

```
In [1]: import pandas as pd
```

```
In [2]: import numpy as np
```

```
In [3]: data=pd.read_csv("Iris.csv")
```

```
In [4]: df=pd.DataFrame(data)
```

```
In [5]: df
```

```
Out[5]:
```

	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
	...	...	...	...	...	...	...
	145	146	6.7	3.0	5.2	2.3	Iris-virginica
	146	147	6.3	2.5	5.0	1.9	Iris-virginica
	147	148	6.5	3.0	5.2	2.0	Iris-virginica
	148	149	6.2	3.4	5.4	2.3	Iris-virginica
	149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

```
In [6]: df.isnull().sum()
```

```
Out[6]: Id          0
SepalLengthCm      0
SepalWidthCm       0
PetalLengthCm      0
PetalWidthCm       0
Species           0
dtype: int64
```

```
In [7]: from sklearn.model_selection import train_test_split
```

```
In [8]: X = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
```

```
In [9]: Y= df[['Species']]
```

```
In [10]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size =.25)
```

```
In [11]: print(X_train.shape)
```

(112, 4)

```
In [13]: print(Y_train.shape)
```

(112, 1)

```
In [14]: print(Y_test.shape)
```

(38, 1)

```
In [15]: print(X_test.shape)
```

(38, 4)

```
In [16]: import matplotlib.pyplot as plt
```

```
In [17]: from sklearn.naive_bayes import GaussianNB
```

```
In [18]: gaussian = GaussianNB()
```

```
In [19]: gaussian.fit(X_train, Y_train)
```

C:\Users\siddh\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1184: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

```
y = column_or_1d(y, warn=True)
```

```
Out[19]: 

▾ GaussianNB



GaussianNB()


```

```
In [20]: Y_pred = gaussian.predict(X_test)
```

```
In [22]: print(Y_pred)
```

```
['Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa'
'Iris-versicolor' 'Iris-setosa' 'Iris-virginica' 'Iris-virginica'
'Iris-virginica' 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor'
'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
'Iris-versicolor' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
'Iris-virginica' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
'Iris-virginica' 'Iris-versicolor']
```

```
In [45]: from sklearn.metrics import precision_score, confusion_matrix, accuracy_score, recall_score, classification_report
```

```
In [25]: cm = confusion_matrix(Y_test, Y_pred)
```

```
In [26]: print(cm)
```

```
[[ 9  0  0]
 [ 0 12  0]
 [ 0  0 17]]
```

```
In [36]: pc = precision_score(Y_test, Y_pred, average='micro')
```

```
In [37]: print(pc)
```

```
1.0
```

```
In [38]: ac = accuracy_score(Y_test, Y_pred)
```

```
In [39]: print(ac)
```

```
1.0
```

```
In [40]: rc = recall_score(Y_test, Y_pred, average='micro')
```

```
In [41]: print(rc)
```

```
1.0
```

```
In [42]: error_rate = 1 - ac
```

```
In [43]: print(error_rate)
```

```
0.0
```

```
In [46]: cr=classification_report(Y_test,Y_pred)
```

```
In [47]: print(cr)
```

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	9
Iris-versicolor	1.00	1.00	1.00	12
Iris-virginica	1.00	1.00	1.00	17
accuracy			1.00	38
macro avg	1.00	1.00	1.00	38
weighted avg	1.00	1.00	1.00	38

```
In [ ]:
```