In [1]:
```
#7. Data Analytics
#Implement Simple Naïve Bayes classification algorithm using Python/R on iris.
#ComputeConfusionmatrixtofindTP,FP,TN,FN,Accuracy,Errorrate,Precision,
#Recall on the given dataset.

#iris.csv
#****1St try
#sssss
```

In [30]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [31]:
```
df = pd.read_csv('iris.csv')
df.head()
```

Out[31]:

	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 [32]:
```
df.rename(columns={"SepalLengthCm":"sepal_length",
                   "SepalWidthCm":"sepal_width",
                   "PetalLengthCm":"petal_length",
                   "PetalWidthCm":"petal_width",
                   "Species":"species"},inplace=True)
df.head()
```

Out[32]:

	Id	sepal_length	sepal_width	petal_length	petal_width	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 [33]:
```
#df = pd.DataFrame(df1)
df.head()
```

Out[33]:

	Id	sepal_length	sepal_width	petal_length	petal_width	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 [34]:
```
#Step 3: Checking for null values
```

In [35]:
```
df.isnull().sum()
```

Out[35]:
```
Id              0
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [36]:
```
x = df.iloc[:,:4].values
y = df['species'].values
```

In [37]:
```
#Step 4: Splitting the dataset into the Training set and Test set
```

In [38]:
```
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 [39]:
```
#Step 5: Feature Scaling
```

In [40]:
```
from sklearn.preprocessing
import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

In [41]:
```
#Step 6: Training the Naive Bayes Classification model on the Training Set
```

In [42]:
```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(x_train, y_train)
```

Out[42]:
```
GaussianNB()
```

In [43]:
```
#In a Jupyter environment, please rerun this cell to show the HTML representat
#On GitHub, the HTML representation is unable to render, please try loading th

#Step 7: Predicting the Test set results
```

In [44]:
```
y_pred = classifier.predict(x_test)
y_pred
```

Out[44]:
```
array(['Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa',
       'Iris-virginica', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-setosa', 'Iris-virginica', 'Iris-virginica'], dtype='<U15')
```

In [45]:
```
#Step 8: Confusion Matrix and Accuracy
```

In [46]:
```
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))
cm
```

```
Accuracy :  1.0
```

Out[46]:
```
array([[ 7,  0,  0],
       [ 0, 12,  0],
       [ 0,  0, 11]], dtype=int64)
```

In [47]:
```
#Step 9: Comparing the Real Values with Predicted Values
```

In [48]:
```
df = pd.DataFrame({'Real Values':y_test,
                   'Predicted Values':y_pred})
df.head()
```

Out[48]:

	Real Values	Predicted Values
0	Iris-virginica	Iris-virginica
1	Iris-versicolor	Iris-versicolor
2	Iris-setosa	Iris-setosa
3	Iris-versicolor	Iris-versicolor
4	Iris-versicolor	Iris-versicolor

In [ ]:
```
```