

```
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/cota63/iris.csv")
```

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

```
Out[4]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

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

```
Out[5]: sepal_length    False
sepal_width      False
petal_length     False
petal_width      False
species          False
dtype: bool
```

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

```
In [8]: y=iris['species'].values
```

```
In [9]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size = 0.2, random_state=0)
```

```
In [10]: 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 [14]: from sklearn.naive_bayes import GaussianNB
gaussian=GaussianNB()
gaussian.fit(x_train,y_train)
```

```
Out[14]: ▼ GaussianNB
GaussianNB()
```

```
In [15]: y_pred=gaussian.predict(x_test)
y_pred
```

```
Out[15]: array(['virginica', 'versicolor', 'setosa', 'virginica', 'setosa',  
        'virginica', 'setosa', 'versicolor', 'versicolor', 'versicolor',  
        'versicolor', 'versicolor', 'versicolor', 'versicolor',  
        'versicolor', 'setosa', 'versicolor', 'versicolor', 'setosa',  
        'setosa', 'virginica', 'versicolor', 'setosa', 'setosa',  
        'virginica', 'setosa', 'setosa', 'versicolor', 'versicolor',  
        'setosa'], dtype='<U10')
```

```
In [16]: from sklearn.metrics import precision_score, confusion_matrix, accuracy_score, recall_score  
cm = confusion_matrix(y_test, y_pred)
```

```
In [17]: 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 [18]: print(accuracy)  
print(precision)  
print(recall)  
print(cm)
```

```
0.9666666666666667  
0.9666666666666667  
0.9666666666666667  
[[11  0  0]  
 [ 0 13  0]  
 [ 0  1  5]]
```

```
In [ ]:
```