In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
from sklearn.model_selection import train_test_split
```

In [3]:
```
iris=pd.read_csv("/home/student/Desktop/Iris.csv")
```

In [4]:
```
iris.head()
```

Out[4]:

	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 [5]:
```
iris.isnull().any()
```

Out[5]:
```
Id                 False
SepalLengthCm      False
SepalWidthCm       False
PetalLengthCm      False
PetalWidthCm       False
Species            False
dtype: bool
```

In [6]:
```
x = iris.iloc[:, :4].values
```

In [7]:
```
y = iris['Species'].values
```

In [8]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size = 0.2, random_state=0)
```

In [9]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(x_train)
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
```

In [10]:
```
# import the class
from sklearn.naive_bayes import GaussianNB
gaussian = GaussianNB()
gaussian.fit(x_train, y_train)
```

Out[10]:
```
▾ GaussianNB
GaussianNB()
```

In [11]:
```
y_pred = gaussian.predict(x_test)
y_pred
```

Out[11]:
```
array(['Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-setosa', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa'], dtype='<U15')
```

In [12]:
```
from sklearn.metrics import precision_score,confusion_matrix,accuracy_score,recall_score
cm = confusion_matrix(y_test, y_pred)
```

In [13]:
```
accuracy = accuracy_score(y_test,y_pred)
precision = precision_score(y_test, y_pred,average='micro')
recall = recall_score(y_test, y_pred,average='micro')
```

In [14]:
```
print(accuracy)
print(precision)
print(recall)
print(cm)
```

```
1.0
1.0
1.0
[[11  0  0]
 [ 0 13  0]
 [ 0  0  6]]
```

In [ ]: