1.Import Libraries

In [86]:
```
import numpy as np
import pandas as pd
from sklearn import tree
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
```

2.Import Datasets

In [40]:
```
data=pd.read_csv('Iris.csv',index_col=0)
data.head()
```

Out[40]:

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	0
2	4.9	3.0	1.4	0.2	0
3	4.7	3.2	1.3	0.2	0
4	4.6	3.1	1.5	0.2	0
5	5.0	3.6	1.4	0.2	0

3.Data Undestanding

In [88]:
```
data.isnull().sum()
```

Out[88]:
```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [87]:
```
data.shape
```

Out[87]:
```
(150, 5)
```

In [41]:
```
le=LabelEncoder()
data['Species']=le.fit_transform(data['Species'])
```

In [42]:
```
data['Species']
```

Out[42]:
```
Id
1      0
2      0
3      0
4      0
5      0
      ..
146    2
147    2
148    2
149    2
150    2
Name: Species, Length: 150, dtype: int64
```

In [43]:
```
x=data.drop(['Species'],axis=1)
y=data['Species']
```

In [44]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,shuffle=True,stratify=y)
```

In [45]:
```
x_train.shape,x_test.shape
```

Out[45]:
```
((105, 4), (45, 4))
```

4.Build Desicion Tree classifier using gini

In [52]:
```
dt=DecisionTreeClassifier(criterion="gini",max_depth=3)
dt.fit(x_train,y_train)
```

Out[52]:
```
DecisionTreeClassifier(max_depth=3)
```

In [53]:
```
y_pred_test=dt.predict(x_test)
```

In [48]:
```
y.value_counts()
```

Out[48]:
```
1    51
0    50
2    49
Name: Species, dtype: int64
```

In [49]:
```
pd.DataFrame({"Actual":y_test,"Predict":y_pred_test})
```

Out[49]:

Id	Actual	Predict
136	2	2
68	1	1
89	1	1
13	0	0
79	1	1
115	2	2
8	0	0
97	1	1
22	0	0
69	1	1
137	2	2
24	0	0
120	2	1
132	2	2
47	0	0
144	2	2
85	1	1
113	2	2
60	1	1
5	0	0
72	1	1
117	2	2
20	0	0
104	2	2
25	0	0
38	0	0
75	1	1
16	0	0
71	1	2
128	2	2
7	0	0
15	0	0
26	0	0
114	2	2
149	2	2
122	2	2
55	1	1
4	0	0
52	1	1
73	1	1
84	1	2
44	0	0
116	2	2
66	1	1
105	2	2

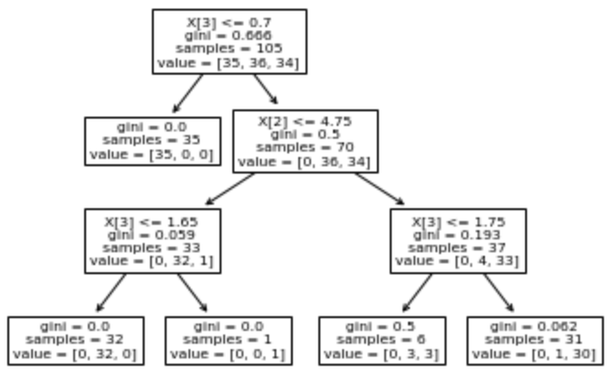
In [70]:
```
accuracy_score(y_pred_test,y_test)
```

Out[70]:
```
0.9555555555555556
```

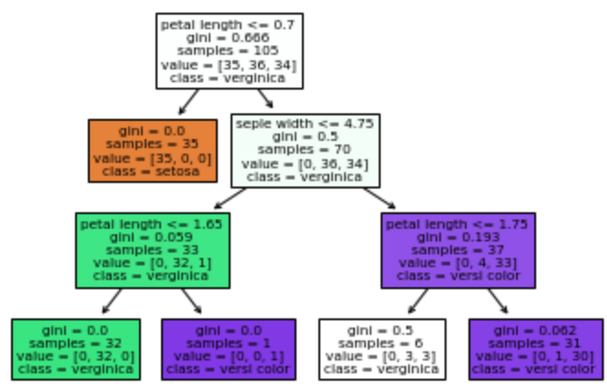
In [71]:
```
print(classification_report(y_test,y_pred_test))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        15
           1       0.93      0.93      0.93        15
           2       0.93      0.93      0.93        15

    accuracy                           0.96        45
   macro avg       0.96      0.96      0.96        45
weighted avg       0.96      0.96      0.96        45
```

In [58]:
```
tree.plot_tree(dt)
plt.show()
```



In [68]:
```
fn=['sepal length',"petal length","seple width","petal length"]
cn=["setosa","verginica","versi color"]
tree.plot_tree(dt,class_names=cn,feature_names=fn,filled=True)
plt.show()
```



5.Build Desicion Tree classifier using entropy

In [76]:
```
model=DecisionTreeClassifier(criterion="entropy",max_depth=3)
```

In [77]:
```
model.fit(x_train,y_train)
```

Out[77]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=3)
```

In [79]:
```
y_test_pr=model.predict(x_test)
y_test_pr
```

Out[79]:
```
array([2, 1, 1, 0, 1, 2, 0, 1, 0, 1, 2, 0, 1, 2, 0, 2, 1, 2, 1, 0, 1, 2,
       0, 2, 0, 0, 1, 0, 2, 2, 0, 0, 0, 2, 2, 2, 1, 0, 1, 1, 1, 0, 2, 1,
       2], dtype=int64)
```

In [80]:
```
accuracy_score(y_test_pr,y_test)
```

Out[80]:
```
0.9555555555555556
```

In [81]:
```
pd.crosstab(y_test,y_pred_test)
```

Out[81]:

col_0	0	1	2
Species			
0	15	0	0
1	0	14	1
2	0	1	14

In [84]:
```
np.mean(y_pred_test==y_test)
```

Out[84]:
```
0.9555555555555556
```

In [85]:
```
y_test[127:]
```

Out[85]:
```
Series([], Name: Species, dtype: int64)
```