In [1]:

```
import pandas as pd
```

In [2]:

```
data = pd.read_csv('Iris.csv')
```

In [3]:

```
data.head()
```

Out[3]:

| | 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 |

In [4]:

```
data.isnull().sum()
```

Out[4]:

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

In [11]:

```
X = data.iloc[:,1:5]
Y = data.iloc[:,-1]
```

In [12]:

```
from sklearn.model_selection import train_test_split
```

In [13]:

```
x_train,x_test,y_train,y_test = train_test_split(X,Y,test_size=0.20)
```

In [14]:

```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(x_train,y_train)
```

Out[14]:

```
GaussianNB()
```

In [16]:

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

Out[16]:

```
array(['Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-setosa', 'Iris-setos
a',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-setosa', 'Iris-versicolor', 'Iris-setos
a',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-virginic
a',
       'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
       'Iris-versicolor', 'Iris-virginica'], dtype='<U15')
```

In [17]:

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

In [18]:

```
accuracy = accuracy_score(y_test,y_pred)*100
confusion_mat = confusion_matrix(y_test,y_pred)
```

In [19]:

```
print(accuracy)
print(confusion_mat)
```

```
100.0
[[11  0  0]
 [ 0  9  0]
 [ 0  0 10]]
```

In [20]:

```
TP = confusion_mat[0][0];
TN =confusion_mat[1][1];
FP = confusion_mat[1][0];
FN = confusion_mat[0][1];

Precision = (TP)/(TP+FP);

Recall = (TP)/(TP+FN)

Error = (FP+FN)/(TP+TN+FN+FP)
```

In [21]:

```
print("TP : ",TP)
print("FP : ",FP)
print("TN : ",TN)
print("FN : ",FN)
print("Precision : ", Precision)
print("Recall : ", Recall)
print("Error : ", Error)
```

```
TP :  11
FP :  0
TN :  9
FN :  0
Precision :  1.0
Recall :  1.0
Error :  0.0
```

In [ ]:

```

```