Subject : AML

Subject Code : 3CS1111

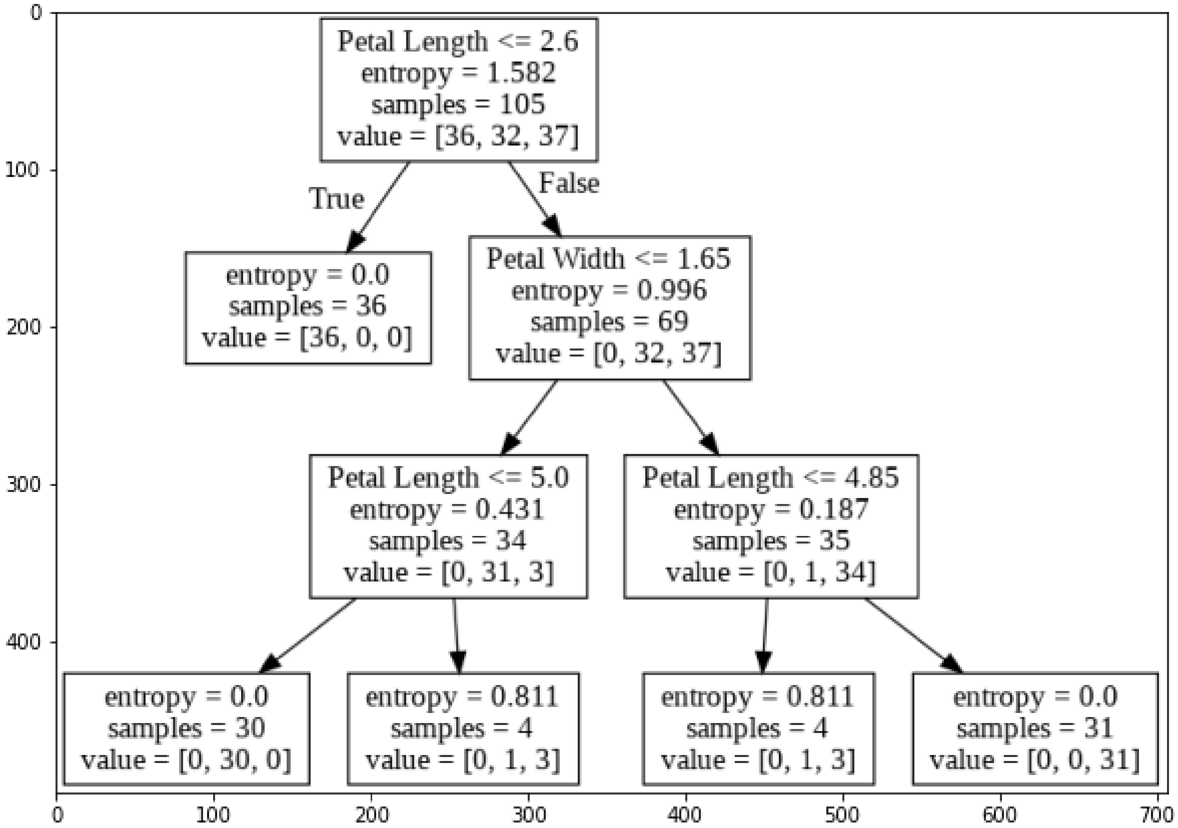
Roll No : 20MCED08

In [ ]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import classification_report,confusion_matrix


from sklearn.datasets import load_iris


FEATURE_NAMES = ['Sepal Length', 'Sepal Width', 'Petal Length', 'Petal Width']

iris = load_iris()
X = pd.DataFrame(iris.data, columns = FEATURE_NAMES)
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.3, random
_state=1)
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classif
ier

# Create Decision Tree classifer object
clf = DecisionTreeClassifier(criterion="entropy", max_depth= 3)
clf = clf.fit(X_train,y_train)

from sklearn.tree import export_graphviz
export_graphviz(clf, 'tree.dot', feature_names = FEATURE_NAMES)
! dot -Tpng tree.dot -o tree.png

#Predict the response for test dataset
y_pred = clf.predict(X_test)
# Model Accuracy, how often is the classifier correct?

print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

import matplotlib.pyplot as plt
import cv2
%matplotlib inline
img = cv2.imread('tree.png')
plt.figure(figsize = (10, 10))
plt.imshow(img)
```

```
Accuracy: 0.9555555555555556
[[14  0  0]
 [ 0 17  1]
 [ 0  1 12]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        14
           1       0.94      0.94      0.94        18
           2       0.92      0.92      0.92        13

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

```
Out[ ]: <matplotlib.image.AxesImage at 0x7f6d9fe0e5c0>
```

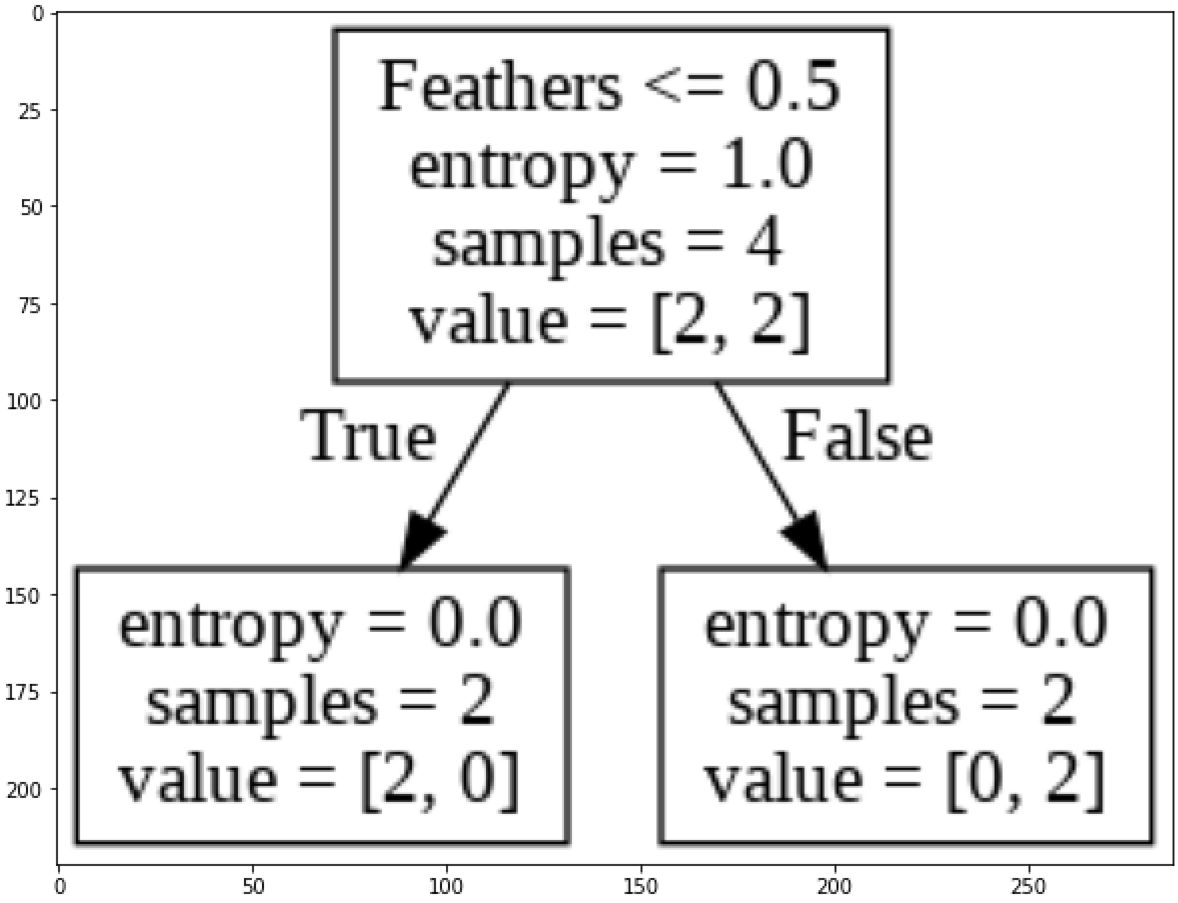


In [ ]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import classification_report,confusion_matrix


df = pd.read_csv('animal.csv')
dummies = pd.get_dummies(df.Animal)
df = pd.concat([df,dummies],axis='columns')
df = df.replace('Yes',1)
df = df.replace('No',0)
#df["Animal_Name"] = df["Animal"].cat.codes
X = df.drop(['Class','Animal'],axis='columns')
FEATURE_NAMES = X.columns
y = df['Class']


X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.3, random
_state=1)
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classif
ier

# Create Decision Tree classifer object
clf = DecisionTreeClassifier(criterion="entropy", max_depth= 3)
clf = clf.fit(X_train,y_train)

from sklearn.tree import export_graphviz
export_graphviz(clf, 'animal.dot', feature_names = FEATURE_NAMES)
! dot -Tpng animal.dot -o animal.png

#Predict the response for test dataset
y_pred = clf.predict(X_test)
# Model Accuracy, how often is the classifier correct?

print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

import matplotlib.pyplot as plt
import cv2
%matplotlib inline
img = cv2.imread('animal.png')
plt.figure(figsize = (10, 10))
plt.imshow(img)
```

```
Accuracy: 0.5
[[0 0]
 [1 1]]
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       1.00      0.50      0.67         2

    accuracy                           0.50         2
   macro avg       0.50      0.25      0.33         2
weighted avg       1.00      0.50      0.67         2


/usr/local/lib/python3.6/dist-packages/sklearn/metrics/_classification.py:127
2: UndefinedMetricWarning: Recall and F-score are ill-defined and being set t
o 0.0 in labels with no true samples. Use `zero_division` parameter to contro
l this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

```
Out[ ]: <matplotlib.image.AxesImage at 0x7f6dbcf09080>
```


Feathers <= 0.5
entropy = 1.0
samples = 4
value = [2, 2]
True
False
entropy = 0.0
samples = 2
value = [2, 0]
entropy = 0.0
samples = 2
value = [0, 2]
0
25
50
75
100
125
150
175
200
0
50
100
150
200
250