northwell\_ds\_exercise

Fang Liu

5/9/2022

Table of Contents

## Step 1 - Data Preparation

#load data  
heart\_data <- read\_csv("./HD.csv")

## Rows: 303 Columns: 15  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (3): ChestPain, Thal, HD  
## dbl (12): ID, Age, Sex, RestBP, Chol, Fbs, RestECG, MaxHR, ExAng, Oldpeak, S...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

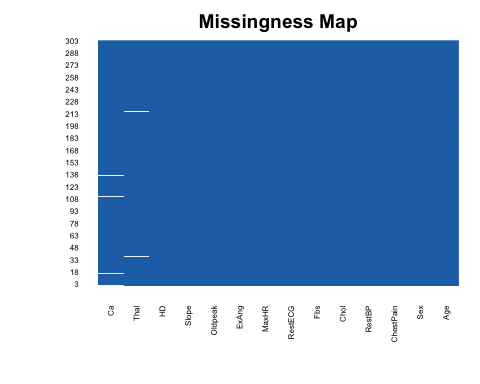
str(heart\_data)

## spec\_tbl\_df [303 × 15] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ID : num [1:303] 1 2 3 4 5 6 7 8 9 10 ...  
## $ Age : num [1:303] 63 67 67 37 41 56 62 57 63 53 ...  
## $ Sex : num [1:303] 1 1 1 1 0 1 0 0 1 1 ...  
## $ ChestPain: chr [1:303] "typical" "asymptomatic" "asymptomatic" "nonanginal" ...  
## $ RestBP : num [1:303] 145 160 120 130 130 120 140 120 130 140 ...  
## $ Chol : num [1:303] 233 286 229 250 204 236 268 354 254 203 ...  
## $ Fbs : num [1:303] 1 0 0 0 0 0 0 0 0 1 ...  
## $ RestECG : num [1:303] 2 2 2 0 2 0 2 0 2 2 ...  
## $ MaxHR : num [1:303] 150 108 129 187 172 178 160 163 147 155 ...  
## $ ExAng : num [1:303] 0 1 1 0 0 0 0 1 0 1 ...  
## $ Oldpeak : num [1:303] 2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...  
## $ Slope : num [1:303] 3 2 2 3 1 1 3 1 2 3 ...  
## $ Ca : num [1:303] 0 3 2 0 0 0 2 0 1 0 ...  
## $ Thal : chr [1:303] "fixed" "normal" "reversable" "normal" ...  
## $ HD : chr [1:303] "No" "Yes" "Yes" "No" ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. ID = col\_double(),  
## .. Age = col\_double(),  
## .. Sex = col\_double(),  
## .. ChestPain = col\_character(),  
## .. RestBP = col\_double(),  
## .. Chol = col\_double(),  
## .. Fbs = col\_double(),  
## .. RestECG = col\_double(),  
## .. MaxHR = col\_double(),  
## .. ExAng = col\_double(),  
## .. Oldpeak = col\_double(),  
## .. Slope = col\_double(),  
## .. Ca = col\_double(),  
## .. Thal = col\_character(),  
## .. HD = col\_character()  
## .. )  
## - attr(\*, "problems")=<externalptr>

#Strip off ID Variable  
heart\_data$ID<-NULL  
  
#make sure the data is the right type  
heart\_data <- heart\_data %>%   
 mutate(Sex = if\_else(Sex == 1, "Male", "Female"),  
 Sex = factor(Sex),  
 ChestPain = factor(ChestPain),  
 Fbs = factor(Fbs),  
 ExAng = factor(ExAng),  
 Thal = factor(Thal),  
 HD = factor(HD))  
  
#check missing data  
missmap(heart\_data, legend = FALSE, x.cex = 0.5, y.cex = 0.5)

## Warning: Unknown or uninitialised column: `arguments`.  
## Unknown or uninitialised column: `arguments`.

## Warning: Unknown or uninitialised column: `imputations`.



#3 IDS missing 'Ca' and 2 IDs missing 'Thal'   
  
#Remove missing (5 observations)  
heart\_data <- na.omit(heart\_data)  
  
#check if data is balanced  
summary(heart\_data$HD) #data is balanced

## No Yes   
## 160 137

## Step 2 - Exploratory Data Analysis

# create a descriptive table for ENTIRE dataset  
heart\_data %>%   
 tbl\_summary(statistic = all\_continuous() ~ "{mean} ({sd})")

## Table printed with `knitr::kable()`, not {gt}. Learn why at  
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

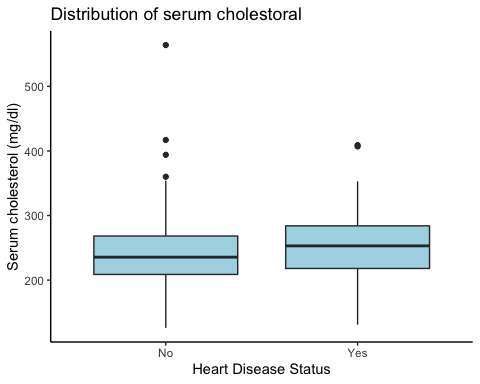
|  |  |
| --- | --- |
| **Characteristic** | **N = 297** |
| Age | 55 (9) |
| Sex |  |
| Female | 96 (32%) |
| Male | 201 (68%) |
| ChestPain |  |
| asymptomatic | 142 (48%) |
| nonanginal | 83 (28%) |
| nontypical | 49 (16%) |
| typical | 23 (7.7%) |
| RestBP | 132 (18) |
| Chol | 247 (52) |
| Fbs |  |
| 0 | 254 (86%) |
| 1 | 43 (14%) |
| RestECG |  |
| 0 | 147 (49%) |
| 1 | 4 (1.3%) |
| 2 | 146 (49%) |
| MaxHR | 150 (23) |
| ExAng |  |
| 0 | 200 (67%) |
| 1 | 97 (33%) |
| Oldpeak | 1.06 (1.17) |
| Slope |  |
| 1 | 139 (47%) |
| 2 | 137 (46%) |
| 3 | 21 (7.1%) |
| Ca |  |
| 0 | 174 (59%) |
| 1 | 65 (22%) |
| 2 | 38 (13%) |
| 3 | 20 (6.7%) |
| Thal |  |
| fixed | 18 (6.1%) |
| normal | 164 (55%) |
| reversable | 115 (39%) |
| HD | 137 (46%) |

# split table by outcome (i.e., heart disease)  
heart\_data %>%   
 tbl\_summary(by = HD,  
 statistic = all\_continuous() ~ "{mean} ({sd})") %>%   
 add\_p(test = all\_continuous() ~ "t.test",  
 test.args = all\_tests("t.test") ~ list(var.equal = TRUE))

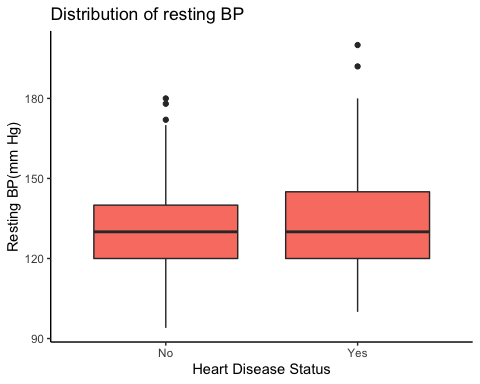
## Table printed with `knitr::kable()`, not {gt}. Learn why at  
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic** | **No**, N = 160 | **Yes**, N = 137 | **p-value** |
| Age | 53 (10) | 57 (8) | <0.001 |
| Sex |  |  | <0.001 |
| Female | 71 (44%) | 25 (18%) |  |
| Male | 89 (56%) | 112 (82%) |  |
| ChestPain |  |  | <0.001 |
| asymptomatic | 39 (24%) | 103 (75%) |  |
| nonanginal | 65 (41%) | 18 (13%) |  |
| nontypical | 40 (25%) | 9 (6.6%) |  |
| typical | 16 (10%) | 7 (5.1%) |  |
| RestBP | 129 (16) | 135 (19) | 0.008 |
| Chol | 243 (54) | 252 (50) | 0.2 |
| Fbs |  |  | >0.9 |
| 0 | 137 (86%) | 117 (85%) |  |
| 1 | 23 (14%) | 20 (15%) |  |
| RestECG |  |  | 0.005 |
| 0 | 92 (57%) | 55 (40%) |  |
| 1 | 1 (0.6%) | 3 (2.2%) |  |
| 2 | 67 (42%) | 79 (58%) |  |
| MaxHR | 159 (19) | 139 (23) | <0.001 |
| ExAng |  |  | <0.001 |
| 0 | 137 (86%) | 63 (46%) |  |
| 1 | 23 (14%) | 74 (54%) |  |
| Oldpeak | 0.60 (0.79) | 1.59 (1.31) | <0.001 |
| Slope |  |  | <0.001 |
| 1 | 103 (64%) | 36 (26%) |  |
| 2 | 48 (30%) | 89 (65%) |  |
| 3 | 9 (5.6%) | 12 (8.8%) |  |
| Ca |  |  | <0.001 |
| 0 | 129 (81%) | 45 (33%) |  |
| 1 | 21 (13%) | 44 (32%) |  |
| 2 | 7 (4.4%) | 31 (23%) |  |
| 3 | 3 (1.9%) | 17 (12%) |  |
| Thal |  |  | <0.001 |
| fixed | 6 (3.8%) | 12 (8.8%) |  |
| normal | 127 (79%) | 37 (27%) |  |
| reversable | 27 (17%) | 88 (64%) |  |

#Quick plot comparing Sex across Outcome Groups  
ggplot(heart\_data) +  
 geom\_boxplot(aes(y=Chol, x=HD), fill = "lightblue") +  
 labs(title = "Distribution of serum cholestoral",  
 x = "Heart Disease Status",  
 y = "Serum cholesterol (mg/dl)") +  
 theme\_classic()



ggplot(heart\_data) +  
 geom\_boxplot(aes(y=RestBP, x=HD), fill = "salmon") +  
 labs(title = "Distribution of resting BP",  
 x = "Heart Disease Status",  
 y = "Resting BP(mm Hg)") +  
 theme\_classic()



From the table above, we can see that individuals with heart diseases are generally older, are more likely to be male, and most of the times experience asymptomatic chest pain, have lower maximum heart rate, more likely to experience exercise-induced angina and have higher ST depression, greater #s of major vessels colored by fluorosopy, and experience more reversable defect. There is no statistically significant difference in resting blood pressure, serum cholesterol, and fasting blood sugar between those with and without heart disease.

## Step 2 - Construct Prediction Models (using caret package)

### Data Partition (70 train: 30 test)

set.seed(100)  
train\_index <- createDataPartition(y=heart\_data$HD, p=0.7, list=FALSE)  
train\_data <- heart\_data[train\_index,]   
test\_data <- heart\_data[-train\_index,]   
  
# Check if the distribution of heart disease status is similar in the training and test sets  
train\_data %>%  
 group\_by(HD) %>%  
 summarize(n = n()) %>%  
 mutate(percent = n/sum(n))

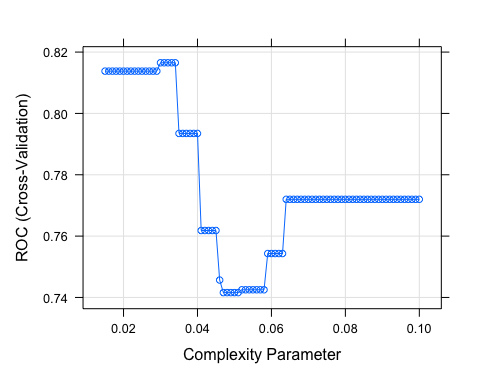
## # A tibble: 2 × 3  
## HD n percent  
## <fct> <int> <dbl>  
## 1 No 112 0.538  
## 2 Yes 96 0.462

test\_data %>%  
 group\_by(HD) %>%  
 summarize(n = n()) %>%  
 mutate(percent = n/sum(n))

## # A tibble: 2 × 3  
## HD n percent  
## <fct> <int> <dbl>  
## 1 No 48 0.539  
## 2 Yes 41 0.461

### Model 1: Classification And Regression Trees (CART)

set.seed(100)  
  
#hyperparmeter tuning: cp  
cp\_grid <- expand.grid(cp = seq(from = 0.015, to = 0.1, by = 0.001))  
  
ctree\_heart <- train(HD ~ ., data = train\_data, method = "rpart",  
 trControl = trainControl("cv", number = 10,  
 summaryFunction = twoClassSummary,  
 classProbs = TRUE,   
 savePredictions = "final"),  
 tuneGrid = cp\_grid,  
 metric = "ROC")  
  
plot(ctree\_heart)



ctree\_heart$bestTune

## cp  
## 20 0.034

confusionMatrix(ctree\_heart)

## Cross-Validated (10 fold) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction No Yes  
## No 46.2 13.5  
## Yes 7.7 32.7  
##   
## Accuracy (average) : 0.7885

### Model 2: Random Forest

set.seed(100)  
  
#hyperparameter tuning: mtry  
mtry\_grid <- expand.grid(.mtry=c(ncol(train\_data)-1, sqrt(ncol(train\_data)-1), 0.5\*ncol(train\_data)-1))   
  
#10-fold cross-validation  
rf\_heart <- train(HD ~ ., data=train\_data, method="rf",  
 trControl=trainControl("cv", number=10,  
 summaryFunction = twoClassSummary,  
 classProbs = TRUE,   
 savePredictions = "final"),  
 tuneGrid=mtry\_grid,   
 ntree=100,  
 metric = "ROC")  
  
rf\_heart$bestTune

## mtry  
## 1 3.605551

rf\_heart$results

## mtry ROC Sens Spec ROCSD SensSD SpecSD  
## 1 3.605551 0.8909764 0.8136364 0.7388889 0.06829533 0.1548126 0.08768475  
## 2 6.000000 0.8779630 0.7856061 0.7288889 0.06467712 0.1875516 0.12049282  
## 3 13.000000 0.8673822 0.7863636 0.7177778 0.08256650 0.2097267 0.12060661

confusionMatrix(rf\_heart)

## Cross-Validated (10 fold) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction No Yes  
## No 43.7 12.0  
## Yes 10.1 34.1  
##   
## Accuracy (average) : 0.7788

### Model 3: Elastic Net (regularized regression)

set.seed(100)  
  
en\_heart <- train(HD ~ .,data=train\_data, method="glmnet", family="binomial", trControl = trainControl("cv", number=10,summaryFunction = twoClassSummary,  
 classProbs = TRUE,   
 savePredictions = "final"),   
 tuneLength=10,  
 metric = "ROC")  
  
en\_heart$bestTune

## alpha lambda  
## 8 0.1 0.04402155

en\_heart$results

## alpha lambda ROC Sens Spec ROCSD SensSD  
## 1 0.1 0.0001253749 0.8983165 0.8651515 0.7822222 0.08716081 0.13672791  
## 2 0.1 0.0002896324 0.8983165 0.8651515 0.7822222 0.08716081 0.13672791  
## 3 0.1 0.0006690884 0.8992256 0.8651515 0.7822222 0.08563405 0.13672791  
## 4 0.1 0.0015456810 0.8992256 0.8651515 0.7922222 0.08563405 0.13672791  
## 5 0.1 0.0035707235 0.9010438 0.8469697 0.7922222 0.08367836 0.14221455  
## 6 0.1 0.0082488343 0.9056229 0.8742424 0.8022222 0.08032890 0.10595233  
## 7 0.1 0.0190558772 0.9131481 0.8833333 0.8022222 0.07500236 0.10449787  
## 8 0.1 0.0440215479 0.9158081 0.8840909 0.8033333 0.07006839 0.10189324  
## 9 0.1 0.1016954852 0.9144781 0.8750000 0.8033333 0.07080734 0.11198292  
## 10 0.1 0.2349297607 0.9108249 0.8750000 0.8033333 0.06578814 0.11198292  
## 11 0.2 0.0001253749 0.8983165 0.8651515 0.7822222 0.08716081 0.13672791  
## 12 0.2 0.0002896324 0.8983165 0.8651515 0.7822222 0.08716081 0.13672791  
## 13 0.2 0.0006690884 0.8992256 0.8651515 0.7822222 0.08563405 0.13672791  
## 14 0.2 0.0015456810 0.8992256 0.8651515 0.7922222 0.08563405 0.13672791  
## 15 0.2 0.0035707235 0.9010438 0.8469697 0.7922222 0.08367836 0.14221455  
## 16 0.2 0.0082488343 0.9038047 0.8742424 0.7922222 0.08336234 0.10595233  
## 17 0.2 0.0190558772 0.9122222 0.8833333 0.7822222 0.07501126 0.10449787  
## 18 0.2 0.0440215479 0.9110269 0.8840909 0.7933333 0.07329270 0.10189324  
## 19 0.2 0.1016954852 0.9080808 0.8750000 0.7933333 0.06968599 0.11198292  
## 20 0.2 0.2349297607 0.9080471 0.8750000 0.8033333 0.07124118 0.11198292  
## 21 0.3 0.0001253749 0.8973906 0.8651515 0.7822222 0.08753904 0.13672791  
## 22 0.3 0.0002896324 0.8973906 0.8651515 0.7822222 0.08753904 0.13672791  
## 23 0.3 0.0006690884 0.8992256 0.8651515 0.7822222 0.08563405 0.13672791  
## 24 0.3 0.0015456810 0.8992256 0.8651515 0.7922222 0.08563405 0.13672791  
## 25 0.3 0.0035707235 0.9010438 0.8469697 0.7922222 0.08367836 0.14221455  
## 26 0.3 0.0082488343 0.9038047 0.8651515 0.7922222 0.08158083 0.12256193  
## 27 0.3 0.0190558772 0.9094781 0.8924242 0.7822222 0.07744661 0.09270119  
## 28 0.3 0.0440215479 0.9072727 0.8840909 0.7933333 0.07356048 0.10189324  
## 29 0.3 0.1016954852 0.9054209 0.8750000 0.8133333 0.07301865 0.11198292  
## 30 0.3 0.2349297607 0.9060101 0.9015152 0.7822222 0.07667032 0.07799720  
## 31 0.4 0.0001253749 0.8973906 0.8651515 0.7822222 0.08753904 0.13672791  
## 32 0.4 0.0002896324 0.8973906 0.8651515 0.7822222 0.08753904 0.13672791  
## 33 0.4 0.0006690884 0.8992256 0.8651515 0.7822222 0.08563405 0.13672791  
## 34 0.4 0.0015456810 0.9001347 0.8651515 0.7922222 0.08461303 0.13672791  
## 35 0.4 0.0035707235 0.9001347 0.8469697 0.7922222 0.08461303 0.14221455  
## 36 0.4 0.0082488343 0.9019529 0.8651515 0.7922222 0.08216512 0.12256193  
## 37 0.4 0.0190558772 0.9086869 0.8924242 0.7822222 0.07626835 0.09270119  
## 38 0.4 0.0440215479 0.9045286 0.8659091 0.8033333 0.07275379 0.12787107  
## 39 0.4 0.1016954852 0.9015488 0.8750000 0.8133333 0.07703061 0.09416517  
## 40 0.4 0.2349297607 0.8958081 0.9106061 0.7400000 0.07457515 0.07229459  
## 41 0.5 0.0001253749 0.8973906 0.8651515 0.7822222 0.08753904 0.13672791  
## 42 0.5 0.0002896324 0.8973906 0.8651515 0.7822222 0.08753904 0.13672791  
## 43 0.5 0.0006690884 0.8992256 0.8651515 0.7922222 0.08563405 0.13672791  
## 44 0.5 0.0015456810 0.9001347 0.8651515 0.7922222 0.08461303 0.13672791  
## 45 0.5 0.0035707235 0.9001347 0.8560606 0.7922222 0.08461303 0.12960230  
## 46 0.5 0.0082488343 0.9019529 0.8651515 0.7922222 0.08216512 0.12256193  
## 47 0.5 0.0190558772 0.9077778 0.8750000 0.7822222 0.07762944 0.11990297  
## 48 0.5 0.0440215479 0.9017845 0.8659091 0.8033333 0.07346814 0.12787107  
## 49 0.5 0.1016954852 0.8978788 0.9015152 0.8133333 0.07340175 0.07799720  
## 50 0.5 0.2349297607 0.8848316 0.9196970 0.7100000 0.07245773 0.09846542  
## 51 0.6 0.0001253749 0.8973906 0.8651515 0.7822222 0.08753904 0.13672791  
## 52 0.6 0.0002896324 0.8973906 0.8651515 0.7822222 0.08753904 0.13672791  
## 53 0.6 0.0006690884 0.8992256 0.8651515 0.7922222 0.08563405 0.13672791  
## 54 0.6 0.0015456810 0.9001347 0.8651515 0.7922222 0.08461303 0.13672791  
## 55 0.6 0.0035707235 0.9001347 0.8560606 0.7922222 0.08461303 0.12960230  
## 56 0.6 0.0082488343 0.9019529 0.8560606 0.7822222 0.08426180 0.12960230  
## 57 0.6 0.0190558772 0.9048485 0.8750000 0.7822222 0.08180027 0.11990297  
## 58 0.6 0.0440215479 0.8997306 0.8666667 0.8033333 0.07688795 0.10449787  
## 59 0.6 0.1016954852 0.9007071 0.8931818 0.8022222 0.07481918 0.07038618  
## 60 0.6 0.2349297607 0.8782997 0.9030303 0.6788889 0.08519385 0.09746297  
## 61 0.7 0.0001253749 0.8973906 0.8651515 0.7822222 0.08753904 0.13672791  
## 62 0.7 0.0002896324 0.8973906 0.8651515 0.7822222 0.08753904 0.13672791  
## 63 0.7 0.0006690884 0.8992256 0.8651515 0.7922222 0.08563405 0.13672791  
## 64 0.7 0.0015456810 0.9001347 0.8651515 0.7922222 0.08461303 0.13672791  
## 65 0.7 0.0035707235 0.9010438 0.8560606 0.7922222 0.08367836 0.12960230  
## 66 0.7 0.0082488343 0.9010438 0.8560606 0.7822222 0.08573804 0.12960230  
## 67 0.7 0.0190558772 0.9029125 0.8659091 0.7822222 0.07920263 0.12787107  
## 68 0.7 0.0440215479 0.8988047 0.8575758 0.8133333 0.07696484 0.09514540  
## 69 0.7 0.1016954852 0.8997306 0.8931818 0.7922222 0.07880911 0.07038618  
## 70 0.7 0.2349297607 0.8764815 0.8946970 0.6677778 0.09370451 0.10716417  
## 71 0.8 0.0001253749 0.8973906 0.8651515 0.7922222 0.08753904 0.13672791  
## 72 0.8 0.0002896324 0.8973906 0.8651515 0.7922222 0.08753904 0.13672791  
## 73 0.8 0.0006690884 0.8992256 0.8651515 0.7922222 0.08563405 0.13672791  
## 74 0.8 0.0015456810 0.9001347 0.8651515 0.7922222 0.08461303 0.13672791  
## 75 0.8 0.0035707235 0.9019529 0.8560606 0.7922222 0.08216512 0.12960230  
## 76 0.8 0.0082488343 0.9001347 0.8560606 0.7822222 0.08675641 0.12960230  
## 77 0.8 0.0190558772 0.8992424 0.8659091 0.7822222 0.07838457 0.12787107  
## 78 0.8 0.0440215479 0.8979125 0.8575758 0.8133333 0.07733785 0.09514540  
## 79 0.8 0.1016954852 0.8949663 0.8931818 0.7600000 0.08021789 0.07038618  
## 80 0.8 0.2349297607 0.8658502 0.8674242 0.6788889 0.09986428 0.09416517  
## 81 0.9 0.0001253749 0.8973906 0.8651515 0.7922222 0.08753904 0.13672791  
## 82 0.9 0.0002896324 0.8973906 0.8651515 0.7922222 0.08753904 0.13672791  
## 83 0.9 0.0006690884 0.8992256 0.8651515 0.7922222 0.08563405 0.13672791  
## 84 0.9 0.0015456810 0.9001347 0.8651515 0.7922222 0.08461303 0.13672791  
## 85 0.9 0.0035707235 0.9019529 0.8560606 0.7922222 0.08216512 0.12960230  
## 86 0.9 0.0082488343 0.8992256 0.8560606 0.7822222 0.08659377 0.12960230  
## 87 0.9 0.0190558772 0.8973232 0.8575758 0.7822222 0.07875855 0.12067431  
## 88 0.9 0.0440215479 0.8987037 0.8575758 0.8133333 0.07533833 0.09514540  
## 89 0.9 0.1016954852 0.8865488 0.9022727 0.7300000 0.08047875 0.07811566  
## 90 0.9 0.2349297607 0.8625253 0.8674242 0.6255556 0.09918684 0.09416517  
## 91 1.0 0.0001253749 0.8973906 0.8651515 0.7922222 0.08753904 0.13672791  
## 92 1.0 0.0002896324 0.8973906 0.8651515 0.7922222 0.08753904 0.13672791  
## 93 1.0 0.0006690884 0.8992256 0.8651515 0.7922222 0.08563405 0.13672791  
## 94 1.0 0.0015456810 0.9001347 0.8560606 0.7922222 0.08461303 0.12960230  
## 95 1.0 0.0035707235 0.9010438 0.8560606 0.7922222 0.08367836 0.12960230  
## 96 1.0 0.0082488343 0.9001515 0.8742424 0.7822222 0.08629995 0.12206142  
## 97 1.0 0.0190558772 0.8982155 0.8575758 0.7722222 0.07874163 0.12067431  
## 98 1.0 0.0440215479 0.8968687 0.8750000 0.8022222 0.07470758 0.07603065  
## 99 1.0 0.1016954852 0.8828620 0.8939394 0.7300000 0.08216047 0.10101010  
## 100 1.0 0.2349297607 0.7887879 0.9037879 0.3466667 0.07927186 0.11225593  
## SpecSD  
## 1 0.09787874  
## 2 0.09787874  
## 3 0.09787874  
## 4 0.09355793  
## 5 0.09355793  
## 6 0.09962894  
## 7 0.09962894  
## 8 0.10887251  
## 9 0.08607427  
## 10 0.09813767  
## 11 0.09787874  
## 12 0.09787874  
## 13 0.09787874  
## 14 0.09355793  
## 15 0.09355793  
## 16 0.09355793  
## 17 0.09787874  
## 18 0.10346533  
## 19 0.09210241  
## 20 0.08607427  
## 21 0.09787874  
## 22 0.09787874  
## 23 0.09787874  
## 24 0.09355793  
## 25 0.09355793  
## 26 0.09355793  
## 27 0.09787874  
## 28 0.12308339  
## 29 0.10274691  
## 30 0.08577893  
## 31 0.09787874  
## 32 0.09787874  
## 33 0.09787874  
## 34 0.09355793  
## 35 0.09355793  
## 36 0.09355793  
## 37 0.09787874  
## 38 0.10887251  
## 39 0.10274691  
## 40 0.05083662  
## 41 0.09787874  
## 42 0.09787874  
## 43 0.09355793  
## 44 0.09355793  
## 45 0.09355793  
## 46 0.09355793  
## 47 0.09787874  
## 48 0.10887251  
## 49 0.10274691  
## 50 0.07322663  
## 51 0.09787874  
## 52 0.09787874  
## 53 0.09355793  
## 54 0.09355793  
## 55 0.09355793  
## 56 0.09787874  
## 57 0.09787874  
## 58 0.10887251  
## 59 0.09962894  
## 60 0.10778351  
## 61 0.09787874  
## 62 0.09787874  
## 63 0.09355793  
## 64 0.09355793  
## 65 0.09355793  
## 66 0.09787874  
## 67 0.09787874  
## 68 0.10274691  
## 69 0.10476311  
## 70 0.10203015  
## 71 0.09355793  
## 72 0.09355793  
## 73 0.09355793  
## 74 0.09355793  
## 75 0.09355793  
## 76 0.09787874  
## 77 0.09787874  
## 78 0.10274691  
## 79 0.08383798  
## 80 0.13713210  
## 81 0.09355793  
## 82 0.09355793  
## 83 0.09355793  
## 84 0.09355793  
## 85 0.09355793  
## 86 0.09787874  
## 87 0.09787874  
## 88 0.10274691  
## 89 0.06688237  
## 90 0.17372305  
## 91 0.09355793  
## 92 0.09355793  
## 93 0.09355793  
## 94 0.09355793  
## 95 0.09355793  
## 96 0.09787874  
## 97 0.10092168  
## 98 0.09962894  
## 99 0.06688237  
## 100 0.30238195

confusionMatrix(en\_heart)

## Cross-Validated (10 fold) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction No Yes  
## No 47.6 9.1  
## Yes 6.2 37.0  
##   
## Accuracy (average) : 0.8462

### Compare model performance

The model with the highest mean cross validation score of AUC will be selected as the final model.

#metric  
confusionMatrix(ctree\_heart) #0.7885

## Cross-Validated (10 fold) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction No Yes  
## No 46.2 13.5  
## Yes 7.7 32.7  
##   
## Accuracy (average) : 0.7885

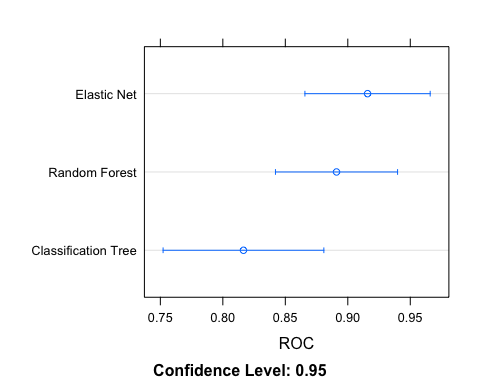
confusionMatrix(rf\_heart) #0.7788

## Cross-Validated (10 fold) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction No Yes  
## No 43.7 12.0  
## Yes 10.1 34.1  
##   
## Accuracy (average) : 0.7788

confusionMatrix(en\_heart) #0.8462

## Cross-Validated (10 fold) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction No Yes  
## No 47.6 9.1  
## Yes 6.2 37.0  
##   
## Accuracy (average) : 0.8462

#cross-validated ROC/AUC   
mod\_comparison <- resamples(list("Classification Tree" = ctree\_heart,  
 "Random Forest" = rf\_heart,  
 "Elastic Net" = en\_heart))  
  
dotplot(mod\_comparison, metric = "ROC")



#en\_heart$results %>% pull(ROC) %>% max()

**Model selection:**  Elastic Net is selected as the final model because it had the highest mean cross validation score (CV) for ROC of 0.9158081 and also the highest accuracy of 0.8462. Elastic net performed better than random forest, though there is a lot of overlap in the 95% confidence intervals.

## Step 3 - Model Evaluation

test\_pred <- predict(en\_heart, newdata = test\_data)  
test\_pred\_prob <- predict(en\_heart, newdata = test\_data, type = "prob")  
test\_pred\_df <- bind\_cols(test\_data, test\_pred\_prob)  
  
#Accuracy  
confusionMatrix(test\_pred, test\_data$HD, positive = "Yes")

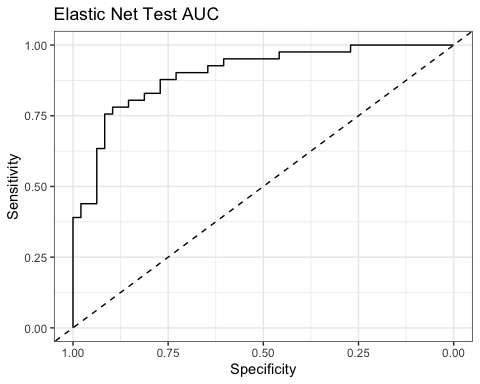
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 41 8  
## Yes 7 33  
##   
## Accuracy : 0.8315   
## 95% CI : (0.7373, 0.9025)  
## No Information Rate : 0.5393   
## P-Value [Acc > NIR] : 6.345e-09   
##   
## Kappa : 0.6602   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.8049   
## Specificity : 0.8542   
## Pos Pred Value : 0.8250   
## Neg Pred Value : 0.8367   
## Prevalence : 0.4607   
## Detection Rate : 0.3708   
## Detection Prevalence : 0.4494   
## Balanced Accuracy : 0.8295   
##   
## 'Positive' Class : Yes   
##

#ROC  
test\_roc <- roc(HD ~ Yes, data = test\_pred\_df)

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

ggroc(test\_roc) +  
 theme\_bw() +  
 labs(x = "Specificity", y = "Sensitivity", title = "Elastic Net Test AUC") +  
 geom\_abline(aes(intercept = 1, slope = 1), linetype = "dashed")



# AUC for the final model on the test set  
test\_auc <- test\_roc$auc  
test\_auc

## Area under the curve: 0.8989

The AUC for the elastic net model in the test set is **0.8988821.** On the testing set, the accuracy is **0.8315** (95% CI: 0.7373, 0.9025), recall or sensitivity was **0.8049** and precision/PPV was **0.825**.