```
import numpy as np
import pandas as pd
 from sklearn.model_selection import train_test_split
from \ sklearn.naive\_bayes \ import \ Gaussian NB
import matplotlib.pyplot as plt
import seaborn as sns
from \ sklearn. metrics \ import \ confusion\_matrix, Confusion Matrix Display, classification\_report, accuracy\_score, \ precision\_score, \ recall\_score, \ f1\_score, \ f1\_sc
 from sklearn.preprocessing import LabelEncoder
data = pd.read_csv('Iris.csv')
data.head(5)
  <del>_</del>
                                            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
```

data.describe(include = 'all')

4.7

4.6

3.2

3.1

2

₹

		sepal_length	sepal_width	petal_length	petal_width	species
	count	150.000000	150.000000	150.000000	150.000000	150
	unique	NaN	NaN	NaN	NaN	3
	top	NaN	NaN	NaN	NaN	setosa
	freq	NaN	NaN	NaN	NaN	50
	mean	5.843333	3.054000	3.758667	1.198667	NaN
	std	0.828066	0.433594	1.764420	0.763161	NaN
	min	4.300000	2.000000	1.000000	0.100000	NaN
	25%	5.100000	2.800000	1.600000	0.300000	NaN
	50%	5.800000	3.000000	4.350000	1.300000	NaN
	75%	6.400000	3.300000	5.100000	1.800000	NaN
	max	7.900000	4.400000	6.900000	2.500000	NaN

1.3

1.5

1.4

0.2

0.2

0.2

setosa

setosa

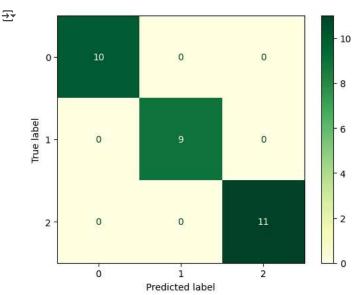
setosa

data.info()

<del>_</del>	<pre><class 'pandas.core.frame.dataframe'="">    RangeIndex: 150 entries, 0 to 149    Data columns (total 5 columns):     # Column Non-Null Count Dtype</class></pre>									
			Non-Null Count							
	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									
<pre>data.columns = data.columns.str.strip().str.lower() print(data['species'].unique())</pre>										
: ['setosa' 'versicolor' 'virginica']										
<pre>data.isnull().sum()</pre>										
<del></del>	sepa peta	l_length 0 l_width 0 l_length 0 l_width 0								

```
species
                0
    dtype: int64
x = data.iloc[:,1:5]
y = data.iloc[:,5:]
from sklearn.preprocessing import LabelEncoder
encode = LabelEncoder()
y = encode.fit_transform(data['species'])
print(y)
2 21
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
# Features and target
X = data.drop(columns='species')
y = LabelEncoder().fit_transform(data['species'])
# Split data
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Train model
naive_bayes = GaussianNB()
naive_bayes.fit(x_train, y_train)
# Predict
pred = naive_bayes.predict(x_test)
print(pred)
pred
\Rightarrow array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2,
         0, 2, 2, 2, 2, 2, 0, 0])
y_test
\Rightarrow array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2,
         0, 2, 2, 2, 2, 2, 0, 0])
matrix = confusion_matrix(y_test,pred,labels = naive_bayes.classes_)
print(matrix)
tp, fn, fp, tn = confusion_matrix(y_test,pred,labels=[1,0]).reshape(-1)
→ [[10 0 0]
    [0 9 0]
    [0 0 11]]
conf_matrix = ConfusionMatrixDisplay(confusion_matrix=matrix,display_labels=naive_bayes.classes_)
conf matrix.plot(cmap=plt.cm.YlGn)
plt.show()
```

3/18/25, 5:29 PM 6.ipynb - Colab



## print(classification\_report(y\_test,pred))

₹	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	9
2	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

```
print('\nAccuracy: {:.2f}'.format(accuracy_score(y_test,pred)))
print('Error Rate: ',(fp+fn)/(tp+tn+fn+fp))
print('Sensitivity (Recall or True positive rate) :',tp/(tp+fn))
print('Specificity (True negative rate) :',tn/(fp+tn))
print('Precision (Positive predictive value) :',tp/(tp+fp))
print('False Positive Rate :',fp/(tn+fp))
```

 $\overline{\Rightarrow}$ 

Accuracy: 1.00 Error Rate: 0.0

Sensitivity (Recall or True positive rate) : 1.0

Specificity (True negative rate): 1.0 Precision (Positive predictive value): 1.0

False Positive Rate : 0.0