Name: Amit Das

Reg. No.: 21BAI1543

Lab Assignment 7

Date: 12-06-2023

Q1. Consider Play Tennis Dataset (see attached csv file) that is used for predicting whether a tennis game is played in the given weather conditions or not. Here the weather conditions are described by features outlook, temperature, humidity, play and wind. The target is play with two class labels Yes and No.

a) Compute information gain for all the attributes and display them.

```
Outlook gain     = 0.24674981977443933
Temperature gain = 0.02922256565895487
Humidity gain    = 0.15183550136234159
Wind gain        = 0.04812703040826949
```

b) Find which attribute will become the root node of the decision tree.

From above results gain of outlook is the maximum. Therefore, Outlook should be the root node

c) Scikit DecisionTreeClassifier:

i) Train using DecisionTreeClassifier using tennis dataset:

Train the model

```
: classifier.fit(x_encoded, y_encoded)
```

```
:
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=2)
```

ii) Classify the test sample <Rain, Cool, High, Weak>.

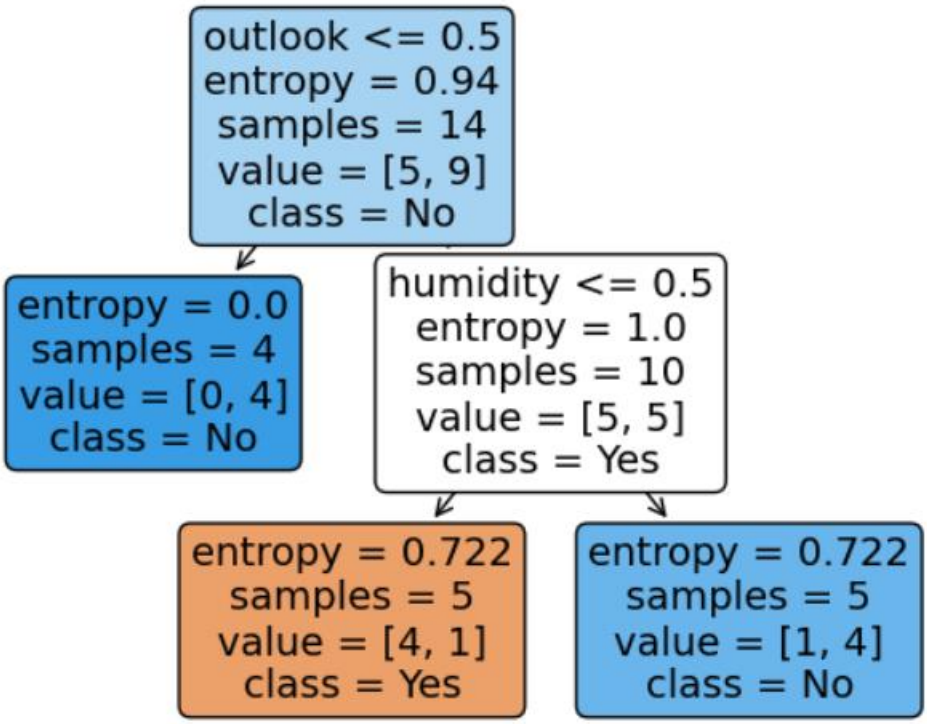
Get the prediction

```
prediction = classifier.predict([[1, 0, 0, 1]])
```

```
if (prediction[0] == 0):
    print("No, play will not be there")
else:
    print("Yes, play will be there")
```

```
No, play will not be there
```

iii) Draw the decision tree for a max depth 2.



d) Check whether root node that you identified is same as the one returned by Scikit DecisionTreeClassifier.

Therefore, we can see that Outlook is the Root node for the Decision Tree as we got while manually training the model.

Q2. Predict heart disease:

a) Apply decision tree classifier on heart disease dataset and report the accuracy.

```
calculate_accuracy(y_test, prediction)
```

```
Accuracy = 74.64622641509435 %
```

Therefore, using Entropy for gain calculation, we got accuracy of around 76%

b) Try to change the following hyperparameters of the decision tree model and report the change in the accuracy:

i) Splitting Criterion from entropy to gini.

```
calculate_accuracy(y_test, prediction)
```

```
Accuracy = 73.23113207547169 %
```

We got an accuracy of around 73% using Gini as the splitting criteria

ii) max_depth

```
calculate_accuracy(y_test, prediction)
```

```
Accuracy = 85.02358490566037 %
```

Amazingly, on increasing the maximum depth the accuracy of our model becomes about 85%

iii) min_samples_leaf

```
calculate_accuracy(y_test, prediction)
```

```
Accuracy = 82.66509433962264 %
```

On making the mininum number of leaf nodes = 20, we notice that our accuracy has become around 83%