Assignment No 6

```
import  pandas as pd
import numpy as np
import io
```

```
from google.colab import files
```

```
upload = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving Iris.csv to Iris.csv
```

```
df=pd.read_csv(io.BytesIO(upload['Iris.csv']))
```

```
df.head()
```

```
   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
```

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

```
Id                 0
SepalLengthCm      0
SepalWidthCm       0
PetalLengthCm      0
PetalWidthCm       0
Species            0
dtype: int64
```

```
x=df.iloc[:,1:5].values
y=df.iloc[:,5].values
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
y=le.fit_transform(y)
print(y)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.25,random_state=0)
```

```
from sklearn.naive_bayes import GaussianNB
gaussian = GaussianNB()
gaussian.fit(x_train,y_train)
```

```
GaussianNB()
```

```
y_predict=gaussian.predict(x_test)
```

```
from sklearn.metrics import confusion_matrix
c_mat = confusion_matrix(y_test,y_predict)
print(c_mat)
```

```
[[13  0  0]
 [ 0 16  0]
 [ 0  0  9]]
```

```
from sklearn.metrics import accuracy_score,precision_score,recall_score
accuracy = accuracy_score(y_test,y_predict)
accuracy = accuracy_score(y_test,y_predict)
precision =precision_score(y_test, y_predict,average='micro')
recall =  recall_score(y_test, y_predict,average='micro')
print('accuracy_Naive Bayes: %.3f' %accuracy)
print('precision_Naive Bayes: %.3f' %precision)
print('recall_Naive Bayes: %.3f' %recall)
# print('f1-score_Naive Bayes : %.3f' %f1)
```

```
accuracy_Naive Bayes: 1.000
precision_Naive Bayes: 1.000
recall_Naive Bayes: 1.000
```