Machine Learning Assignment 7

Alice Tivarovsky

3/5/2020

## Setup

knitr::opts\_chunk$set(echo = TRUE)  
  
library(randomForest)

## randomForest 4.6-14

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

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

##   
## Attaching package: 'ggplot2'

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

library(tidyverse)

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

## ✔ tibble 2.1.3 ✔ purrr 0.3.2  
## ✔ tidyr 1.0.0 ✔ dplyr 0.8.3  
## ✔ readr 1.3.1 ✔ stringr 1.4.0  
## ✔ tibble 2.1.3 ✔ forcats 0.4.0

## ── Conflicts ───────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::combine() masks randomForest::combine()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ purrr::lift() masks caret::lift()  
## ✖ ggplot2::margin() masks randomForest::margin()

library(rpart)  
library(caret)  
library(rpart.plot)  
library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library(gbm)

## Loaded gbm 2.1.5

## Number 1

**Cleaning and tidying the Cleveland Heart Disease dataset.**

heart.data <- read.csv("/Users/AliceTivarovsky/Documents/Grad School/Spring 2020/Machine Learning/processed.cleveland.data", header = FALSE)  
  
var.names <- c("age", "sex", "pain\_type", "resting\_sysbp", "chol", "fast\_blsugar\_gt120", "rest\_ecg", "max\_hr", "exerc\_angina", "ST\_depression", "ST\_slope", "vessels\_colorflu", "defect", "heart\_disease\_present")  
  
colnames(heart.data) <- var.names  
str(heart.data)

## 'data.frame': 303 obs. of 14 variables:  
## $ age : num 63 67 67 37 41 56 62 57 63 53 ...  
## $ sex : num 1 1 1 1 0 1 0 0 1 1 ...  
## $ pain\_type : num 1 4 4 3 2 2 4 4 4 4 ...  
## $ resting\_sysbp : num 145 160 120 130 130 120 140 120 130 140 ...  
## $ chol : num 233 286 229 250 204 236 268 354 254 203 ...  
## $ fast\_blsugar\_gt120 : num 1 0 0 0 0 0 0 0 0 1 ...  
## $ rest\_ecg : num 2 2 2 0 2 0 2 0 2 2 ...  
## $ max\_hr : num 150 108 129 187 172 178 160 163 147 155 ...  
## $ exerc\_angina : num 0 1 1 0 0 0 0 1 0 1 ...  
## $ ST\_depression : num 2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...  
## $ ST\_slope : num 3 2 2 3 1 1 3 1 2 3 ...  
## $ vessels\_colorflu : Factor w/ 5 levels "?","0.0","1.0",..: 2 5 4 2 2 2 4 2 3 2 ...  
## $ defect : Factor w/ 4 levels "?","3.0","6.0",..: 3 2 4 2 2 2 2 2 4 4 ...  
## $ heart\_disease\_present: int 0 2 1 0 0 0 3 0 2 1 ...

heart.data[heart.data == "?"] <- NA  
  
heart.data$defect <- as.numeric(factor(heart.data$defect))  
heart.data$vessels\_colorflu <- as.numeric(factor(heart.data$vessels\_colorflu))  
  
heart.data$outcome <- ifelse(heart.data$heart\_disease\_present == 0, 0,1)  
heart.data$heart\_disease\_present <- NULL  
heart.data$outcome <- factor(heart.data$outcome)  
levels(heart.data$outcome) <- c("HD Not Present", "HD Present")  
str(heart.data)

## 'data.frame': 303 obs. of 14 variables:  
## $ age : num 63 67 67 37 41 56 62 57 63 53 ...  
## $ sex : num 1 1 1 1 0 1 0 0 1 1 ...  
## $ pain\_type : num 1 4 4 3 2 2 4 4 4 4 ...  
## $ resting\_sysbp : num 145 160 120 130 130 120 140 120 130 140 ...  
## $ chol : num 233 286 229 250 204 236 268 354 254 203 ...  
## $ fast\_blsugar\_gt120: num 1 0 0 0 0 0 0 0 0 1 ...  
## $ rest\_ecg : num 2 2 2 0 2 0 2 0 2 2 ...  
## $ max\_hr : num 150 108 129 187 172 178 160 163 147 155 ...  
## $ exerc\_angina : num 0 1 1 0 0 0 0 1 0 1 ...  
## $ ST\_depression : num 2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...  
## $ ST\_slope : num 3 2 2 3 1 1 3 1 2 3 ...  
## $ vessels\_colorflu : num 1 4 3 1 1 1 3 1 2 1 ...  
## $ defect : num 2 1 3 1 1 1 1 1 3 3 ...  
## $ outcome : Factor w/ 2 levels "HD Not Present",..: 1 2 2 1 1 1 2 1 2 2 ...

summary(heart.data)

## age sex pain\_type resting\_sysbp   
## Min. :29.00 Min. :0.0000 Min. :1.000 Min. : 94.0   
## 1st Qu.:48.00 1st Qu.:0.0000 1st Qu.:3.000 1st Qu.:120.0   
## Median :56.00 Median :1.0000 Median :3.000 Median :130.0   
## Mean :54.44 Mean :0.6799 Mean :3.158 Mean :131.7   
## 3rd Qu.:61.00 3rd Qu.:1.0000 3rd Qu.:4.000 3rd Qu.:140.0   
## Max. :77.00 Max. :1.0000 Max. :4.000 Max. :200.0   
##   
## chol fast\_blsugar\_gt120 rest\_ecg max\_hr   
## Min. :126.0 Min. :0.0000 Min. :0.0000 Min. : 71.0   
## 1st Qu.:211.0 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:133.5   
## Median :241.0 Median :0.0000 Median :1.0000 Median :153.0   
## Mean :246.7 Mean :0.1485 Mean :0.9901 Mean :149.6   
## 3rd Qu.:275.0 3rd Qu.:0.0000 3rd Qu.:2.0000 3rd Qu.:166.0   
## Max. :564.0 Max. :1.0000 Max. :2.0000 Max. :202.0   
##   
## exerc\_angina ST\_depression ST\_slope vessels\_colorflu  
## Min. :0.0000 Min. :0.00 Min. :1.000 Min. :1.000   
## 1st Qu.:0.0000 1st Qu.:0.00 1st Qu.:1.000 1st Qu.:1.000   
## Median :0.0000 Median :0.80 Median :2.000 Median :1.000   
## Mean :0.3267 Mean :1.04 Mean :1.601 Mean :1.672   
## 3rd Qu.:1.0000 3rd Qu.:1.60 3rd Qu.:2.000 3rd Qu.:2.000   
## Max. :1.0000 Max. :6.20 Max. :3.000 Max. :4.000   
## NA's :4   
## defect outcome   
## Min. :1.000 HD Not Present:164   
## 1st Qu.:1.000 HD Present :139   
## Median :1.000   
## Mean :1.837   
## 3rd Qu.:3.000   
## Max. :3.000   
## NA's :2

#Remove the missings  
heart.data.nomiss <- na.omit(heart.data)  
  
#Set No Heart Disease as Reference Level  
heart.data.nomiss$outcome <- relevel(heart.data.nomiss$outcome, ref = "HD Not Present")  
  
str(heart.data.nomiss)

## 'data.frame': 297 obs. of 14 variables:  
## $ age : num 63 67 67 37 41 56 62 57 63 53 ...  
## $ sex : num 1 1 1 1 0 1 0 0 1 1 ...  
## $ pain\_type : num 1 4 4 3 2 2 4 4 4 4 ...  
## $ resting\_sysbp : num 145 160 120 130 130 120 140 120 130 140 ...  
## $ chol : num 233 286 229 250 204 236 268 354 254 203 ...  
## $ fast\_blsugar\_gt120: num 1 0 0 0 0 0 0 0 0 1 ...  
## $ rest\_ecg : num 2 2 2 0 2 0 2 0 2 2 ...  
## $ max\_hr : num 150 108 129 187 172 178 160 163 147 155 ...  
## $ exerc\_angina : num 0 1 1 0 0 0 0 1 0 1 ...  
## $ ST\_depression : num 2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...  
## $ ST\_slope : num 3 2 2 3 1 1 3 1 2 3 ...  
## $ vessels\_colorflu : num 1 4 3 1 1 1 3 1 2 1 ...  
## $ defect : num 2 1 3 1 1 1 1 1 3 3 ...  
## $ outcome : Factor w/ 2 levels "HD Not Present",..: 1 2 2 1 1 1 2 1 2 2 ...  
## - attr(\*, "na.action")= 'omit' Named int 88 167 193 267 288 303  
## ..- attr(\*, "names")= chr "88" "167" "193" "267" ...

## Number 2

**Run a single classification tree using all of the features available in the dataset. Calculate evaluation metrics and output the variable importance metrics.**

training.data <- heart.data.nomiss$outcome %>% createDataPartition(p = 0.7, list = F)  
train.data <- heart.data.nomiss[training.data, ]  
test.data <- heart.data.nomiss[-training.data, ]

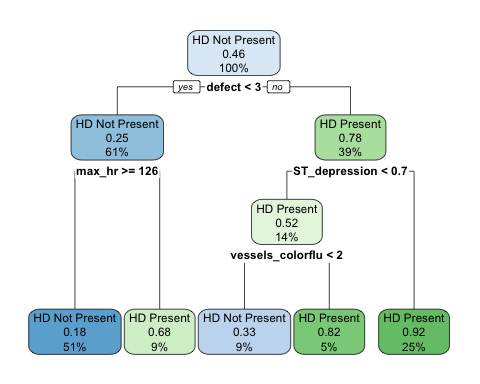
train.control <- trainControl(method = "cv", number = 10)  
grid.2 <- expand.grid(cp = seq(0.001, 0.3, by = 0.01))  
hd.tree <- train(outcome~., data = train.data, method = "rpart",trControl = train.control, tuneGrid = grid.2)  
hd.tree$bestTune

## cp  
## 3 0.021

# variable importance  
varImp(hd.tree)

## rpart variable importance  
##   
## Overall  
## pain\_type 100.000  
## vessels\_colorflu 93.829  
## ST\_slope 79.696  
## defect 75.732  
## exerc\_angina 48.133  
## max\_hr 32.017  
## age 24.111  
## ST\_depression 17.100  
## resting\_sysbp 6.247  
## chol 6.099  
## sex 0.000  
## fast\_blsugar\_gt120 0.000  
## rest\_ecg 0.000

# tree plot  
rpart.plot(hd.tree$finalModel)



accuracy.train.singletree <- hd.tree$results[which.max(hd.tree$results[,"Accuracy"]), "Accuracy"]

The variables vessels\_colorflu, pain\_type, max\_hr, defect and exerc\_angina have the highest variable importance. The accuracy of the model is 81.76%.

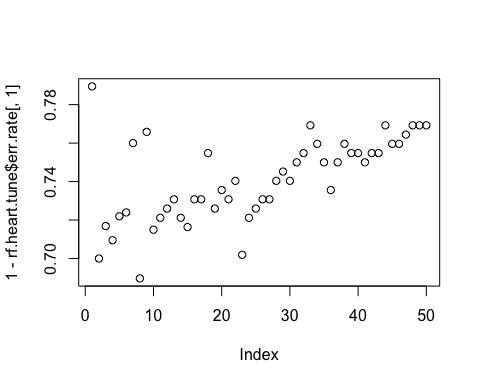
## Number 3

**Use random forest to classify heart disease. Set up a pipeline to try different values of mtry and different numbers of trees to obtain your optimal model. Again, calculate appropriate evaluation metrics and output the variable importance metrics.**

set.seed(100)  
  
possible\_predictors = heart.data.nomiss %>% select(-"outcome")  
  
## using tuneRF to find optimal mtry  
bestMtry=tuneRF(x = possible\_predictors, y = heart.data.nomiss$outcome, ntreeTry = 50, stepFactor = 1, improve=0.0001, trace=FALSE, plot=FALSE, doBest=FALSE)   
  
# running random forest with 50 trees  
rf.heart.tune = randomForest(outcome ~., data = train.data, mtry = bestMtry, importance = TRUE, ntree = 50)  
  
print(rf.heart.tune)

##   
## Call:  
## randomForest(formula = outcome ~ ., data = train.data, mtry = bestMtry, importance = TRUE, ntree = 50)   
## Type of random forest: classification  
## Number of trees: 50  
## No. of variables tried at each split: 1  
##   
## OOB estimate of error rate: 23.08%  
## Confusion matrix:  
## HD Not Present HD Present class.error  
## HD Not Present 89 23 0.2053571  
## HD Present 25 71 0.2604167

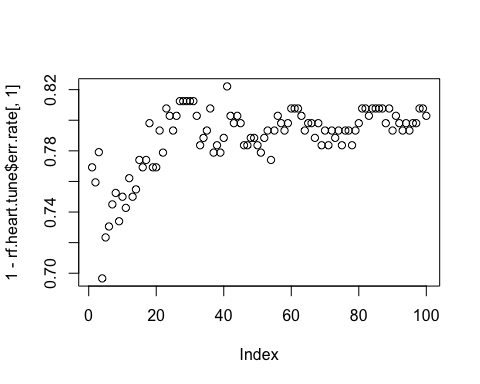
plot(1-rf.heart.tune$err.rate[,1])



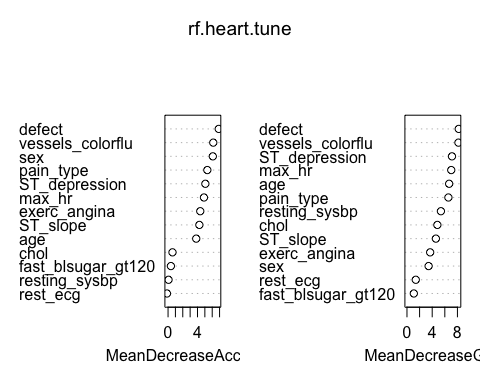
# running random forest with 100 trees  
rf.heart.tune = randomForest(outcome ~., data = train.data, mtry = bestMtry, importance = TRUE, ntree = 100)  
  
print(rf.heart.tune)

##   
## Call:  
## randomForest(formula = outcome ~ ., data = train.data, mtry = bestMtry, importance = TRUE, ntree = 100)   
## Type of random forest: classification  
## Number of trees: 100  
## No. of variables tried at each split: 1  
##   
## OOB estimate of error rate: 19.71%  
## Confusion matrix:  
## HD Not Present HD Present class.error  
## HD Not Present 96 16 0.1428571  
## HD Present 25 71 0.2604167

plot(1-rf.heart.tune$err.rate[,1])



varImpPlot(rf.heart.tune)



Out of bag error rate with 50 trees = 12.98% Error rate with 100 trees = 16.35%

We see from the plots that the error rate does not improve after about 50 trees. Looking at the variable importance plots, vessels\_colorflu, defect, pain\_type, and max\_hr are the most important variables.

## Number 4

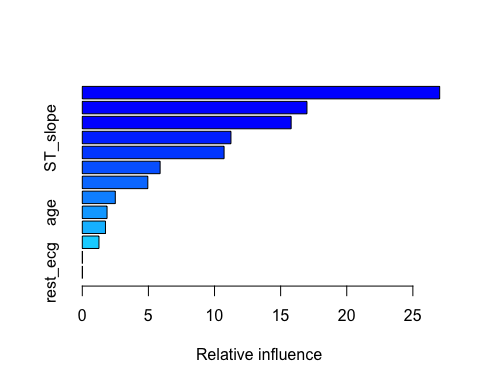
**Answer the questions: Are there differences in variable importance that you see between a single tree and an ensemble metric? Are there differences observed across the different variable importance metrics output from the ensemble? How do you interpret those differences?**

For the most part the variables with the highest importance are the same between the single tree and the random forest: vessels, defect, chest pain type, and maximum heart rate. The order is slightly different, but the most important variable remains vessels. This consistency suggests that the variable importance computed by the single tree algorithm is accurate, since random forests are more accurate than single trees.

## Number 5

**Use a boosting algorithm and tune to obtain your optimal model. Compare to the results from the single classification tree and the random forest.**

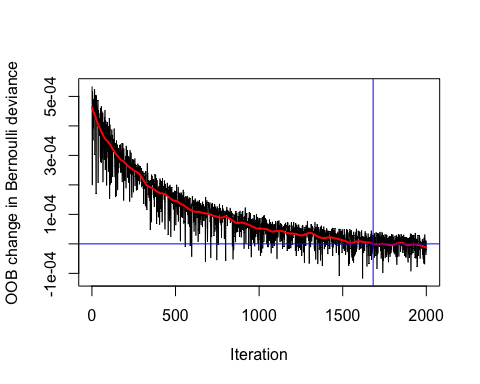
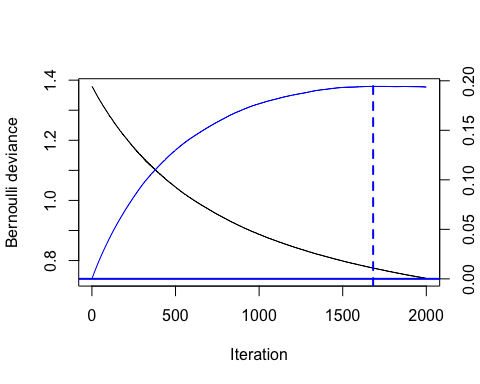
# convert outcome to numeric  
set.seed(100)  
train.data$outcome.num = (as.numeric(train.data$outcome)[train.data$outcome]) - 1  
  
train.data$outcome = NULL  
  
gbm.heart = gbm(outcome.num ~., data = train.data, distribution = 'bernoulli', n.trees = 2000, shrinkage = 0.002)  
  
summary(gbm.heart)



## var rel.inf  
## defect defect 27.028195  
## pain\_type pain\_type 16.990944  
## vessels\_colorflu vessels\_colorflu 15.794262  
## ST\_slope ST\_slope 11.241835  
## ST\_depression ST\_depression 10.722454  
## max\_hr max\_hr 5.885205  
## exerc\_angina exerc\_angina 4.955247  
## sex sex 2.495534  
## age age 1.872242  
## chol chol 1.752724  
## resting\_sysbp resting\_sysbp 1.261357  
## fast\_blsugar\_gt120 fast\_blsugar\_gt120 0.000000  
## rest\_ecg rest\_ecg 0.000000

gbm.perf(gbm.heart, plot.it=TRUE, oobag.curve=TRUE, overlay=TRUE, method='OOB')

## OOB generally underestimates the optimal number of iterations although predictive performance is reasonably competitive. Using cv\_folds>1 when calling gbm usually results in improved predictive performance.



## [1] 1682  
## attr(,"smoother")  
## Call:  
## loess(formula = object$oobag.improve ~ x, enp.target = min(max(4,   
## length(x)/10), 50))  
##   
## Number of Observations: 2000   
## Equivalent Number of Parameters: 39.99   
## Residual Standard Error: 3.688e-05

pred.gbm.heart<-predict(gbm.heart, train.data, n.trees=2000, type="response")  
  
pred.gbm.class<-round(pred.gbm.heart)  
  
misClasificError <- mean(pred.gbm.class != train.data$outcome.num)  
print(paste('Accuracy Model',1-misClasificError))

## [1] "Accuracy Model 0.860576923076923"

The boosting algorithm with 2000 trees and a lambda of 0.002 yields a highly accurate model (94.7%). This is a big improvment from both the single tree and random forest models.

## Number 6

**Which model performs the best? Provide justification for your answer.**

The boosted model performs best. It has the lowest error rate (100 - 94.7 = 5.3%), versus 16% in the random forest and 19% in the single tree model. However, in order to achieve this accuracy in the boosted algorithm, we need 2000 trees. If we drop to 1000 trees, the accuracy falls to 90%.

## Number 7

**How do these results compare to the SVC analysis we did back in Class 6?**

The SVC analysis we ran in Class 6 resulted in a model with 87% accuracy, which is higher than both the single tree and random forest models, but lower than the boosted algorithm.