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

In [2]:
```
df = pd.read_csv('iris-data.csv')
```

In [3]:
```
df
```

Out[3]:

	sepal length	sepal width	petal length	petal width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [6]:
```
x = df.loc[:,['sepal length','sepal width','petal length','petal width']].values
y = df['class'].values
```

In [7]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size = 0.25)
```

In [10]:
```
x_train[0:4]
```

Out[10]:
```
array([[6.4, 3.2, 5.3, 2.3],
       [6. , 2.2, 4. , 1. ],
       [4.8, 3.1, 1.6, 0.2],
       [5.5, 2.4, 3.8, 1.1]])
```

In [11]:
```
x_test[0:4]
```

Out[11]:
```
array([[5.8, 2.7, 3.9, 1.2],
       [5.1, 3.4, 1.5, 0.2],
       [6.5, 3. , 5.2, 2. ],
       [4.8, 3. , 1.4, 0.3]])
```

In [12]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
```

```
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

In [13]:
```
x_train[0:4]
```

Out[13]:
```
array([[ 0.65857567,  0.32139187,  0.84067336,  1.43264996],
       [ 0.15850465, -1.94250055,  0.10063684, -0.2984192 ],
       [-1.34170839,  0.09500263, -1.26558444, -1.36369254],
       [-0.46658412, -1.48972207, -0.01321494, -0.16526004]])
```

In [14]:
```
x_test[0:4]
```

Out[14]:
```
array([[-0.09153086, -0.81055434,  0.04371095, -0.03210087],
       [-0.96665513,  0.77417035, -1.32251033, -1.36369254],
       [ 0.78359342, -0.13138661,  0.78374747,  1.03317246],
       [-1.34170839, -0.13138661, -1.37943621, -1.23053337]])
```

In [15]:
```
from sklearn.naive_bayes import GaussianNB
reg = GaussianNB()
reg.fit(x_train,y_train)
y_pred = reg.predict(x_test)
```

In [16]:
```
y_pred
```

Out[16]:
```
array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-virginica', 'Iris-setosa', 'Iris-virginica',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor'], dtype='<U15')
```

In [17]:
```
y_test
```

Out[17]:
```
array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-virginica', 'Iris-versicolor',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-virginica', 'Iris-setosa', 'Iris-virginica',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor'], dtype=object)
```

In [18]:
```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,y_pred))
```

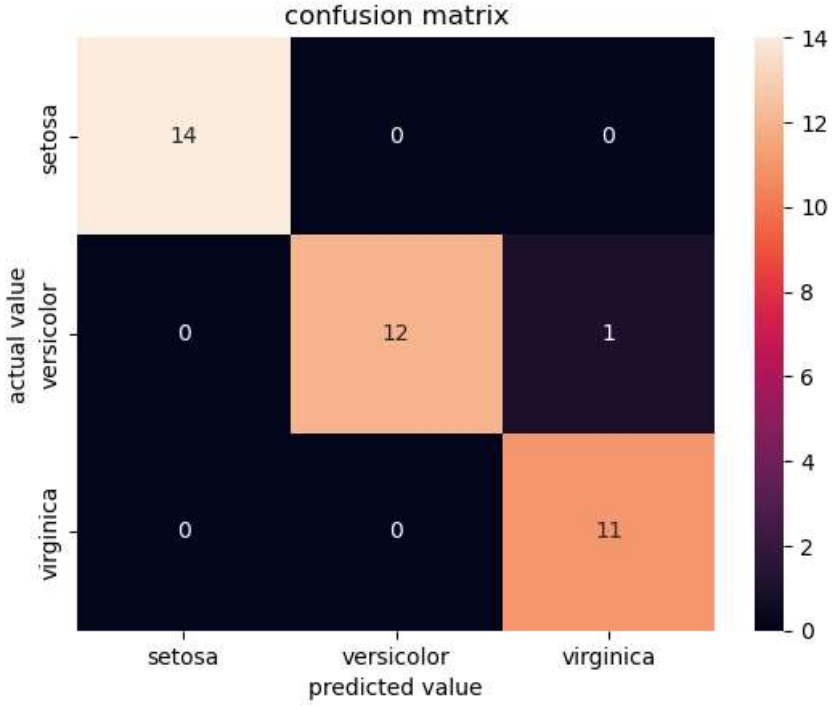
```
0.9736842105263158
```

In [41]:
```
import seaborn as sns
from sklearn.metrics import confusion_matrix
cm1 = confusion_matrix(y_test,y_pred)
cm1
```

Out[41]:
```
array([[14,  0,  0],
       [ 0, 12,  1],
       [ 0,  0, 11]], dtype=int64)
```

In [33]:
```
cm = pd.DataFrame(cm,
                  index = ['setosa','versicolor','virginica'],
                  columns = ['setosa','versicolor','virginica'])
```

In [34]:
```
from matplotlib import pyplot as plt
sns.heatmap(cm, annot = True)
plt.title('confusion matrix')
plt.xlabel('predicted value')
plt.ylabel('actual value')
plt.show()
```



In [42]:
```
# VIRGINICA
TP = cm1[2][2]
FN = cm1[2][0] + cm1[2][1]
FP = cm1[0][2] + cm1[1][2]
TN = cm1[0][0] + cm1[0][1] + cm1[1][0] + cm1[1][1]
```

In [43]:
```
TP
```

Out[43]:
```
11
```

In [44]:
```
# VIRSICOLOR
TP = cm1[1][1]
FN = cm1[1][0] + cm1[1][2]
FP = cm1[0][1] + cm1[2][1]
TN = cm1[0][0] + cm1[0][2] + cm1[2][0] + cm1[2][2]
```

In [48]:
```
print("tp:{},fn:{},fp:{},tn:{}".format(TP,FN,FP,TN))
```

```
tp:12,fn:1,fp:0,tn:25
```

In [56]:
```
accuracy = (TP+TN)/(TP+TN+FP+FN)
accuracy
```

Out[56]:
```
1.0
```

In [54]:
```
precision = TP/(TP+FP)
precision
```

Out[54]:
```
0.9230769230769231
```

In [55]:
```
recall = TP/(TP+FN)
recall
```

Out[55]:
```
0.9230769230769231
```

In [ ]: