Assignment 2 writeup

Santina Lin

Sunday, October 12, 2014

First loading some libraries and my helper file

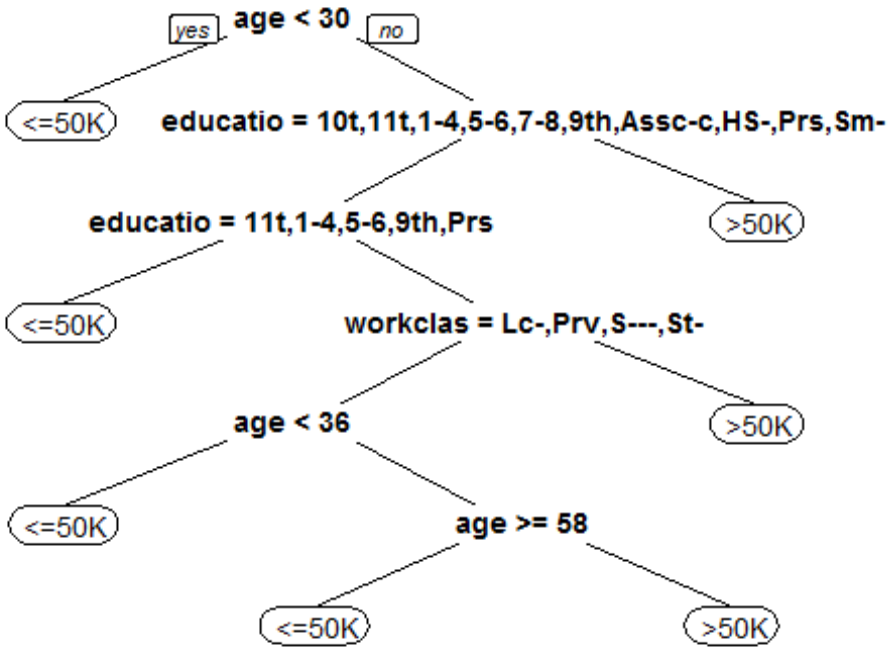
```
library(rpart) #need this for classification, decision tree
library(rpart.plot) # to make pretty tree, thanks to Jonathan Stiansen's
suggestion
library(ggplot2) #for plotting the graph
library(reshape) #for melting dataframe

#for calculating TP and TN and making the confusion matrix more clear
source('./helperFuns_A2_SantinaLin.R')

#===================
trainingData<- read.table("2014CensusTraining.csv", sep = ",", header = TRUE)
training50  <- read.table("2014HalfCensusTraining.csv", sep = ",", header =
TRUE)
testingData <- read.table("2014NewCensusTest.csv", sep = ",", header = TRUE)
```

Show trees with split points

Full data: F5, F10, F14

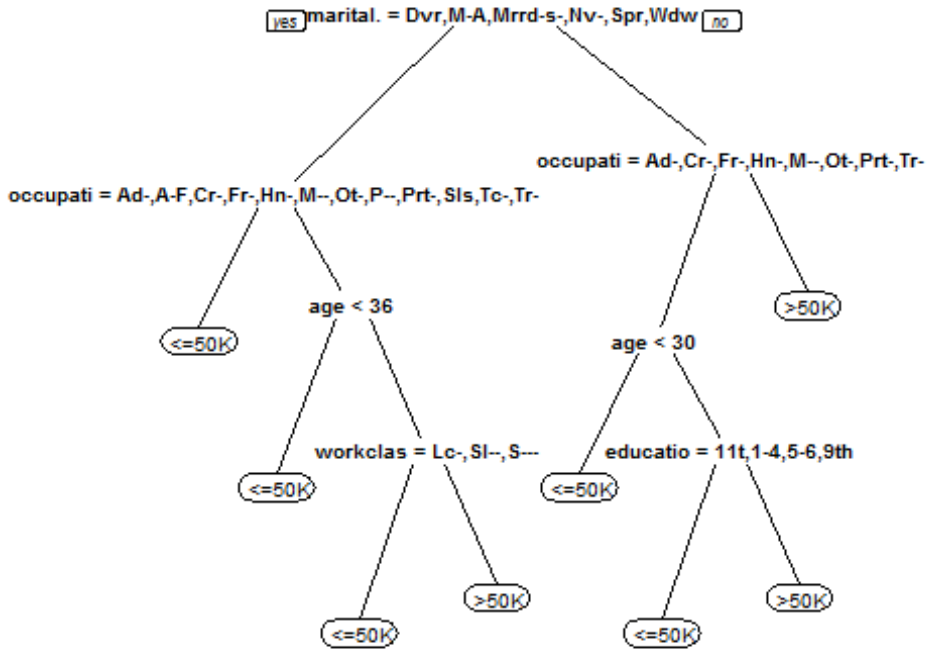
For F5:

```
tree_F5 <- rpart(class ~ workclass + age + fnlwgt + education +
                     education.num,trainingData,method="class")
prp(tree_F5) #plot a pretty tree
```



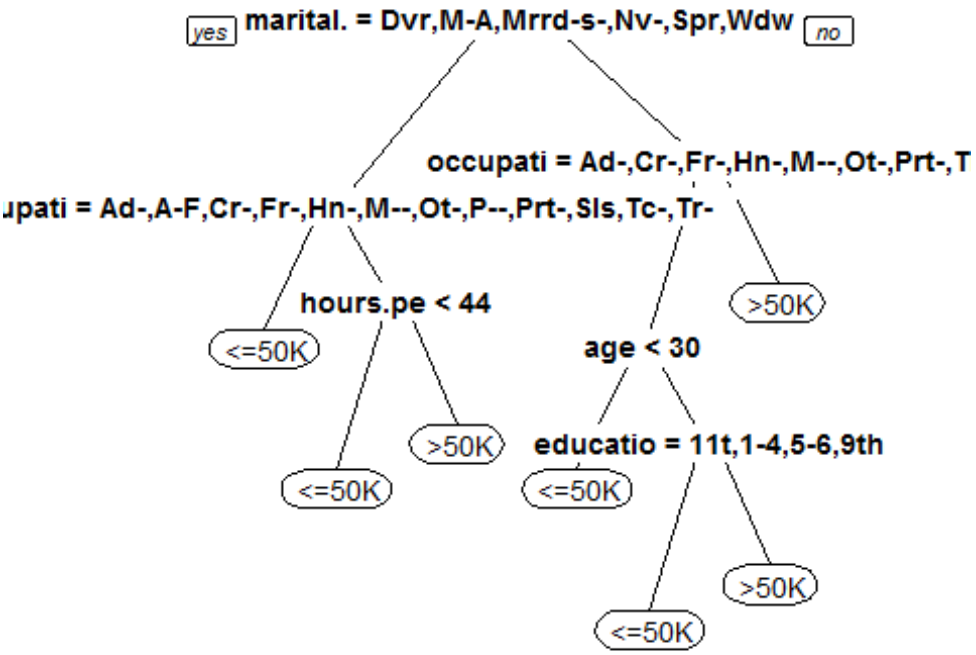
For F10

```
trainingData_F10 <- trainingData[,c(1:10,15)]
#make and plot a pretty tree
tree_F10 <- rpart(class ~ . ,trainingData_F10,method="class")
prp(tree_F10)
```



For F14

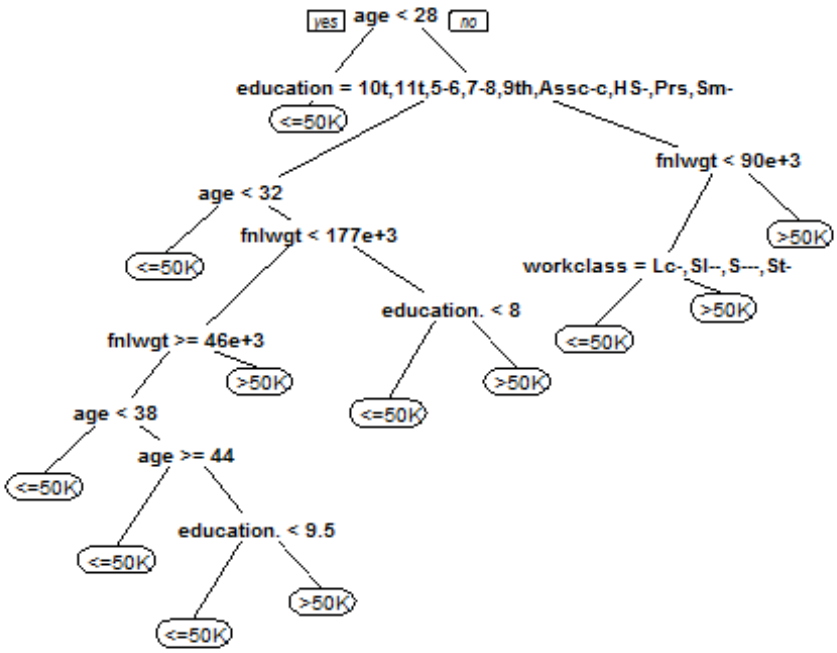
```
tree_F14 <- rpart(class ~ . ,trainingData,method="class")
prp(tree_F14)
```



Half data : H5, H10, H14

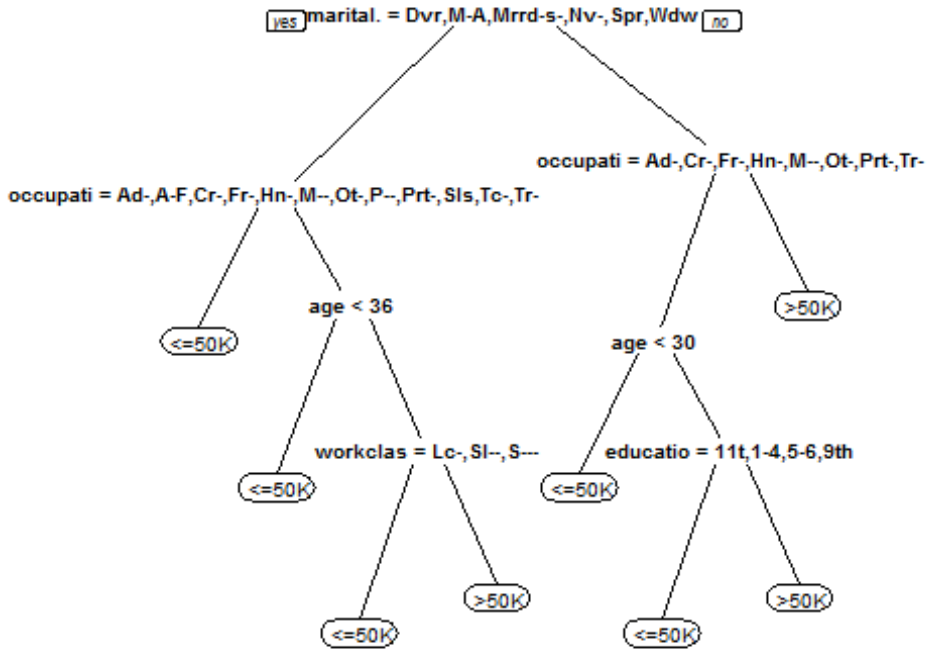
H5:

```
tree_H5 <- rpart(class ~ workclass + age + fnlwgt + education +
                   education.num,training50,method="class")
prp(tree_H5)
```



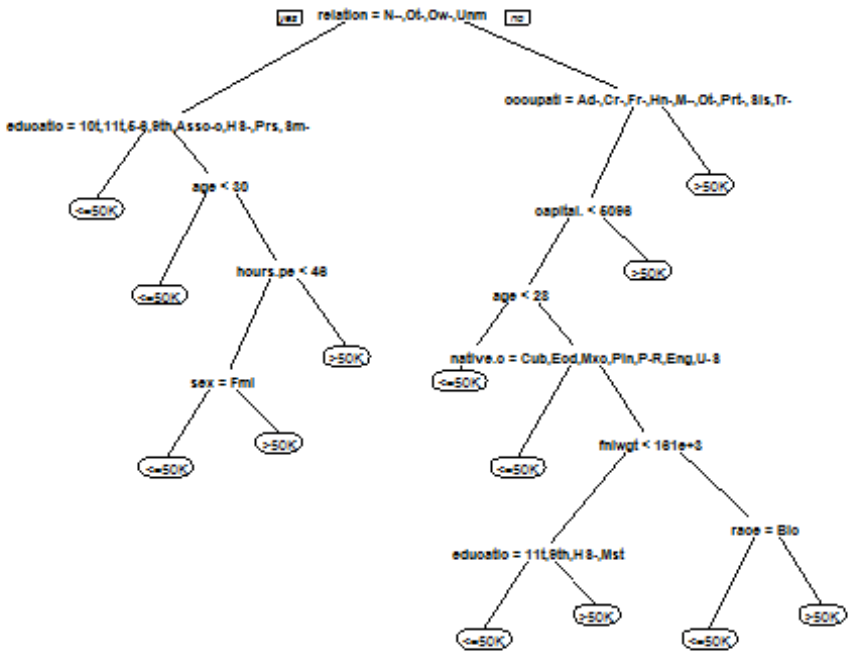
H10:

```
training50_H10 <- trainingData[,c(1:10,15)]
tree_H10 <- rpart(class ~ . ,training50_H10,method="class")
prp(tree_H10)
```



H14:

```
tree_H14 <- rpart(class ~ . ,training50,method="class")
prp(tree_H14)
```

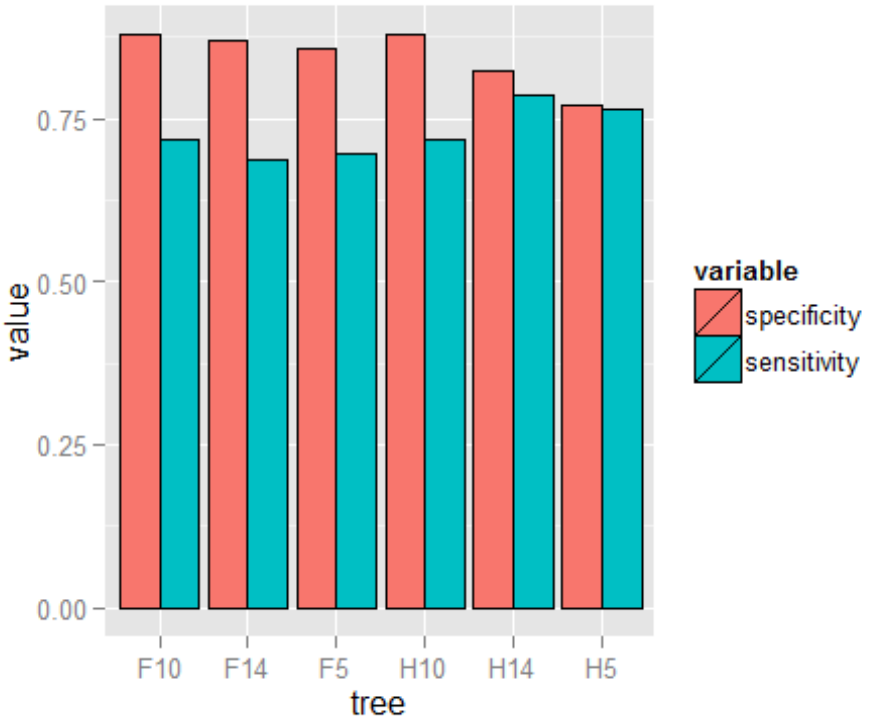


Summary of the result

```
trees <- list(tree_F5, tree_F10, tree_F14, tree_H5, tree_H10, tree_H14)
names <- c("F5", "F10", "F14", "H5", "H10", "H14")
predictionQuality <-  computeAllQualities(trees, testingData, names)

#inspect the results
predictionQuality
```

```
##     specificity sensitivity
## F5       0.8571      0.6966
## F10      0.8791      0.7191
## F14      0.8681      0.6854
## H5       0.7692      0.7640
## H10      0.8791      0.7191
## H14      0.8242      0.7865
```

```
#make a histo graph of 'predictionQuality'
p <- mutate(predictionQuality, tree=rownames(predictionQuality))
p <- melt(p, id="tree")
ggplot(data=p, aes(x=tree, y=value, fill=variable)) +
geom_bar(stat="identity", position=position_dodge(), colour="black")
```



Observations

In this assignment we define "true positive" as number of cases of <=50K that are correctly classified by the model, and define "true negative" as those that are >50K are classified into the >50K bin.

When a full data set is used, specificity seems to be slightly higher than when using half a data set. However, sensitivity seems a bit lower in training with full data. This implies that using full data generates a model that is more capable of excluding those who make less than 50K than when using half the training data. However, at the same time the model is less capable of actually picking out people in the category of <=50K.

As for using different number of predictors to train the model, there doesn't seem to be a pattern as far as the datasets we're given are concerned... In th e case of using half the training dataset, sensitivity is the highest when all predictors are included in the model. However, in the case of using a full dataset, sensitivity is the lowest when all predictors are used.

When using either the full training dataset or half dataset, specificty is the highest when the first 10 predictors are used. This tells us that the first 10 predictors are more related to whether a person makes >50K a year than all predictors together.

The lack of a concrete pattern in the number of predictors and the scores of specificity and sensitivity shows that a model performance depends highly on how representative the

training dataset is to the testing dataset, and that too much information could be more noise than useful information.