TASK 6 : PREDICTION USING DECISION TREE ALGORITHM

SUBMITTED BY RIDHANYA S

Create the Decision Tree classifier and visualize it graphically

In [1]:
```
# IMPORTED NECESSARY LIBRARIES

import numpy as np
import pandas as pd
import sklearn
import matplotlib.pyplot as plt
from sklearn import tree, model_selection, metrics
```

In [2]:
```
# LOADING THE DATA

from sklearn.datasets import load_iris
df = load_iris()
print("------ Data Loaded Successfully -----")
```

```
------ Data Loaded Successfully -----
```

In [3]:
```
# DIMENSION OF THE DATA

df.data.shape
```

Out[3]:
```
(150, 4)
```

In [4]:
```
# ATTRIBUTES DATA AND THE CORRESPONDING LABELS

print("Prediction Classes : \n",df.target_names)
print("\nFeatures :\n",df.feature_names)
```

```
Prediction Classes :
 ['setosa' 'versicolor' 'virginica']

Features :
 ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

In [5]:
```
# DEFINING DEPENDENT AND INDEPENDENT VARIABLES

X=pd.DataFrame(df.data,columns=df.feature_names)
y=df.target
print(X)
print(y)
```

```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
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
..                 ...               ...                ...               ...
145                6.7               3.0                5.2               2.3
146                6.3               2.5                5.0               1.9
147                6.5               3.0                5.2               2.0
148                6.2               3.4                5.4               2.3
149                5.9               3.0                5.1               1.8

[150 rows x 4 columns]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
```

In [6]:
```
# SPLITTING THE DATA INTO TRAIN AND TEST SETS

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X, y, test_size = 0.3, random_state=0)
```

In [7]:
```
# DEFINING DECISION TREE CLASSIFIER

from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier()
```

In [8]:
```
# FITTING TRAINING DATA

clf.fit(X_train, y_train)
print("----- Model Trained -----")
```

```
----- Model Trained -----
```

In [9]:
```
# PREDICTION ON TEST DATA

y_pred = clf.predict(X_test)
print(y_pred)
```

```
[2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0
 2 1 1 2 0 2 0 0]
```

In [10]:
```
# ACCURACY ON TEST DATA

from sklearn.metrics import accuracy_score, confusion_matrix
accuracy = accuracy_score(y_test, y_pred)
print("Accuarcy : {:.2f}".format(accuracy))

cm = np.array(confusion_matrix(y_test,y_pred))
cm
```

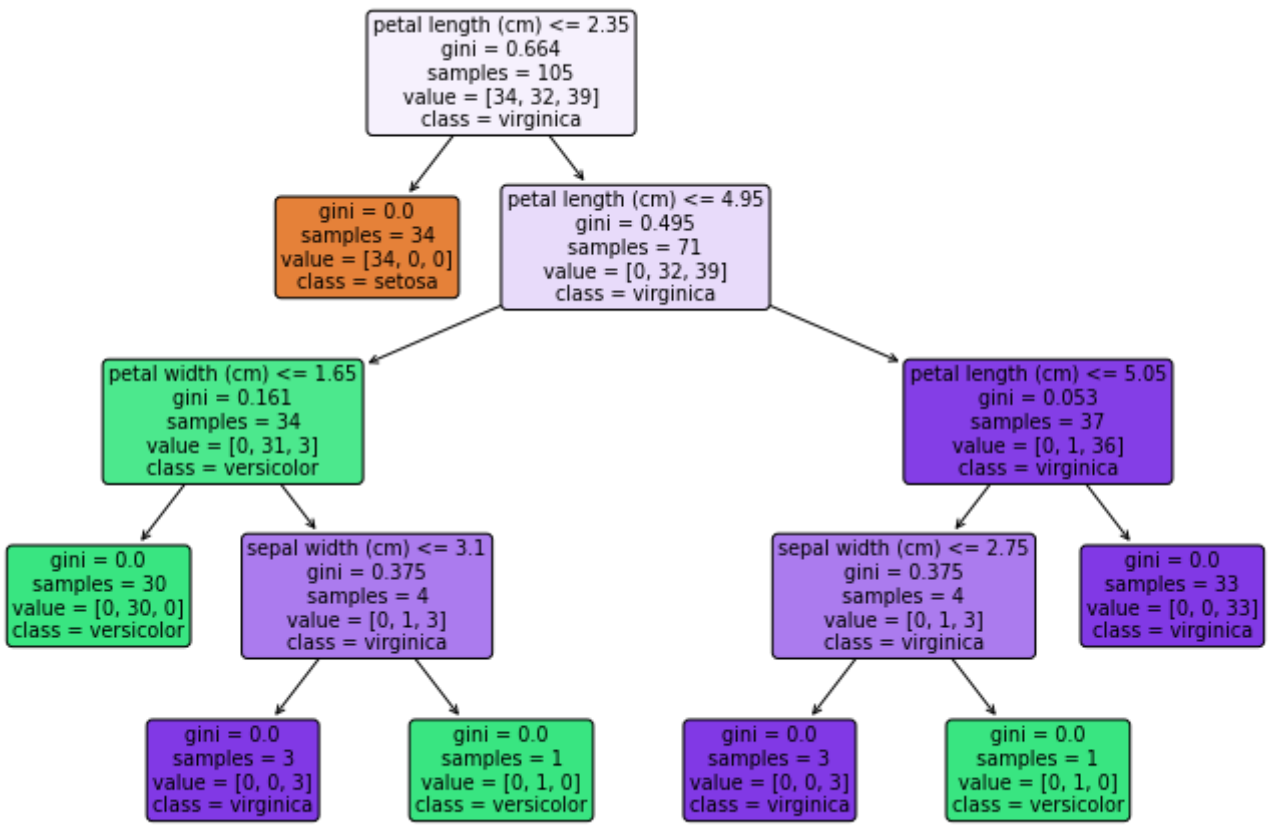
```
Accuarcy : 0.98
```

Out[10]:
```
array([[16,  0,  0],
       [ 0, 17,  1],
       [ 0,  0, 11]], dtype=int64)
```

This model predicts with 98% accuracy

In [11]:
```
# VISUALIZING THE DECISION TREE ON IRIS DATASET

import matplotlib.pyplot as plt
from sklearn import tree
plt.figure(figsize = (12,8))
tree.plot_tree(clf.fit(X_train, y_train), filled=True, rounded = True,
               feature_names = df.feature_names, class_names = df.target_names)
plt.show()
```



THANK YOU :)