Spark Foundation

Task 1:Create the Decision Tree classifier and visualize it graphically. ¶

Author : Sujata Sambhaji Gaikwad

In [1]:
```
# Importing libraries in python
import pandas as pd
import numpy as np
```

In [2]:
```
# Loading Dataset
data=pd.read_csv(r"C:\Users\HP\Downloads\Iris.csv")
data.head()
```

Out[2]:

	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 [3]:
```
# Data Preprocessing
data.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 [4]:
```
data.isnull().sum()
```

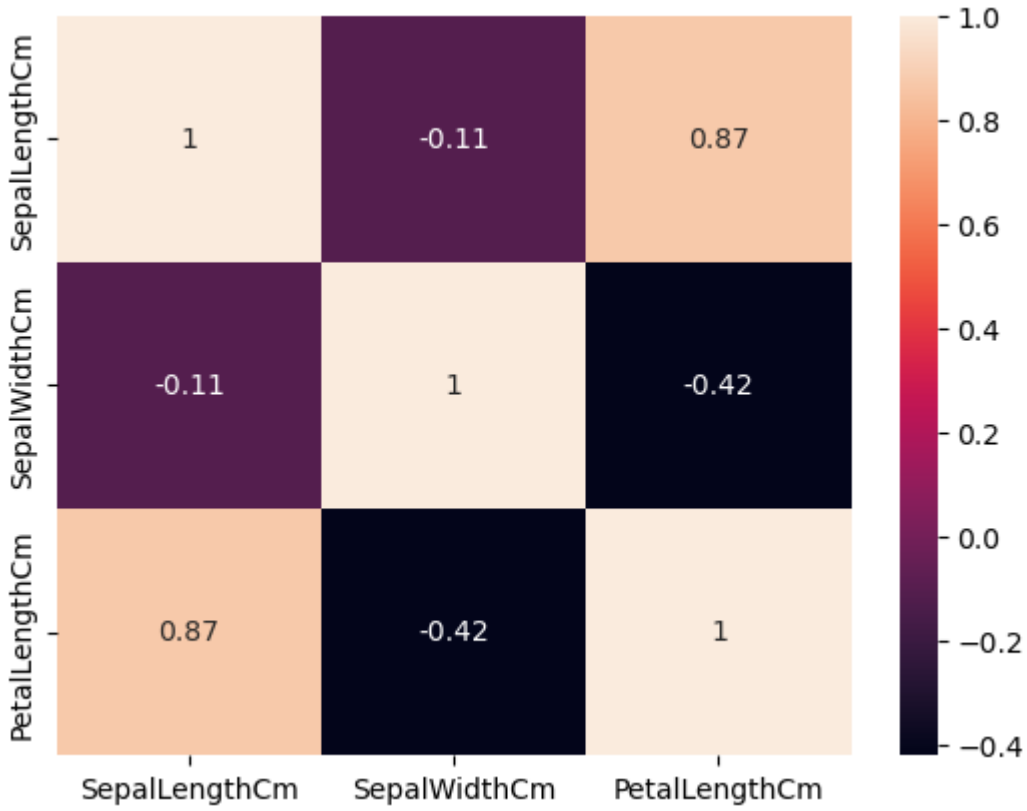
Out[4]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [5]:
```
data["Species"].value_counts()
```

Out[5]:
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

In [6]:
```
import seaborn as sns
sns.heatmap(data.iloc[:,1:4].corr(),annot=True)
```

Out[6]:
```
<AxesSubplot:>
```



In [7]:
```
# spliting the dataset
x=data.iloc[:,:-1].values
y=data.iloc[:,-1].values
```

In [8]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=
```

In [9]:
```
# Fitting model
from sklearn.tree import DecisionTreeClassifier
d1=DecisionTreeClassifier()
d1.fit(x_test,y_test)
```

Out[9]:
```
DecisionTreeClassifier()
```

In [10]:
```
y_pred=d1.predict(x_test)
y_pred
```

Out[10]:
```
array(['Iris-versicolor', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-setosa', 'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica'],
      dtype=object)
```

In [11]:
```
from sklearn.metrics import accuracy_score
acc=accuracy_score(y_pred,y_test)*100
print(acc)
```

```
100.0
```

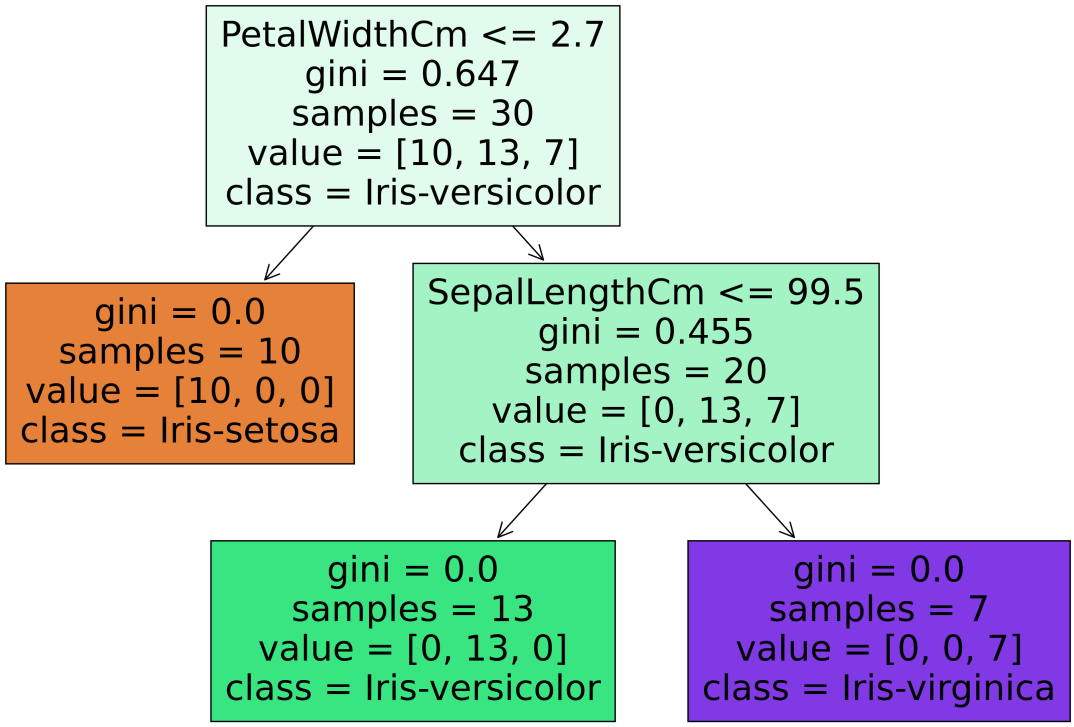
In [12]:
```
# Classification Report
import sklearn.metrics as metrics
print(metrics.classification_report(y_test,y_pred))
```

```
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00        13
 Iris-virginica       1.00      1.00      1.00         7

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

In [13]:
```
from sklearn import tree
import matplotlib.pyplot as plt
fn=["SepalLengthCm","SepalWidthCm","PetalLengthCm","PetalWidthCm"]
cn=["Iris-setosa","Iris-versicolor","Iris-virginica"]
fig,axes=plt.subplots(nrows=1,ncols=1,figsize=(15,10),dpi=300)
tree.plot_tree(d1,
               feature_names=fn,
               class_names=cn,
               filled=True)
fig.savefig("imagename.png")
```



In [ ]:

In [ ]: