April\_HW

Set up

#install.packages("tidyverse")  
#install.packages("caret")  
#install.packages("caTools")  
#install.packages("rpart")  
#install.packages("rpart.plot")  
#install.packages("randomForest")  
#install.packages("e1071")  
  
library(tidyverse)

## ── Attaching packages ──────────────────────────────────────────────────────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 2.2.1 ✔ purrr 0.2.4  
## ✔ tibble 1.4.1 ✔ dplyr 0.7.4  
## ✔ tidyr 0.7.2 ✔ stringr 1.3.0  
## ✔ readr 1.1.1 ✔ forcats 0.2.0

## ── Conflicts ─────────────────────────────────────────────────────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(caret)

## Warning: package 'caret' was built under R version 3.4.4

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(caTools)  
library(rpart)  
library(rpart.plot)  
library(randomForest)

## Warning: package 'randomForest' was built under R version 3.4.4

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(e1071)

Get data frame

file = "framingham\_edit(1).csv"  
heart = read.csv(file, header=TRUE)

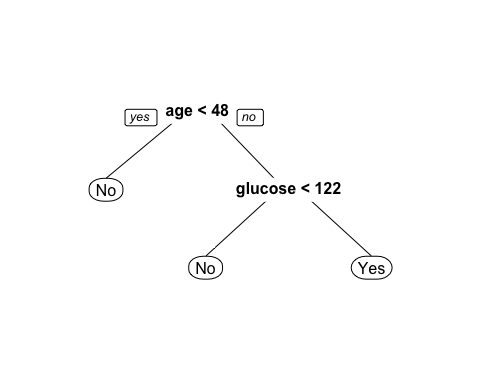
Split dataset

set.seed(999)  
split =sample.split(heart$TenYearCHD, SplitRatio =0.7)  
train =subset(heart, split ==TRUE)  
test =subset(heart, split ==FALSE)

## Q1

Create a classification tree to predict TenYearCHD (coronary heart disease in the next tne years after health measurements taken). Use the training set and all variables. Plot the tree.

fit = rpart(TenYearCHD ~ ., data = train, method = "class")  
prp(fit)



printcp(fit)

##   
## Classification tree:  
## rpart(formula = TenYearCHD ~ ., data = train, method = "class")  
##   
## Variables actually used in tree construction:  
## [1] age glucose  
##   
## Root node error: 451/2968 = 0.15195  
##   
## n= 2968   
##   
## CP nsplit rel error xerror xstd  
## 1 0.012195 0 1.00000 1.00000 0.043363  
## 2 0.010000 2 0.97561 0.99557 0.043284

treepred = predict(fit, train, type = "class")

## Q2

Briefly describe how you would use the tree to classify a person as to whether or not they would be predicted to have coronary heart disease in the next ten years.  
**Answer:** I would start with the top node and considr whether or not the example was 48 years of age or less. If the example was younger than 48, I would predict, using the model, that they would not have coronanry heart disease. If the example was 48 or older, the model would direct me to consider whether or not they had a glucose of 122 or less. If they had a glucose of less than 122, they would be predicted to not hav e coronoary hearyt disease, but if they did have a glucose of 122 or higher, I would predict that they did, in fact, have coronary heart disease.

## Q3

What is the naive accuracy on the training set?

table(train$TenYearCHD, treepred)

## treepred  
## No Yes  
## No 2495 22  
## Yes 418 33

(2495+22)/(nrow(train))

## [1] 0.8480458

**Answer:** Our naive accuracy is 0.848.

## Q4

What is the accuracy of the tree that you developed in Q1? How does this accuracy compare to the naive accuracy?

table(train$TenYearCHD, treepred)

## treepred  
## No Yes  
## No 2495 22  
## Yes 418 33

(2495+33)/(nrow(train))

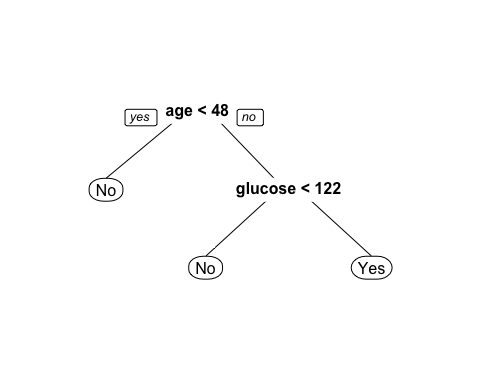
## [1] 0.851752

**Answer:** Our accuracy is 0.8517. This is relatively close to the naive accuracy found in Q3.

## Q5

Repeat the classification tree development, but this time use caret with 5-fold cross validation. Use a tuneLength of 30 as we did in class. Use a seed of 999. Plot your tree and comment.

cvCtrl = trainControl(method = "cv", number = 5)  
  
tree1 = train(TenYearCHD ~ .,   
 data = train,   
 method = "rpart",  
 tuneLength = 30,  
 trControl = cvCtrl,  
 na.action = na.pass  
 )  
  
prp(tree1$finalModel)

 **Answer:** This model is the same as the one produced in Q1. This may indicate that both models are accurate ones.

## Q6

Repeat Q5, but this time use a random forest. Use a seed of 999, 5-fold cross validation, and test mtry values of 2, 3, 4, and 5. As the random forest function cannot handle missing values, use na.action = na.omit. This skips any row in the dataset with missing values.

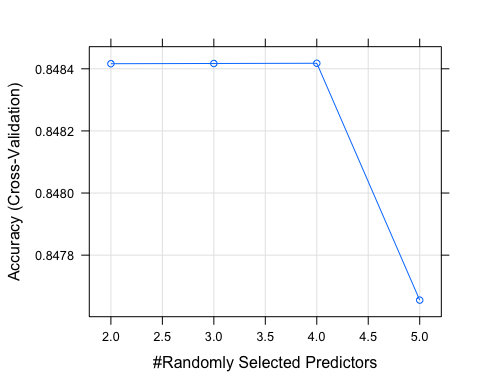
trainpred = predict(tree1$finalModel,type="class")  
confusionMatrix(trainpred,train$TenYearCHD)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 2495 418  
## Yes 22 33  
##   
## Accuracy : 0.8518   
## 95% CI : (0.8385, 0.8644)  
## No Information Rate : 0.848   
## P-Value [Acc > NIR] : 0.2971   
##   
## Kappa : 0.1007   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.99126   
## Specificity : 0.07317   
## Pos Pred Value : 0.85651   
## Neg Pred Value : 0.60000   
## Prevalence : 0.84805   
## Detection Rate : 0.84063   
## Detection Prevalence : 0.98147   
## Balanced Accuracy : 0.53222   
##   
## 'Positive' Class : No   
##

manualgrid = expand.grid(mtry=c(2,3,4,5))  
  
rf1 = train(TenYearCHD ~ .,   
 data = train,   
 method = "rf",   
 tuneGrid = manualgrid,  
 trControl = cvCtrl,  
 na.action = na.omit)  
  
print(rf1)

## Random Forest   
##   
## 2968 samples  
## 14 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 2095, 2095, 2095, 2095, 2096   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 2 0.8484164 0.04429820  
## 3 0.8484171 0.08399327  
## 4 0.8484178 0.10728863  
## 5 0.8476552 0.11358117  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 4.

plot(rf1)



varImp(rf1)

## rf variable importance  
##   
## Overall  
## sysBP 100.000  
## BMI 97.182  
## age 95.567  
## glucose 94.936  
## totChol 94.420  
## diaBP 88.292  
## heartRate 71.776  
## cigsPerDay 36.355  
## genderMale 13.417  
## prevalentHypYes 12.406  
## currentSmokerYes 6.346  
## diabetesYes 3.089  
## BPMedsYes 2.210  
## prevalentStrokeYes 0.000

## Q7

Use the code below to develop predictions on the training set using the random forest from Q6. Note that you will need to change rf1 in the code below to reflect the name of your random forest model. How does your model performance compare to the naive and to your tree from Q5?

trainpredrf = predict(rf1$finalModel)  
complete = train[complete.cases(train),]  
confusionMatrix(trainpredrf,complete$TenYearCHD)

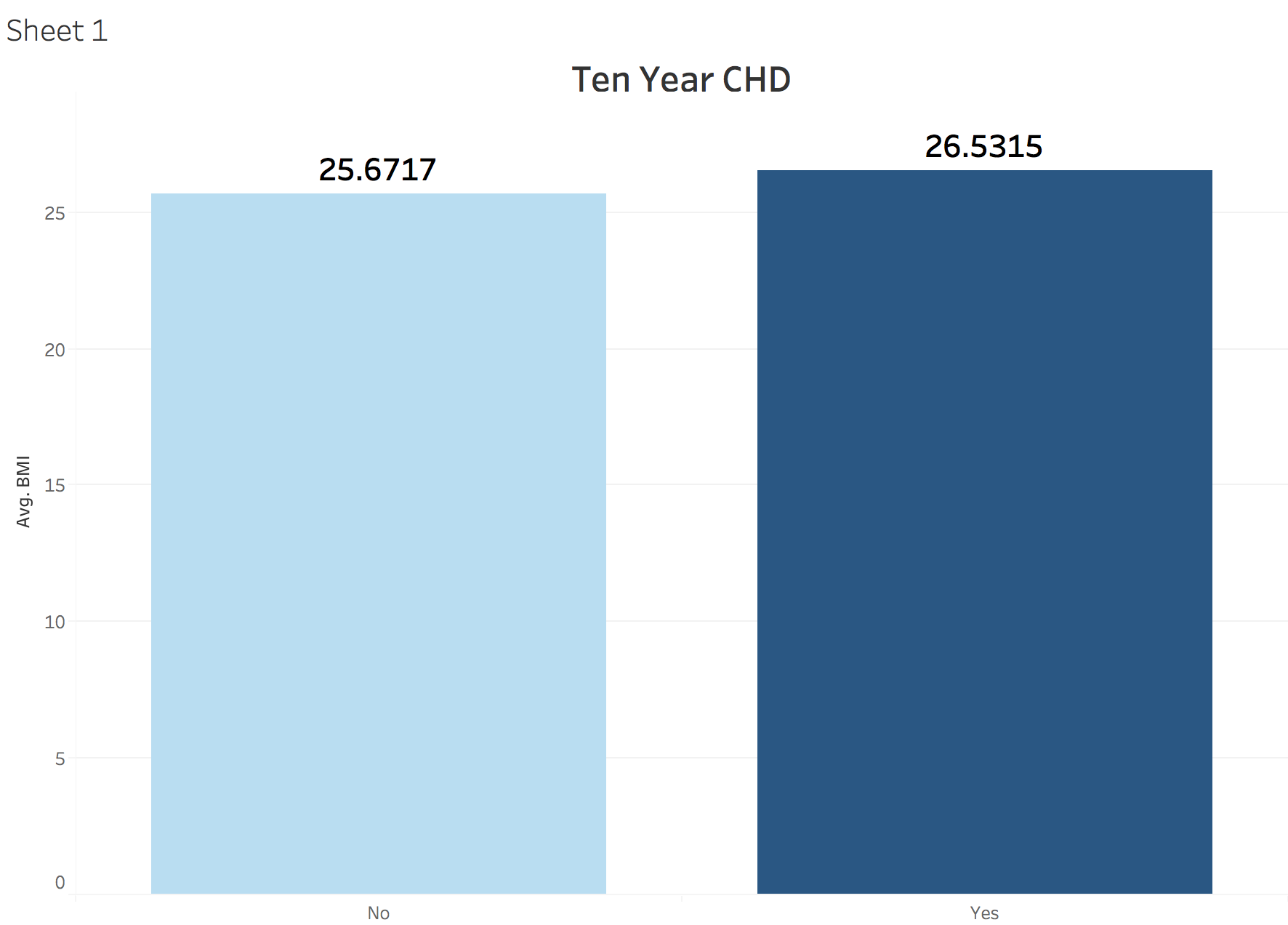
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 2189 366  
## Yes 31 33  
##   
## Accuracy : 0.8484   
## 95% CI : (0.8341, 0.8619)  
## No Information Rate : 0.8477   
## P-Value [Acc > NIR] : 0.47   
##   
## Kappa : 0.1048   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.98604   
## Specificity : 0.08271   
## Pos Pred Value : 0.85675   
## Neg Pred Value : 0.51563   
## Prevalence : 0.84765   
## Detection Rate : 0.83582   
## Detection Prevalence : 0.97556   
## Balanced Accuracy : 0.53437   
##   
## 'Positive' Class : No   
##

**Answer:** The accuracy indicated above is slightly less than the naive and actual accuracy displayed in Q3 and Q4 but close enough to indicate that the model used is a good one that shows consistency.

## 

## Q8

Create an appropriate Tableau plot to show the relationship between BMI and TenYearCHD.



## Q9

Create an appropriate Tableau plot to show the relationship between totChol (total cholesterol) and cigsPerDay (cigarettes smoked per day).

