Exercise 7
Classification

CLASSIFICATION

Often in data mining, one randomly divides the training data into three subsets called the *training*, *tuning*, and *testing* sets. As seen in the course the training set is used to construct the decision tree and the testing set to evaluate the accuracy. The tuning set is used to avoid over-fitting, as described below.

Assume a sample of data which categorizes a DRINK as YES or NO, based on whether people with certain AGE and GENDER would drink it or not. The attributes can take the following values.

```
DRINK  possible values: Chocolate, Egg Milk, Strawberry Juice, Tea

AGE    possible values: <=25, >25

GENDER possible values: Male, Female
```

(Since each feature value starts with a different letter, for shorthand we'll just use that initial letter, e.g., 'C' for Chocolate)

The category labels are marked as YES or NO.

Here is our **TRAINING** set:

```
DRINK = C AGE <= 25 GENDER = M CATEGORY = NO
DRINK = E AGE > 25 GENDER = M CATEGORY = NO
DRINK = C AGE <= 25 GENDER = F CATEGORY = NO
DRINK = C AGE > 25 GENDER = F CATEGORY = NO
DRINK = S AGE > 25 GENDER = F CATEGORY = YES
DRINK = E AGE <= 25 GENDER = M CATEGORY = YES
DRINK = E AGE <= 25 GENDER = F CATEGORY = YES
```

Our **TUNING** set:

```
DRINK = C AGE > 25 GENDER = M CATEGORY = NO
DRINK = S AGE > 25 GENDER = F CATEGORY = YES
DRINK = E AGE > 25 GENDER = F CATEGORY = YES
DRINK = E AGE <= 25 GENDER = F CATEGORY = NO
DRINK = C AGE <= 25 GENDER = M CATEGORY = NO
```

Our **TESTING** set:

```
DRINK = C AGE > 25 GENDER = F CATEGORY = NO
DRINK = C AGE <= 25 GENDER = M CATEGORY = NO
DRINK = T AGE <= 25 GENDER = F CATEGORY = YES
DRINK = E AGE <= 25 GENDER = M CATEGORY = YES
DRINK = E AGE > 25 GENDER = F CATEGORY = YES
```

With these information answer the following:

Question 1. Inducing the initial decision tree

First, apply the decision-tree algorithm on the TRAINING set.

When more than one attribute turns out to be the best, choose the one whose name appears earliest in alphabetical order (e.g., AGE before DRINK before GENDER). When there is a tie in majority voting, choose “NO” as the categorical label. (Majority voting labels a node with a categorical label value shared by maximum no. of data samples.)

Question 2. Pruning the tree to reduce overfitting

Overfitting occurs when a decision tree conforms too closely to the training data and does not accurately model the underlying concept. One way to address this problem in decision-tree induction is to use a tuning set in conjunction with a pruning algorithm. Here we will use a 'greedy' algorithm sketched in Figure 1 below.

Apply the pruning algorithm to the decision tree produced in (a).

Question 3. Estimating future accuracy

Apply the decision tree produced in (b) to the TESTING data samples and report its accuracy. What is the accuracy of the unpruned tree (produced in (a)) for these data samples. Briefly discuss your results.

Pruning Algorithm: BEGIN

Let *bestTree* be the tree produced by decision tree induction on the TRAINING set.
Let *bestAccuracy* be the accuracy of *bestTree* on the TUNING set. Let *progressMade* = true

```
While (progressMade) // Continue as long as improvement on TUNING SET

{

  progressMade = false; currentTree = bestTree

  For each non-leaf node N  in currentTree   {

      /* consider various pruned versions of the current tree and see if any, is
     better than the best tree found so far */

    (STEP-1) prunedTree = currentTree

     (STEP-2)Replace node N in prunedTree by a leaf node and label it with the
     category label of majority class among TRAINING set examples that reached
     node N (break ties in favor of 'NO')

    (STEP-3) newAccuracy = accuracy of prunedTree  on the TUNING set

      /* is this pruned tree an improvement, based on the TUNE set? In case of a
     tie, go with the smaller tree */

    (STEP-4)If (newAccuracy >= bestAccuracy) {

         bestAccuracy = newAccuracy; bestTree = prunedTree

         progressMade = true

       }

   }

 } return bestTree;
```

Pruning Algorithm: END

Figure 1 Pruning Algorithm for question (b)