In [2]:
```
import pandas as pd
import numpy as np
df=pd.read_csv('IRIS.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	sepal_length	sepal_width	petal_length	petal_width	species
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

In [4]:
```
df.tail()
```

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width	species
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

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   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 [6]:
```
df.describe()
```

Out[6]:

	sepal_length	sepal_width	petal_length	petal_width
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 [7]:
```
df.isnull().sum()
```

Out[7]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [8]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [9]:
```
df['species']=df['species'].astype('category')
df['species']=df['species'].cat.codes
```

In [10]:
```
df.head()
```

Out[10]:

	sepal_length	sepal_width	petal_length	petal_width	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

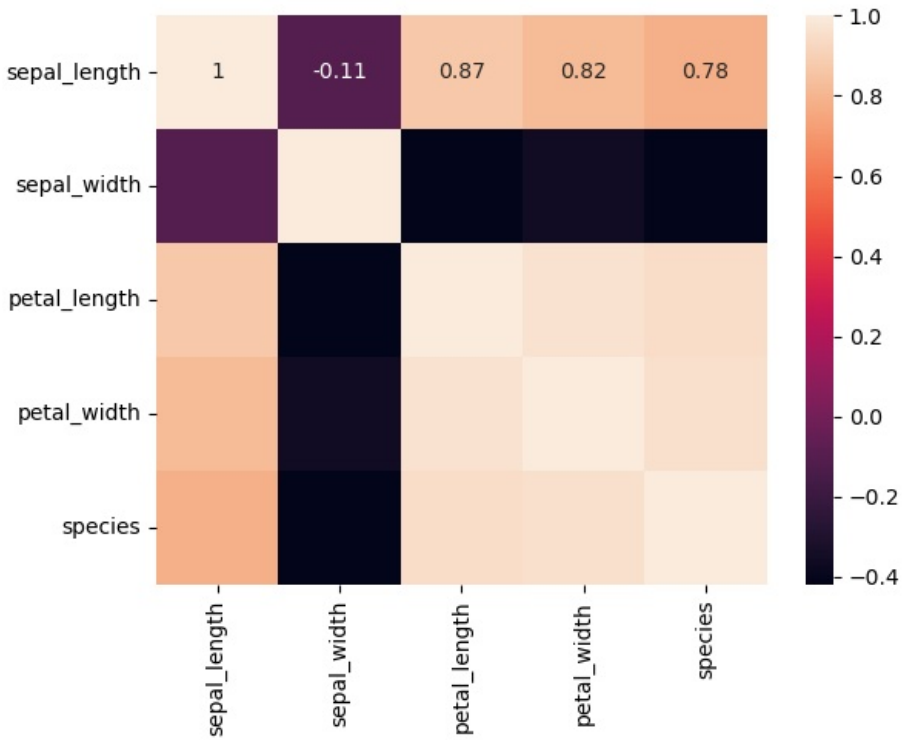
In [11]:
```
df.tail()
```

Out[11]:

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

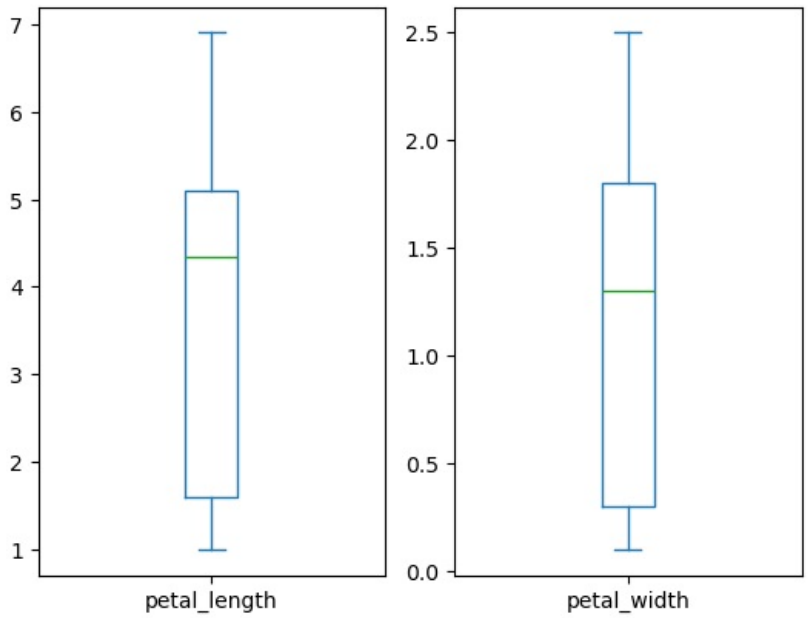
In [12]:
```
correaltion_matrix=df.corr()
sns.heatmap(data=correaltion_matrix,annot=True)
```

Out[12]:
```
<Axes: >
```



In [13]:
```
plt.subplot(1,2,1)
df['petal_length'].plot.box()
plt.subplot(1,2,2)
df['petal_width'].plot.box()
```

Out[13]:
```
<Axes: >
```



In [14]:
```
x=df.loc[:,['petal_length','petal_width']]
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=0.2)
```

In [16]:
```
from sklearn.naive_bayes import GaussianNB
```

In [17]:
```
model=GaussianNB()
model.fit(x_train,y_train)
```

Out[17]:
```
▾ GaussianNB
GaussianNB()
```

In [18]:
```
y_pred=model.predict(x_test)
```

In [19]:
```
y_test.head()
```

Out[19]:
```
134    2
8      0
17     0
106    2
116    2
Name: species, dtype: int8
```

In [20]:
```
y_pred
```

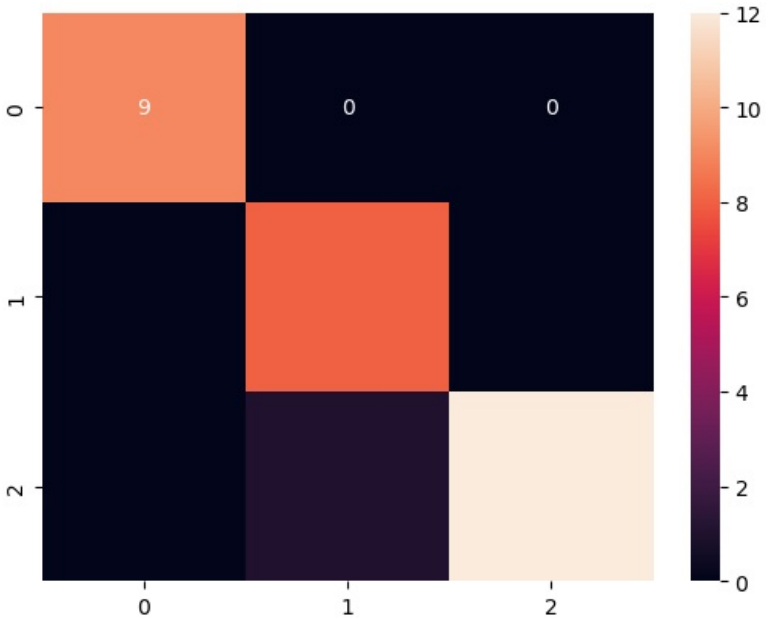
Out[20]:
```
array([2, 0, 0, 1, 2, 2, 1, 2, 2, 1, 2, 2, 0, 1, 1, 1, 0, 2, 2, 0, 0, 2,
       1, 2, 0, 0, 2, 1, 0, 1], dtype=int8)
```

In [ ]:
```
p
```

In [22]:
```
from sklearn.metrics import confusion_matrix
confu_matrix=confusion_matrix(y_test,y_pred)
print(confu_matrix)
```

```
[[ 9  0  0]
 [ 0  8  0]
 [ 0  1 12]]
```

In [25]:
```
dataplot=sns.heatmap(data=confu_matrix,annot=True)
plt.show()
```



In [41]:
```
accuracy=(9+8+12)/(9+8+12+1)
print('Accuracy based on the confusion matrix is : ',(accuracy*100),' %')
```

```
Accuracy based on the confusion matrix is :  96.66666666666667  %
```

In [38]:
```
from sklearn.metrics import mean_squared_error
mse=mean_squared_error(y_test,y_pred)
print("Mean square error is : ",(mse*100).round(2), ' %')
```

```
Mean square error is :  3.33  %
```

In [39]:
```
rmse=np.sqrt(mse)
print("Root mean square error is : ",(rmse*100).round(2),' %')
```

```
Root mean square error is :  18.26  %
```

In [43]:
```
r2_score=model.score(x_test,y_test)
print("R2 score is : ",r2_score)
```

```
R2 score is :  0.9666666666666667
```