6) Data Analytics III

1. Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset.
2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv('IRIS.csv')
df.head()
```

Out[2]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [3]:
```
df.isna().sum()
```

Out[3]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [4]:
```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import precision_score,accuracy_score,confusion_matrix,recall_score,ConfusionMatrixDisplay
```

In [5]:
```
labelencoder = LabelEncoder()
df['species'] = labelencoder.fit_transform(df['species'])
```

In [6]:
```
x = df.drop(columns=['species'])
y = df['species']
```

In [7]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2)
```

In [8]:
```
model = GaussianNB()
model.fit(x_train, y_train)
```

Out[8]:
```
GaussianNB()
```

In [9]:
```
predicted = model.predict(x_test)
predicted
```

Out[9]:
```
array([0, 0, 0, 2, 2, 1, 1, 1, 1, 2, 2, 2, 0, 0, 2, 2, 2, 1, 0, 1, 1, 0,
       1, 2, 1, 2, 1, 0, 0, 1])
```

In [10]:
```
precision_score(y_test, predicted, average='micro')
```

Out[10]:
```
0.9333333333333333
```

In [12]:
```
recall_score(y_test, predicted, average='macro')
```

Out[12]:
```
0.9487179487179488
```

In [13]:
```
accuracy_score(y_test, predicted)
```

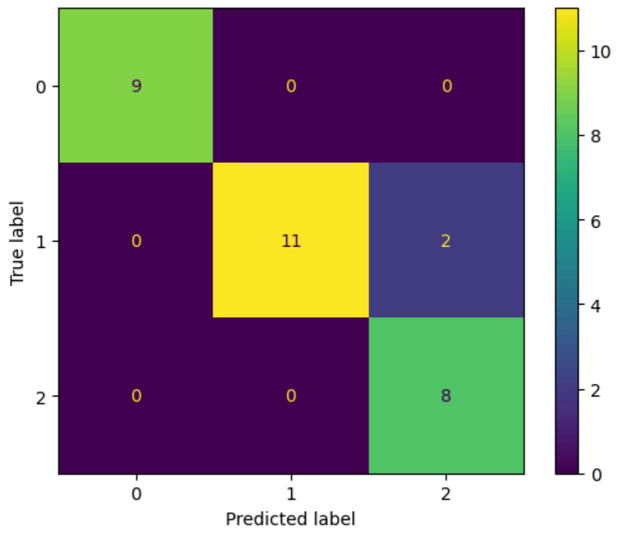
Out[13]:
```
0.9333333333333333
```

In [14]:
```
cm = confusion_matrix(y_test, predicted)
cm
```

Out[14]:
```
array([[ 9,  0,  0],
       [ 0, 11,  2],
       [ 0,  0,  8]], dtype=int64)
```

In [15]:
```
ConfusionMatrixDisplay(confusion_matrix=cm).plot()
```

Out[15]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1bd5d076d20>
```



In [ ]: