a) Jack Xu and Brandon Chen

b) Role and Assignment

   i) Brandon Chen

      1) Understanding the assignment:

         (a) Finding the relevant powerpoint slides and extracting key criterias

      2) Reading the input data and quantizing it

      3) Implementing the skeleton for the decision tree

      4) Implementing writing the classifier file

      5) Completing part of the writeup

   ii) Jack Xu

      1) Implementing how to calculate Gini Index

      2) Implementing decision tree logic

      3) Implementing how to calculate weighted Gini Index

      4) Completing part of the writeup

c) The maximum call depth of our final classifier was 15 and it did not use the maximum number of levels.

d) Our decision tree consisted of 153 decisions. We noticed that the most important attribute which was utilized the most frequently in our decision tree was age. The second most important attribute was height. An attribute that was only utilized once in our decision tree was whether or not the abominable snowfolk had an earlobe. The rest of the attributes were relatively evenly distributed in how many times they were used to split the data.

e) Confusion Matrix for original training data
The top row is the Actual and the first column is Expected

	Assam	Bhuttan
Assam	1320	60
Bhuttan	34	1316

f) The accuracy of our resulting classifier on the training data is 0.9655677655677656

g) The most difficult part of the homework was figuring out how to split the data and how to implement the weighted gini index properly. We had an issue with the weighted gini index where we had a division by zero when the length of the right split was 0. This occurred when we were looking at the maximum range for the thresholds of any attribute. We solved it by adding a conditional that set the gini index of that dataset to be 0. This practically nullified the effect of the empty set on the weighted gini index. Another difficult part of the homework was figuring out how to write our decision tree to the classifier file. We had issues figuring out how to index our if statements correctly. We ended up solving this problem by tabbing a depth amount of times + 2.

Conclusions:

We discovered that by using the training data to build a decision tree, we were able to classify only the decision tree to a certain degree of accuracy. However, when applied to other sets of data, the decision tree we built may not perform as well as it did on the training data. In order to make the decision tree better, we would need to test on more data and adjust the parameters to our decision tree.

The accuracy paradox states that just because our decision tree is accurate, it does not mean that it is working well. Just because our data is 96% represented by the decision tree, we do not know what is broken. We tested this by creating smaller testing suites to see if the output is what we expected. While doing this, we determined that by adjusting the parameters to the decision tree by even a little, the decisions we make could be completely different.

The data was completely separated into nodes where there were fewer than 3 data points or the data was 95% fitted to either Assam or Bhuttan. There was never a case where the decision tree had to terminate into leaf nodes because we reached the maximum depth of 26 levels.