

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

```
In [2]: #import dataset
```

```
In [3]: dataset = pd.read_csv("https://raw.githubusercontent.com/mk-gurucharan/Class
```

```
In [4]: X=dataset.iloc[:,4].values
y = dataset['species'].values
print(dataset.head(5))
```

	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 [ ]: # Splitting the dataset into the Training set and Test set
```

```
In [5]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

```
In [6]: #Feature Scaling
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
In [7]: #Training the Naive Bayes Classification model on the Training Set
```

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

```
Out[7]: GaussianNB()
```

```
In [8]: #: Predicting the Test set results
```

```
y_pred = classifier.predict(X_test)
```

```
In [9]: #Confusion Matrix and Accuracy
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import accuracy_score
print("Accuracy : ", accuracy_score(y_test,y_pred))
print(cm)
```

```
Accuracy : 0.9333333333333333
[[14  0  0]
 [ 0  7  1]
 [ 0  1  7]]
```

```
In [10]: #Comparing the Real Values with Predicted Values
df = pd.DataFrame({'Real Values':y_test, 'Predicted Values':y_pred})
print(df)
```

	Real Values	Predicted Values
0	virginica	virginica
1	virginica	versicolor
2	virginica	virginica
3	setosa	setosa
4	virginica	virginica
5	versicolor	versicolor
6	setosa	setosa
7	setosa	setosa
8	virginica	virginica
9	setosa	setosa
10	setosa	setosa
11	virginica	virginica
12	versicolor	versicolor
13	versicolor	versicolor
14	setosa	setosa
15	versicolor	versicolor
16	setosa	setosa
17	setosa	setosa
18	setosa	setosa
19	setosa	setosa
20	versicolor	versicolor
21	versicolor	versicolor
22	setosa	setosa
23	versicolor	virginica
24	setosa	setosa
25	virginica	virginica
26	virginica	virginica
27	setosa	setosa
28	setosa	setosa
29	versicolor	versicolor

```
In [11]: #: Compute Error rate, Precision and Recall

from sklearn.metrics import precision_score, recall_score, accuracy_score
```

```
In [ ]: #Error Rate
```

```
In [12]: m=accuracy_score(y_test, y_pred)
print("error rate:-",1-m)
```

```
error rate:- 0.06666666666666665
```

```
In [ ]: #precision
```

```
In [13]: print('Precision:',precision_score(y_test,y_pred,average='micro'))
```

```
File "C:\Users\avcom\AppData\Local\Temp\ipykernel_9664\163736583.py", line 1
```

```
    print('Precision:',precision_score(y_test,y_pred,average='micro'))
```

```
SyntaxError: unexpected EOF while parsing
```

```
In [14]: print('Precision:',precision_score(y_test,y_pred,average='micro'))
```

Precision: 0.9333333333333333

```
In [ ]: #Recall
```

```
In [15]: print("Recall Score:",recall_score(y_test,y_pred,average='micro'))
```

Recall Score: 0.9333333333333333

```
In [ ]:
```