Decision Tree classification

project : Iris flower species

1)Import required libraries

In [ ]:
```
import numpy as np
import pandas as pd
df=pd.read_csv('/content/Iris.csv')
df
```

Out[ ]:

	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
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [ ]:
```
df.head()
```

Out[ ]:

	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 [ ]:
```
df.tail()
```

Out[ ]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

In [ ]:
```
df.dtypes
```

Out[ ]:
```
Id                int64
SepalLengthCm    float64
SepalWidthCm     float64
PetalLengthCm    float64
PetalWidthCm     float64
Species           object
dtype: object
```

In [ ]:
```
df.isna().sum()
```

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

In [ ]:
```
df=df.drop(['Id'],axis=1)
df
```

Out[ ]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [ ]:
```
x=df.iloc[:,:-1].values
y=df.iloc[:,-1].values
```

2)Split the data into Training & Testing data

In [ ]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
x_train
```

Out[ ]:
```
array([[5.5, 2.4, 3.7, 1. ],
       [6.3, 2.8, 5.1, 1.5],
       [6.4, 3.1, 5.5, 1.8],
       [6.6, 3. , 4.4, 1.4],
       [7.2, 3.6, 6.1, 2.5],
       [5.7, 2.9, 4.2, 1.3],
       [7.6, 3. , 6.6, 2.1],
       [5.6, 3. , 4.5, 1.5],
       [5.1, 3.5, 1.4, 0.2],
       [7.7, 2.8, 6.7, 2. ],
       [5.8, 2.7, 4.1, 1. ],
       [5.2, 3.4, 1.4, 0.2],
       [5. , 3.5, 1.3, 0.3],
       [5.1, 3.8, 1.9, 0.4],
       [5. , 2. , 3.5, 1. ],
       [6.3, 2.7, 4.9, 1.8],
       [4.8, 3.4, 1.9, 0.2],
       [5. , 3. , 1.6, 0.2],
       [5.1, 3.3, 1.7, 0.5],
       [5.6, 2.7, 4.2, 1.3],
       [5.1, 3.4, 1.5, 0.2],
       [5.7, 3. , 4.2, 1.2],
       [7.7, 3.8, 6.7, 2.2],
       [4.6, 3.2, 1.4, 0.2],
       [6.2, 2.9, 4.3, 1.3],
       [5.7, 2.5, 5. , 2. ],
       [5.5, 4.2, 1.4, 0.2],
       [6. , 3. , 4.8, 1.8],
       [5.8, 2.7, 5.1, 1.9],
       [6. , 2.2, 4. , 1. ],
       [5.4, 3. , 4.5, 1.5],
       [6.2, 3.4, 5.4, 2.3],
       [5.5, 2.3, 4. , 1.3],
       [5.4, 3.9, 1.7, 0.4],
       [5. , 2.3, 3.3, 1. ],
       [6.4, 2.7, 5.3, 1.9],
       [5. , 3.3, 1.4, 0.2],
       [5. , 3.2, 1.2, 0.2],
       [5.5, 2.4, 3.8, 1.1],
       [6.7, 3. , 5. , 1.7],
       [4.9, 3.1, 1.5, 0.1],
       [5.8, 2.8, 5.1, 2.4],
       [5. , 3.4, 1.5, 0.2],
       [5. , 3.5, 1.6, 0.6],
       [5.9, 3.2, 4.8, 1.8],
       [5.1, 2.5, 3. , 1.1],
       [6.9, 3.2, 5.7, 2.3],
       [6. , 2.7, 5.1, 1.6],
       [6.1, 2.6, 5.6, 1.4],
       [7.7, 3. , 6.1, 2.3],
       [5.5, 2.5, 4. , 1.3],
       [4.4, 2.9, 1.4, 0.2],
       [4.3, 3. , 1.1, 0.1],
       [6. , 2.2, 5. , 1.5],
       [7.2, 3.2, 6. , 1.8],
       [4.6, 3.1, 1.5, 0.2],
       [5.1, 3.5, 1.4, 0.3],
       [4.4, 3. , 1.3, 0.2],
       [6.3, 2.5, 4.9, 1.5],
       [6.3, 3.4, 5.6, 2.4],
```

```
[4.6, 3.4, 1.4, 0.3],
[6.8, 3. , 5.5, 2.1],
[6.3, 3.3, 6. , 2.5],
[4.7, 3.2, 1.3, 0.2],
[6.1, 2.9, 4.7, 1.4],
[6.5, 2.8, 4.6, 1.5],
[6.2, 2.8, 4.8, 1.8],
[7. , 3.2, 4.7, 1.4],
[6.4, 3.2, 5.3, 2.3],
[5.1, 3.8, 1.6, 0.2],
[6.9, 3.1, 5.4, 2.1],
[5.9, 3. , 4.2, 1.5],
[6.5, 3. , 5.2, 2. ],
[5.7, 2.6, 3.5, 1. ],
[5.2, 2.7, 3.9, 1.4],
[6.1, 3. , 4.6, 1.4],
[4.5, 2.3, 1.3, 0.3],
[6.6, 2.9, 4.6, 1.3],
[5.5, 2.6, 4.4, 1.2],
[5.3, 3.7, 1.5, 0.2],
[5.6, 3. , 4.1, 1.3],
[7.3, 2.9, 6.3, 1.8],
[6.7, 3.3, 5.7, 2.1],
[5.1, 3.7, 1.5, 0.4],
[4.9, 2.4, 3.3, 1. ],
[6.7, 3.3, 5.7, 2.5],
[7.2, 3. , 5.8, 1.6],
[4.9, 3.1, 1.5, 0.1],
[6.7, 3.1, 5.6, 2.4],
[4.9, 3. , 1.4, 0.2],
[6.9, 3.1, 4.9, 1.5],
[7.4, 2.8, 6.1, 1.9],
[6.3, 2.9, 5.6, 1.8],
[5.7, 2.8, 4.1, 1.3],
[6.5, 3. , 5.5, 1.8],
[6.3, 2.3, 4.4, 1.3],
[6.4, 2.9, 4.3, 1.3],
[5.6, 2.8, 4.9, 2. ],
[5.9, 3. , 5.1, 1.8],
[5.4, 3.4, 1.7, 0.2],
[6.1, 2.8, 4. , 1.3],
[4.9, 2.5, 4.5, 1.7],
[5.8, 4. , 1.2, 0.2],
[5.8, 2.6, 4. , 1.2],
[7.1, 3. , 5.9, 2.1]])
```

In [ ]:

```
x_test
print(y_train)
y_test
```

```
['Iris-versicolor' 'Iris-virginica' 'Iris-virginica' 'Iris-versicolor'
 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor'
 'Iris-virginica' 'Iris-setosa' 'Iris-virginica' 'Iris-virginica'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica'
 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica'
 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-virginica'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor'
 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica']
```

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

3)Data Normalization

In [ ]:
```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
scaler.fit(x_train)
x_test=scaler.transform(x_test)
x_train=scaler.transform(x_train)
```

4)Model Creation using Decision Tree

In [ ]:
```
from sklearn.tree import DecisionTreeClassifier
dec=DecisionTreeClassifier(criterion='entropy')
dec.fit(x_train,y_train)
y_pred=dec.predict(x_test)
y_pred
```

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

5)Performance Evaluation

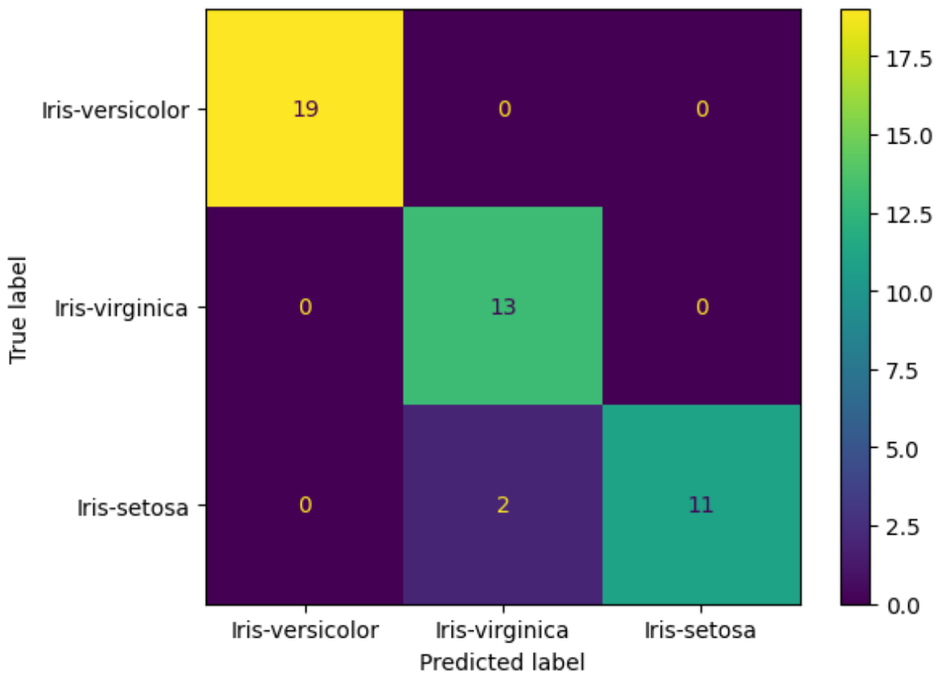
In [ ]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
matr=confusion_matrix(y_test,y_pred)
print(matr)
score=accuracy_score(y_test,y_pred)
score
```

```
[[19  0  0]
 [ 0 13  0]
 [ 0  2 11]]
```

Out[ ]:
```
0.9555555555555556
```

In [ ]:
```
from sklearn.metrics import ConfusionMatrixDisplay
labels=['Iris-versicolor','Iris-virginica','Iris-setosa']
cmd=ConfusionMatrixDisplay(matr,display_labels=labels)
cmd.plot()
```

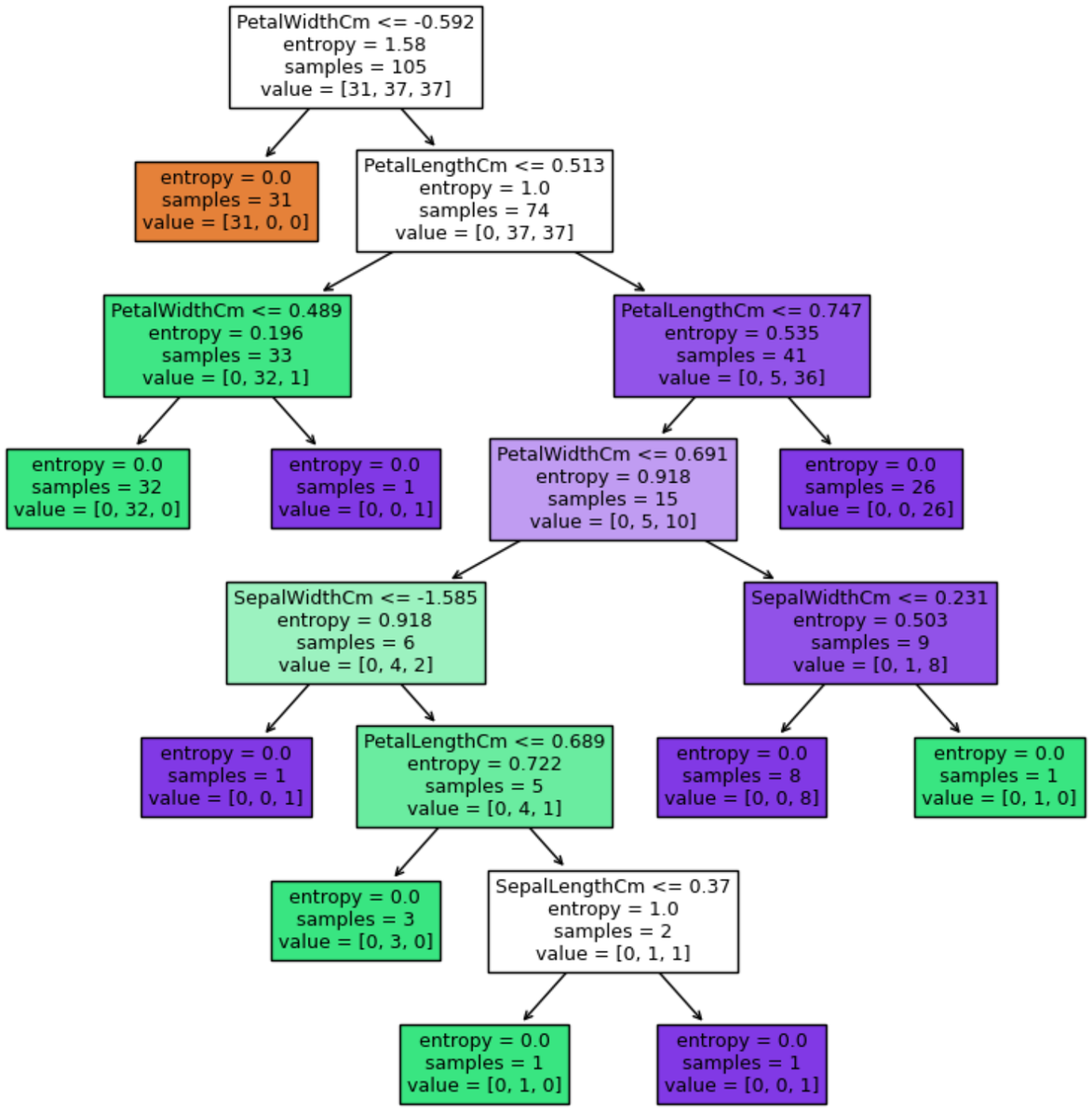
Out[ ]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7e204adc3430>
```



6)Plot Tree

In [ ]:
```
from sklearn import tree
import matplotlib.pyplot as plt
plt.figure(figsize=(10,10))
tree.plot_tree(dec,feature_names=['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm'],filled=True)
```

Out[ ]:
```
[Text(0.3333333333333333, 0.9375, 'PetalWidthCm <= -0.592\nentropy = 1.58\nsamples = 105\nvalue = [31, 37, 37]'),
 Text(0.2222222222222222, 0.8125, 'entropy = 0.0\nsamples = 31\nvalue = [31, 0, 0]'),
 Text(0.4444444444444444, 0.8125, 'PetalLengthCm <= 0.513\nentropy = 1.0\nsamples = 74\nvalue = [0, 37, 37]'),
 Text(0.2222222222222222, 0.6875, 'PetalWidthCm <= 0.489\nentropy = 0.196\nsamples = 33\nvalue = [0, 32, 1]'),
 Text(0.1111111111111111, 0.5625, 'entropy = 0.0\nsamples = 32\nvalue = [0, 32, 0]'),
 Text(0.3333333333333333, 0.5625, 'entropy = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(0.6666666666666666, 0.6875, 'PetalLengthCm <= 0.747\nentropy = 0.535\nsamples = 41\nvalue = [0, 5, 36]'),
 Text(0.5555555555555556, 0.5625, 'PetalWidthCm <= 0.691\nentropy = 0.918\nsamples = 15\nvalue = [0, 5, 10]'),
 Text(0.3333333333333333, 0.4375, 'SepalWidthCm <= -1.585\nentropy = 0.918\nsamples = 6\nvalue = [0, 4, 2]'),
 Text(0.2222222222222222, 0.3125, 'entropy = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(0.4444444444444444, 0.3125, 'PetalLengthCm <= 0.689\nentropy = 0.722\nsamples = 5\nvalue = [0, 4, 1]'),
 Text(0.3333333333333333, 0.1875, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3, 0]'),
 Text(0.5555555555555556, 0.1875, 'SepalLengthCm <= 0.37\nentropy = 1.0\nsamples = 2\nvalue = [0, 1, 1]'),
 Text(0.4444444444444444, 0.0625, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),
 Text(0.6666666666666666, 0.0625, 'entropy = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(0.7777777777777778, 0.4375, 'SepalWidthCm <= 0.231\nentropy = 0.503\nsamples = 9\nvalue = [0, 1, 8]'),
 Text(0.6666666666666666, 0.3125, 'entropy = 0.0\nsamples = 8\nvalue = [0, 0, 8]'),
 Text(0.8888888888888888, 0.3125, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),
 Text(0.7777777777777778, 0.5625, 'entropy = 0.0\nsamples = 26\nvalue = [0, 0, 26]')]
```


PetalWidthCm <= -0.592
entropy = 1.58
samples = 105
value = [31, 37, 37]
entropy = 0.0
samples = 31
value = [31, 0, 0]
PetalLengthCm <= 0.513
entropy = 1.0
samples = 74
value = [0, 37, 37]
PetalWidthCm <= 0.489
entropy = 0.196
samples = 33
value = [0, 32, 1]
PetalLengthCm <= 0.747
entropy = 0.535
samples = 41
value = [0, 5, 36]
entropy = 0.0
samples = 32
value = [0, 32, 0]
entropy = 0.0
samples = 1
value = [0, 0, 1]
PetalWidthCm <= 0.691
entropy = 0.918
samples = 15
value = [0, 5, 10]
entropy = 0.0
samples = 26
value = [0, 0, 26]
SepalWidthCm <= -1.585
entropy = 0.918
samples = 6
value = [0, 4, 2]
SepalWidthCm <= 0.231
entropy = 0.503
samples = 9
value = [0, 1, 8]
entropy = 0.0
samples = 1
value = [0, 0, 1]
PetalLengthCm <= 0.689
entropy = 0.722
samples = 5
value = [0, 4, 1]
entropy = 0.0
samples = 8
value = [0, 0, 8]
entropy = 0.0
samples = 1
value = [0, 1, 0]
entropy = 0.0
samples = 3
value = [0, 3, 0]
SepalLengthCm <= 0.37
entropy = 1.0
samples = 2
value = [0, 1, 1]
entropy = 0.0
samples = 1
value = [0, 1, 0]
entropy = 0.0
samples = 1
value = [0, 0, 1]

In [ ]:

In [ ]: