# **Decision Tree Assignment**

# Introduction

A decision tree is an important type of supervised machine learning model. It is a classification tool that identifies collections of features that lead to different outcomes. For this assignment you will run a decision tree and interpret the outcomes. There are three main tasks in the assignment. The first is to produce a decision tree in R and, the second is to correctly report the results, and the third is to test the model you have created on a new dataset.

For the assignment, you will use the rate of violent crime as the target variable. In the assignment you will gain experience in working with decision tree models by interpreting the meaning of the branches and nodes that produce a given outcome. You will also use measures of model fit to assess the predictive capacity of the model given the set of input variables that you have selected.

## Directions

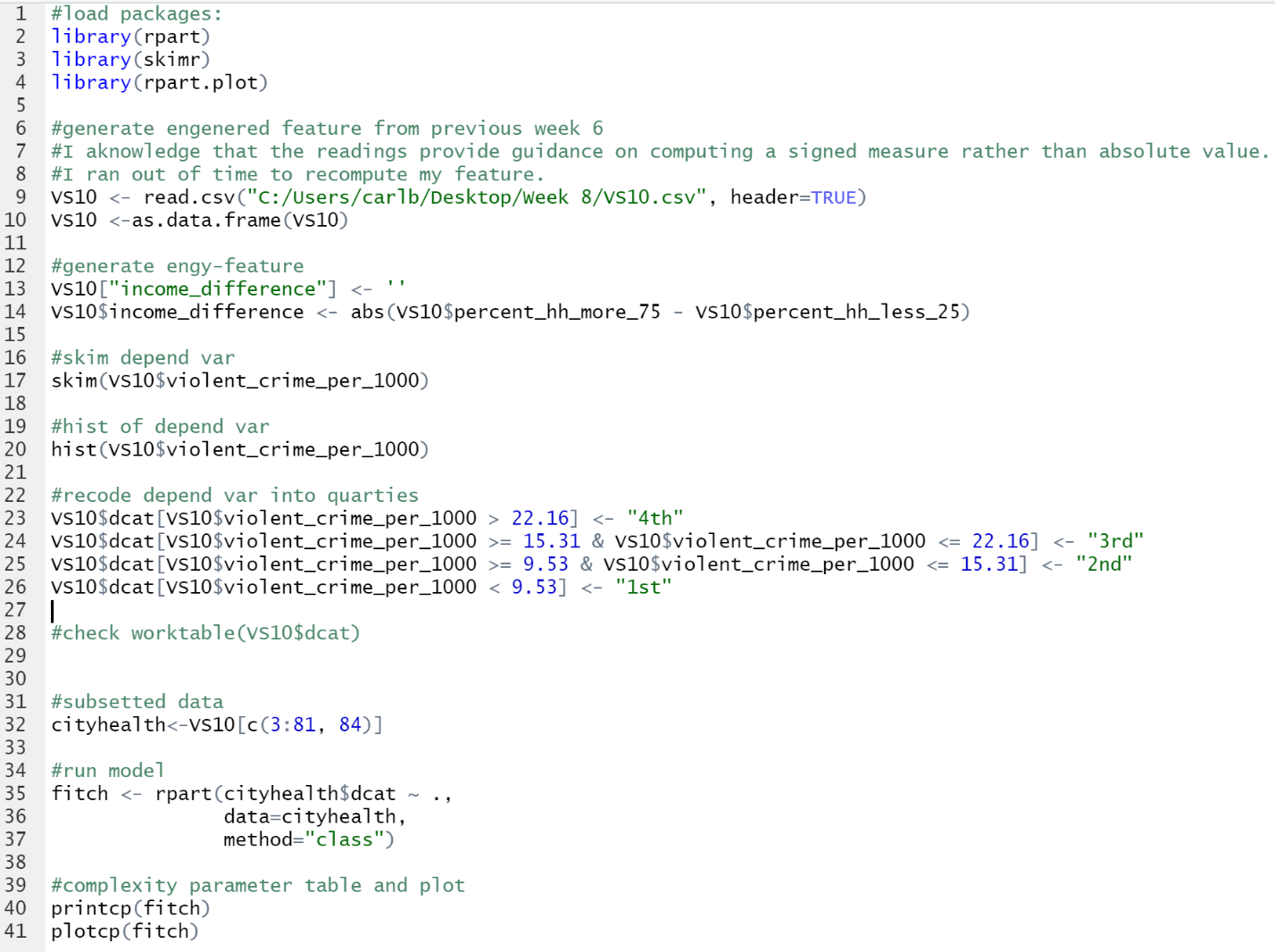
Use the lecture material from Lesson 2 and Lesson 8 and the readings from Lesson 8 to complete the assignment. These readings include *Practical Data Science with R*, Part 2, Chapter 6, Sections 6.3.2, *Beginning Data Science with R*, Chapter 6, Section 6.3.3, Chapter 7, Section 7.3.2, *R in Action*, Part 4, Chapter 17, Section 17.3, and "Plotting rpart Trees with the rpart.plot".

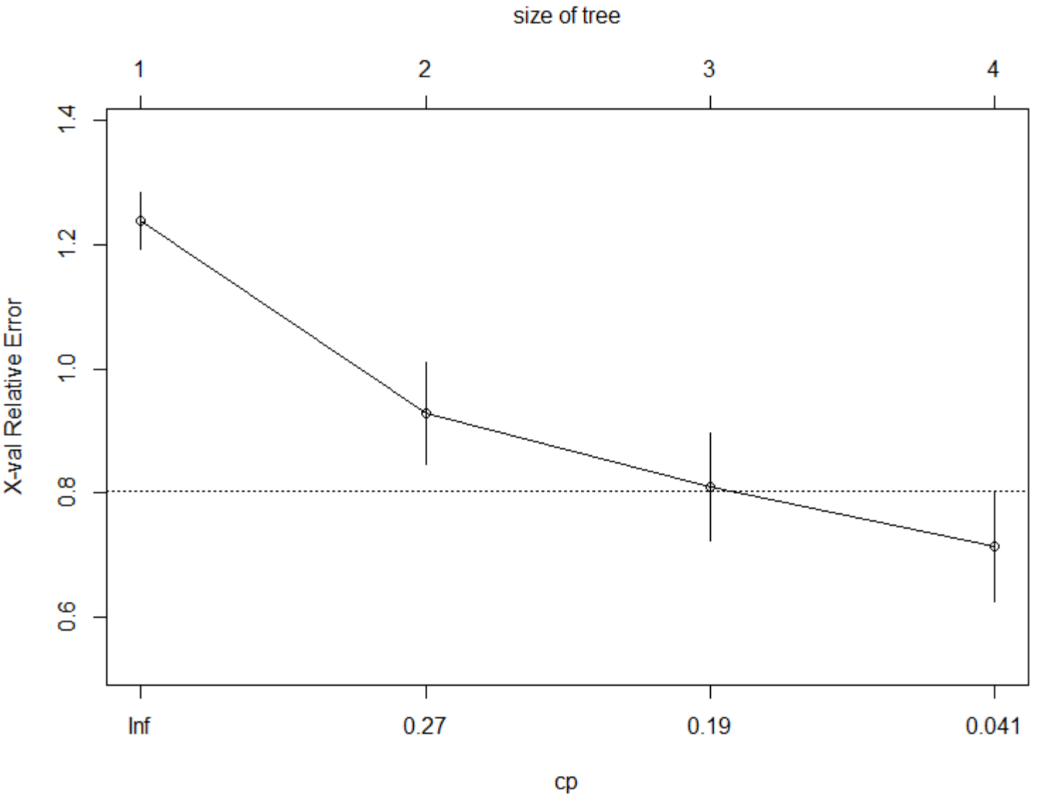
The completed assignment should adhere to the following guidelines:

* 1. Include your answers on the assignment document.
  2. Write your answers using complete sentences with correct punctuation, grammar, and spelling.
  3. Submit your completed assignment through the Blackboard portal.

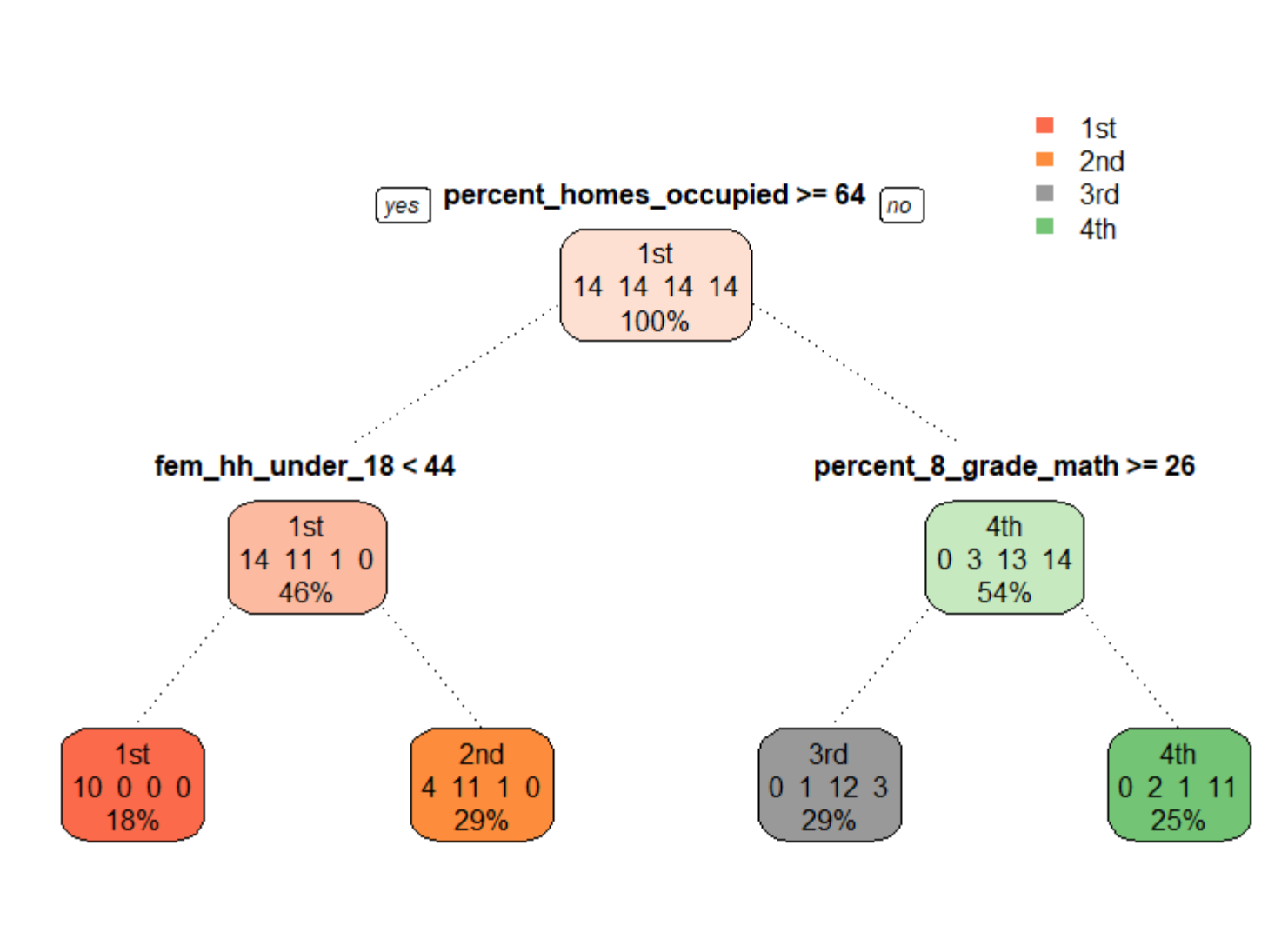
Use the Vital Signs 2010 data to answer Questions 1-3. Use Vital Signs 2017 along with Vital Signs 2010 to answer Question 4.

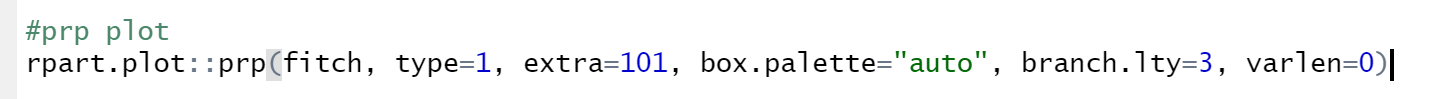
1. Recode the variable Violent Crime Rate per 1,000 Residents into four categories based on the quartiles. Using this as your target variable, create a decision tree to model rates of violent crime. Make sure to omit the Part 1 Crime Rate per 1,000 Residents variable. And make sure to include the feature you created for the Assignment03 Feature Selection and Feature Engineering. Produce a visualization for the tree. Create a bitmap image of the visualization and paste your code and the visualization below. *(3 points)*



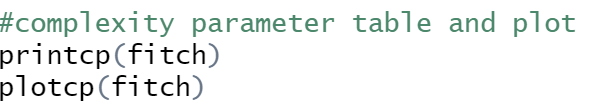


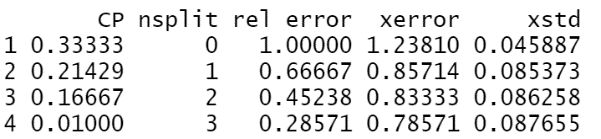
1. Print the rules for the tree. Paste your code and the rules below. Report the attributes of the terminal nodes including the threshold values for each split and the percent of cases that are correctly predicted within the terminal node. *(3 points)*

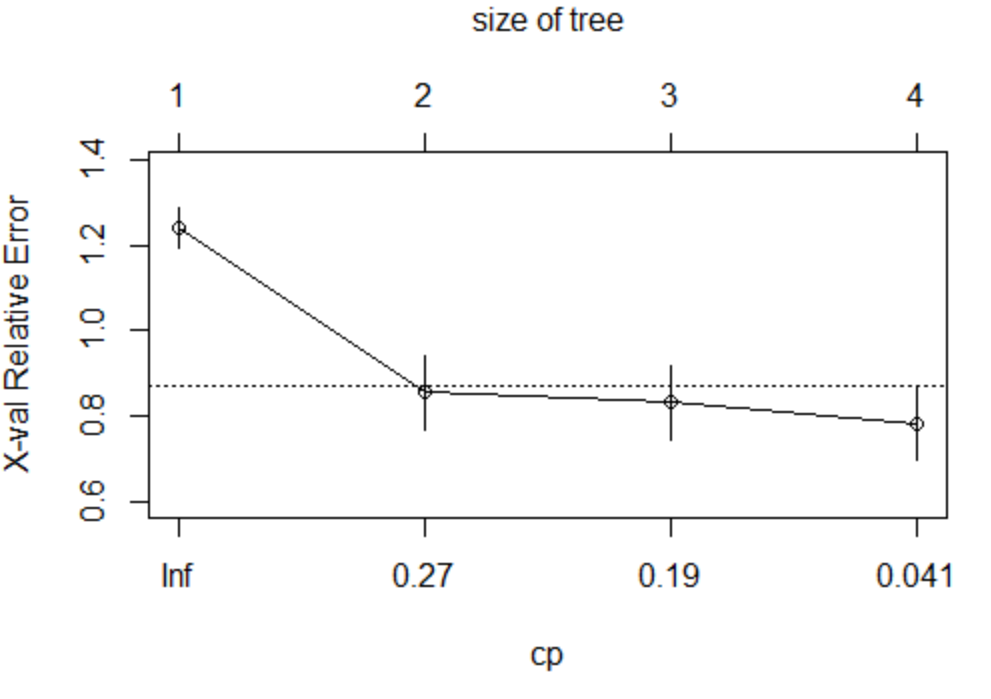




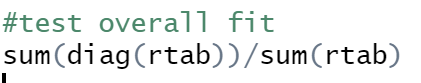
1. Run a complexity table and plot. Report what you observe from the table and plot and what it indicates about the results of your model. Paste your code, the output from the complexity parameter table, and the plot below. Create an error matrix and a measure of the overall error rate for your tree. Revise the tree to using any of the parameters minsplit, minbucket, cp, or maxdepth in order to potentially improve the results. Re-run the error matrix and overall error rate for your revised model. Paste the code and the output below. Which model performed better? *(5 points)*





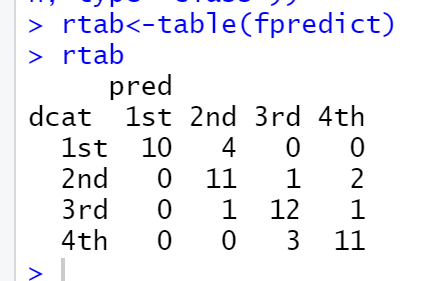


Overall Fit:



0.785

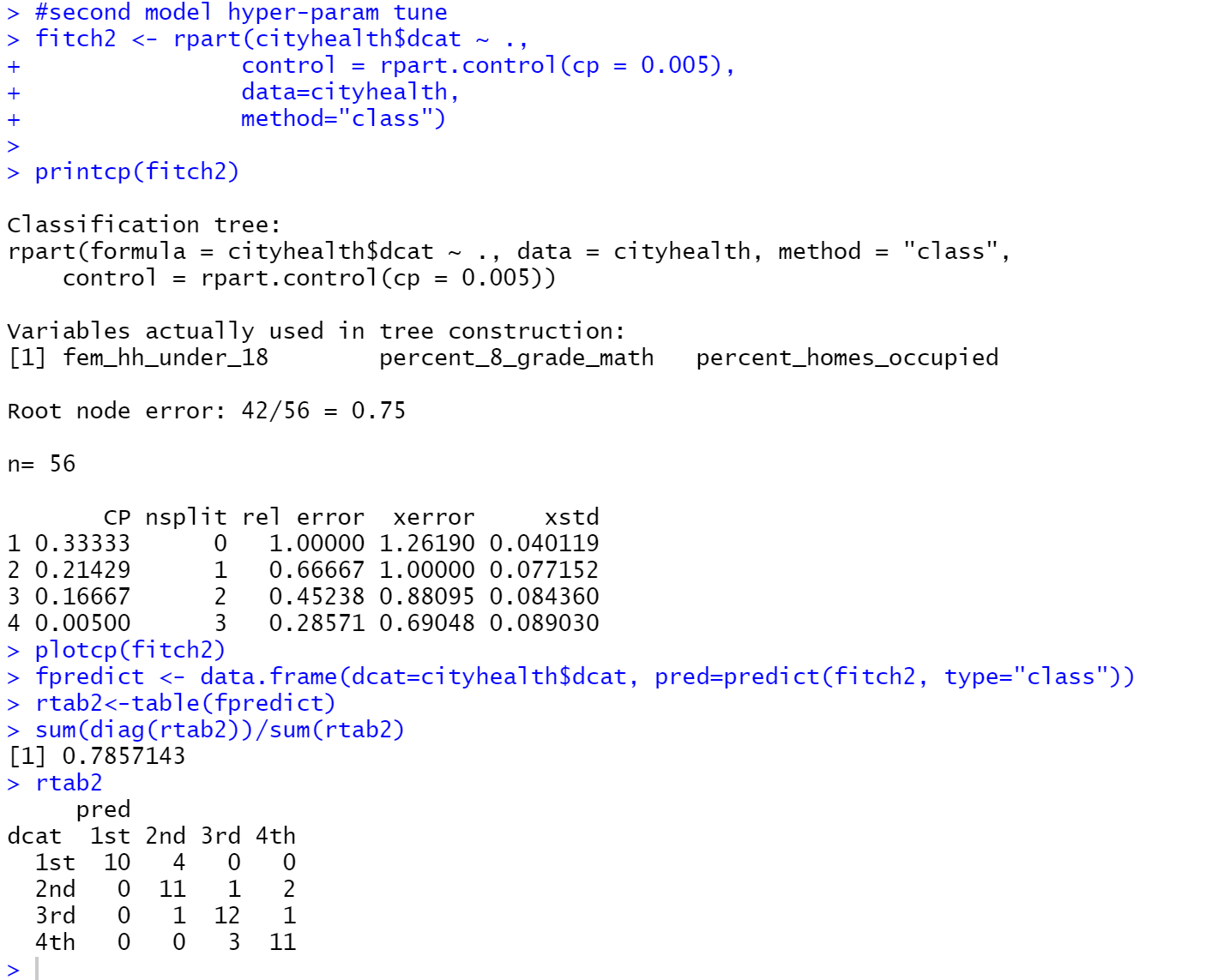
Complexity Table:

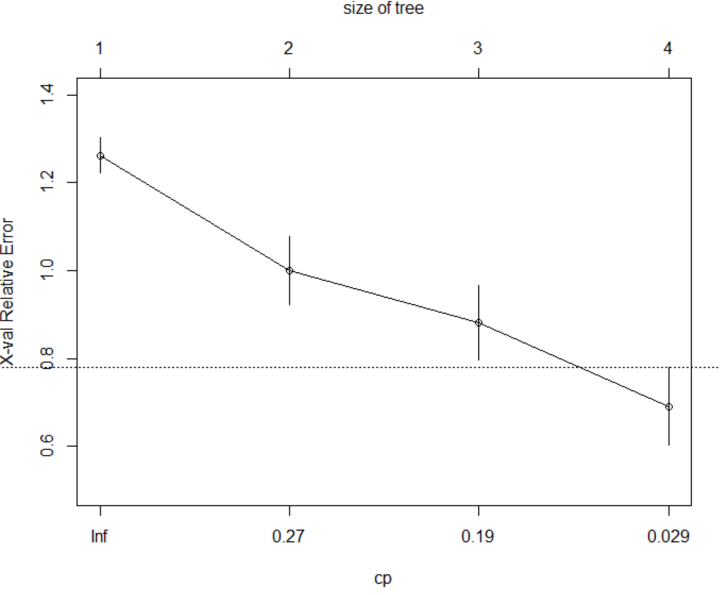


Observations:

There are 4 splits and we stop seeing gains at the second split. The overall fit is strong, at 0.78.

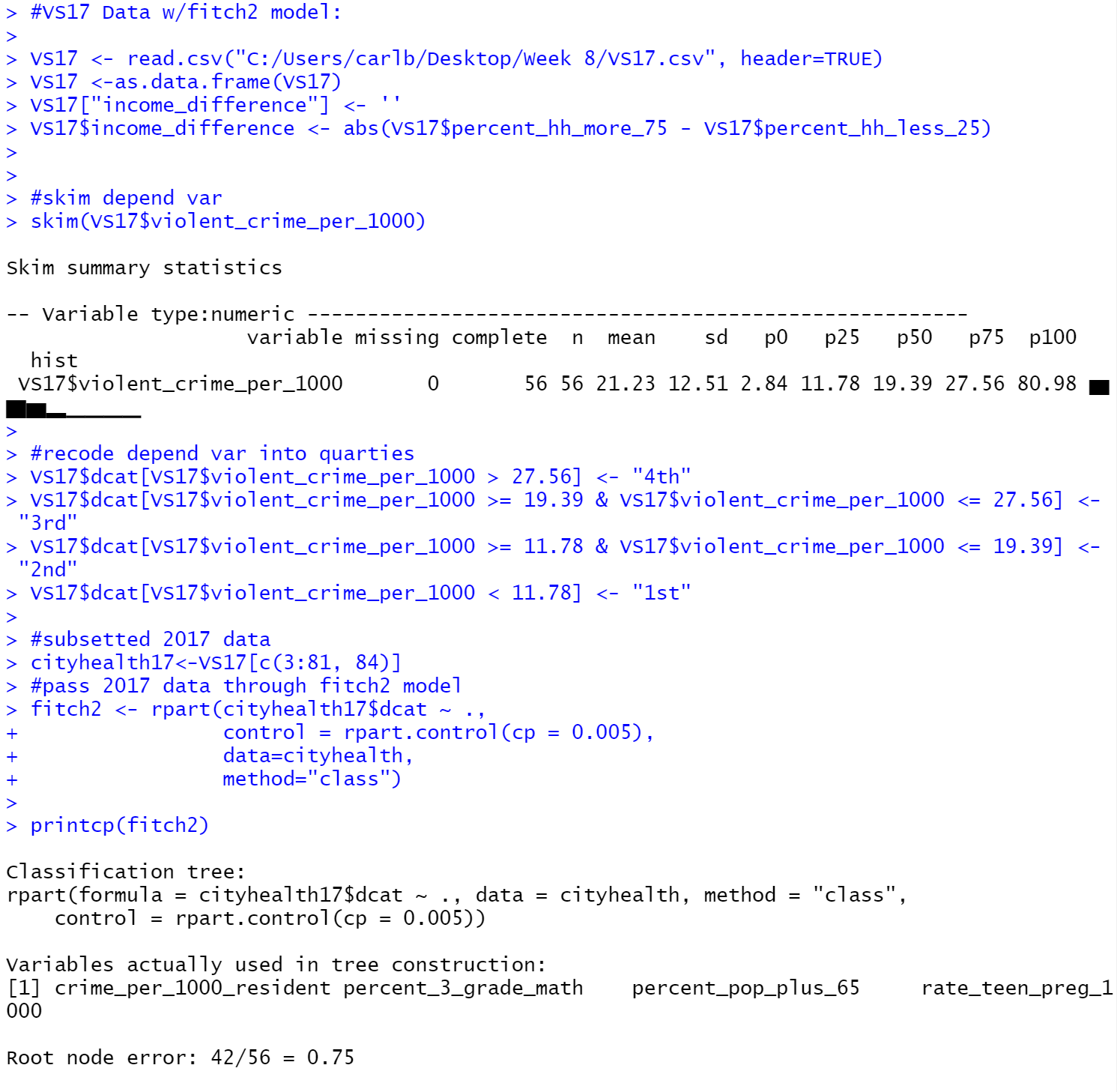
**SECOND MODEL:**

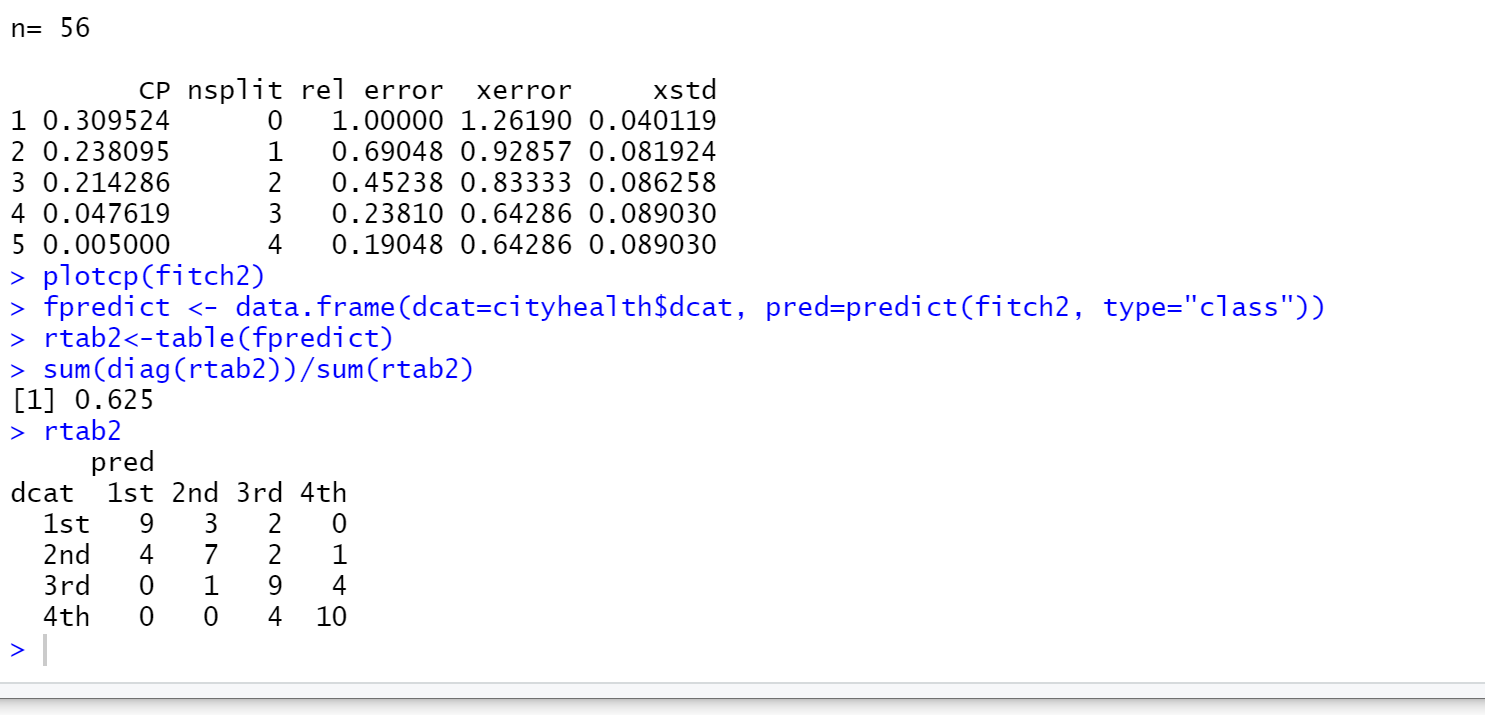
****



The second model preformed better, lower error (not by much).

1. Using the better of the two models from Question 3, test your model using the 2017 Vital Signs data. Create an error matrix and a measure of the overall error rate for the model with the 2017 data. Paste your code and output below. Report your comparison of the results for the model for the 2010 and 2017 data. How well does your model fit the data from seven years later? *(4 points)*





Overall fit went down ~0.2. I’m surprised that seven years later this model is still, somewhat, applicable/useful.

## Scoring

The assignment is worth 15 points. The code for the visualization for the decision tree must include the elements from "Plotting rpart Trees with the rpart.plot Package" to receive full credit. Tests for the best model for Questions 1-3 must be thorough and complete for full credit. The content of the rules must be interpreted and reported full y and correctly for full credit.

In case I missed something… here’s a copy/paste of my entire script:

#load packages:

library(rpart)

library(skimr)

library(rpart.plot)

#generate engenered feature from previous week 6

#I aknowledge that the readings provide guidance on computing a signed measure rather than absolute value.

#I ran out of time to recompute my feature.

VS10 <- read.csv("C:/Users/carlb/Desktop/Week 8/VS10.csv", header=TRUE)

VS10 <-as.data.frame(VS10)

#generate engy-feature

VS10["income\_difference"] <- ''

VS10$income\_difference <- abs(VS10$percent\_hh\_more\_75 - VS10$percent\_hh\_less\_25)

#skim depend var

skim(VS10$violent\_crime\_per\_1000)

#hist of depend var

hist(VS10$violent\_crime\_per\_1000)

#recode depend var into quarties

VS10$dcat[VS10$violent\_crime\_per\_1000 > 22.16] <- "4th"

VS10$dcat[VS10$violent\_crime\_per\_1000 >= 15.31 & VS10$violent\_crime\_per\_1000 <= 22.16] <- "3rd"

VS10$dcat[VS10$violent\_crime\_per\_1000 >= 9.53 & VS10$violent\_crime\_per\_1000 <= 15.31] <- "2nd"

VS10$dcat[VS10$violent\_crime\_per\_1000 < 9.53] <- "1st"

#subsetted data

cityhealth<-VS10[c(3:81, 84)]

#run model

fitch <- rpart(cityhealth$dcat ~ .,

data=cityhealth,

method="class")

#complexity parameter table and plot

printcp(fitch)

plotcp(fitch)

#prp plot

rpart.plot::prp(fitch, type=1, extra=101, box.palette="auto", branch.lty=3, varlen=0)

#create an error matrix

fpredict <- data.frame(dcat=cityhealth$dcat, pred=predict(fitch, type="class"))

rtab<-table(fpredict)

rtab

#test overall fit

sum(diag(rtab))/sum(rtab)

#test within ros for error rate for specific outcomes

sum((rtab[1,1]))/sum(rtab[1,])

sum((rtab[2,2]))/sum(rtab[2,])

sum((rtab[3,3]))/sum(rtab[3,])

sum((rtab[4,4]))/sum(rtab[4,])

#second model hyper-param tune

fitch2 <- rpart(cityhealth$dcat ~ .,

control = rpart.control(cp = 0.005),

data=cityhealth,

method="class")

printcp(fitch2)

plotcp(fitch2)

fpredict <- data.frame(dcat=cityhealth$dcat, pred=predict(fitch2, type="class"))

rtab2<-table(fpredict)

sum(diag(rtab2))/sum(rtab2)

rtab2

#VS17 Data w/fitch2 model:

VS17 <- read.csv("C:/Users/carlb/Desktop/Week 8/VS17.csv", header=TRUE)

VS17 <-as.data.frame(VS17)

VS17["income\_difference"] <- ''

VS17$income\_difference <- abs(VS17$percent\_hh\_more\_75 - VS17$percent\_hh\_less\_25)

#skim depend var

skim(VS17$violent\_crime\_per\_1000)

#recode depend var into quarties

VS17$dcat[VS17$violent\_crime\_per\_1000 > 27.56] <- "4th"

VS17$dcat[VS17$violent\_crime\_per\_1000 >= 19.39 & VS17$violent\_crime\_per\_1000 <= 27.56] <- "3rd"

VS17$dcat[VS17$violent\_crime\_per\_1000 >= 11.78 & VS17$violent\_crime\_per\_1000 <= 19.39] <- "2nd"

VS17$dcat[VS17$violent\_crime\_per\_1000 < 11.78] <- "1st"

#subsetted 2017 data

cityhealth17<-VS17[c(3:81, 84)]

#pass 2017 data through fitch2 model

fitch2 <- rpart(cityhealth17$dcat ~ .,

control = rpart.control(cp = 0.005),

data=cityhealth,

method="class")

printcp(fitch2)

plotcp(fitch2)

fpredict <- data.frame(dcat=cityhealth$dcat, pred=predict(fitch2, type="class"))

rtab2<-table(fpredict)

sum(diag(rtab2))/sum(rtab2)

rtab2