Tanvi Bhosle

Data Science Intern @ LGM Virtual Intrnship 2021 (October)

Intermediate Level Task

Task 2: Prediction using Decision Tree Algorithm

In [ ]:

Import Libraries

In [18]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
import graphviz
from sklearn import tree
```

Import Dataset

In [19]:
```
data=pd.read_csv("C:\\Users\\Admin\\Desktop\\Iris.csv",encoding=("ISO-8859-1"),low_memory=False)
data.head(5)
```

Out[19]:

	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 [20]:
```
data.shape
```

Out[20]:
```
(150, 6)
```

In [21]:
```
data["Species"].value_counts()
```

Out[21]:
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

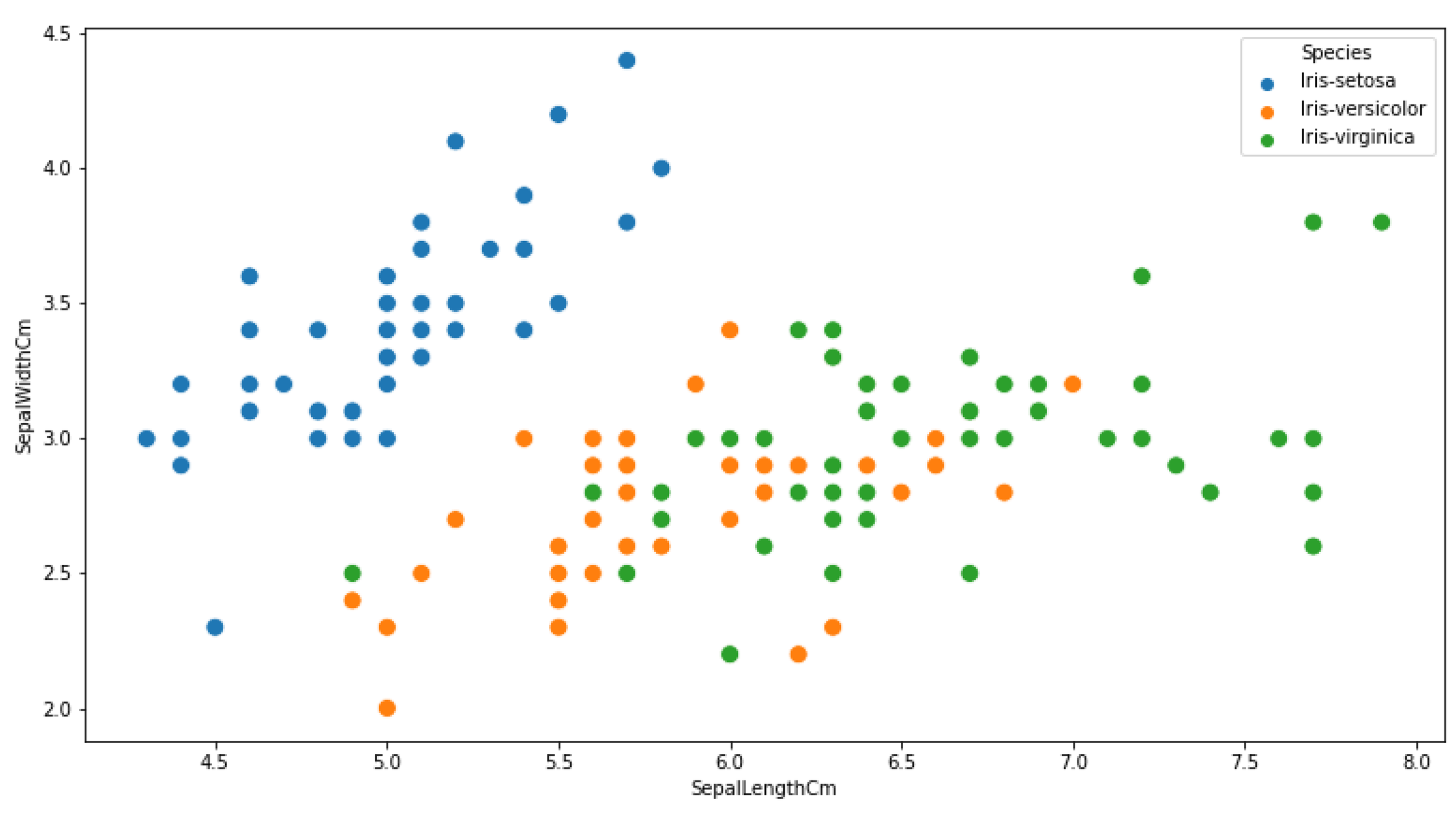
In [22]:
```
data.isnull().sum()
```

Out[22]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

Sepal Dimensions

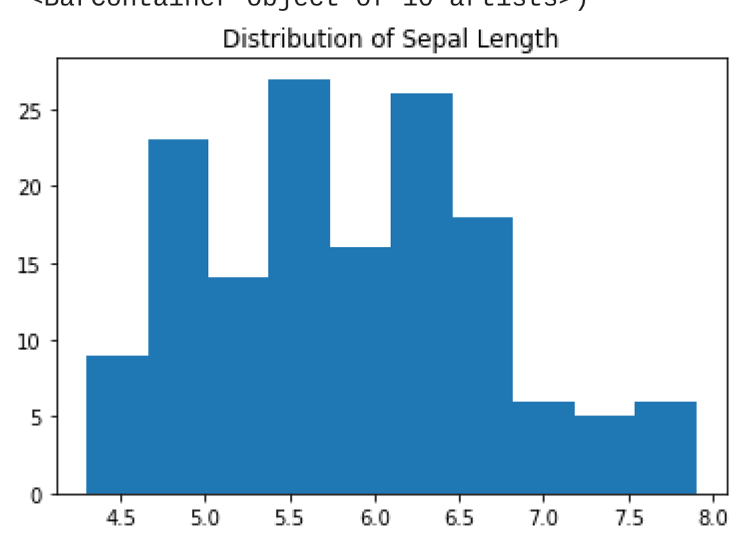
In [23]:
```
plt.figure(figsize=(13,7))
sns.scatterplot(x=data["SepalLengthCm"],y=data["SepalWidthCm"],hue=data["Species"],s=100)
```

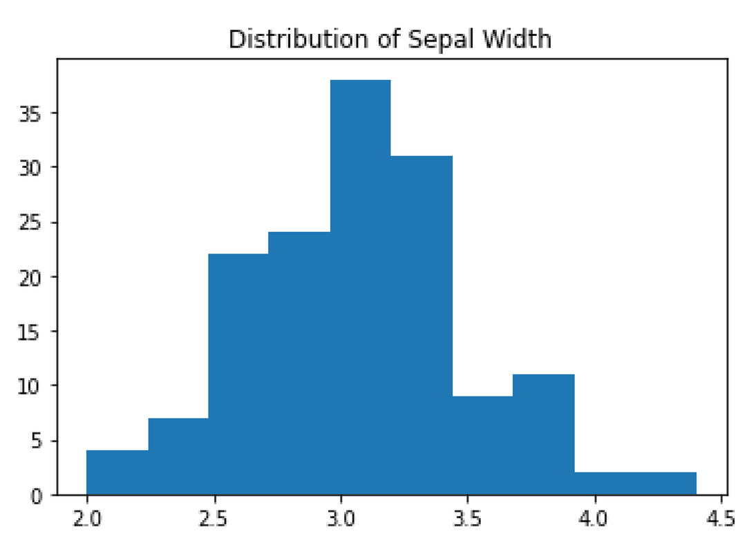
Out[23]:
```
<AxesSubplot:xlabel='SepalLengthCm', ylabel='SepalWidthCm'>
```



In [24]:
```
fig,axes=plt.subplots(figsize=(6,4))
plt.title("Distribution of Sepal Length")
plt.hist(data["SepalLengthCm"])
fig,axes=plt.subplots(figsize=(6,4))
plt.title("Distribution of Sepal Width")
plt.hist(data["SepalWidthCm"])
```

Out[24]:
```
(array([ 4.,  7., 22., 24., 38., 31.,  9., 11.,  2.,  2.]),
 array([2.  , 2.24, 2.48, 2.72, 2.96, 3.2 , 3.44, 3.68, 3.92, 4.16, 4.4 ]),
 <BarContainer object of 10 artists>)
```

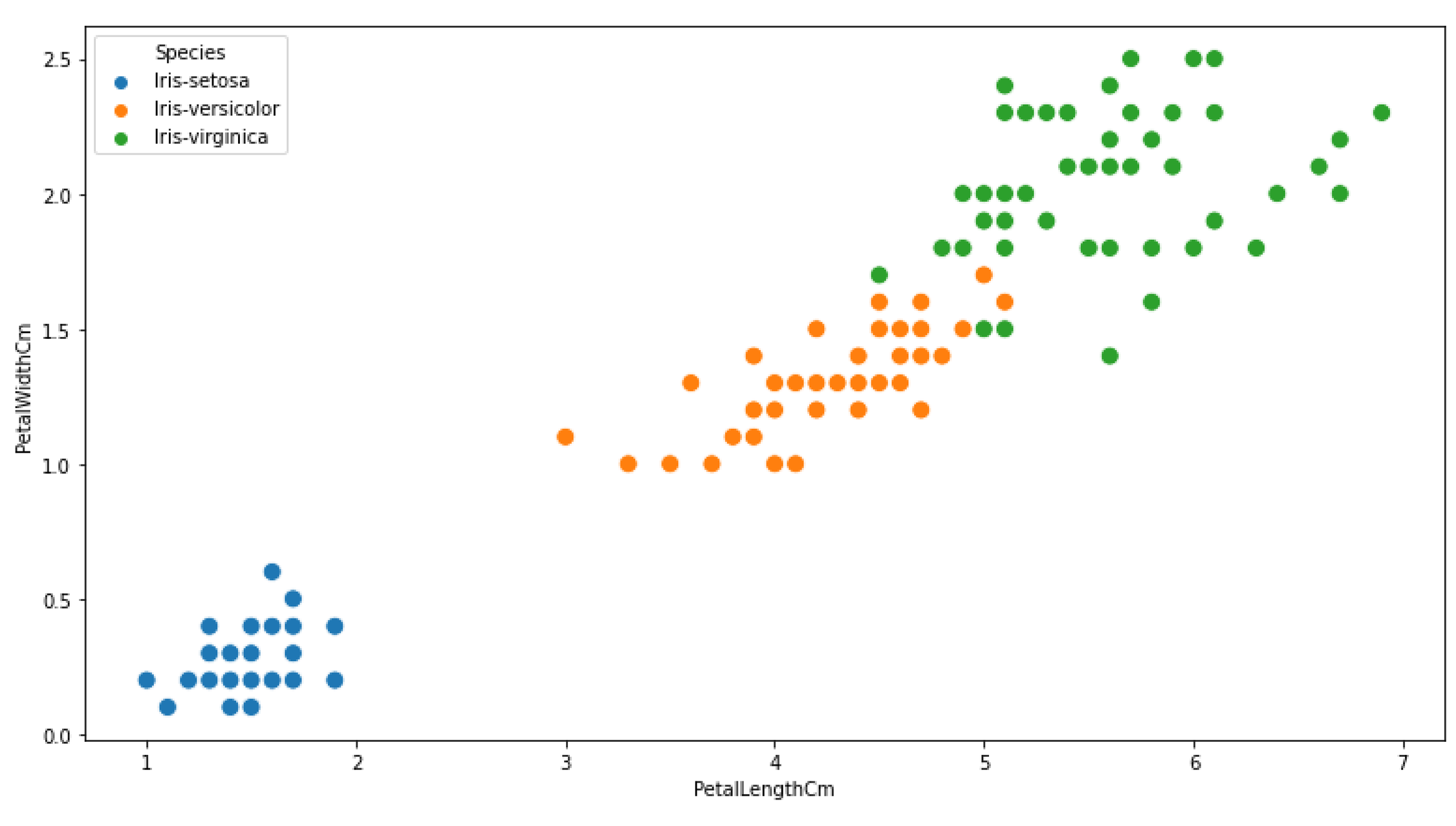




Petal Dimensions

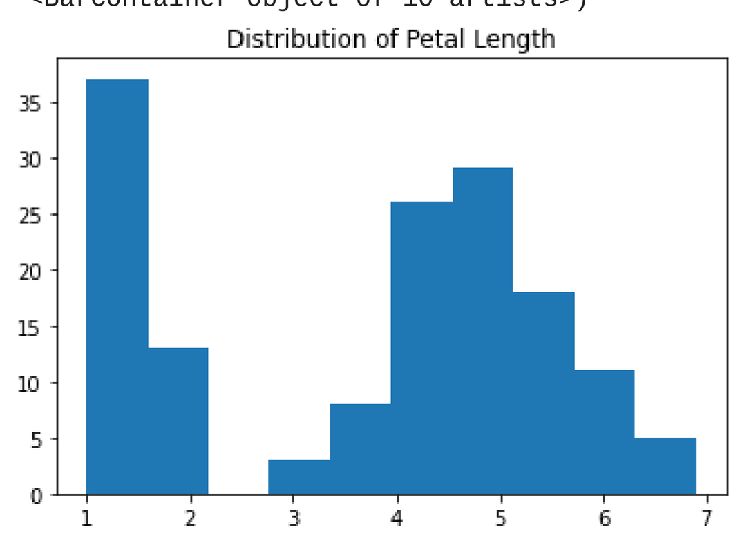
In [25]:
```
plt.figure(figsize=(13,7))
sns.scatterplot(x=data["PetalLengthCm"],y=data["PetalWidthCm"],hue=data["Species"],s=100)
```

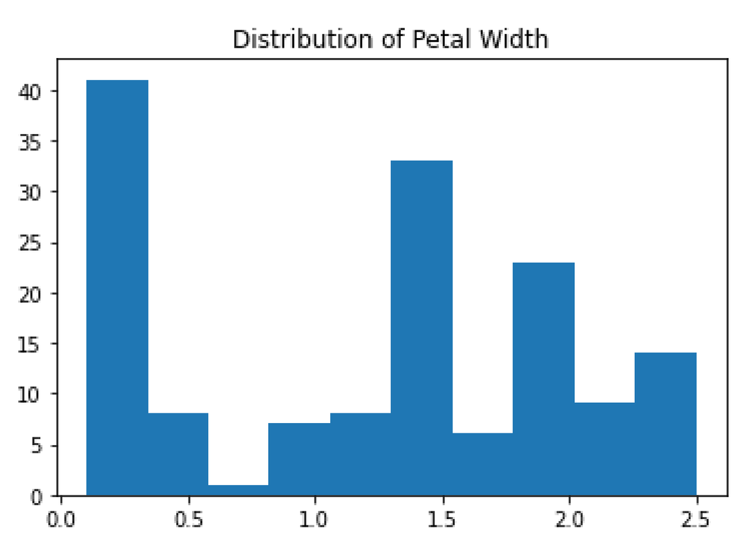
Out[25]:
```
<AxesSubplot:xlabel='PetalLengthCm', ylabel='PetalWidthCm'>
```



In [26]:
```
fig,axes=plt.subplots(figsize=(6,4))
plt.title("Distribution of Petal Length")
plt.hist(data["PetalLengthCm"])
fig,axes=plt.subplots(figsize=(6,4))
plt.title("Distribution of Petal Width")
plt.hist(data["PetalWidthCm"])
```

Out[26]:
```
(array([41.,  8.,  1.,  7.,  8., 33.,  6., 23.,  9., 14.]),
 array([0.1 , 0.34, 0.58, 0.82, 1.06, 1.3 , 1.54, 1.78, 2.02, 2.26, 2.5 ]),
 <BarContainer object of 10 artists>)
```





Correlation between Dependent and Independent variables

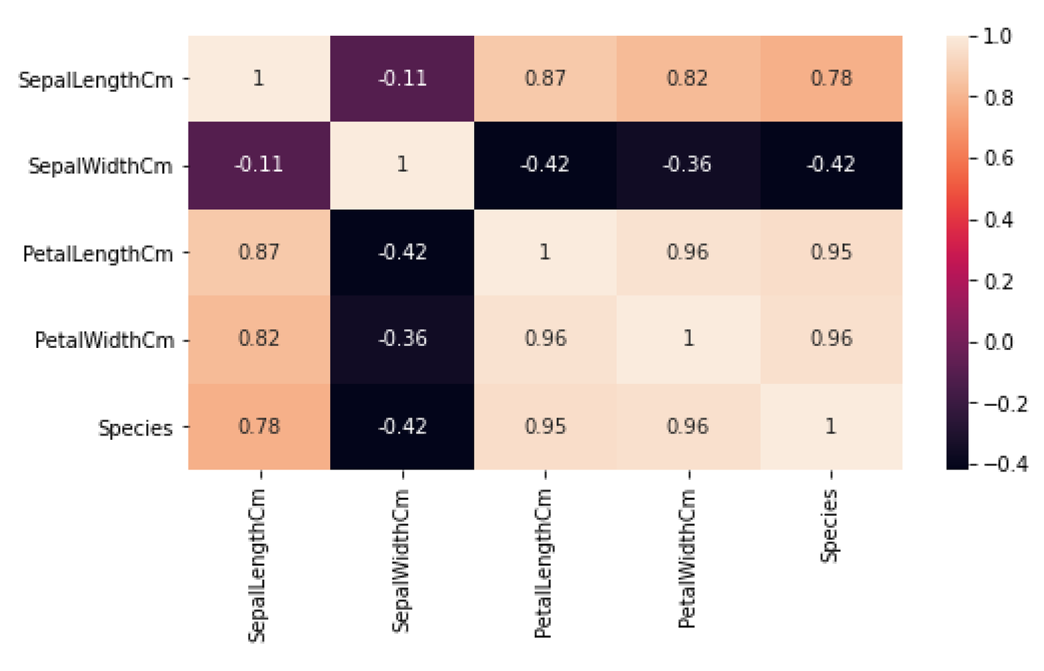
In [27]:
```
data1=data.drop("Id",axis=1)
data1.head()
data1["Species"].replace("Iris-setosa","0",inplace=True)
data1["Species"].replace("Iris-versicolor","1",inplace=True)
data1["Species"].replace("Iris-virginica","2",inplace=True)
data1["Species"]=data1.Species.astype(int)
data1.head()
```

Out[27]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	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 [28]:
```
plt.figure(figsize=(8,4))
sns.heatmap(data1.corr(),annot=True)
```

Out[28]:
```
<AxesSubplot:>
```



Model Building

In [29]:
```
train,test=train_test_split(data1,test_size=0.3)
```

In [30]:
```
train.shape,test.shape
```

Out[30]:
```
((105, 5), (45, 5))
```

In [31]:
```
train_x=train[["SepalLengthCm","SepalWidthCm","PetalLengthCm","PetalWidthCm"]]
train_y=train.Species
test_x=test[["SepalLengthCm","SepalWidthCm","PetalLengthCm","PetalWidthCm"]]
test_y=test.Species
```

In [32]:
```
dtree=DecisionTreeClassifier()
dtree.fit(train_x,train_y)
predictions=dtree.predict(test_x)
accuracy=metrics.accuracy_score(predictions,test_y)
accuracy
```

Out[32]:
```
0.9777777777777777
```

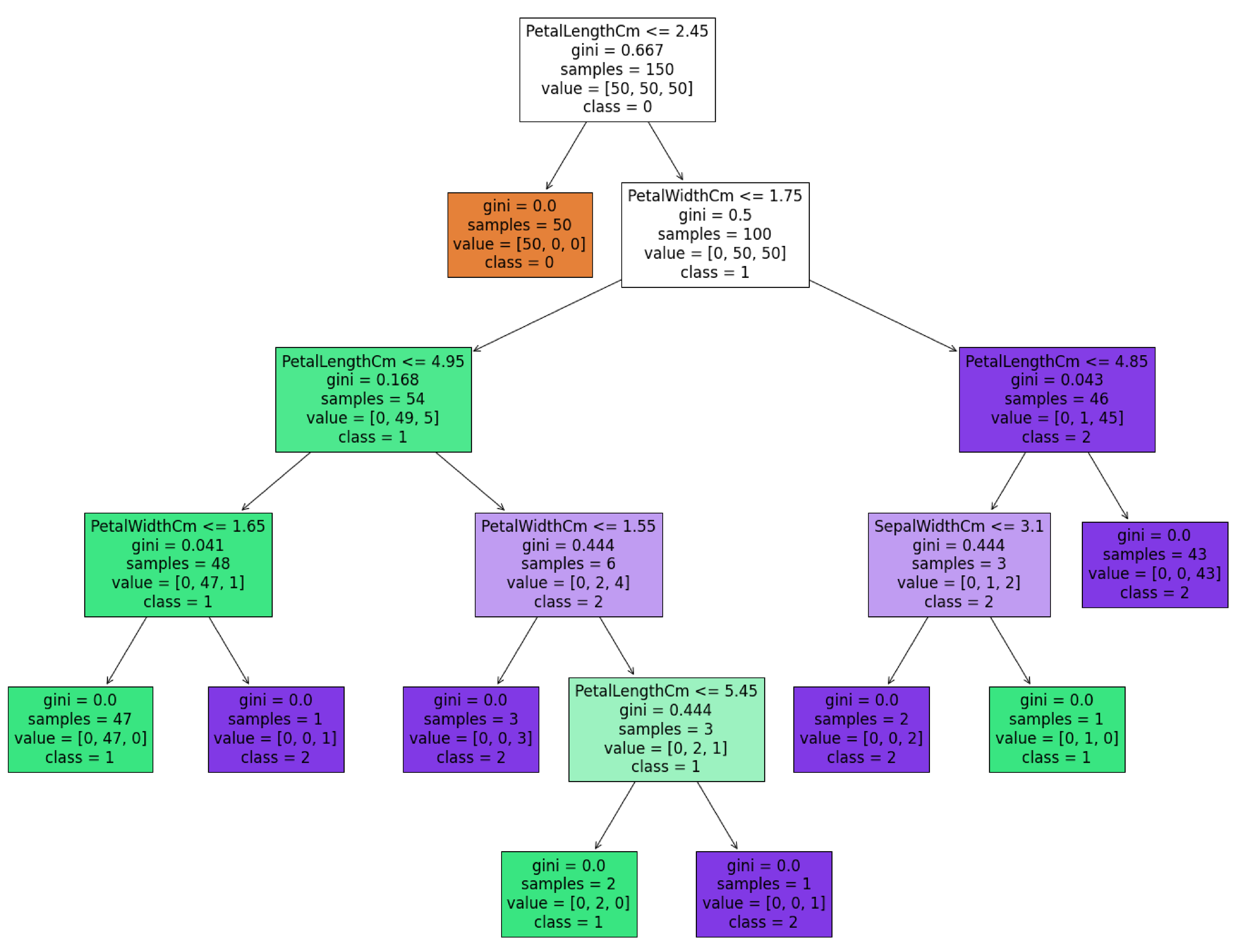
In [33]:
```
x=data1[["SepalLengthCm","SepalWidthCm","PetalLengthCm","PetalWidthCm"]]
y=data1.Species
dtree1=DecisionTreeClassifier()
dtree1.fit(x,y)
```

Out[33]:
```
DecisionTreeClassifier()
```

Visualization

In [34]:
```
fig = plt.figure(figsize=(25,20))
tree.plot_tree(dtree1,feature_names=["SepalLengthCm","SepalWidthCm","PetalLengthCm","PetalWidthCm"],class_names=["0","1","2"],
filled=True)
```

Out[34]:
```
[Text(697.5, 996.6, 'PetalLengthCm <= 2.45\ngini = 0.667\nsamples = 150\nvalue = [50, 50, 50]\nclass = 0'),
 Text(590.1923076923077, 815.4000000000001, 'gini = 0.0\nsamples = 50\nvalue = [50, 0, 0]\nclass = 0'),
 Text(804.8076923076923, 815.4000000000001, 'PetalWidthCm <= 1.75\ngini = 0.5\nsamples = 100\nvalue = [0, 50, 50]\nclass = 1'),
 Text(429.2307692307692, 634.2, 'PetalLengthCm <= 4.95\ngini = 0.168\nsamples = 54\nvalue = [0, 49, 5]\nclass = 1'),
 Text(214.6153846153846, 453.0, 'PetalWidthCm <= 1.65\ngini = 0.041\nsamples = 48\nvalue = [0, 47, 1]\nclass = 1'),
 Text(107.3076923076923, 271.79999999999995, 'gini = 0.0\nsamples = 47\nvalue = [0, 47, 0]\nclass = 1'),
 Text(321.9230769230769, 271.79999999999995, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]\nclass = 2'),
 Text(643.8461538461538, 453.0, 'PetalWidthCm <= 1.55\ngini = 0.444\nsamples = 6\nvalue = [0, 2, 4]\nclass = 2'),
 Text(536.5384615384615, 271.79999999999995, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]\nclass = 2'),
 Text(751.1538461538462, 271.79999999999995, 'PetalLengthCm <= 5.45\ngini = 0.444\nsamples = 3\nvalue = [0, 2, 1]\nclass = 1'),
 Text(643.8461538461538, 90.59999999999991, 'gini = 0.0\nsamples = 2\nvalue = [0, 2, 0]\nclass = 1'),
 Text(858.4615384615385, 90.59999999999991, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]\nclass = 2'),
 Text(1180.3846153846155, 634.2, 'PetalLengthCm <= 4.85\ngini = 0.043\nsamples = 46\nvalue = [0, 1, 45]\nclass = 2'),
 Text(1073.076923076923, 453.0, 'SepalWidthCm <= 3.1\ngini = 0.444\nsamples = 3\nvalue = [0, 1, 2]\nclass = 2'),
 Text(965.7692307692307, 271.79999999999995, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]\nclass = 2'),
 Text(1180.3846153846155, 271.79999999999995, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]\nclass = 1'),
 Text(1287.6923076923076, 453.0, 'gini = 0.0\nsamples = 43\nvalue = [0, 0, 43]\nclass = 2')]
```



In [ ]: