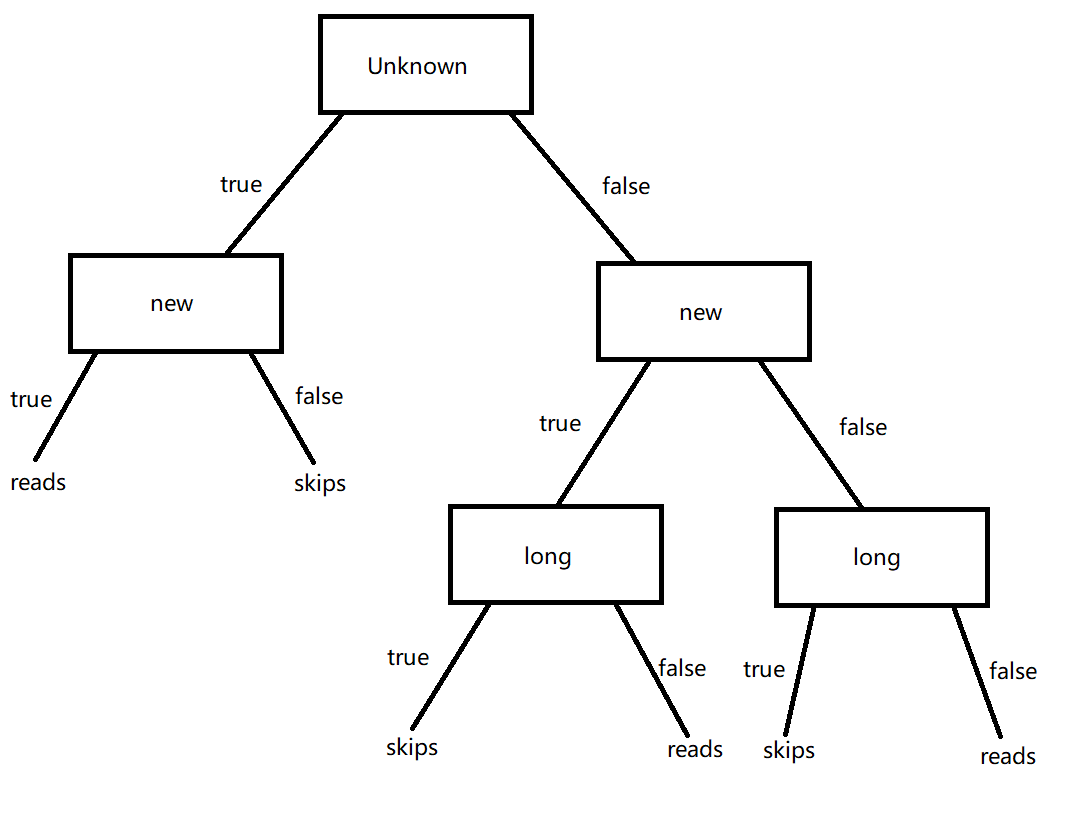
# Question 1: Decision Trees

Before we start, we can simplify the representation of the data by eliminating duplicated data. Then construct a “truth table” to enumerate all the possible data to compare out put of different decision trees. Figure 1.3 clearly shows the comparison among the tree generated by the maximum information gain split and those obtained from question a) and b).

1. Figure1.1.1 is the tree found when the features are in the order [*Author, Thread, Length, WhereRead*]. This tree can be simplified to Figure 1.2. It represents the different function from that found with the maximum information gain split as for any features, these two trees can’t give identical UserActions as output. For example, decision for example <*unknown, new, long, work*> given by this tree is *reads*, but that given by the tree found with the maximum information gain split is *skips*.

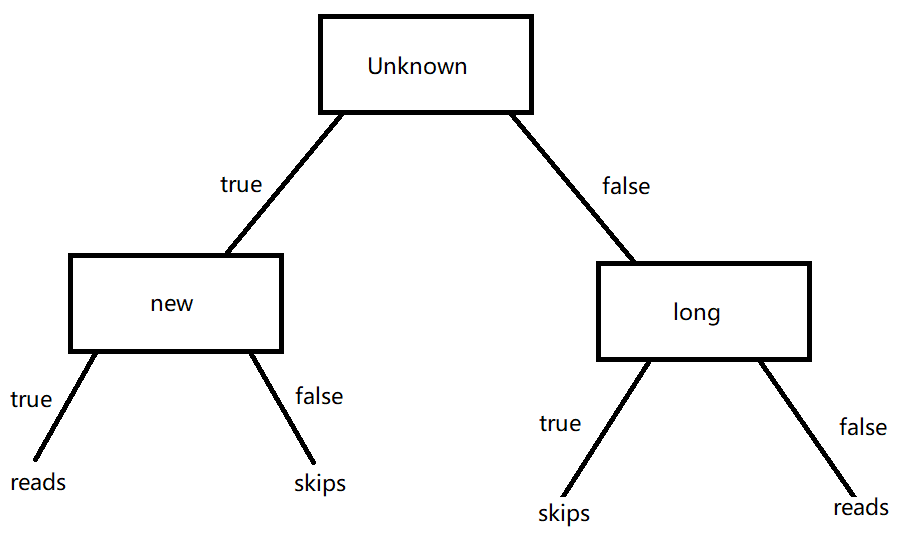
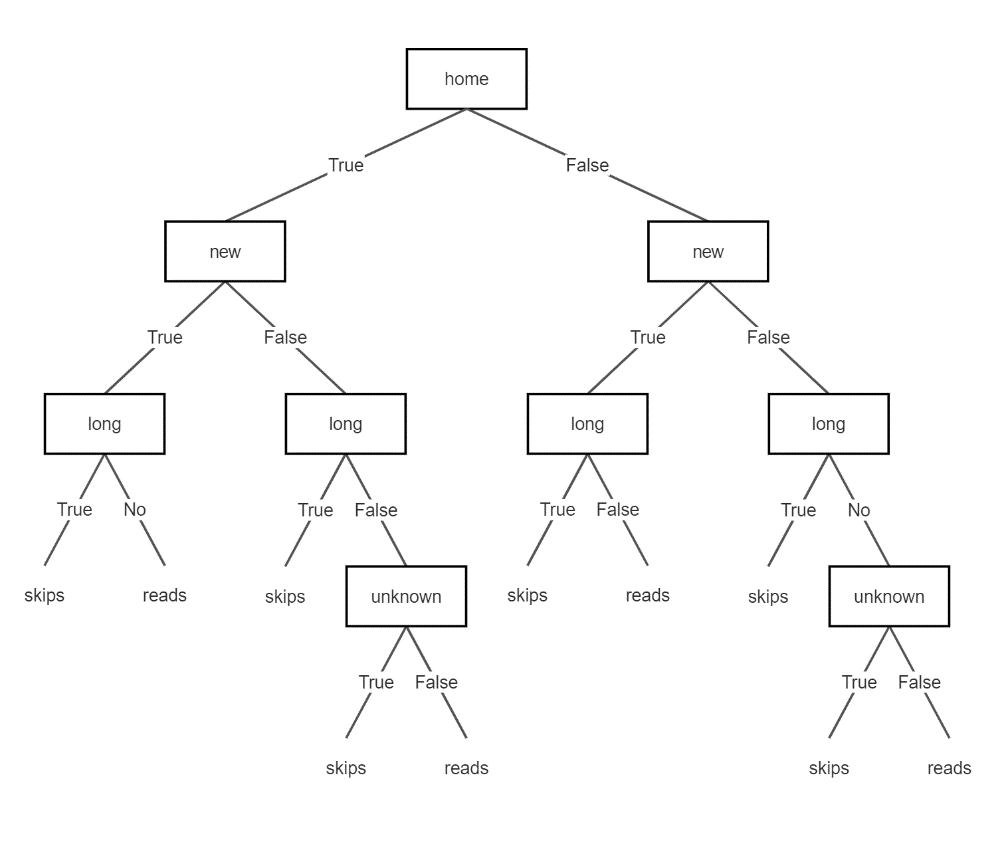
Figure 1.1.1

Figure 1.1.2

1. Figure 1.2.1 is the tree found when the features are in the order [*WhereRead, Thread, Length, Author*]. This tree can be simplified to Figure 1.2.2, which represents the same function as that found with the maximum information gain split because it has identical output according to the “truth table” in Figure 1.3. Hence, it represent a different function with the one given for the preceding part as that function is different from the one found with the maximum information gain split.

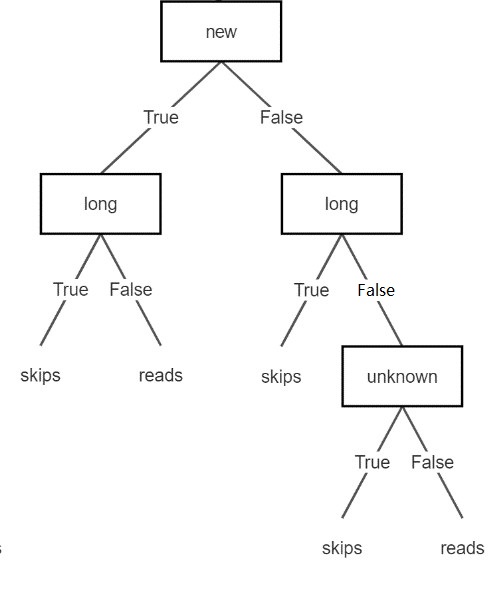
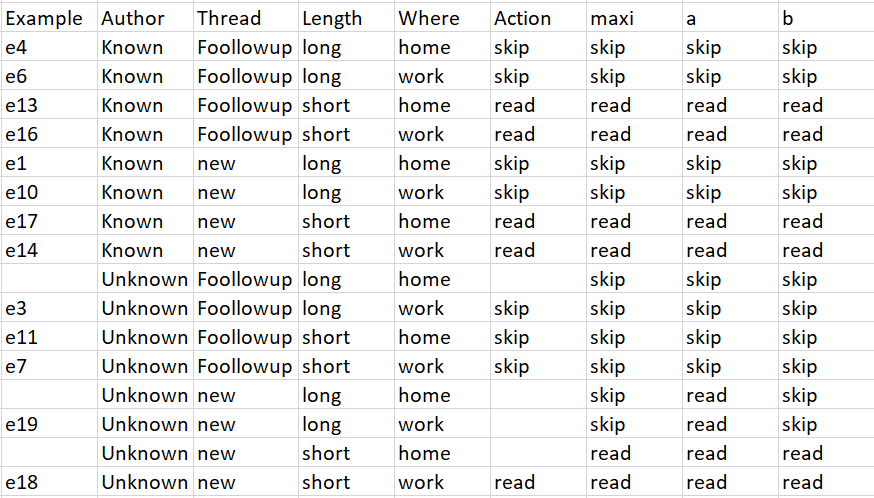
Figure 1.2.1

Figure 1.2.2

Figure 1.3

1. Figure 1.4 contains a tree that correctly classifies the training examples but represents a different function than those found by the preceding examples. It is obtained by adding branches to Figure 1.1.2. It has exactly the same actions as the training examples but gives different output with any tree mentioned before. For example, decision for <*Unknown, new, short, home*> is *skips* but that of any tree found by the preceding examples is *reads*.

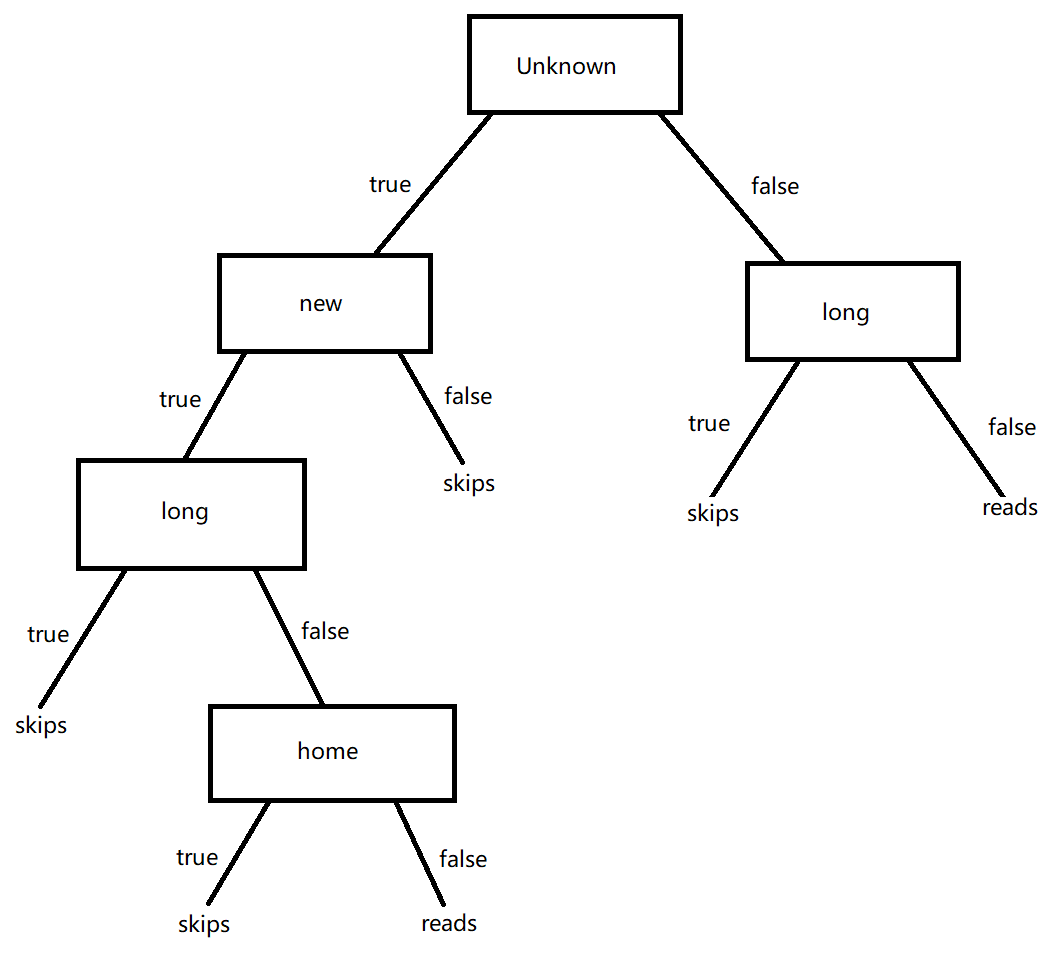


Figure 1.4

# Question 2: Decision Trees

I tried sklearn and Weka as tools and finally choose Weka to show my work as there is a lower possibility to get a wrong model using a GUI program as a beginner and it is more intuitive to show the steps.

Firstly, I prepared the data to train by the following steps:

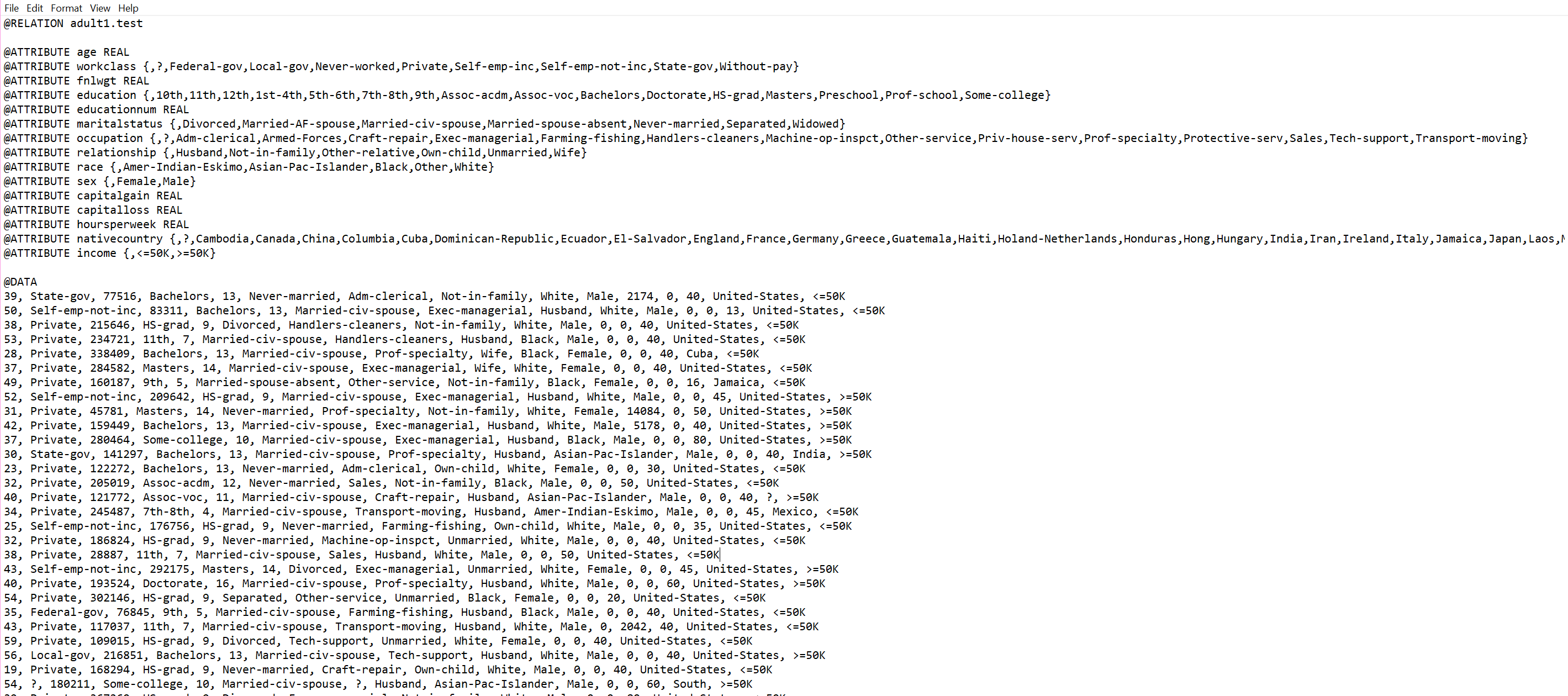
1. Download the data from <http://archive.ics.uci.edu/ml/machine-learning-databases/adult/>
2. Combine .data and .test together according to the requirement of the instruction on WebCMS3.
3. Unify the dataset by eliminating ‘.’ at the end of some rows and Converting ‘>50K’ to ‘>=50K’
4. Add header to each column according to adult.names
5. Convert the file type CSV to ARFF, result is shown in Figure 2.1

Figure 2. 1 First several rows of the data

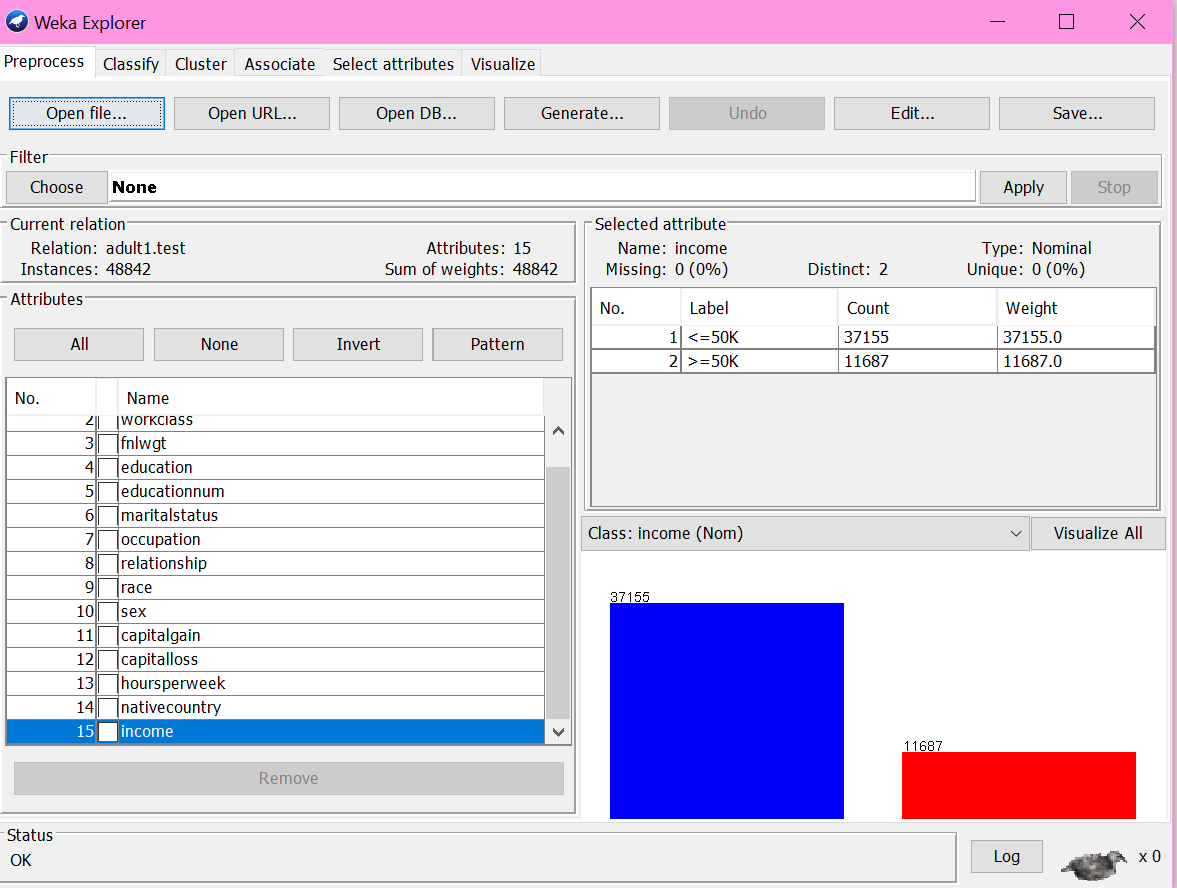
Secondly, I imported the data to Weka, shown in Figure 2.2. Then I chose J48 to classify the dataset with Percentage Split of 66%. The reason why I chose 66% split is that the size of training set is two times larger than that of testing dataset. So, we could use 1x3 cross validation to split the combined file. Figure 2.3 is the result I got with 85% of correctly

Figure 2.2

classified Instances.

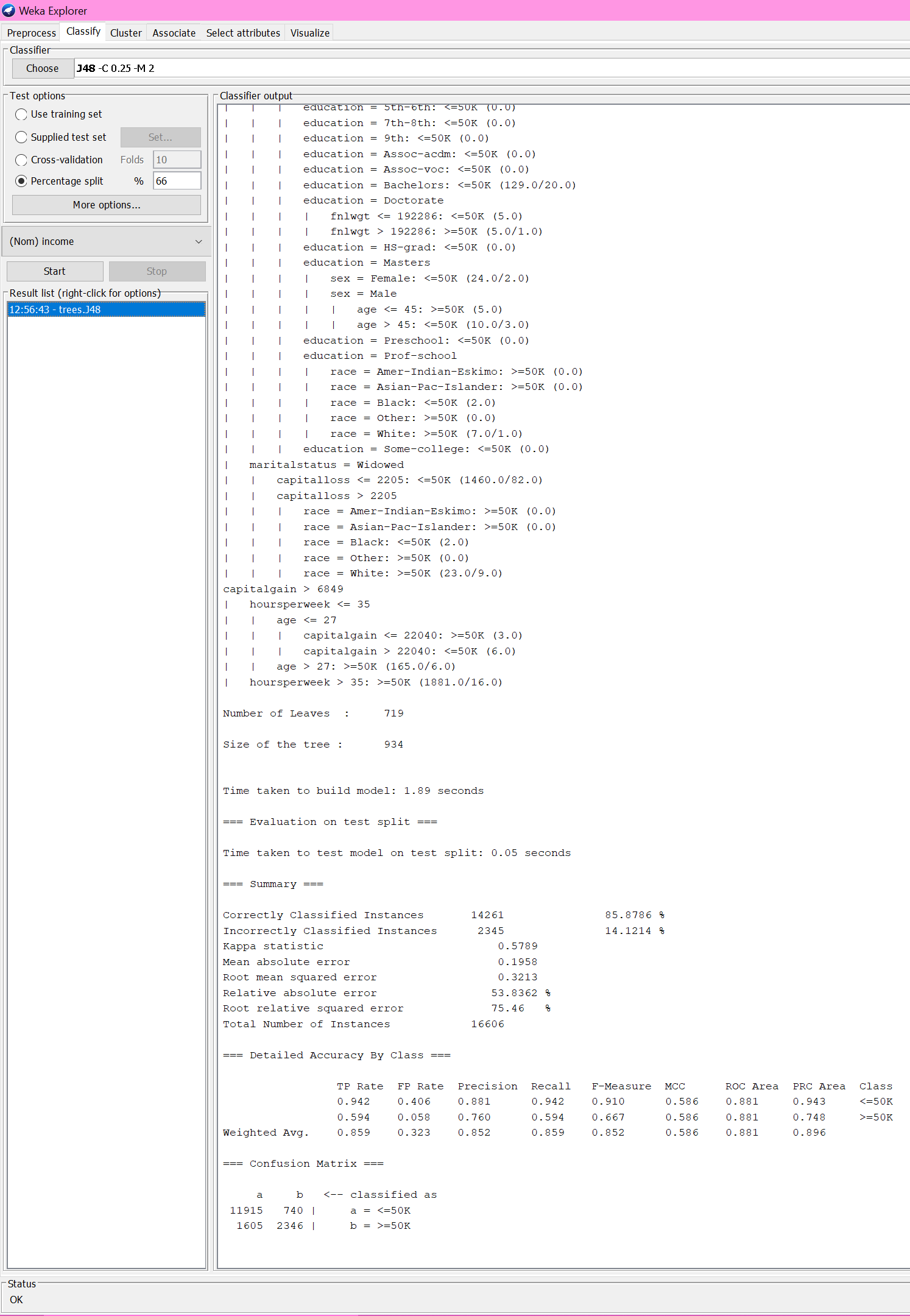


Figure 2.3