```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score,
precision_score, recall_score

df=pd.read_csv('iris.csv')

df
```

	sepal.length	sepal.width	petal.length	petal.width	variety
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
..	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

```
[150 rows x 5 columns]

df.columns

Index(['sepal.length', 'sepal.width', 'petal.length', 'petal.width',
       'variety'],
      dtype='object')

x=df[['sepal.length', 'sepal.width', 'petal.length', 'petal.width']]

y=df['variety']

X_train, X_test, y_train, y_test = train_test_split(x, y,
test_size=0.3, random_state=42)

model = GaussianNB()
model.fit(X_train, y_train)

GaussianNB()

y_predict = model.predict(X_test)
y_predict

array(['Versicolor', 'Setosa', 'Virginica', 'Versicolor',
'Versicolor',
       'Setosa', 'Versicolor', 'Virginica', 'Versicolor',
'Versicolor',
       'Virginica', 'Setosa', 'Setosa', 'Setosa', 'Setosa',
```

```
'Virginica',
       'Virginica', 'Versicolor', 'Versicolor', 'Virginica', 'Setosa',
       'Virginica', 'Setosa', 'Virginica', 'Virginica', 'Virginica',
       'Virginica', 'Virginica', 'Setosa', 'Setosa', 'Setosa',
'Setosa',
       'Versicolor', 'Setosa', 'Setosa', 'Virginica', 'Versicolor',
       'Setosa', 'Setosa', 'Setosa', 'Virginica', 'Versicolor',
       'Versicolor', 'Setosa', 'Setosa'], dtype='<U10')
```

```
model.score(X_train,y_train)
```

```
0.9428571428571428
```

```
model.score(X_test,y_test)
```

```
0.9777777777777777
```

```
cm=confusion_matrix(y_test,y_predict)
```

```
cm
```

```
array([[19,  0,  0],
       [ 0, 12,  1],
       [ 0,  0, 13]], dtype=int64)
```

```
TP = np.diag(cm)  # True Positives (diagonal elements)
FP = cm.sum(axis=0) - TP  # False Positives
FN = cm.sum(axis=1) - TP  # False Negatives
TN = cm.sum() - (TP + FP + FN)
```

```
print(TP,FP,FN,TN)
```

```
[19 12 13] [0 0 1] [0 1 0] [26 32 31]
```

```
accuracy = accuracy_score(y_test, y_predict)
error_rate = 1 - accuracy
precision = precision_score(y_test, y_predict, average='macro')  # Use
'macro' for multiple classes
recall = recall_score(y_test, y_predict, average='macro')

print(f"Accuracy: {accuracy:.2f}")
print(f"Error Rate: {error_rate:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
```

```
Accuracy: 0.98
Error Rate: 0.02
Precision: 0.98
Recall: 0.97
```