In [2]:

```
import pandas as pd
import numpy as np
import matplotlib as pl
import seaborn as sns
```

In [3]:

```
df = pd.read_csv("/home/anaconda_user/Downloads/Dataset/iris.csv")
df
```

Out[3]:

	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
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa
10	5.4	3.7	1.5	0.2	setosa
11	4.8	3.4	1.6	0.2	setosa
12	4.8	3.0	1.4	0.1	setosa
13	4.3	3.0	1.1	0.1	setosa
14	5.8	4.0	1.2	0.2	setosa
15	5.7	4.4	1.5	0.4	setosa
16	5.4	3.9	1.3	0.4	setosa
17	5.1	3.5	1.4	0.3	setosa
18	5.7	3.8	1.7	0.3	setosa
19	5.1	3.8	1.5	0.3	setosa
20	5.4	3.4	1.7	0.2	setosa
21	5.1	3.7	1.5	0.4	setosa
22	4.6	3.6	1.0	0.2	setosa
23	5.1	3.3	1.7	0.5	setosa
24	4.8	3.4	1.9	0.2	setosa
25	5.0	3.0	1.6	0.2	setosa
26	5.0	3.4	1.6	0.4	setosa
27	5.2	3.5	1.5	0.2	setosa
28	5.2	3.4	1.4	0.2	setosa
29	4.7	3.2	1.6	0.2	setosa
...	...	...	...	...	...
120	6.9	3.2	5.7	2.3	virginica
121	5.6	2.8	4.9	2.0	virginica
122	7.7	2.8	6.7	2.0	virginica
123	6.3	2.7	4.9	1.8	virginica
124	6.7	3.3	5.7	2.1	virginica
125	7.2	3.2	6.0	1.8	virginica
126	6.2	2.8	4.8	1.8	virginica
127	6.1	3.0	4.9	1.8	virginica
128	6.4	2.8	5.6	2.1	virginica
129	7.2	3.0	5.8	1.6	virginica
130	7.4	2.8	6.1	1.9	virginica
131	7.9	3.8	6.4	2.0	virginica

132	6.4	2.8	5.6	2.2	virginica
133	6.3	2.8	5.1	1.5	virginica
134	6.1	2.6	5.6	1.4	virginica
135	7.7	3.0	6.1	2.3	virginica
136	6.3	3.4	5.6	2.4	virginica
137	6.4	3.1	5.5	1.8	virginica
138	6.0	3.0	4.8	1.8	virginica
139	6.9	3.1	5.4	2.1	virginica
140	6.7	3.1	5.6	2.4	virginica
141	6.9	3.1	5.1	2.3	virginica
142	5.8	2.7	5.1	1.9	virginica
143	6.8	3.2	5.9	2.3	virginica
144	6.7	3.3	5.7	2.5	virginica
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 × 5 columns

In [4]:

```
df.shape
```

Out[4]:

```
(150, 5)
```

In [5]:

```
df.columns
```

Out[5]:

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [6]:

```
df.dtypes
```

Out[6]:

```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
species          object
dtype: object
```

In [7]:

```
df.describe()
```

Out[7]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
sepal_length    150 non-null float64
sepal_width     150 non-null float64
petal_length    150 non-null float64
petal_width     150 non-null float64
species         150 non-null object
dtypes: float64(4), object(1)
memory usage: 5.9+ KB
```

In [9]:

```
df.isnull().sum()
```

Out[9]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [10]:

```
X = df.iloc[:,[0,1,2,3]]
X
```

Out[10]:

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5.0	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1
10	5.4	3.7	1.5	0.2
11	4.8	3.4	1.6	0.2
12	4.8	3.0	1.4	0.1
13	4.3	3.0	1.1	0.1
14	5.8	4.0	1.2	0.2
15	5.7	4.4	1.5	0.4
16	5.4	3.9	1.3	0.4
17	5.1	3.5	1.4	0.3
18	5.7	3.8	1.7	0.3
19	5.1	3.8	1.5	0.3
20	5.4	3.4	1.7	0.2
21	5.1	3.7	1.5	0.4
22	4.6	3.6	1.0	0.2
23	5.1	3.3	1.7	0.5
24	4.8	3.4	1.9	0.2
25	5.0	3.0	1.6	0.2
26	5.0	3.4	1.6	0.4
27	5.2	3.5	1.5	0.2
28	5.2	3.4	1.4	0.2
29	4.7	3.2	1.6	0.2
...	...	...	...	...

120	6.9	3.2	5.7	2.3
121	5.6	2.8	4.9	2.0
122	7.7	2.8	6.7	2.0
123	6.3	2.7	4.9	1.8
124	6.7	3.3	5.7	2.1
125	7.2	3.2	6.0	1.8
126	6.2	2.8	4.8	1.8
127	6.1	3.0	4.9	1.8
128	6.4	2.8	5.6	2.1
129	7.2	3.0	5.8	1.6
130	7.4	2.8	6.1	1.9
131	7.9	3.8	6.4	2.0
132	6.4	2.8	5.6	2.2
133	6.3	2.8	5.1	1.5
134	6.1	2.6	5.6	1.4
135	7.7	3.0	6.1	2.3
136	6.3	3.4	5.6	2.4
137	6.4	3.1	5.5	1.8
138	6.0	3.0	4.8	1.8
139	6.9	3.1	5.4	2.1
140	6.7	3.1	5.6	2.4
141	6.9	3.1	5.1	2.3
142	5.8	2.7	5.1	1.9
143	6.8	3.2	5.9	2.3
144	6.7	3.3	5.7	2.5
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [11]:

```
Y = df.iloc[:,4]
Y
```

Out[11]:

```
0         setosa
1         setosa
2         setosa
3         setosa
4         setosa
5         setosa
6         setosa
7         setosa
8         setosa
9         setosa
10        setosa
11        setosa
12        setosa
13        setosa
14        setosa
15        setosa
16        setosa
17        setosa
18        setosa
19        setosa
20        setosa
21        setosa
22        setosa
23        setosa
24        setosa
25        setosa
26        setosa
27        setosa
28        setosa
29        setosa
         ...
120    virginica
121    virginica
122    virginica
123    virginica
124    virginica
125    virginica
126    virginica
127    virginica
128    virginica
129    virginica
130    virginica
131    virginica
132    virginica
133    virginica
134    virginica
135    virginica
136    virginica
137    virginica
138    virginica
139    virginica
140    virginica
141    virginica
142    virginica
143    virginica
144    virginica
145    virginica
146    virginica
147    virginica
148    virginica
149    virginica
Name: species, Length: 150, dtype: object
```

In [12]:

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
```

In [13]:

```
X_train , X_test, Y_train , Y_test = train_test_split(X,Y,test_size = 0.25)
```

In [14]:

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
```

In [15]:

```
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.fit_transform(X_test)
```

In [16]:

```
Classifier = GaussianNB()
Classifier.fit(X_train, Y_train)
```

Out[16]:

```
GaussianNB(priors=None)
```

In [17]:

```
Y_pred = Classifier.predict(X_test)
```

In [19]:

```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(Y_test, Y_pred)*100
accuracy
```

Out[19]:

```
92.10526315789474
```

In [18]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test,Y_pred)
```

In [22]:

```
cm
```

Out[22]:

```
array([[15,  1,  0],
       [ 0,  9,  2],
       [ 0,  0, 11]])
```

In [23]:

```
TP = cm[0,[0]]
print(TP)
```

```
[15]
```

In [36]:

```
TN = cm[[[1],[2]],[1,2]]
print(TN)
```

```
[[ 9  2]
 [ 0 11]]
```

In [26]:

```
FP = cm[[1,2],0]
print(FP)
```

```
[0 0]
```

In [25]:

```
FN = cm[0,[1,2]]
print(FN)
```

```
[1 0]
```

In [37]:

```
accuracy = (TP+TN)/(TP+TN+FP+FN)
print(accuracy)
```

```
[[0.96   1.    ]
 [0.9375 1.    ]]
```

In [38]:

```
error_rate =(FP+FN)/(TP+TN+FP+FN)
print(error_rate)
```

```
[[0.04   0.    ]
 [0.0625 0.    ]]
```

In [39]:

```
precision = TP/(TP+FP)
print(precision)
```

```
[1. 1.]
```

In [40]:

```
Recall = (TP/(TP+FN))
print(Recall)
```

```
[0.9375 1.    ]
```

In [42]:

```
specificity = (TN/(TN+FP))
print(specificity)
```

```
[[ 1.  1.]
 [nan  1.]]

/home/anaconda_user/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: RuntimeWarning: i
nvalid value encountered in true_divide
  """Entry point for launching an IPython kernel.
```

In [48]:

```
data = [[5.4,3.9,1.7,0.4]]
spec = Classifier.predict(data)
spec
```

Out[48]:

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

In [45]:

```
import pickle
```

In [46]:

```
pickle.dump(Classifier, open('Classifier.pkl','wb'))
```

In [49]:

```
pickled_model = pickle.load(open('Classifier.pkl','rb'))
data = [[5.4,3.9,1.7,0.4]]
pickled_model.predict(data)
```

Out[49]:

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