# IRIS implementation

In [1]:
```
import numpy as np
import pandas as pd        #importing the header files required for our project
import matplotlib.pyplot as plt
```

In [3]:
```
data=pd.read_csv("G:/csvfiles/iris.csv")  #loadind the data by using the pandas library
```

In [ ]:
```
#understanding the dataa
```

In [4]:
```
data.shape
```

Out[4]:
```
(150, 5)
```

In [5]:
```
data.head(5)
```

Out[5]:

| | sepal.length | sepal.width | petal.length | petal.width | variety |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | Setosa |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | Setosa |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | Setosa |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | Setosa |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | Setosa |

In [6]:
```
data.isnull().sum()
```

Out[6]:
```
sepal.length    0
sepal.width     0
petal.length    0
petal.width     0
variety         0
dtype: int64
```

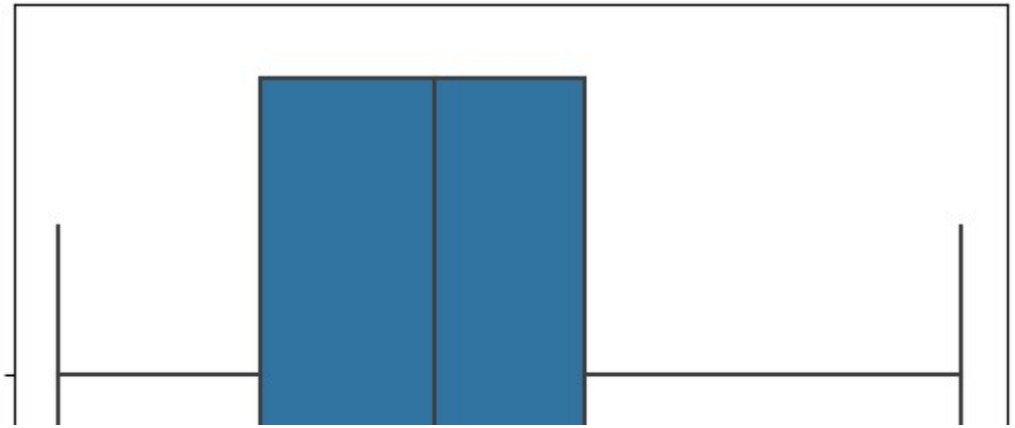
In [7]:
```
import seaborn as sns
```

In [ ]:
```
#data preprocessing
```

In [45]:
```
sns.boxplot(data['sepal.length'])
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a ke
yword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without
an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

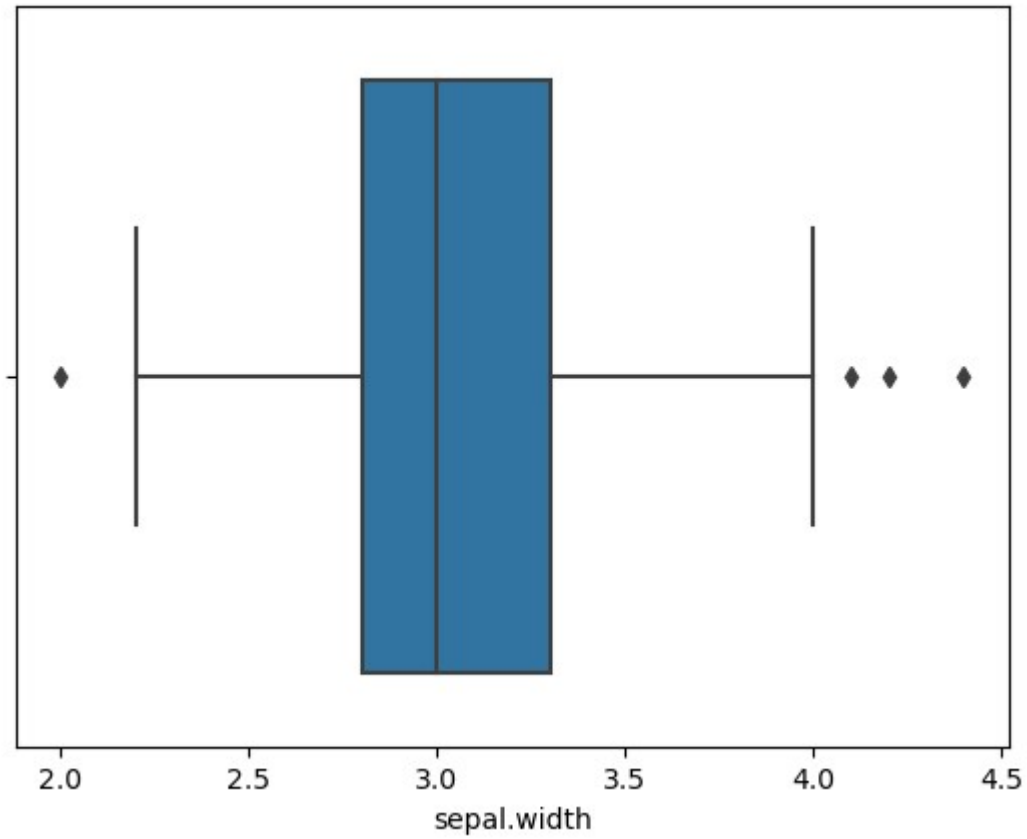
Out[45]:
```
<AxesSubplot:xlabel='sepal.length'>
```



In [9]:
```
sns.boxplot(data['sepal.width'])
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a ke
yword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without
an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[9]: <AxesSubplot:xlabel='sepal.width'>




In [10]:
```
q1=data['sepal.width'].quantile(0.25)
```

In [11]:
```
q3=data['sepal.width'].quantile(0.75)
```

In [12]:
```
iqr=q3-q1
```

In [17]:
```
upper_limit=iqr*1.5+q3
lower_limit=q1-1.5*iqr
```

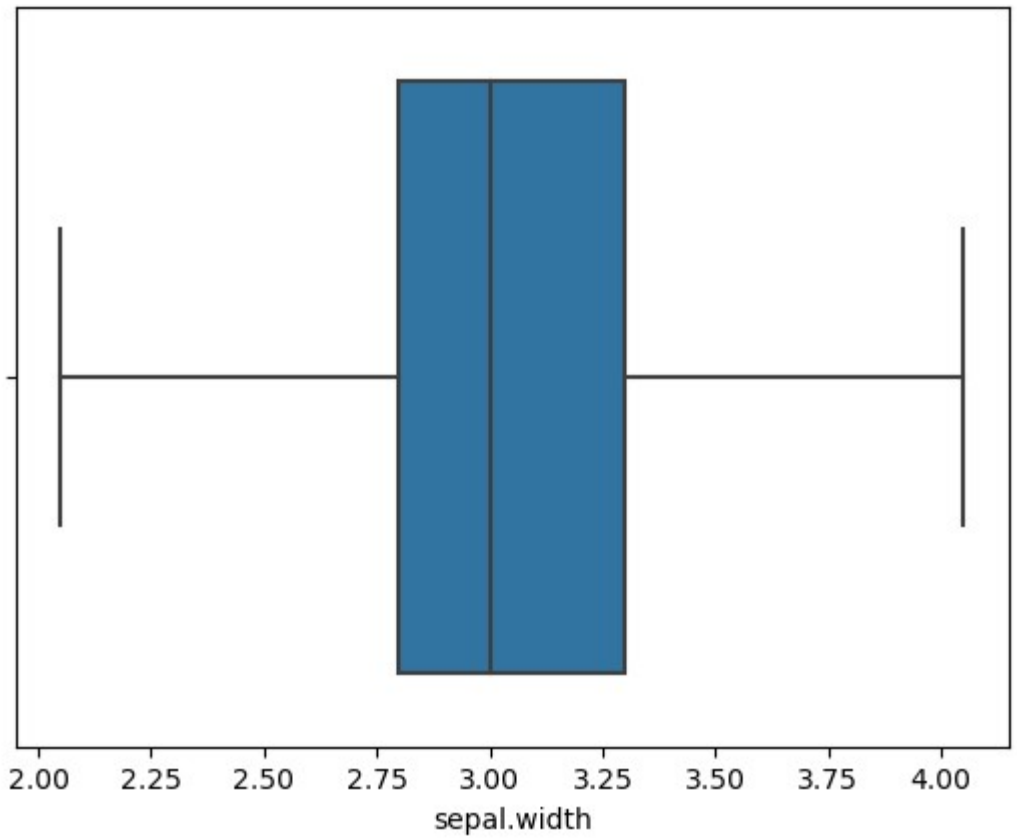
In [18]:
```
def Imputation(value):
    if value > upper_limit:
        return upper_limit
    elif value < lower_limit:
        return lower_limit
    else:
        return value
```

In [19]:
```
data['sepal.width']=data['sepal.width'].apply(Imputation)
```

In [20]:
```
sns.boxplot(data['sepal.width'])
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a ke
yword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without
an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

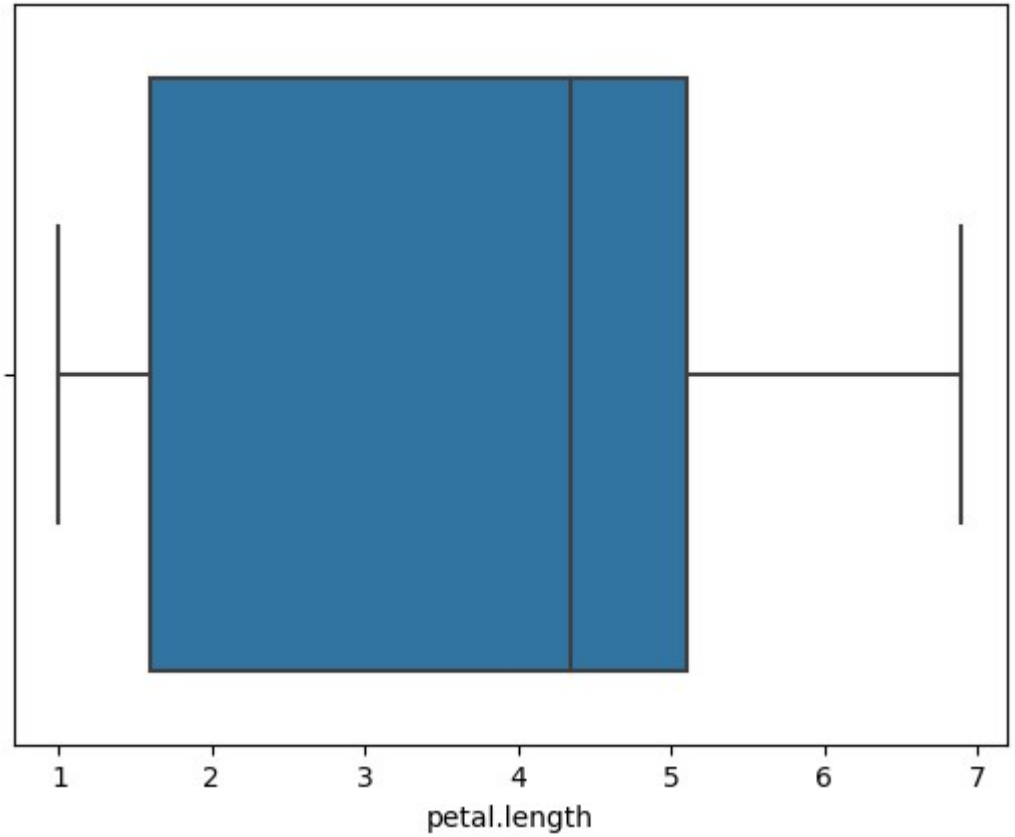
Out[20]:
```
<AxesSubplot:xlabel='sepal.width'>
```



In [21]:
```
sns.boxplot(data['petal.length'])
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a ke
yword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without
an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

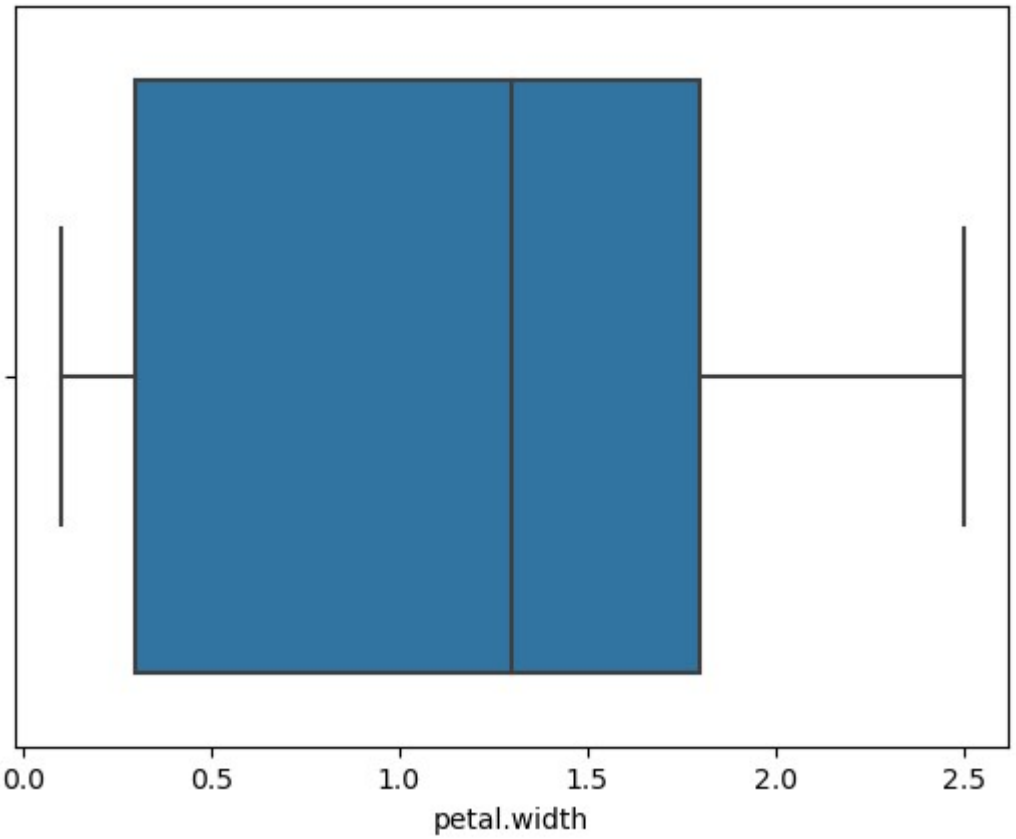
Out[21]: <AxesSubplot:xlabel='petal.length'>



In [22]:
```
sns.boxplot(data['petal.width'])
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a ke
yword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without
an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[22]:
```
<AxesSubplot:xlabel='petal.width'>
```




In [23]:
```
data['petal.width'].unique()
```

Out[23]:
```
array([0.2, 0.4, 0.3, 0.1, 0.5, 0.6, 1.4, 1.5, 1.3, 1.6, 1. , 1.1, 1.8,
       1.2, 1.7, 2.5, 1.9, 2.1, 2.2, 2. , 2.4, 2.3])
```

In [25]:
```
data['petal.length'].unique()
```

Out[25]:
```
array([1.4, 1.3, 1.5, 1.7, 1.6, 1.1, 1.2, 1. , 1.9, 4.7, 4.5, 4.9, 4. ,
       4.6, 3.3, 3.9, 3.5, 4.2, 3.6, 4.4, 4.1, 4.8, 4.3, 5. , 3.8, 3.7,
       5.1, 3. , 6. , 5.9, 5.6, 5.8, 6.6, 6.3, 6.1, 5.3, 5.5, 6.7, 6.9,
       5.7, 6.4, 5.4, 5.2])
```

In [26]:
```
data['sepal.length'].unique()
```

Out[26]:
```
array([5.1, 4.9, 4.7, 4.6, 5. , 5.4, 4.4, 4.8, 4.3, 5.8, 5.7, 5.2, 5.5,
       4.5, 5.3, 7. , 6.4, 6.9, 6.5, 6.3, 6.6, 5.9, 6. , 6.1, 5.6, 6.7,
       6.2, 6.8, 7.1, 7.6, 7.3, 7.2, 7.7, 7.4, 7.9])
```

In [27]:
```
data['sepal.width'].unique()
```

Out[27]:
```
array([3.5 , 3.  , 3.2 , 3.1 , 3.6 , 3.9 , 3.4 , 2.9 , 3.7 , 4.  , 4.05,
       3.8 , 3.3 , 2.3 , 2.8 , 2.4 , 2.7 , 2.05, 2.2 , 2.5 , 2.6 ])
```

In [28]:
```
data.shape
```

Out[28]:
```
(150, 5)
```

In [29]:
```
from sklearn.tree import DecisionTreeClassifier  #importing the model
```

In [30]:
```
tree=DecisionTreeClassifier()
```

In [32]:
```
from sklearn.model_selection import train_test_split
```

In [35]:
```
x=data.iloc[:,0:4]
```

In [36]:
```
x.head(5)
```

Out[36]:

| | sepal.length | sepal.width | petal.length | petal.width |
|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 |

In [37]:
```
y=data['variety']
```

In [38]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=42,test_size=0.2) #training the model
```

In [39]:
```
tree.fit(x_train,y_train)
```

Out[39]:
```
DecisionTreeClassifier()
```

In [40]:
```
y_pred=tree.predict(x_test)   #predicting the labels by using model
```

In [41]:
```
from sklearn.metrics import accuracy_score
```

In [42]:
```
accuracy=(accuracy_score(y_test,y_pred)*100)  #finding the accuracy of our model
```

In [43]:
```
print(accuracy)
```

```
100.0
```

In [44]:
```
print(y_test,y_pred)
```

```
73     Versicolor
18         Setosa
118     Virginica
78     Versicolor
76     Versicolor
31         Setosa
64     Versicolor
141     Virginica
68     Versicolor
82     Versicolor
110     Virginica
12         Setosa
36         Setosa
9          Setosa
19         Setosa
56     Versicolor
104     Virginica
69     Versicolor
55     Versicolor
132     Virginica
29         Setosa
127     Virginica
26         Setosa
128     Virginica
131     Virginica
145     Virginica
108     Virginica
143     Virginica
45         Setosa
30         Setosa
Name: variety, dtype: object ['Versicolor' 'Setosa' 'Virginica' 'Versicolor' 'Versicolor' 'Setosa'
 'Versicolor' 'Virginica' 'Versicolor' 'Versicolor' 'Virginica' 'Setosa'
 'Setosa' 'Setosa' 'Setosa' 'Versicolor' 'Virginica' 'Versicolor'
 'Versicolor' 'Virginica' 'Setosa' 'Virginica' 'Setosa' 'Virginica'
 'Virginica' 'Virginica' 'Virginica' 'Virginica' 'Setosa' 'Setosa']
```

In [ ]: