CPS 844 Lab 3: Decision tree classifier

02/16/2022
Section 1
Ryan Soliven
500840952

 # Import the packages

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

 # 1) (5 points) Read the vertebrate.csv data

```
data = pd.read_csv('D:\Downloads\CPS844\\vertebrate.csv', header = None)
```

 # 2) (15 points) The number of records is limited. Convert the data into a binary classification: mammals versus non-mammals

```
data = data.replace(['fishes','birds','amphibians','reptiles'], 'non-mammals')
```

 # 3) (15 points) We want to classify animals based on the attributes: Warm-blooded,Gives Birth,Aquatic Creature,Aerial Creature,Has Legs,Hibernates
 # For training, keep only the attributes of interest, and seperate the target class from the class attributes

```
classData = data.iloc[1:,[7]]
attributeData = data.iloc[1:,1:7]
```

 # 4) (10 points) Create a decision tree classifier object. The impurity measure should be based on entropy. Constrain the generated tree with a maximum depth of 3

```
clf = tree.DecisionTreeClassifier(criterion = 'entropy', max_depth=3)
```

 # 5) (10 points) Train the classifier

```
clf = clf.fit(attributeData, classData)
```

 # 6) (25 points) Suppose we have the following data

```
testData = [['lizard',0,0,0,0,1,1,'non-mammals'],
        ['monotreme',1,0,0,0,1,1,'mammals'],
        ['dove',1,0,0,1,1,0,'non-mammals'],
        ['whale',1,1,1,0,0,0,'mammals']]
testData = pd.DataFrame(testData, columns=data.columns)
```

 # Prepare the test data and apply the decision tree to classify the test records.
 # Extract the class attributes and target class from 'testData'

```
classTest = testData.iloc[:,[7]]
dataTest = testData.iloc[:,1:7]
predC = clf.predict(dataTest)
```

 # 7) (10 points) Compute and print out the accuracy of the classifier on 'testData'

```
print('The accuracy of the classifier is', accuracy_score(classTest, predC))
```

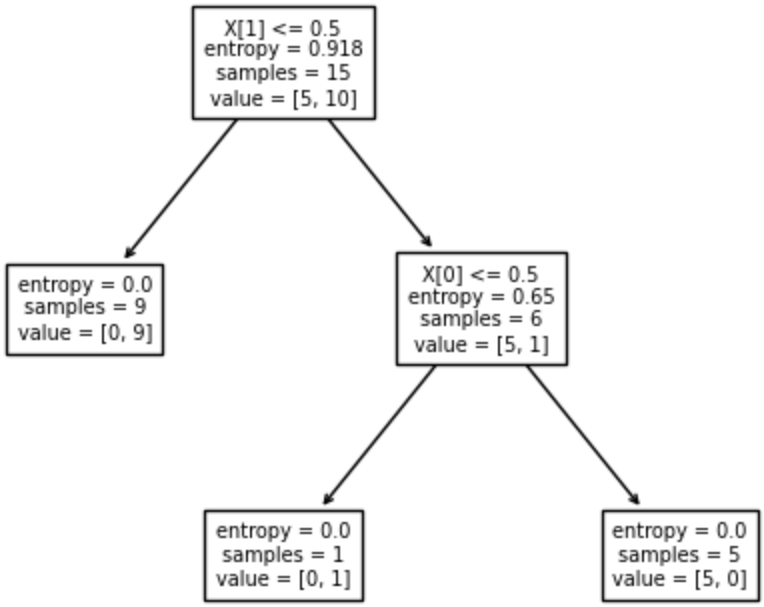
Result

The accuracy of the classifier is 0.75

 # 8) (10 points) Plot your decision tree

```
tree.plot_tree(clf, fontsize = 7)
plt.show()
```

Result


X[1] <= 0.5
entropy = 0.918
samples = 15
value = [5, 10]
entropy = 0.0
samples = 9
value = [0, 9]
X[0] <= 0.5
entropy = 0.65
samples = 6
value = [5, 1]
entropy = 0.0
samples = 1
value = [0, 1]
entropy = 0.0
samples = 5
value = [5, 0]