In [1]:

```
import numpy as np
import pandas as pd
import sklearn.metrics as sm
import seaborn as sns
import matplotlib.pyplot as mt
%matplotlib inline

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.tree import plot_tree
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix, classification_report
```

In [5]:

```
data=pd.read_csv(r'C:\Users\RITIKA SHUKLA\Downloads\Iris.csv',index_col=0)
data.head()
```

Out[5]:

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

In [6]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 150 entries, 1 to 150
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   SepalLengthCm  150 non-null    float64
 1   SepalWidthCm   150 non-null    float64
 2   PetalLengthCm  150 non-null    float64
 3   PetalWidthCm   150 non-null    float64
 4   Species        150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 7.0+ KB
```

In [7]:

```
data.describe()
```

Out[7]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
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 [8]:

```
target=data['Species']
df=data.copy()
df=df.drop('Species', axis=1)
df.shape
```

Out[8]:

```
(150, 4)
```

In [9]:

```
#defingi the attributes and labels
X=data.iloc[:, [0,1,2,3]].values
le=LabelEncoder()
data['Species']=le.fit_transform(data['Species'])
y=data['Species'].values
data.shape
```

Out[9]:

```
(150, 5)
```

In [10]:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
print("Traingin split:",X_train.shape)
print("Testin spllit:",X_test.shape)
```

```
Traingin split: (120, 4)
Testin spllit: (30, 4)
```

In [11]:

```
dtree=DecisionTreeClassifier()
dtree.fit(X_train,y_train)
print("Decision Tree Classifier created!")
```

```
Decision Tree Classifier created!
```

In [12]:

```
y_pred=dtree.predict(X_test)
print("Classification report:\n",classification_report(y_test,y_pred))
```

```
Classification report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

In [13]:

```
print("Accuracy:",sm.accuracy_score(y_test,y_pred))
```

```
Accuracy: 1.0
```

In [14]:

```
#confusion matrix
cm=confusion_matrix(y_test,y_pred)
cm
```

Out[14]:

```
array([[10,  0,  0],
       [ 0,  9,  0],
       [ 0,  0, 11]], dtype=int64)
```

In [ ]:

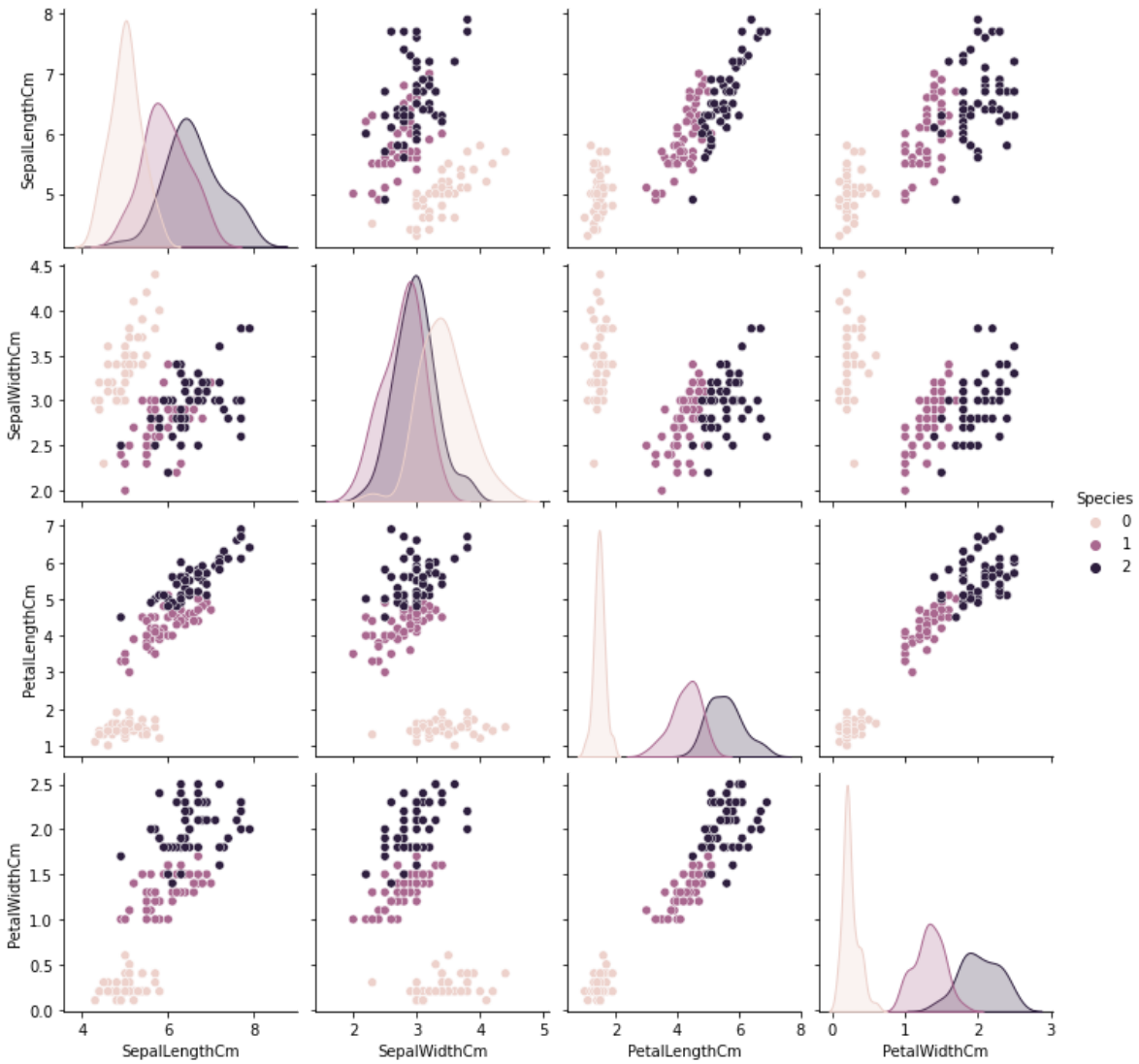
```

```

In [16]:

```
sns.pairplot(data, hue='Species')
```

Out[16]:

```
<seaborn.axisgrid.PairGrid at 0x1bfcbca3048>
```

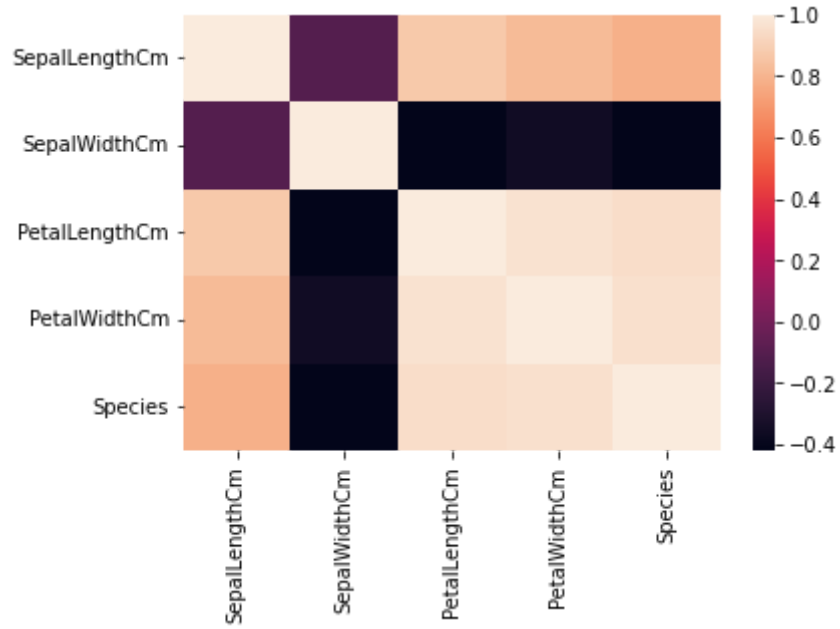


In [17]:

```
sns.heatmap(data.corr())
```

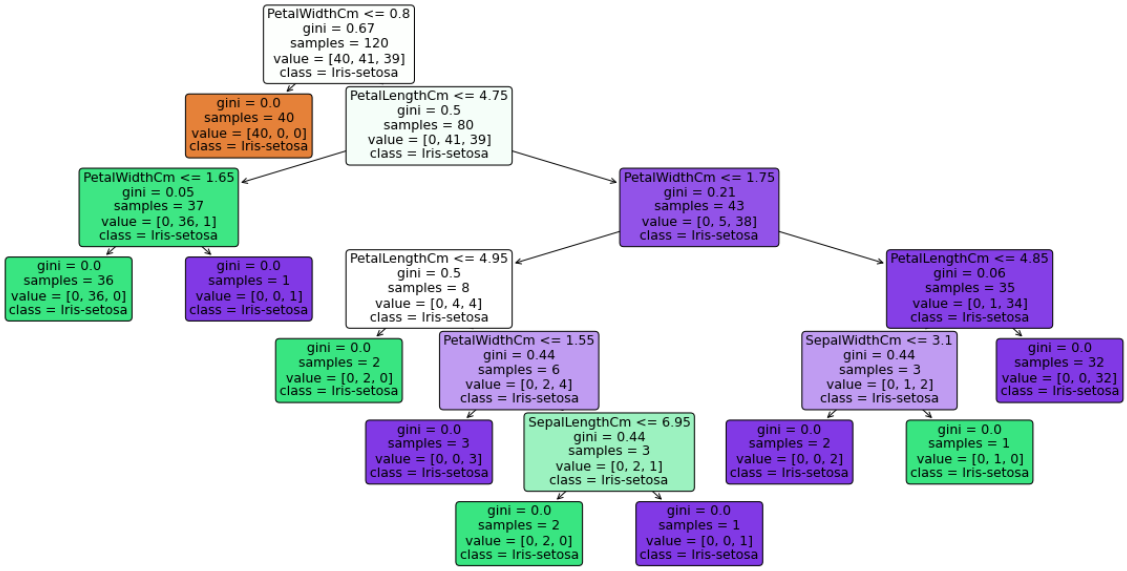
Out[17]:

<AxesSubplot:>



In [18]:

```
#visualizing the graph
mt.figure(figsize=(20,10))
tree=plot_tree(dtree,feature_names=df.columns,precision=2,rounded=True,filled=True,class_na
```



In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```