Graduate Rotational Internship Program

The Sparks Foundation

Data Science & Business Analytics Tasks

TASK # 6 Prediction using Decision Tree Algorithm

IMPORTING THE LIBRARIES

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

Dataset_name -- IRIS DATASET

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

In [3]:
```
# first five elements
data.head()
```

Out[3]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [4]:
```
#shape of data
data.shape
```

Out[4]:
```
(150, 5)
```

In [5]:
```
#Column name
data.columns
```

Out[5]:
```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [6]:
```
X = data.iloc[:,0:4].values
```

In [7]:
```
X.shape
```

Out[7]:
```
(150, 4)
```

In [8]:
```
y = data.iloc[:,-1].values
```

In [9]:
```
y.shape
```

Out[9]:
```
(150,)
```

LabelEncoding used for label the output columns which contains the categorical values

In [10]:
```
from sklearn.preprocessing import LabelEncoder
```

In [11]:
```
le = LabelEncoder()
```

In [12]:
```
y = le.fit_transform(y)
```

In [13]:
```
le.classes_
```

Out[13]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

After label Encoding 0-Iris-setosa 1-Iris-versicolor 2-Iris-virginica

In [14]:
```
y
```

Out[14]:
```
array([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, 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 [15]:
```
#Train_test_split
from sklearn.model_selection import train_test_split
```

In [16]:
```
 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.40, random_stat
```

In [17]:
```
print(X_train.shape,X_test.shape)
```
```
(90, 4) (60, 4)
```

DecisionTreeClassifier

In [18]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [19]:
```
dtc = DecisionTreeClassifier()
```

In [20]:
```
dtc.fit(X_train,y_train)
```

Out[20]:
```
DecisionTreeClassifier()
```

In [21]:
```
y_pred = dtc.predict(X_test)
```

In [22]:
```
y_pred
```

Out[22]:
```
array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2,
       0, 2, 2, 2, 2, 2, 0, 0, 0, 0, 1, 0, 0, 2, 1, 0, 0, 0, 2, 1, 1, 0,
       0, 1, 1, 2, 1, 2, 1, 2, 1, 0, 2, 1, 0, 0, 0, 1])
```

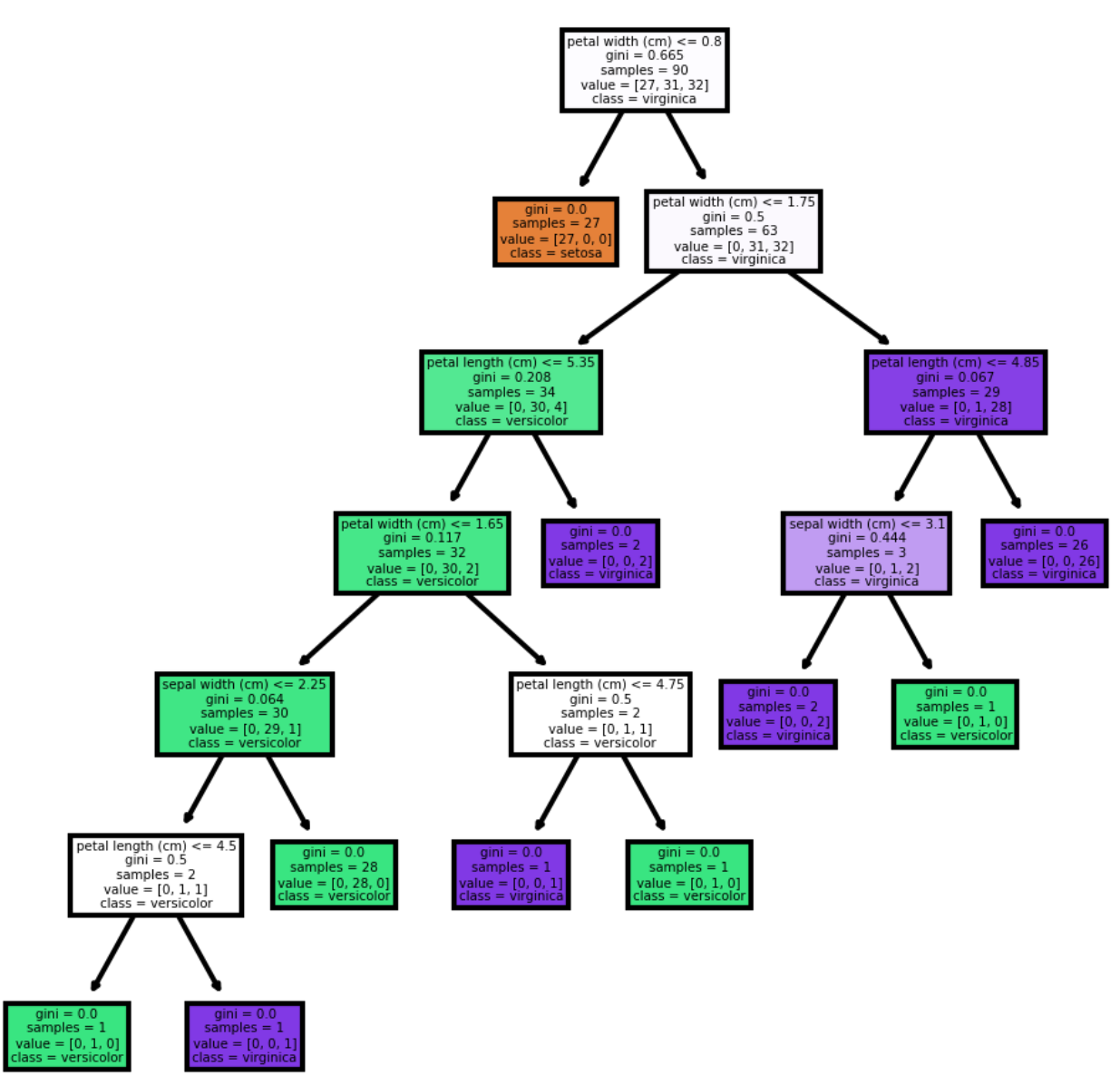
In [23]:
```
from sklearn.metrics import accuracy_score  # Evbaluation metric - accuracy_score
```

In [24]:
```
print("ACCURACY SCORE",accuracy_score(y_pred,y_test))
```
```
ACCURACY SCORE 0.9833333333333333
```

VISUALIZATION OF DECISION TREES

In [25]:
```
from sklearn import tree
```

In [26]:
```
fn=['sepal length (cm)','sepal width (cm)','petal length (cm)','petal width (cm)']
cn=['setosa', 'versicolor', 'virginica']
fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (4,4), dpi=300)
tree.plot_tree(dtc,
               feature_names = fn,
               class_names=cn,
               filled = True);
fig.savefig('imagename.png')
```



TO FEED A NEW DATA SO THAT IT DECIDES ITS OWN SPECIES

As we feed data in the list ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']

In [27]:
```
characteristic_data = [[5.0,3.2,4.0,3.2],[4.5,3.4,4.4,2.0],[5.0,3.0,1.0,0.2],[5.9,3.0,
print(characteristic_data)
```
```
[[5.0, 3.2, 4.0, 3.2], [4.5, 3.4, 4.4, 2.0], [5.0, 3.0, 1.0, 0.2], [5.9, 3.0, 5.2, 1.
8]]
```

In [28]:
```
species_data = dtc.predict(characteristic_data)
```

In [29]:
```
species_data
```

Out[29]:
```
array([1, 1, 0, 2])
```

0 - Iris-setosa 1 - Iris-versicolor 2 - Iris-virginica

output_data shows the species which it belongs for the following lengths.

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```