In [7]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

In [8]:
```
df = pd.read_csv('iris.csv')
print(df)
```

```
     sepal length in cm  sepal width in cm  petal length in cm  \
0                   5.1                3.5                 1.4
1                   4.9                3.0                 1.4
2                   4.7                3.2                 1.3
3                   4.6                3.1                 1.5
4                   5.0                3.6                 1.4
..                  ...                ...                 ...
145                 6.7                3.0                 5.2
146                 6.3                2.5                 5.0
147                 6.5                3.0                 5.2
148                 6.2                3.4                 5.4
149                 5.9                3.0                 5.1

     petal width in cm         species
0                  0.2     Iris-setosa
1                  0.2     Iris-setosa
2                  0.2     Iris-setosa
3                  0.2     Iris-setosa
4                  0.2     Iris-setosa
..                 ...             ...
145                2.3  Iris-virginica
146                1.9  Iris-virginica
147                2.0  Iris-virginica
148                2.3  Iris-virginica
149                1.8  Iris-virginica

[150 rows x 5 columns]
```

In [9]:
```
df.describe()
```

Out[9]:

	sepal length in cm	sepal width in cm	petal length in cm	petal width in cm
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
sepal length in cm    0
sepal width in cm     0
petal length in cm    0
petal width in cm     0
species               0
dtype: int64
```

In [11]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

In [12]:
```
df['species']=le.fit_transform(df['species'])
```

In [13]:
```
df.head(7)
```

Out[13]:

	sepal length in cm	sepal width in cm	petal length in cm	petal width in cm	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
5	5.4	3.9	1.7	0.4	0
6	4.6	3.4	1.4	0.3	0

In [14]:
```
x=df.drop(labels='species',axis=1)
y=df['species']
```

In [15]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=0.3,random_state=42)
```

In [16]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
```

In [17]:
```
xtr_scale = sc.fit_transform(x_train)
xte_scale = sc.transform(x_test)
```

In [18]:
```
xtr_scale
```

Out[18]:
```
array([[-0.4134164 , -1.46440146, -0.10014569, -0.32149987],
       [ 0.55122187, -0.49582097,  0.71771076,  0.35364985],
       [ 0.67180165,  0.2306144 ,  0.95138404,  0.75873969],
       [ 0.91296121, -0.01153072,  0.30878254,  0.21861991],
       [ 1.63643991,  1.44134002,  1.30189395,  1.7039493 ],
       [-0.17225683, -0.25367584,  0.1919459 ,  0.08358997],
       [ 2.11875905, -0.01153072,  1.59398554,  1.16382952],
       [-0.29283662, -0.01153072,  0.36720086,  0.35364985],
       [-0.89573553,  1.19919489, -1.443767  , -1.40173942],
       [ 2.23933883, -0.49582097,  1.65240385,  1.02879957],
       [-0.05167705, -0.73796609,  0.13352758, -0.32149987],
       [-0.77515575,  0.95704977, -1.443767  , -1.40173942],
       [-1.01631531,  1.19919489, -1.50218532, -1.26670948],
       [-0.89573553,  1.92563026, -1.15167541, -1.13167953],
       [-1.01631531, -2.43298195, -0.21698232, -0.32149987],
       [ 0.55122187, -0.73796609,  0.60087413,  0.75873969],
       [-1.25747488,  0.95704977, -1.15167541, -1.40173942],
       [-1.01631531, -0.01153072, -1.32693037, -1.40173942],
       [-0.89573553,  0.71490465, -1.26851205, -0.99664959],
       [-0.29283662, -0.73796609,  0.1919459 ,  0.08358997],
       [-0.89573553,  0.95704977, -1.38534869, -1.40173942],
       [-0.17225683, -0.01153072,  0.1919459 , -0.05143998],
       [ 2.23933883,  1.92563026,  1.65240385,  1.29885946],
       [-1.49863445,  0.47275953, -1.443767  , -1.40173942],
       [ 0.43064208, -0.25367584,  0.25036422,  0.08358997],
       [-0.17225683, -1.22225633,  0.65929245,  1.02879957],
       [-0.4134164 ,  2.89421075, -1.443767  , -1.40173942],
       [ 0.18948252, -0.01153072,  0.54245581,  0.75873969],
       [-0.05167705, -0.73796609,  0.71771076,  0.89376963],
       [ 0.18948252, -1.9486917 ,  0.07510927, -0.32149987],
       [-0.53399618, -0.01153072,  0.36720086,  0.35364985],
       [ 0.43064208,  0.95704977,  0.89296572,  1.43388941],
       [-0.4134164 , -1.70654658,  0.07510927,  0.08358997],
       [-0.53399618,  2.16777538, -1.26851205, -1.13167953],
       [-1.01631531, -1.70654658, -0.33381896, -0.32149987],
       [ 0.67180165, -0.73796609,  0.8345474 ,  0.89376963],
       [-1.01631531,  0.71490465, -1.443767  , -1.40173942],
       [-1.01631531,  0.47275953, -1.56060364, -1.40173942],
       [-0.4134164 , -1.46440146, -0.04172737, -0.18646992],
       [ 1.033541  , -0.01153072,  0.65929245,  0.62370974],
       [-1.1368951 ,  0.2306144 , -1.38534869, -1.53676936],
```

```
[-0.05167705, -0.49582097,  0.71771076,  1.56891935],
[-1.01631531,  0.95704977, -1.38534869, -1.40173942],
[-1.01631531,  1.19919489, -1.32693037, -0.86161964],
[ 0.06890273,  0.47275953,  0.54245581,  0.75873969],
[-0.89573553, -1.22225633, -0.50907391, -0.18646992],
[ 1.27470056,  0.47275953,  1.06822067,  1.43388941],
[ 0.18948252, -0.73796609,  0.71771076,  0.4886798 ],
[ 0.3100623 , -0.98011121,  1.00980236,  0.21861991],
[ 2.23933883, -0.01153072,  1.30189395,  1.43388941],
[-0.4134164 , -1.22225633,  0.07510927,  0.08358997],
[-1.73979401, -0.25367584, -1.443767  , -1.40173942],
[-1.8603738 , -0.01153072, -1.61902196, -1.53676936],
[ 0.18948252, -1.9486917 ,  0.65929245,  0.35364985],
[ 1.63643991,  0.47275953,  1.24347563,  0.75873969],
[-1.49863445,  0.2306144 , -1.38534869, -1.40173942],
[-0.89573553,  1.19919489, -1.443767  , -1.26670948],
[-1.73979401, -0.01153072, -1.50218532, -1.40173942],
[ 0.55122187, -1.22225633,  0.60087413,  0.35364985],
[ 0.55122187,  0.95704977,  1.00980236,  1.56891935],
[-1.49863445,  0.95704977, -1.443767  , -1.26670948],
[ 1.15412078, -0.01153072,  0.95138404,  1.16382952],
[ 0.55122187,  0.71490465,  1.24347563,  1.7039493 ],
[-1.37805466,  0.47275953, -1.50218532, -1.40173942],
[ 0.3100623 , -0.25367584,  0.48403749,  0.21861991],
[ 0.79238143, -0.49582097,  0.42561917,  0.35364985],
[ 0.43064208, -0.49582097,  0.54245581,  0.75873969],
[ 1.39528035,  0.47275953,  0.48403749,  0.21861991],
[ 0.67180165,  0.47275953,  0.8345474 ,  1.43388941],
[-0.89573553,  1.92563026, -1.32693037, -1.40173942],
[ 1.27470056,  0.2306144 ,  0.89296572,  1.16382952],
[ 0.06890273, -0.01153072,  0.1919459 ,  0.35364985],
[ 0.79238143, -0.01153072,  0.77612908,  1.02879957],
[-0.17225683, -0.98011121, -0.21698232, -0.32149987],
[-0.77515575, -0.73796609,  0.01669095,  0.21861991],
[ 0.3100623 , -0.01153072,  0.42561917,  0.21861991],
[-1.61921423, -1.70654658, -1.50218532, -1.26670948],
[ 0.91296121, -0.25367584,  0.42561917,  0.08358997],
[-0.4134164 , -0.98011121,  0.30878254, -0.05143998],
[-0.65457597,  1.68348514, -1.38534869, -1.40173942],
[-0.29283662, -0.01153072,  0.13352758,  0.08358997],
[ 1.7570197 , -0.25367584,  1.41873058,  0.75873969],
[ 1.033541  ,  0.71490465,  1.06822067,  1.16382952],
[-0.89573553,  1.68348514, -1.38534869, -1.13167953],
[-1.1368951 , -1.46440146, -0.33381896, -0.32149987],
```

```
[ 1.033541  ,  0.71490465,  1.06822067,  1.7039493 ],
[ 1.63643991, -0.01153072,  1.12663899,  0.4886798 ],
[-1.1368951 ,  0.2306144 , -1.38534869, -1.53676936],
[ 1.033541  ,  0.2306144 ,  1.00980236,  1.56891935],
[-1.1368951 , -0.01153072, -1.443767  , -1.40173942],
[ 1.27470056,  0.2306144 ,  0.60087413,  0.35364985],
[ 1.87759948, -0.49582097,  1.30189395,  0.89376963],
[ 0.55122187, -0.25367584,  1.00980236,  0.75873969],
[-0.17225683, -0.49582097,  0.13352758,  0.08358997],
[ 0.79238143, -0.01153072,  0.95138404,  0.75873969],
[ 0.55122187, -1.70654658,  0.30878254,  0.08358997],
[ 0.67180165, -0.25367584,  0.25036422,  0.08358997],
[-0.29283662, -0.49582097,  0.60087413,  1.02879957],
[ 0.06890273, -0.01153072,  0.71771076,  0.75873969],
[-0.53399618,  0.95704977, -1.26851205, -1.40173942],
[ 0.3100623 , -0.49582097,  0.07510927,  0.08358997],
[-1.1368951 , -1.22225633,  0.36720086,  0.62370974],
[-0.05167705,  2.40992051, -1.56060364, -1.40173942],
[-0.05167705, -0.98011121,  0.07510927, -0.05143998],
[ 1.51586013, -0.01153072,  1.18505731,  1.16382952]])
```

In [19]:
```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
```

In [20]:
```
gnb.fit(xtr_scale,y_train)
```

Out[20]:
```
GaussianNB()
```

In [21]:
```
y_pred = gnb.predict(xte_scale)
y_pred
```

Out[21]:
```
array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 2, 2, 1, 1, 2, 0, 2,
       0, 2, 2, 2, 2, 2, 0, 0, 0, 0, 1, 0, 0, 2, 1, 0, 0, 0, 2, 1, 1, 0,
       0])
```

In [22]:
```
pred_df=pd.DataFrame(np.c_[ y_test, y_pred], columns = ["original","predicted"])
pred_df
```

Out[22]:

	original	predicted
0	1	1
1	0	0
2	2	2
3	1	1
4	1	1
5	0	0
6	1	1
7	2	2
8	1	1
9	1	1
10	2	2
11	0	0
12	0	0
13	0	0
14	0	0
15	1	2
16	2	2
17	1	1
18	1	1
19	2	2
20	0	0
21	2	2
22	0	0
23	2	2

	original	predicted
24	2	2
25	2	2
26	2	2
27	2	2
28	0	0
29	0	0
30	0	0
31	0	0
32	1	1
33	0	0
34	0	0
35	2	2
36	1	1
37	0	0
38	0	0
39	0	0
40	2	2
41	1	1
42	1	1
43	0	0
44	0	0

In [23]:
```
from sklearn.metrics import accuracy_score
gnb.score(xte_scale,y_test)
```

Out[23]: 0.9777777777777777

In [24]:
```
from sklearn.metrics import confusion_matrix
```

In [25]:
```
confusion_matrix(y_test,y_pred)
```

Out[25]:
```
array([[19,  0,  0],
       [ 0, 12,  1],
       [ 0,  0, 13]], dtype=int64)
```

In [26]:
```
from sklearn.metrics import classification_report as cr
print(cr(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       1.00      0.92      0.96        13
           2       0.93      1.00      0.96        13

    accuracy                           0.98        45
   macro avg       0.98      0.97      0.97        45
weighted avg       0.98      0.98      0.98        45
```