [ 0 13  0]
 [ 0  1  5]]
```

In [14]:

```
cr=classification_report(ytest,Y_pred)
print(cr)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        11
           1       0.93      1.00      0.96        13
           2       1.00      0.83      0.91         6

    accuracy                           0.97        30
   macro avg       0.98      0.94      0.96        30
weighted avg       0.97      0.97      0.97        30
```

In [15]:

```
accuracy = accuracy_score(ytest,Y_pred)
```

In [16]:

```
accuracy
```

Out[16]:

```
0.9666666666666667
```

In [ ]:

```

```