In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [3]:
```
dataset=pd.read_csv("iris.csv")
```

In [4]:
```
dataset.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]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [6]:
```
X=dataset.iloc[:,:4].values
y=dataset["species"].values
```

In [7]:
```
y
```

Out[7]:
```
array(['setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'virginica', 'virginica',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
```

```
       'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
       'virginica', 'virginica', 'virginica'], dtype=object)
```

In [8]:
```
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 [15]:
```
from sklearn.preprocessing import StandardScaler,normalize
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
print(X_test)
```

```
[[-0.11774441  2.19548765 -1.52234863 -1.34836924]
 [-0.11774441 -0.862513    0.16430315 -0.30193879]
 [-1.65912576  0.78410273 -1.40602782 -1.21756544]
 [-1.7875742  -1.80343628 -1.46418823 -1.21756544]
 [-1.91602265  0.31364109 -1.46418823 -1.34836924]
 [ 0.78139471  0.31364109  0.74590721  1.00609928]
 [ 1.93743072 -0.62728218  1.32751127  0.87529548]
 [-1.53067731  0.31364109 -1.46418823 -1.34836924]
 [-1.27378042 -1.56820546 -0.3009801  -0.30193879]
 [ 0.13915248 -0.39205137  0.39694477  0.35208025]
 [ 2.32277606 -0.62728218  1.67647371  1.00609928]
 [ 0.52449782 -0.39205137  1.03670924  0.74449167]
 [ 0.65294627  0.07841027  0.97854884  0.74449167]
 [-1.14533197  0.31364109 -1.52234863 -1.34836924]
 [-1.14533197 -1.80343628 -0.3009801  -0.30193879]
 [-1.65912576  0.07841027 -1.34786742 -1.34836924]
 [ 2.57967295  1.72502601  1.50199249  1.00609928]
 [ 0.52449782 -1.33297464  0.6877468   0.87529548]
 [-1.01688353  1.72502601 -1.11522579 -1.08676163]
 [-0.3746413  -0.862513    0.22246355  0.09047263]
 [-1.27378042 -1.33297464  0.39694477  0.61368786]
 [-1.01688353  1.48979519 -1.34786742 -1.08676163]
 [-0.24619285 -0.15682055  0.22246355 -0.04033117]
 [-0.50308975 -1.80343628  0.10614274  0.09047263]
 [-1.40222886  0.07841027 -1.28970701 -1.34836924]
 [ 1.0382916  -1.33297464  1.15303005  0.74449167]
 [ 1.0382916   0.54887191  1.09486965  1.13690309]
 [ 0.13915248 -0.15682055  0.57142599  0.74449167]
 [-1.91602265 -0.15682055 -1.46418823 -1.34836924]
 [-0.88843508  0.78410273 -1.40602782 -1.34836924]]
```

In [41]:
```
from sklearn.naive_bayes import GaussianNB, MultinomialNB
model = GaussianNB()
model.fit(X_train, y_train)
```

Out[41]:
```
GaussianNB()
```

In [31]:
```
y_pred = model.predict(X_test)
y_pred
```

Out[31]:
```
array(['setosa', 'versicolor', 'setosa', 'setosa', 'setosa', 'virginica',
       'virginica', 'setosa', 'versicolor', 'versicolor', 'virginica',
       'virginica', 'virginica', 'setosa', 'versicolor', 'setosa',
       'virginica', 'virginica', 'setosa', 'versicolor', 'versicolor',
       'setosa', 'versicolor', 'versicolor', 'setosa', 'virginica',
       'virginica', 'virginica', 'setosa', 'setosa'], dtype='<U10')
```

In [36]:
```
model.predict([[-0.11774441,  2.19548765, -1.52234863, -1.34836924]])
#X_test[0]
```

```
array(['setosa'], dtype='<U10')
```

Out[36]:

In [26]:
```
from sklearn.metrics import confusion_matrix,precision_score,recall_score
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import accuracy_score
print ("Accuracy : ", accuracy_score(y_test, y_pred))
```

```
Accuracy :  0.9666666666666667
```

In [27]:
```
df = pd.DataFrame({'original Values':y_test, 'Predicted Values':y_pred})
df
```

Out[27]:

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

	original Values	Predicted Values
29	setosa	setosa

In [28]:
```
print("Precision Score : ",precision_score(y_test, y_pred,pos_label='positive', aver
print("Recall Score : ",recall_score(y_test, y_pred, pos_label='positive',average='m
```

```
Precision Score :  0.9666666666666667
Recall Score :  0.9666666666666667
/home/pktc-320/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_classificatio
n.py:1298: UserWarning: Note that pos_label (set to 'positive') is ignored when aver
age != 'binary' (got 'micro'). You may use labels=[pos_label] to specify a single po
sitive class.
  warnings.warn("Note that pos_label (set to %r) is ignored when "
/home/pktc-320/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_classificatio
n.py:1298: UserWarning: Note that pos_label (set to 'positive') is ignored when aver
age != 'binary' (got 'micro'). You may use labels=[pos_label] to specify a single po
sitive class.
  warnings.warn("Note that pos_label (set to %r) is ignored when "
```

In [ ]:

In [ ]: