In [1]:

**import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

In [3]:

data **=** pd**.**read\_csv(r"C:\Users\ARYAN SURESH PAWAR\Desktop\datasets\datasets\iris.csv data**.**head()

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

In [5]:

data**.**shape

Out[5]:

In [6]:

Out[6]:

In [7]:

(150, 5)

data**.**tail

<bound method NDFrame.tail of 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 |
| .. | ... | | ... | ... | ... | ... |
| 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 x 5 | columns]> | | | | |

<class 'pandas.core.frame.DataFrame'> RangeIndex: 150 entries, 0 to 149 Data columns (total 5 columns):

data**.**info()

# 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 [8]:

data**.**describe

Out[8]:

In [9]:

<bound method NDFrame.describe of sepal\_length sepal\_width petal\_length pe

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| tal\_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 |
| .. | ... | ... | ... | ... | ... |
| 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 x 5 columns]>

data**.**isnull()**.**sum()

|  |  |  |
| --- | --- | --- |
| Out[9]: | sepal\_length | 0 |
|  | sepal\_width | 0 |
|  | petal\_length | 0 |
|  | petal\_width | 0 |
|  | species | 0 |
|  | dtype: int64 |  |

In [16]:

X **=** data**.**drop(['species'], axis**=**1)

y **=** data**.**drop(['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width'], ax print(X)

print(y) print(X**.**shape) print(y**.**shape)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 |
| .. | ... | ... | ... | ... |
| 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 x 4 columns] species

1. setosa
2. setosa
3. setosa
4. setosa
5. setosa

.. ...

145 virginica

|  |  |
| --- | --- |
| 146 | virginica |
| 147 | virginica |
| 148 | virginica |
| 149 | virginica |
| [150 | rows x 1 columns] |

(150, 4)

(150, 1)

In [17]:

**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, shuffle**=Tr** print(X\_train**.**shape)

print(X\_test**.**shape) print(y\_train**.**shape) print(y\_test**.**shape)

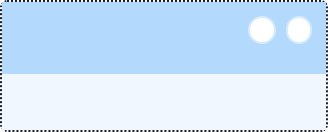
(120, 4)

(30, 4)

(120, 1)

(30, 1)

In [18]:



▾ GaussianNB i

GaussianNB()

[?](https://scikit-learn.org/1.4/modules/generated/sklearn.naive_bayes.GaussianNB.html)

C:\Users\ARYAN SURESH PAWAR\AppData\Local\Programs\Python\Python312\Lib\site-package s\sklearn\utils\validation.py:1300: DataConversionWarning: A column-vector y was pas sed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

y = column\_or\_1d(y, warn=True)

**from** sklearn.naive\_bayes **import** GaussianNB model **=** GaussianNB()

model**.**fit(X\_train, y\_train)

Out[18]:

In [19]:

y\_pred **=** model**.**predict(X\_test) model**.**score(X\_test,y\_test)

Out[19]: 0.9666666666666667

In [20]:

**from** sklearn.metrics **import** accuracy\_score, confusion\_matrix, ConfusionMatrixDispla print(accuracy\_score(y\_test, y\_pred))

0.9666666666666667

In [21]:

cm **=** confusion\_matrix(y\_test, y\_pred)

disp **=** ConfusionMatrixDisplay(confusion\_matrix **=** cm) print("Confusion matrix:")

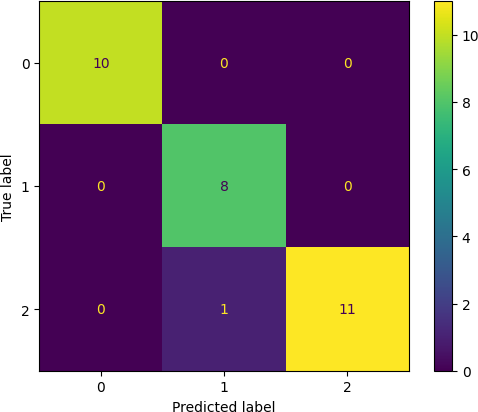
print(cm)

Confusion matrix:

|  |  |  |
| --- | --- | --- |
| [[10 | 0 | 0] |
| [ 0 | 8 | 0] |
| [ 0 | 1 | 11]] |

In [22]:

disp**.**plot() plt**.**show()



In [23]:

**def** get\_confusion\_matrix\_values(y\_true, y\_pred): cm **=** confusion\_matrix(y\_true, y\_pred) **return**(cm[0][0], cm[0][1], cm[1][0], cm[1][1])

TP, FP, FN, TN **=** get\_confusion\_matrix\_values(y\_test, y\_pred) print("TP: ", TP)

print("FP: ", FP)

print("FN: ", FN)

print("TN: ", TN)

|  |  |
| --- | --- |
| TP: | 10 |
| FP: | 0 |
| FN: | 0 |
| TN: | 8 |

In [24]:

print("The Accuracy is ", (TP**+**TN)**/**(TP**+**TN**+**FP**+**FN)) print("The precision is ", TP**/**(TP**+**FP)) print("The recall is ", TP**/**(TP**+**FN))

The Accuracy is 1.0 The precision is 1.0 The recall is 1.0

In [ ]: