Name : Yun Zhou
ID: 300442776

PART 2:

Q1: You should first apply your program to the hepatitis-training and hepatitis-test files and report the classification accuracy in terms of the fraction of the test instances that it classified correctly. Report the constructed decision tree classifier printed by your program. Compare the accuracy of your decision tree program to the baseline classifier (which always predicts the most frequent class in the training set), and comment on any difference.

```
-----------Decision Tree Accuracy result-----------------------------
----------------------------------------
Total number of correct guessed instances: 21.0
Total test instances: 25.0
The accuracy is: 21.0 out of 25.0
Accuracy: 84.00%
----------------------------------------

-----------BaseLine classifier Accuracy result-----------------------------
----------------------------------------
Total number of correct guessed instances: 20.0
Total test instances: 25.0
The accuracy is: 20.0 out of 25.0
Accuracy: 80.00%
----------------------------------------
```

The full log result **including the constructed Decision Tree** can be **seen** in the **Q1_sampleoutput.txt** which is inside the **part2 folder,** also, you can see the tree and accuracy result if you run my program in the terminal.

From the output accuracy result shown above, we can see that, after the decision tree is made by the training, it can get 21 out of 25 correct guessed instances, meanwhile, the baseline classifier get 20 out of 25.
Therefore, I can get that my decision tree algorithm model can perform better prediction on the 25 instances test set. This is not surprise since the baseline classifier is a stupid and simple approach which always predicts the most frequent class in the training set, and the constructed decision tree will predict the result by making the decision up to each attribute value, which means more accurate and smart.

2. You should then apply 10-fold cross-validation to evaluate the robustness of your algorithm. We have provided files for the split training and test sets. The files are named as hepatitis-training-run-*, and hepatitis-test-run-*. Each training set has 107 instances and each test set has the remaining 30 instances. You should train and test your classifier on each pair, and calculate the average accuracy of the classifiers across the 10 folds (show your working).

By following this question instruction, in order to evaluate the robustness of my decision tree algorithm, I go back to add a block of the code, which run the 10-fold cross-validation.

```
/* below for the report q2, 10 fold cross validation */
double totalAccuracy_decisionTree = 0.0;
double times = 0.0;
for (int i = 0; i < 10; i++) {
    String trainPath_10fold = "/Users/11973/git/comp307_a1_yun/comp307_a1_yun/ass1_data/part2/hepatitis-training-run-"
                            + String.valueOf(i);
    String testfilePath_10fold = "/Users/11973/git/comp307_a1_yun/comp307_a1_yun/ass1_data/part2/hepatitis-test-run-"
                               + String.valueOf(i);

    System.out.println("----------------------------------------");
    System.out.println("----------------------------------------");
    System.out
            .println("The training set: \n\t  hepatitis-training-run-" + String.valueOf(i));
    System.out.println("The test set: \n\t  hepatitis-test-run-" + String.valueOf(i));

    DecisionTree dtree_10fold = new DecisionTree();
    dtree_10fold.loadFiles(trainPath_10fold, testfilePath_10fold);
    TreeNode tree_1 = dtree_10fold.buildTree(train_instances, Tool2.categoryNames);
    totalAccuracy_decisionTree += printAccuracyResult(tree_1);
    times++;
    // tree_1.drawTree("");
}
double average_accuracy_decisionTree = totalAccuracy_decisionTree / times;

System.out.println("----------------------------------------");
System.out.println("It runs " + times + " times");
System.out.printf("\nThe average accuracy for the decision tree is  %.2f%%",
        average_accuracy_decisionTree);
System.out.println("\n----------------------------------------");
```

As you can see that, I use the for loop to iterate through each pair of the train/test set, add each of the accuracy results together and finally divided by running times.

```
----------------------------------------
It runs 10.0 times

The average accuracy for the decision tree is 74.00%
----------------------------------------
```

Here you can see the result, as we can see that, the average accuracy of the classifiers across the 10 folds is 74%, which proves the robustness of my decision tree algorithm.
(p.s. The full log can be seen in **Q2_sampleoutput.txt** which is inside the **part2 folder**)

3. “Pruning” (removing) some of leaves of the decision tree will always make the decision tree less accurate on the training set. Explain:

(a) how you could prune leaves from the decision tree;

There are lots of approaches that can prune some of leaves of the decision tree, pruning will result less accurate on the training set, but can run better on testing set which avoid the overfitting.
From the tutorial website below, I can use the approach: error complexity pruning. This approach will generate a series of trees, and each one is made by pruning the full tree by different amounts, and finally select one of these by assessing its performance with an independent data set.

https://alanjeffares.wordpress.com/tutorials/decision-tree/

(b) why it reduces accuracy on the training set;

The pruning can avoid the case of overfitting. In the case of overfitting, the decision tree algorithm is constantly dividing nodes in order to classify the training samples as accurately as possible, which will lead the result of it can run well on training, but not on testing.
Pruning can make sure that the algorithm is not too biased on training, so that it can run better on testing, but it will reduce the accuracy on the training set.

and (c) why it might improve accuracy on the test set.

The purpose of the Pruning is to avoid the case of the overfitting, which means the decision tree runs well on training, but not on testing. The overfitting is caused due to the decision tree algorithm is constantly dividing the nodes in order to classify the training samples as accurately as possible.
During the learning process, this will cause the whole tree to have too many branches, which leads to overfitting.
The pruning can avoid the case of the overfitting, less branch means it’s not too biased on training, which might improve the accuracy on the test set.

4. Explain why the impurity measure (from lectures) is not an appropriate measure to use if there are three or more classes in the dataset.

The impurity measure from the lecture works well on the set that has only 2 classes since 2 class labels are easy to calculate, only pure and impure two options, so that it can get a smooth graph.
However, in the real world, three and more classes are common. If the impurity measure is still in use, the more the classes, the less the accuracy, the graph is not smooth anymore, so that the impurity measure doesn’t make any scene. For more classes, we need to use Gini impurity or the Entropy.