In [1]:

```
import pandas as pd
from sklearn.metrics import confusion_matrix, classification_report, accurac
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
```

In [2]:

```
df = pd.read_csv('Iris.csv')
df
```

Out[2]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Spe
0	1	5.1	3.5	1.4	0.2	se
1	2	4.9	3.0	1.4	0.2	se
2	3	4.7	3.2	1.3	0.2	se
3	4	4.6	3.1	1.5	0.2	se
4	5	5.0	3.6	1.4	0.2	se
...	...	...	...	...	...	
145	146	6.7	3.0	5.2	2.3	virg
146	147	6.3	2.5	5.0	1.9	virg
147	148	6.5	3.0	5.2	2.0	virg
148	149	6.2	3.4	5.4	2.3	virg
149	150	5.9	3.0	5.1	1.8	virg

150 rows × 6 columns

In [3]:

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

Out[3]:

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

In [4]:

```
label_encoder = LabelEncoder()
df['Species'] = label_encoder.fit_transform(df['Species'])
df
```

Out[4]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Spe
0	1	5.1	3.5	1.4	0.2	
1	2	4.9	3.0	1.4	0.2	
2	3	4.7	3.2	1.3	0.2	
3	4	4.6	3.1	1.5	0.2	
4	5	5.0	3.6	1.4	0.2	
...	...	...	...	...	...	
145	146	6.7	3.0	5.2	2.3	
146	147	6.3	2.5	5.0	1.9	
147	148	6.5	3.0	5.2	2.0	
148	149	6.2	3.4	5.4	2.3	
149	150	5.9	3.0	5.1	1.8	

150 rows × 6 columns

In [5]:

```
x = df.drop('Species', axis=1)
x
```

Out[5]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	1	5.1	3.5	1.4	0.2
1	2	4.9	3.0	1.4	0.2
2	3	4.7	3.2	1.3	0.2
3	4	4.6	3.1	1.5	0.2
4	5	5.0	3.6	1.4	0.2
...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3
146	147	6.3	2.5	5.0	1.9
147	148	6.5	3.0	5.2	2.0
148	149	6.2	3.4	5.4	2.3
149	150	5.9	3.0	5.1	1.8

150 rows × 5 columns

In [6]:

```
y = df.Species
y
```

```
Out[6]: 0      0
        1      0
        2      0
        3      0
        4      0
              ..
        145    2
        146    2
        147    2
        148    2
        149    2
        Name: Species, Length: 150, dtype: int32
```

```
In [11]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, rar
         gaussian = GaussianNB()
         gaussian.fit(x_train, y_train)
```

```
Out[11]: ▾  GaussianNB
         GaussianNB()
```

```
In [12]: y_pred = gaussian.predict(x_test)
```

```
In [13]: matrix = confusion_matrix(y_test, y_pred)
         matrix
```

```
Out[13]: array([[11,  0,  0],
                [ 0, 13,  0],
                [ 0,  0,  6]], dtype=int64)
```

```
In [14]: print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        11
           1       1.00      1.00      1.00        13
           2       1.00      1.00      1.00         6

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

```
In [15]: accuracy = accuracy_score(y_test, y_pred)
         accuracy
```

```
Out[15]: 1.0
```

```
In [ ]:
```

This notebook was converted with convert.ploomber.io