FINDING OUTLIERS AND WORKING ON THEM :

Outliers are the extreme values that are outside the range of expected values of data and are unlike other data points. Presence of outliers in the data could affect the accuracy of the classification model. There are a number of ways in which the outliers can be detected and handled.

DETECTING OUTLIERS IF ANY:

Importing the Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Loading the dataset obtained by data cleaning:

In [2]:
```
data = pd.read_csv('iris_dataset_cleaned.csv')
```

Displaying properties of the cleaned dataset for further processing

In [3]:
```
data.info()
data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 93 entries, 0 to 92
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  93 non-null     float64
 1   sepal_width   93 non-null     float64
 2   petal_length  93 non-null     float64
 3   petal_width   93 non-null     float64
 4   species       93 non-null     object 
dtypes: float64(4), object(1)
memory usage: 3.8+ KB
```

Out[3]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.045070	2.508203	3.018024	1.164924	Iris-versicolor
1	6.325517	2.115481	4.542052	1.413651	Iris-versicolor
2	5.257497	3.814303	1.470660	0.395348	Iris-setosa
3	6.675168	3.201700	5.785461	2.362764	Iris-virginica
4	5.595237	2.678166	4.077750	1.369266	Iris-versicolor

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

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width	species
88	4.874848	3.217348	1.592887	0.123588	Iris-setosa
89	5.564197	2.771731	3.483588	1.074754	Iris-versicolor
90	5.548047	4.249211	1.453466	0.214527	Iris-setosa
91	5.510482	2.652867	4.276817	1.298032	Iris-versicolor
92	4.538713	3.056142	1.545136	0.241424	Iris-setosa

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

Out[5]:

	sepal_length	sepal_width	petal_length	petal_width
count	93.000000	93.000000	93.000000	93.000000
mean	5.867894	3.054935	3.808118	1.236858
std	0.892271	0.439463	1.811399	0.770872
min	4.344007	1.946010	1.033031	0.020731
25%	5.152435	2.794790	1.541564	0.343669
50%	5.636744	3.049459	4.192791	1.369266
75%	6.478961	3.239682	5.098860	1.837925
max	7.795561	4.249211	6.768611	2.603123

In [6]:
```
data.shape
```

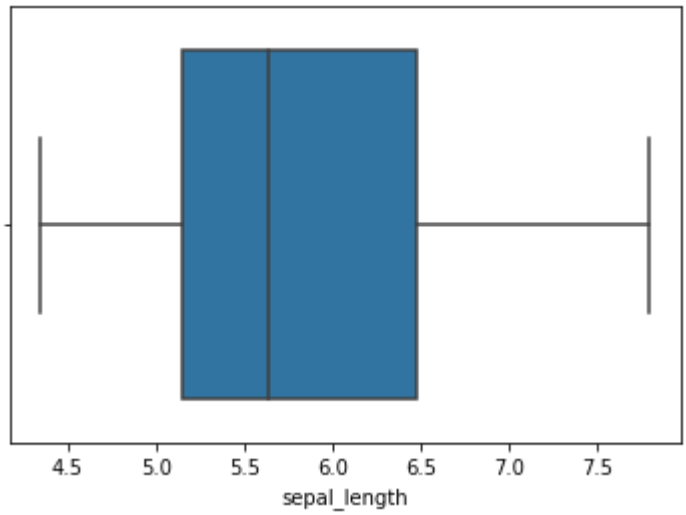
Out[6]:
```
(93, 5)
```

FINDING IF OUTLIERS ARE PRESENT IN THE DATA OF THE FEATURES WITH THE HELP OF BOXPLOT:

sepal_length:

In [7]:
```
sns.boxplot(x=data["sepal_length"])
```

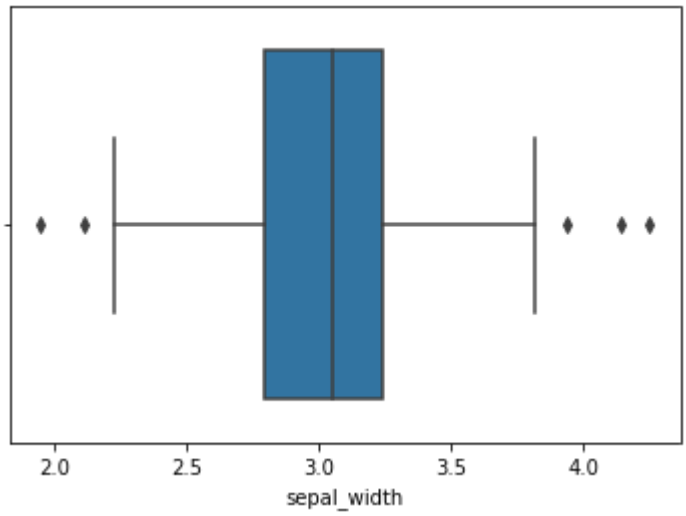
Out[7]:
```
<AxesSubplot:xlabel='sepal_length'>
```



sepal_width:

In [8]:
```
sns.boxplot(x=data["sepal_width"])
```

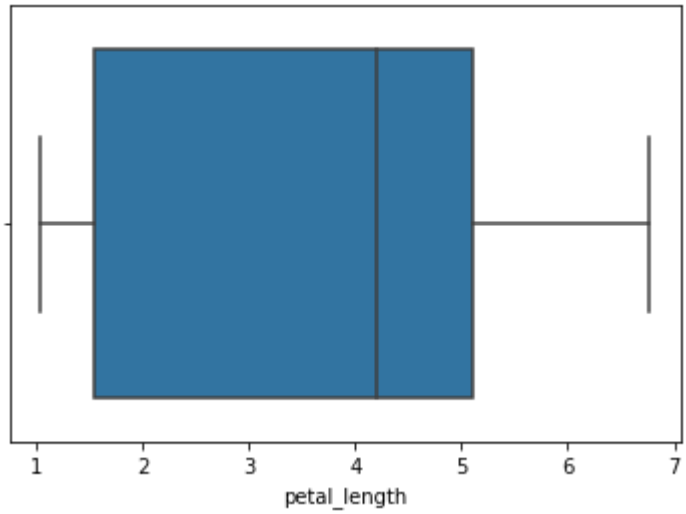
Out[8]:
```
<AxesSubplot:xlabel='sepal_width'>
```



petal_length:

In [9]:
```
sns.boxplot(x=data["petal_length"])
```

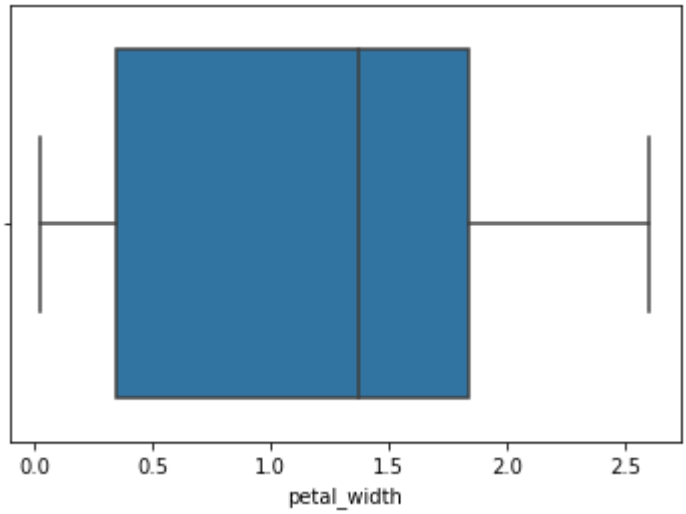
Out[9]:
```
<AxesSubplot:xlabel='petal_length'>
```



petal_width:

In [10]:
```
sns.boxplot(x=data["petal_width"])
```

Out[10]:
```
<AxesSubplot:xlabel='petal_width'>
```



From the above box plots, we can observe that the feature **"sepal_width"** has **5** outliers. We handled the outliers in this data with the help of interquarantile concept.

In [11]:
```
print(np.percentile(data['sepal_width'], 10))
print(np.percentile(data['sepal_width'], 90))
```

```
2.4984958663582804
3.673501217663288
```

In [12]:
```
data["sepal_width"] = np.where(data["sepal_width"] <2.4984958663582804, 2.4984958
data["sepal_width"] = np.where(data["sepal_width"] >3.673501217663288, 3.6735012
```

Here, initially we have defined the limits for the feature's data by calculating the percentiles of 10% and 90%.

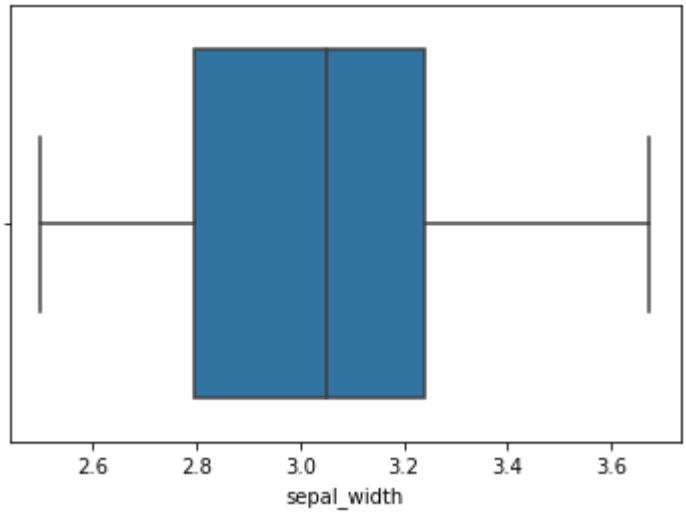
Now in order to remove the outliers, we have replaced the sample values of the feature **"sepal_width"** that have,

-- Less than 10% percentile with a data value of **"2.4984958663582804"**

-- More than 90% percentile with a data value of **"3.673501217663288"**

In [13]:
```
sns.boxplot(x=data["sepal_width"])
```

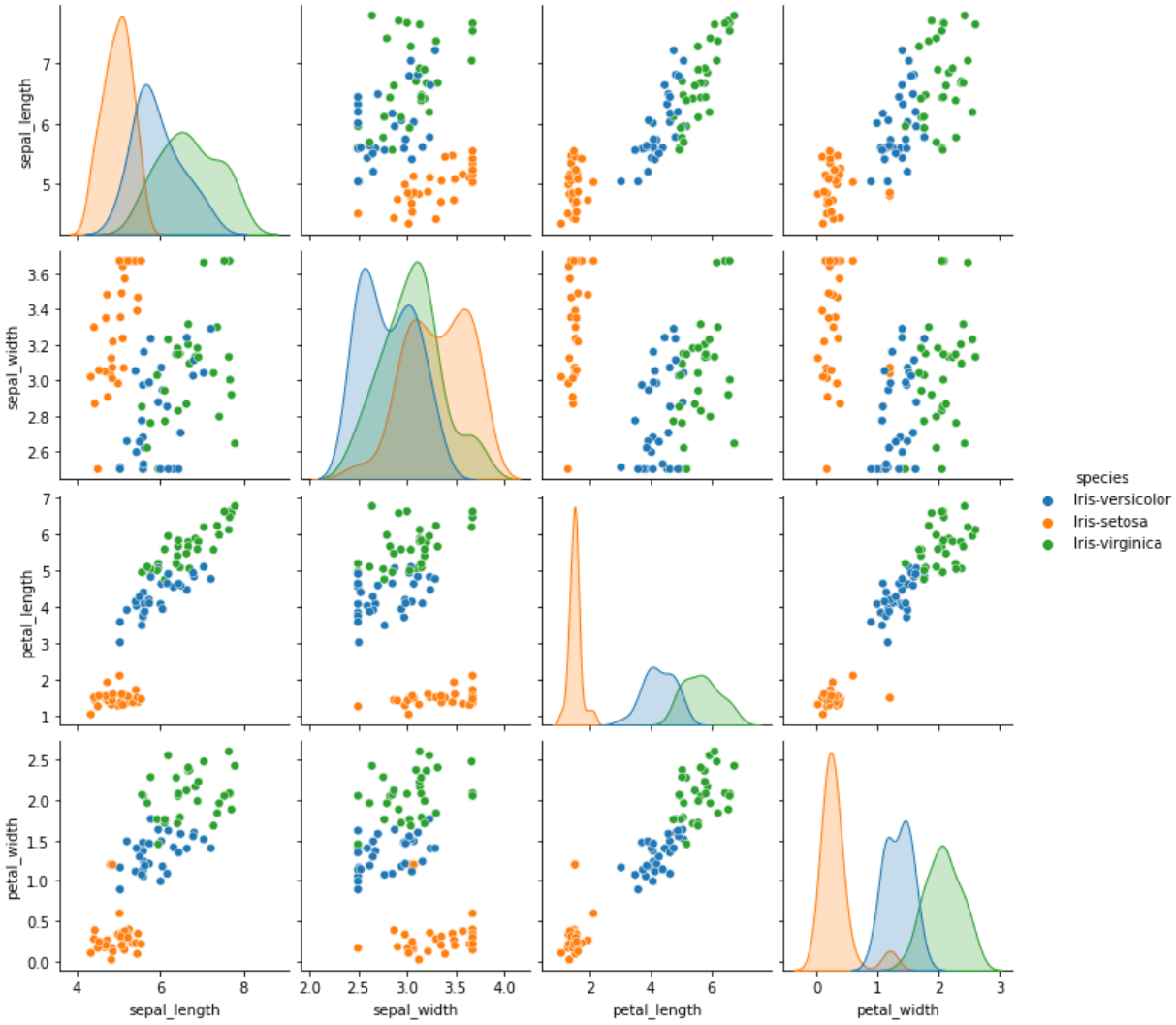
Out[13]:
```
<AxesSubplot:xlabel='sepal_width'>
```



Once the outliers are handled, the boxplot for the feature **"sepal_width"** looks as shown above.

In [14]:
```
sns.pairplot(data[['sepal_length','sepal_width','petal_length','petal_width','spe
```

Out[14]:
```
<seaborn.axisgrid.PairGrid at 0x293ef7758b0>
```



In [15]:
```
data.to_csv("cleaned_iris_dataset.csv",index=False)
```

At this stage, the dataset is free from outliers and is now saved in a new CSV filed named "cleaned_iris_dataset.csv".

This new CSV file is given as an input to the Question CM5 for developing the KNN classifier model.