

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

```
In [3]: df = pd.read_csv('C:/Users/prajw/Desktop/Indexes/DSBDA print/Assignment 6 (Data Analytics-III)/Iris.csv')
```

```
In [4]: df
```

Out[4]:

	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

	145	146	6.7	3.0	5.2	2.3	Iris-virginica
	146	147	6.3	2.5	5.0	1.9	Iris-virginica
	147	148	6.5	3.0	5.2	2.0	Iris-virginica
	148	149	6.2	3.4	5.4	2.3	Iris-virginica
	149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

```
In [5]: df.shape
```

Out[5]: (150, 6)

```
In [6]: df.describe()
```

Out[6]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

```
In [7]: df.isnull()
```

Out[7]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
	0	False	False	False	False	False
	1	False	False	False	False	False
	2	False	False	False	False	False
	3	False	False	False	False	False
	4	False	False	False	False	False

	145	False	False	False	False	False
	146	False	False	False	False	False
	147	False	False	False	False	False
	148	False	False	False	False	False
	149	False	False	False	False	False

150 rows × 6 columns

```
In [8]: df.isnull().sum()
```

```
Out[8]: Id      0
        SepalLengthCm  0
        SepalWidthCm   0
        PetalLengthCm  0
        PetalWidthCm   0
        Species      0
        dtype: int64
```

```
In [9]: x = df.drop(["Species"],axis=1)
        y = df["Species"]
```

```
In [10]: 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, random_state=0)

         print(x_train.shape)
         print(y_train.shape)
         print(x_test.shape)
         print(y_test.shape)

(120, 5)
(120,)
(30, 5)
(30,)
```

```
In [11]: from sklearn.naive_bayes import MultinomialNB
```

```
In [12]: classifier = MultinomialNB()
         classifier.fit(x_train,y_train)
```

```
Out[12]: ▾ MultinomialNB ⓘ ?
         MultinomialNB()
```

```
In [13]: classifier.score(x_test, y_test)
```

```
Out[13]: 0.8333333333333334
```

```
In [14]: y_pred = classifier.predict(x_test)
```

```
In [15]: y_pred
```

```
Out[15]: array(['Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
                'Iris-virginica', 'Iris-setosa', 'Iris-versicolor', 'Iris-setosa',
                'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
                'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
                'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
                'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
                'Iris-setosa', 'Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
                'Iris-setosa', 'Iris-virginica', 'Iris-setosa', 'Iris-setosa',
                'Iris-virginica', 'Iris-virginica', 'Iris-setosa'], dtype='<U15')
```

```
In [16]: y_test
```

```
Out[16]: 114    Iris-virginica
         62    Iris-versicolor
         33      Iris-setosa
        107    Iris-virginica
          7      Iris-setosa
        100    Iris-virginica
         40      Iris-setosa
         86    Iris-versicolor
         76    Iris-versicolor
         71    Iris-versicolor
        134    Iris-virginica
         51    Iris-versicolor
         73    Iris-versicolor
         54    Iris-versicolor
         63    Iris-versicolor
         37      Iris-setosa
         78    Iris-versicolor
         90    Iris-versicolor
         45      Iris-setosa
         16      Iris-setosa
        121    Iris-virginica
         66    Iris-versicolor
         24      Iris-setosa
          8      Iris-setosa
        126    Iris-virginica
         22      Iris-setosa
         44      Iris-setosa
         97    Iris-versicolor
         93    Iris-versicolor
         26      Iris-setosa
         Name: Species, dtype: object
```

```
In [17]: import sklearn.metrics

lbs = ['Iris-versicolor', 'Iris-setosa', 'Iris-virginica']
print(sklearn.metrics.confusion_matrix(y_test, y_pred, labels = lbs))
```

```
[[10  0  3]
 [ 1 10  0]
 [ 1  0  5]]
```

```
In [18]: from sklearn.metrics import classification_report
```

```
In [19]: print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
Iris-setosa	1.00	0.91	0.95	11
Iris-versicolor	0.83	0.77	0.80	13
Iris-virginica	0.62	0.83	0.71	6
accuracy			0.83	30
macro avg	0.82	0.84	0.82	30
weighted avg	0.85	0.83	0.84	30

```
In [20]: from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(x_train, y_train)
```

Out[20]:

▼ GaussianNB ⓘ ?

GaussianNB()

```
In [21]: classifier.score(x_test, y_test)
```

Out[21]: 0.8333333333333334

```
In [22]: y_pred = gnb.predict(x_test)
```

```
In [23]: y_test
```

Out[23]: 114 Iris-virginica
62 Iris-versicolor
33 Iris-setosa
107 Iris-virginica
7 Iris-setosa
100 Iris-virginica
40 Iris-setosa
86 Iris-versicolor
76 Iris-versicolor
71 Iris-versicolor
134 Iris-virginica
51 Iris-versicolor
73 Iris-versicolor
54 Iris-versicolor
63 Iris-versicolor
37 Iris-setosa
78 Iris-versicolor
90 Iris-versicolor
45 Iris-setosa
16 Iris-setosa
121 Iris-virginica
66 Iris-versicolor
24 Iris-setosa
8 Iris-setosa
126 Iris-virginica
22 Iris-setosa
44 Iris-setosa
97 Iris-versicolor
93 Iris-versicolor
26 Iris-setosa
Name: Species, dtype: object