In [ ]:

```
import pandas as pd
import numpy as np
```

In [ ]:

```
# Load the dataset
df=pd.read_csv("Iris.csv")
```

In [ ]:

```
df.head()
```

Out[ ]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [ ]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [ ]:

```
df.describe()
```

Out[ ]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [ ]:

```
df.isnull().sum()
```

Out[ ]:

```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [ ]:

```
df['Species'].unique()
```

Out[ ]:

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

In [ ]:
```
df['Target']=df['Species'].replace({'Iris-setosa':0, 'Iris-versicolor':1, 'Iris-virginica':2})
```

In [ ]:
```
df.drop("Species",inplace=True, axis=1)
```

In [ ]:
```
df.drop("Id", inplace=True, axis=1 )
```

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

Out[ ]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Target
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

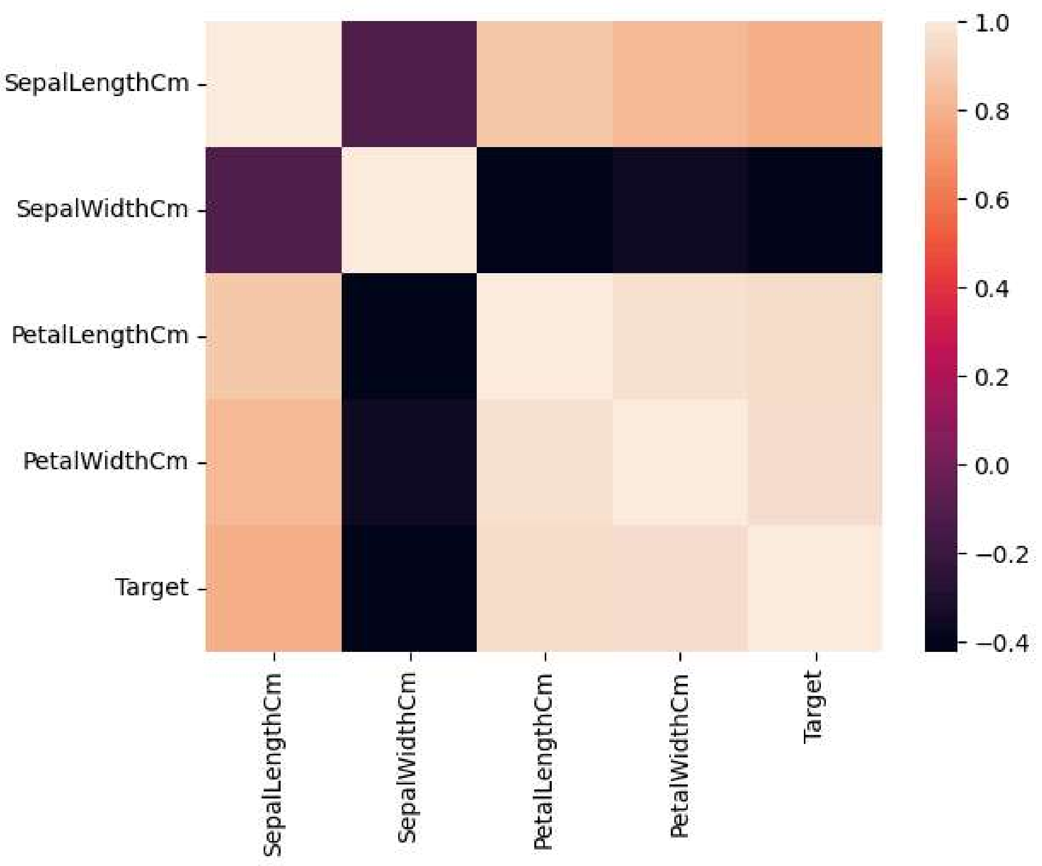
In [ ]:
```
df.corr()
```

Out[ ]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Target
SepalLengthCm	1.000000	-0.109369	0.871754	0.817954	0.782561
SepalWidthCm	-0.109369	1.000000	-0.420516	-0.356544	-0.419446
PetalLengthCm	0.871754	-0.420516	1.000000	0.962757	0.949043
PetalWidthCm	0.817954	-0.356544	0.962757	1.000000	0.956464
Target	0.782561	-0.419446	0.949043	0.956464	1.000000

In [ ]:
```
import seaborn as sns
sns.heatmap(df.corr())
```

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

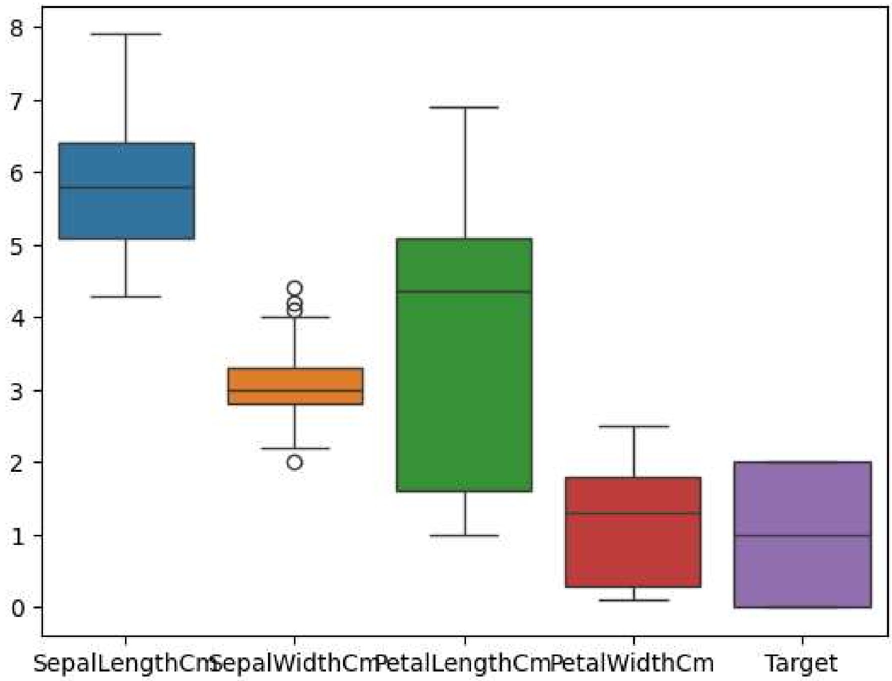


In [ ]:

```
sns.boxplot(df)
```

Out[ ]:

```
<Axes: >
```



In [ ]:

```
#dependent and independent Variable
x=df.drop('Target', axis=1)
y=df[['Target']]
```

In [ ]:

```
#train Test slipt
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=42)
```

In [ ]:

```
# Initialize the model
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier()
```

In [ ]:

```
# Fit the model
clf.fit(x_train,y_train)
```

Out[ ]:

```
▾ DecisionTreeClassifier

DecisionTreeClassifier()
```

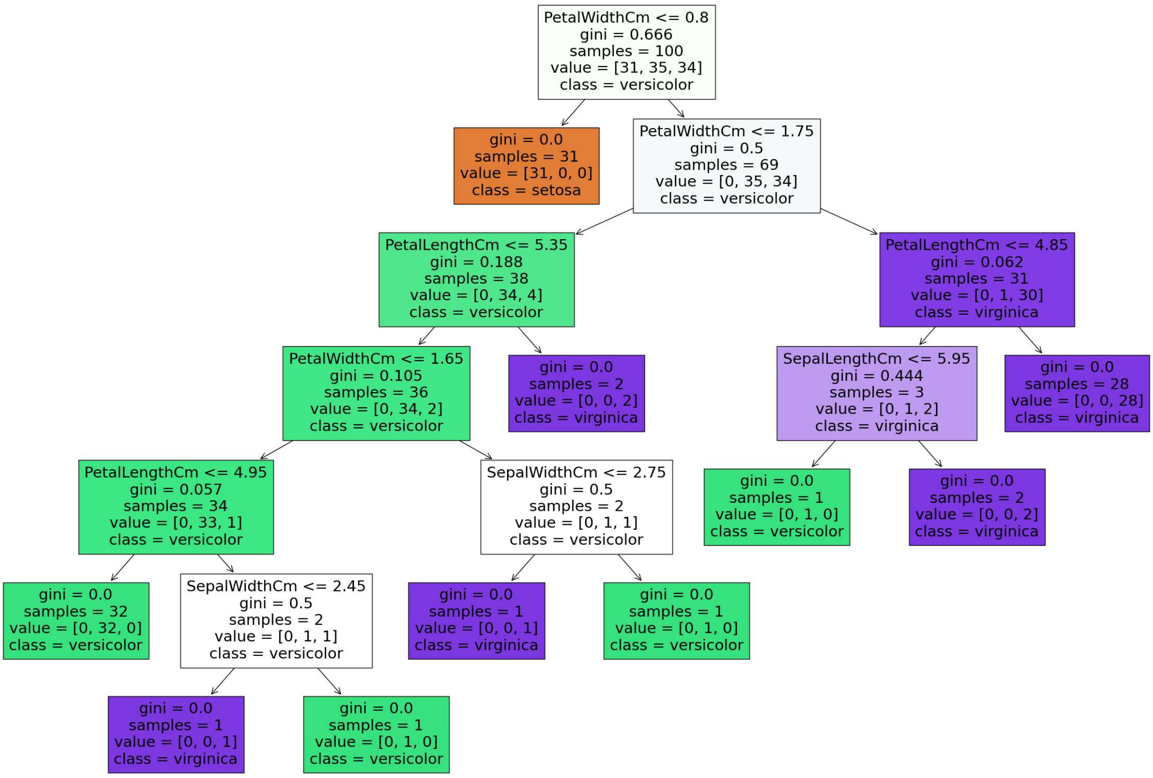
In [ ]:

```
y_pred=clf.predict(x_test)
```

In [ ]:

```
from sklearn import tree
import matplotlib.pyplot as plt

# Plot the decision tree
plt.figure(figsize=(30,20))
tree.plot_tree(clf, feature_names=x.columns, class_names=['setosa', 'versicolor', 'virginica'], f
plt.show()
```



In [ ]:

```
from sklearn.metrics import classification_report, confusion_matrix
classification_report1=classification_report(y_test, y_pred)
print(classification_report1)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       0.94      1.00      0.97        15
           2       1.00      0.94      0.97        16

    accuracy                           0.98        50
   macro avg       0.98      0.98      0.98        50
weighted avg       0.98      0.98      0.98        50
```

In [ ]:

```
confusion_matrix(y_test, y_pred)
```

Out[ ]:

```
array([[19,  0,  0],
       [ 0, 15,  0],
       [ 0,  1, 15]], dtype=int64)
```