

Project 2 - Regression and CART

Group 3

3/13/2021

Group Leader: Brianna Johnson

Member Names: Marco Sousa, Ben Pfeffer, Nikita Seleznev, Brianna Johnson

Introduction to the Heart Dataset

Observing the data:

##	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca
## 1	63	1	typical	145	233	1	2	150	0	2.3	3	0
## 2	67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3
## 3	67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2
## 4	37	1	nonanginal	130	250	0	0	187	0	3.5	3	0
## 5	41	0	nontypical	130	204	0	2	172	0	1.4	1	0
## 6	56	1	nontypical	120	236	0	0	178	0	0.8	1	0
##			Thal	AHD	AHDBinary							
## 1			fixed	No		0						
## 2			normal	Yes		1						
## 3			reversable	Yes		1						
## 4			normal	No		0						
## 5			normal	No		0						
## 6			normal	No		0						

Describing the data, and (1) identify Y and X.

The Heart dataset considers AHD, their binary relationship of having heart disease or not, and their other associated properties. The dataset contains 303 observations and 14 attributes.

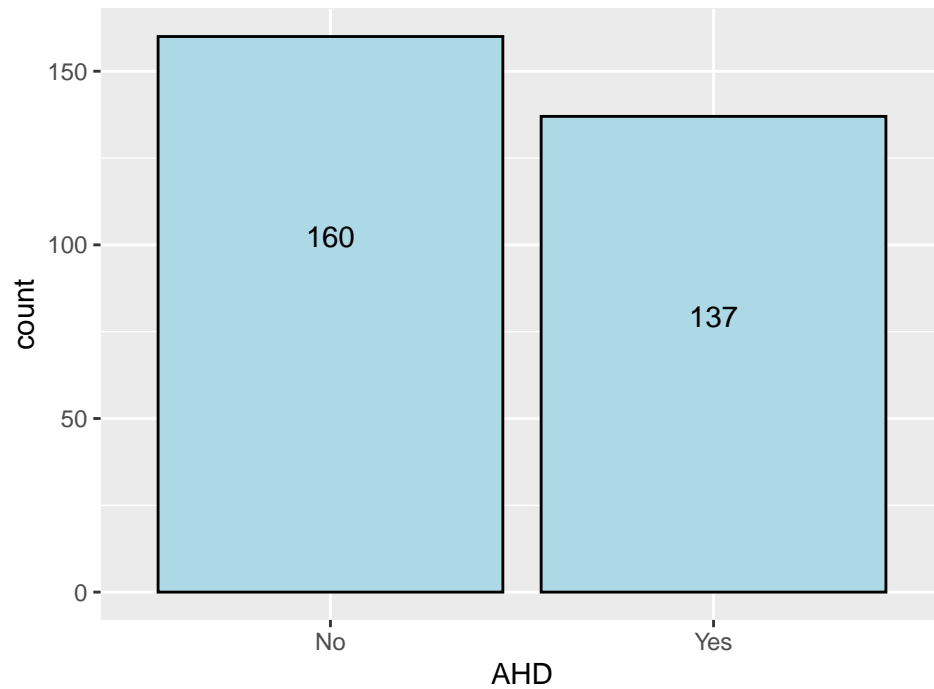
More to be added/edited

Exploratory Data Analysis

Count of Binary Outcome

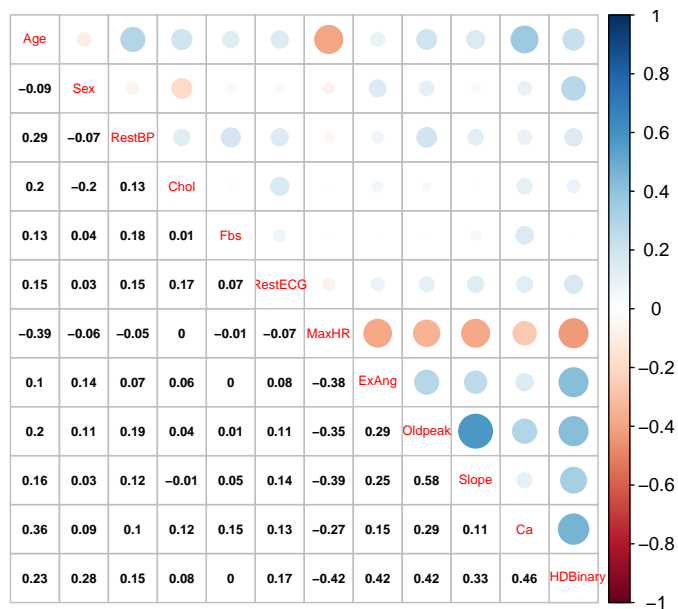
The following is a simple barplot of the count for the binary AHD outcome. We can see there are some more “No”, than “Yes”.

Count of AHD



Correlation matrix

No correlation among AHD Binary exceeds 0.5, naturally.



Logistic Model

Logistic Model

Constructing a logistic model for AHD based on other features. training and testing sets were separated randomly in a 70/30 split. To construct a final logistic model, features with the highest p values were removed until the remaining attributes were all significant. Some summaries were skipped over for brevity.

Splitting the data and setting seed.

```
set.seed(1)

indices <- sample(1:nrow(data), 0.7 * nrow(data), replace = TRUE)

training <- data[indices,]
test <- data[-indices,]
```

Carrying out the feature selection and yielding a final regression model.

```
# Entire glm fit for numeric data
glm.fit <- glm(AHDBinary ~ Age+Sex+RestBP+Chol+Fbs+RestECG+MaxHR+ExAng+Oldpeak+Slope+Ca, data = training)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = AHDBinary ~ Age + Sex + RestBP + Chol + Fbs + RestECG +
##      MaxHR + ExAng + Oldpeak + Slope + Ca, family = binomial,
##      data = training)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.8883  -0.5778  -0.1832   0.4322   2.6547
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.427167   3.212744  -1.378 0.168203
## Age         -0.046653   0.029954  -1.558 0.119350
## Sex          1.770038   0.524263   3.376 0.000735 ***
## RestBP       0.024170   0.014339   1.686 0.091854 .
## Chol         0.012568   0.004439   2.831 0.004637 **
## Fbs         -1.254899   0.670752  -1.871 0.061361 .
## RestECG     -0.077942   0.218381  -0.357 0.721158
## MaxHR       -0.027156   0.012998  -2.089 0.036686 *
## ExAng        1.792944   0.503486   3.561 0.000369 ***
## Oldpeak      0.037720   0.245289   0.154 0.877785
## Slope        1.234978   0.411564   3.001 0.002694 **
## Ca           1.695576   0.362786   4.674 2.96e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 282.89  on 206  degrees of freedom
## Residual deviance: 154.98  on 195  degrees of freedom
## AIC: 178.98
##
## Number of Fisher Scoring iterations: 6
```

```
# Considering non-numeric categorical data
```

```
glm.fit.cat <- glm(AHDBinary ~ChestPain+Thal, data = training, family = binomial)
summary(glm.fit.cat)
```

```
##
## Call:
## glm(formula = AHDBinary ~ ChestPain + Thal, family = binomial,
##      data = training)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.1222  -0.5724  -0.1844   0.4715   2.8576
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.8879     0.7777   1.142  0.2536
## ChestPainnonanginal -1.8308     0.4452  -4.113 3.91e-05 ***
## ChestPainnontypical -3.8667     0.8128  -4.757 1.96e-06 ***
## ChestPaintypical    -1.0123     0.6070  -1.668  0.0954 .
## Thalnormal         -1.0871     0.8114  -1.340  0.1803
## Thalreversible      1.2529     0.8419   1.488  0.1367
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 282.89  on 206  degrees of freedom
## Residual deviance: 176.17  on 201  degrees of freedom
## AIC: 188.17
##
## Number of Fisher Scoring iterations: 5
```

```
# Altogether
```

```
glm.fit <- glm(AHDBinary ~ Age+Sex+RestBP+Chol+Fbs+RestECG+MaxHR+ExAng+Oldpeak+Slope+Ca+ChestPain+Thal
summary(glm.fit)
```

```
##
## Call:
## glm(formula = AHDBinary ~ Age + Sex + RestBP + Chol + Fbs + RestECG +
##      MaxHR + ExAng + Oldpeak + Slope + Ca + ChestPain + Thal,
##      family = binomial, data = training)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.0063  -0.4291  -0.1109   0.2791   2.5940
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -8.522904    3.965361  -2.149 0.031608 *
## Age            -0.012372    0.034110  -0.363 0.716829
## Sex             1.837955    0.666409   2.758 0.005816 **
## RestBP          0.026716    0.015309   1.745 0.080956 .
## Chol            0.010956    0.004845   2.261 0.023756 *
```

```
## Fbs -1.015934 0.808116 -1.257 0.208695
## RestECG -0.119485 0.254429 -0.470 0.638628
## MaxHR -0.012725 0.015260 -0.834 0.404342
## ExAng 1.294977 0.600853 2.155 0.031144 *
## Oldpeak -0.045660 0.303666 -0.150 0.880479
## Slope 1.370356 0.518486 2.643 0.008218 **
## Ca 1.700924 0.407042 4.179 2.93e-05 ***
## ChestPainnonanginal -2.045326 0.619620 -3.301 0.000964 ***
## ChestPainnontypical -3.158668 0.935319 -3.377 0.000733 ***
## ChestPaintypical -1.983980 0.763728 -2.598 0.009383 **
## Thalnormal 1.010428 1.224139 0.825 0.409134
## Thalreversible 2.097811 1.203044 1.744 0.081202 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 282.89 on 206 degrees of freedom
## Residual deviance: 123.51 on 190 degrees of freedom
## AIC: 157.51
##
## Number of Fisher Scoring iterations: 6
```

Removing highest param (Oldpeak)

```
glm.fit <- glm(AHDBinary ~ Age+Sex+RestBP+Chol+Fbs+RestECG+MaxHR+ExAng+Slope+Ca+ChestPain+Thal, data =
summary(glm.fit))
```

```
##
## Call:
## glm(formula = AHDBinary ~ Age + Sex + RestBP + Chol + Fbs + RestECG +
## MaxHR + ExAng + Slope + Ca + ChestPain + Thal, family = binomial,
## data = training)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
## -3.0063 -0.4277 -0.1120 0.2755 2.5968
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.579505 3.950732 -2.172 0.029884 *
## Age -0.011101 0.033006 -0.336 0.736619
## Sex 1.842278 0.666528 2.764 0.005710 **
## RestBP 0.026199 0.014899 1.758 0.078675 .
## Chol 0.010847 0.004786 2.266 0.023432 *
## Fbs -1.005109 0.807074 -1.245 0.212994
## RestECG -0.112775 0.250381 -0.450 0.652411
## MaxHR -0.012364 0.015084 -0.820 0.412395
## ExAng 1.285547 0.597140 2.153 0.031331 *
## Slope 1.347793 0.496717 2.713 0.006660 **
## Ca 1.688378 0.398209 4.240 2.24e-05 ***
## ChestPainnonanginal -2.052046 0.619517 -3.312 0.000925 ***
## ChestPainnontypical -3.136389 0.921067 -3.405 0.000661 ***
## ChestPaintypical -2.012248 0.741042 -2.715 0.006619 **
## Thalnormal 1.040552 1.205186 0.863 0.387920
```

```
## Thalreversible      2.118402   1.193274   1.775 0.075851 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 282.89  on 206  degrees of freedom
## Residual deviance: 123.53  on 191  degrees of freedom
## AIC: 155.53
##
## Number of Fisher Scoring iterations: 6
```

```
# Removing highest param (Age?!)
glm.fit <- glm(AHDBinary ~ Sex+RestBP+Chol+Fbs+RestECG+MaxHR+ExAng+Slope+Ca+ChestPain+Thal, data = tra

# Removing highest param (RestECG)
glm.fit <- glm(AHDBinary ~ Sex+RestBP+Chol+Fbs+MaxHR+ExAng+Slope+Ca+ChestPain+Thal, data = training, f

# Removing highest params (MaxHR)
glm.fit <- glm(AHDBinary ~ Sex+RestBP+Chol+Fbs+ExAng+Slope+Ca+ChestPain+Thal, data = training, family =

# Removing rest of nonsignificant params in order (skipping ahead)
glm.fit5 <- glm(AHDBinary ~ Sex+Chol+ExAng+Slope+Ca+ChestPain+Thal, data = training, family = binomial,
summary(glm.fit5)
```

```
##
## Call:
## glm(formula = AHDBinary ~ Sex + Chol + ExAng + Slope + Ca + ChestPain +
## Thal, family = binomial, data = training)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.8391  -0.4416  -0.1502   0.2739   2.5677
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -7.075262    2.146256  -3.297 0.000979 ***
## Sex              1.560495    0.581312   2.684 0.007265 **
## Chol             0.009321    0.004352   2.142 0.032189 *
## ExAng            1.448660    0.550374   2.632 0.008485 **
## Slope            1.210650    0.447351   2.706 0.006804 **
## Ca               1.633395    0.369119   4.425 9.64e-06 ***
## ChestPainnonanginal -2.184537    0.594776  -3.673 0.000240 ***
## ChestPainnontypical -3.058687    0.840206  -3.640 0.000272 ***
## ChestPaintypical   -1.884966    0.702410  -2.684 0.007284 **
## Thalnormal         0.964779    1.132919   0.852 0.394443
## Thalreversible     2.236792    1.108621   2.018 0.043629 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 282.89  on 206  degrees of freedom
## Residual deviance: 128.12  on 196  degrees of freedom
```

```
## AIC: 150.12
##
## Number of Fisher Scoring iterations: 6
```

Confusion Table

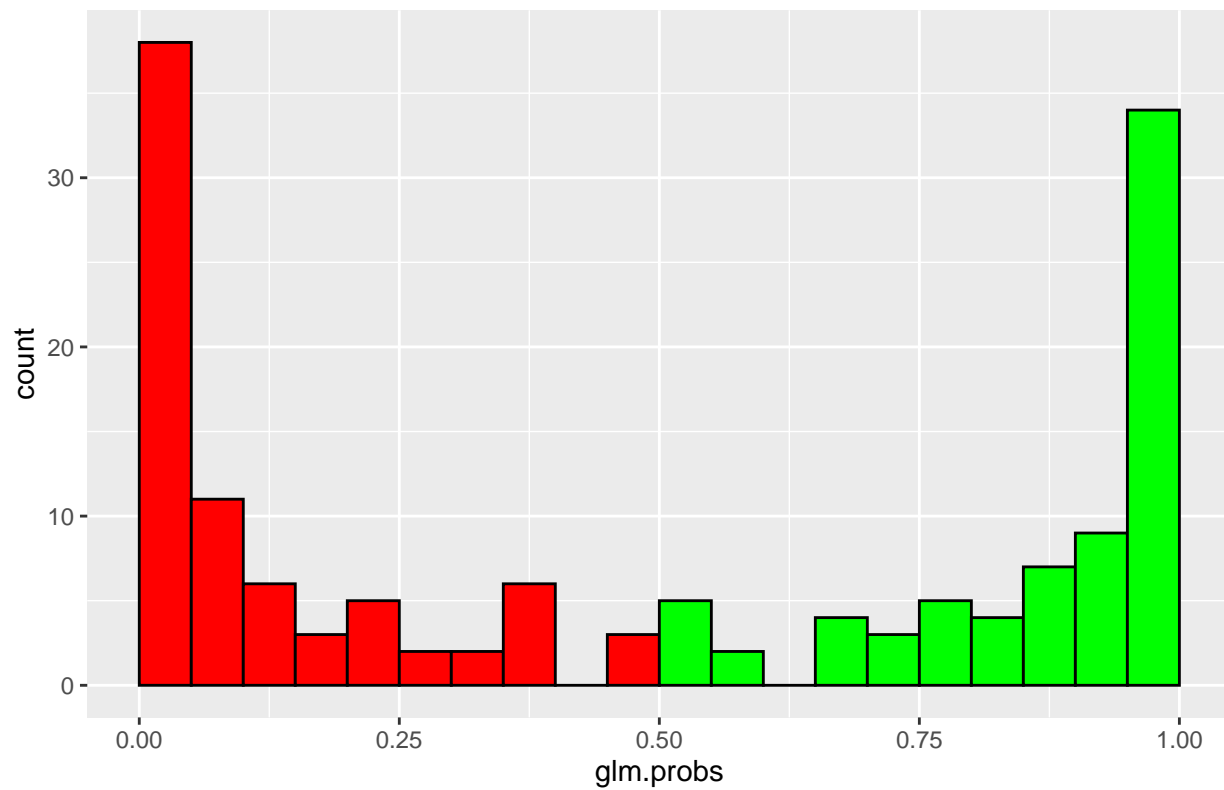
Constructing probability distribution for predictions on each test observation.

```
glm.probs = predict(glm.fit5, test, type = "response")
#The first 10 predicted probabilities
glm.probs[1:10]
```

```
##          2          3          4          5          6          7
## 0.998979144 0.997511014 0.316001177 0.002336079 0.014800821 0.963959548
##          8          9          10         11
## 0.462227736 0.958686279 0.975752401 0.213535076
```

```
#Visualizing our probability distribution
colors <- c(rep("red",10), rep("green",10))
probHist = ggplot(mapping = aes(glm.probs)) + geom_histogram(binwidth=0.05,boundary = 0,color="black",
probHist = probHist + ggtitle("Histogram of Probabilities") + theme(plot.title = element_text(hjust = 0
probHist
```

Histogram of Probabilities



Generate confusion table based off of 0.5 prediction cutoff.

```

#Choosing 0.5 as the cutoff for prediction
glm.pred <- ifelse(glm.probs > 0.5,1,0)
#Constructing the table
glm.table = table(glm.pred,test$AHDBinary)
glm.table

```

```

##
## glm.pred  0  1
##           0 65 11
##           1 14 59

```

Mode-Test Statistics

Calculate classification accuracy and error, sensitivity, specificity, PPV and NPV.

```

#Accuracy
table.trace = sum(diag(glm.table))
table.sum = sum(glm.table)
acc = table.trace / table.sum
acc

```

```

## [1] 0.8322148

```

```

#0.8754209

```

```

#error
err = 1 - acc
err

```

```

## [1] 0.1677852

```

```

#sensitivity
sens = glm.table[1]/(glm.table[1] + glm.table[2])
sens

```

```

## [1] 0.8227848

```

```

#Specificity
spec = glm.table[4]/(glm.table[4] + glm.table[3])
spec

```

```

## [1] 0.8428571

```

```

#PPV - Positive Predictive Value
PPV = glm.table[1]/(glm.table[1] + glm.table[3])
PPV

```

```

## [1] 0.8552632

```



```
#NPV - Negative Predictive Value
PPV = glm.table[4]/(glm.table[4] + glm.table[2])
PPV
```

```
## [1] 0.8082192
```

ROC and AUC

Generate ROC and compute AUC for each model

```
test_prob = predict(glm.fit5, newdata = test, type = "response")
```

```
test_roc = roc(test$AHDBinary, test_prob)
```

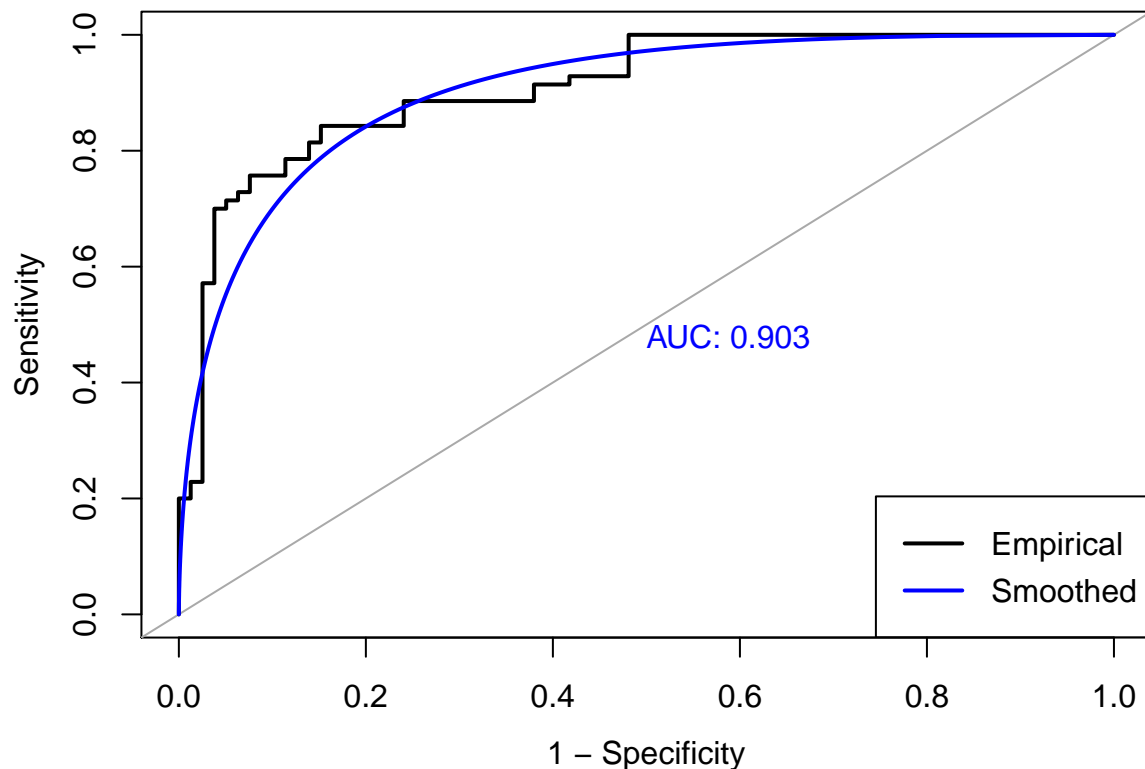
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot.roc(test_roc, col=par("fg"), print.auc=FALSE, legacy.axes=TRUE, asp =NA)
```

```
plot.roc(smooth(test_roc), col="blue", add=TRUE, print.auc=TRUE, legacy.axes = TRUE, asp =NA)
legend("bottomright", legend=c("Empirical", "Smoothed"), col=c(par("fg"), "blue"), lwd=2)
```

```
abline(v = -coef(glm.fit5)[1] / coef(glm.fit5)[2], lwd = 3)
```



S sigmoid curve

Generate s-curve for Y against one attribute (you can pick any one attribute), and interpret your findings

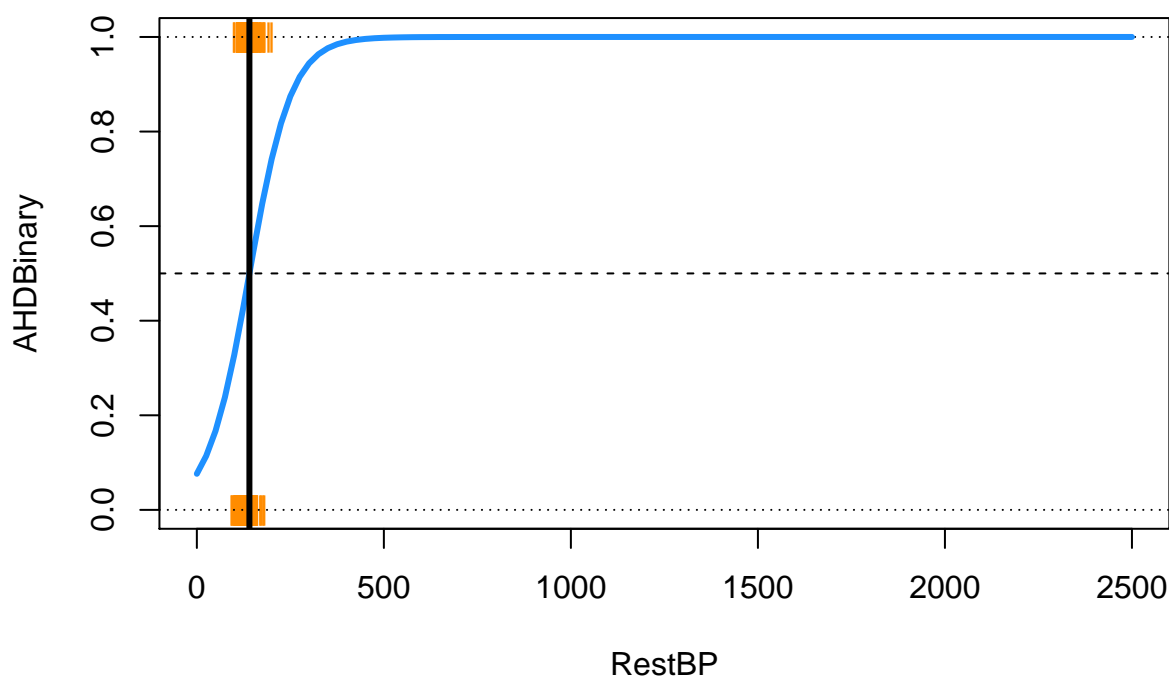
I used RestBP TEMPORARILY. This will be replaced with something more appropriate when we try our CART model.

```
single.glm <- glm(AHDBinary ~ RestBP, data = data, family = "binomial")

plot(AHDBinary ~ RestBP, data = data, col = "darkorange", pch = "|", xlim = c(0, 2500), ylim = c(0, 1),
     abline(h = 0, lty = 3)
     abline(h = 1, lty = 3)
     abline(h = 0.5, lty = 2)

curve(predict(single.glm, data.frame(RestBP = x), type = "response"), add = TRUE, lwd = 3, col = "dodgerblue",
      abline(v = -coef(single.glm)[1] / coef(single.glm)[2], lwd = 3))
```

Using Logistic Regression for Classification



CART Model

CART MODEL

Other subsections

Other Models (Bagging, Random Forests, Boosting)

```
Heart <- read.csv('Heart.csv')
Heart <- na.omit(Heart) #Remove NA for demo
data <- Heart[, -1]
set.seed(490)
split <- sample(1:nrow(data), size=nrow(data) * 0.7)
train <- data[split,]
test <- data[-split,]
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.4
```

```
## 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
```

Bagged Trees

```
train$AHD = factor(train$AHD)
```

```
bag.class <- randomForest(AHD~., data = train, mtry=ncol(train) - 1, importance=TRUE, xtest=test[, -14], ytest=test[, 14])
```

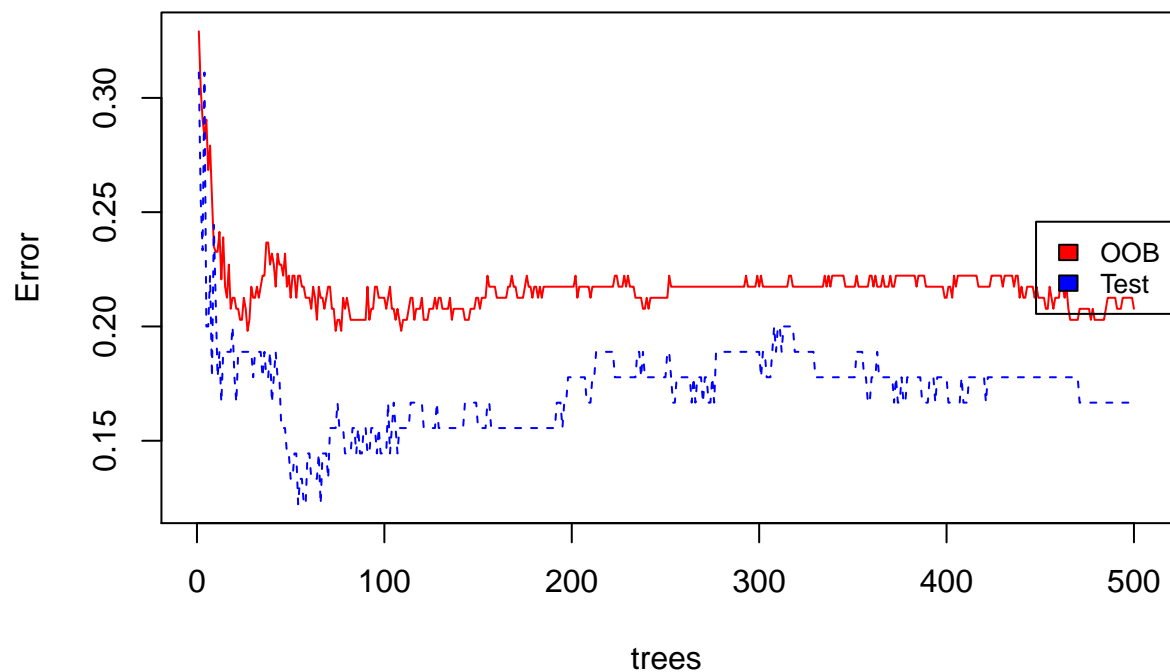
```
err <- bag.class$herr.rate[, 1]
```

```
bag.err <- cbind(err, bag.class$test$herr.rate[, 1])
```

```
colnames(bag.err) <- c("OOB", "Test")
```

```
matplot(1:bag.class$ntree, bag.err, type = "l", xlab="trees", ylab="Error", col = c("red", "blue"))
```

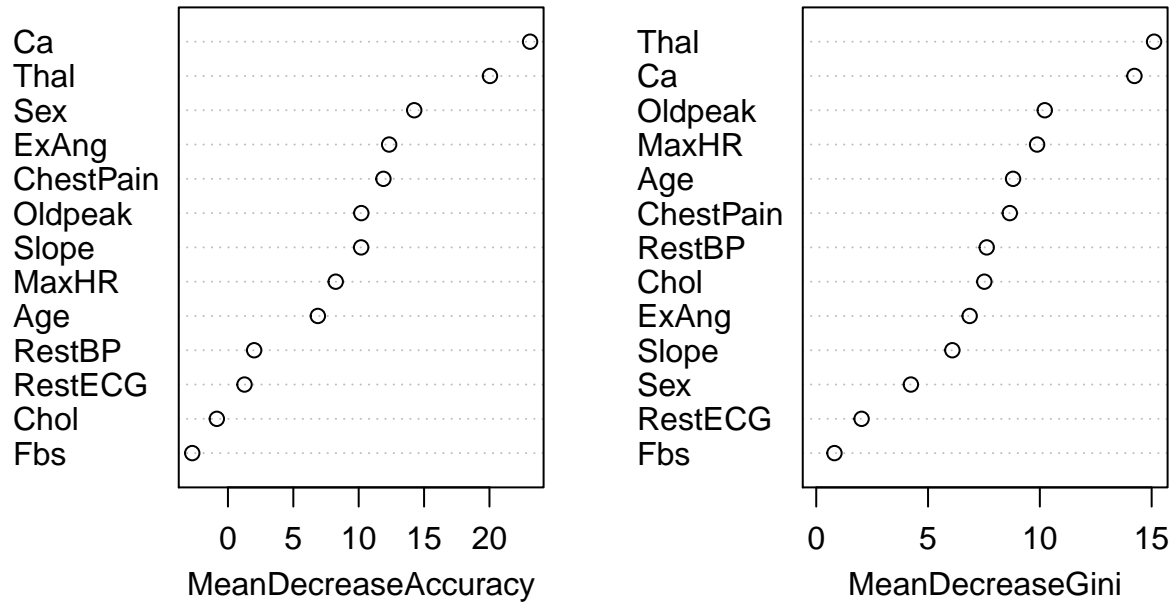
```
legend("right", c('OOB', 'Test'), col=seq_len(2), cex=0.8, fill=c("red", "blue"))
```



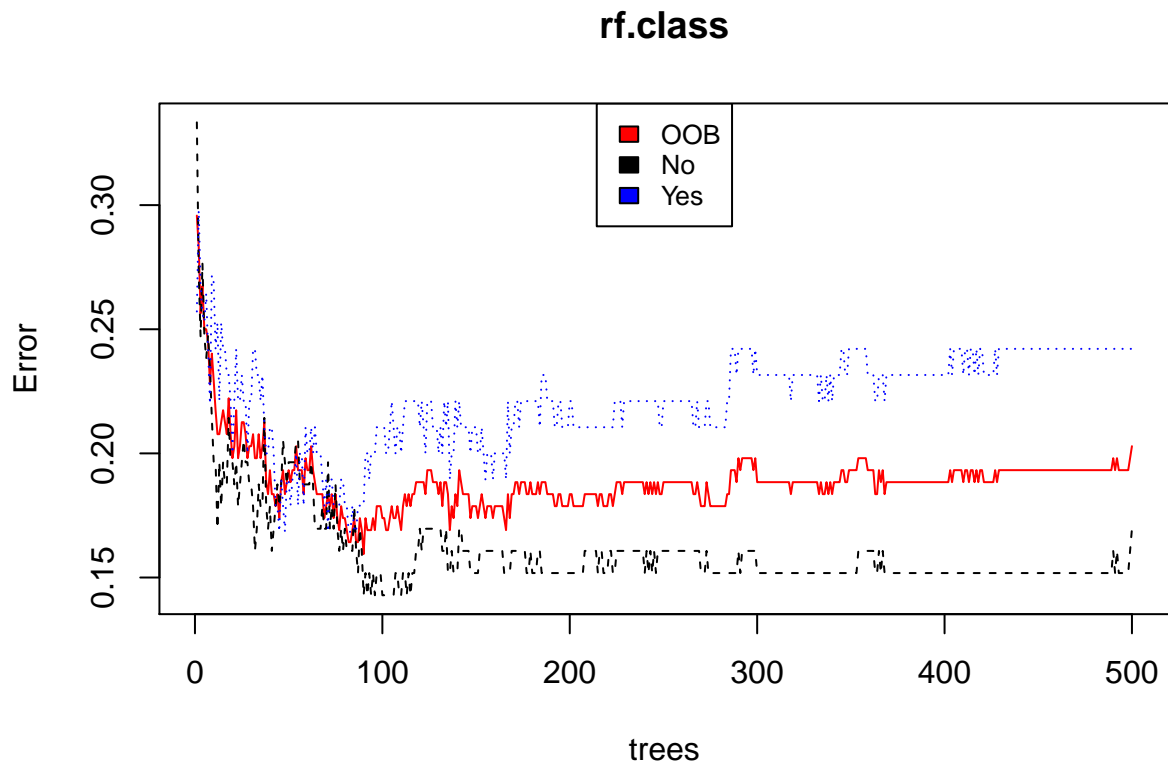
Random Forests

```
# setting mtry to sqrt(p) is a rule of thumb, this number can be set by
# k fold CV as well
rf.class <- randomForest(AHD~., data=train, mtry=round(sqrt(ncol(train)) - 1), importance=TRUE)
#importance(rf.class)
varImpPlot(rf.class)
```

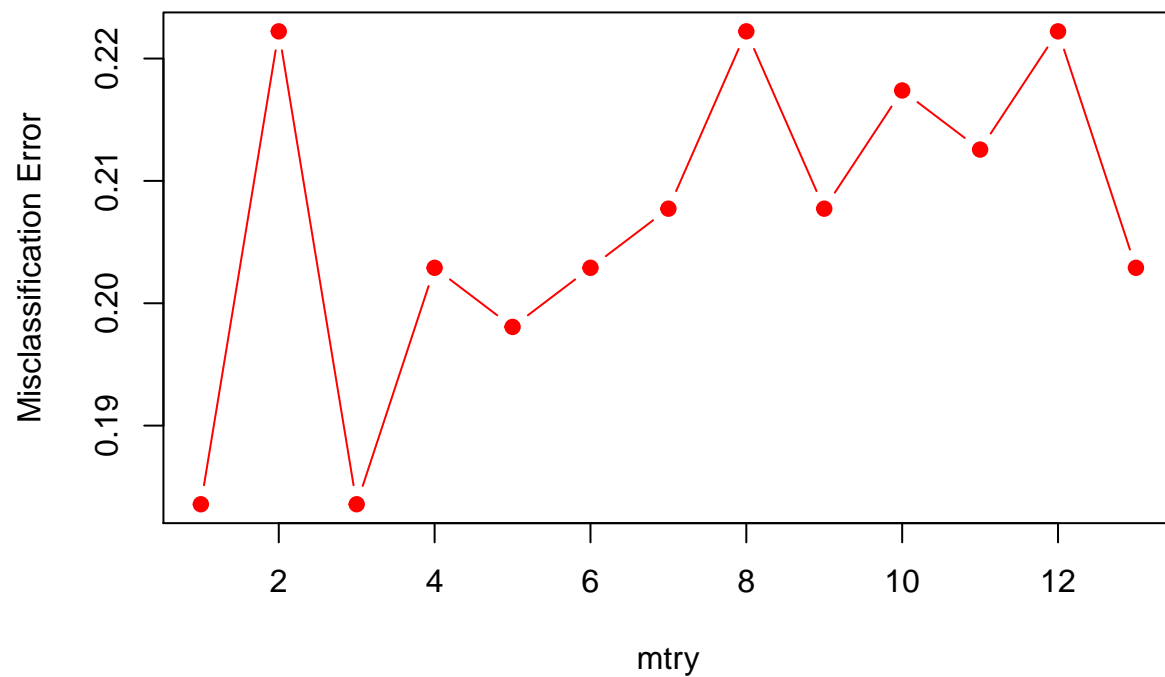
rf.class



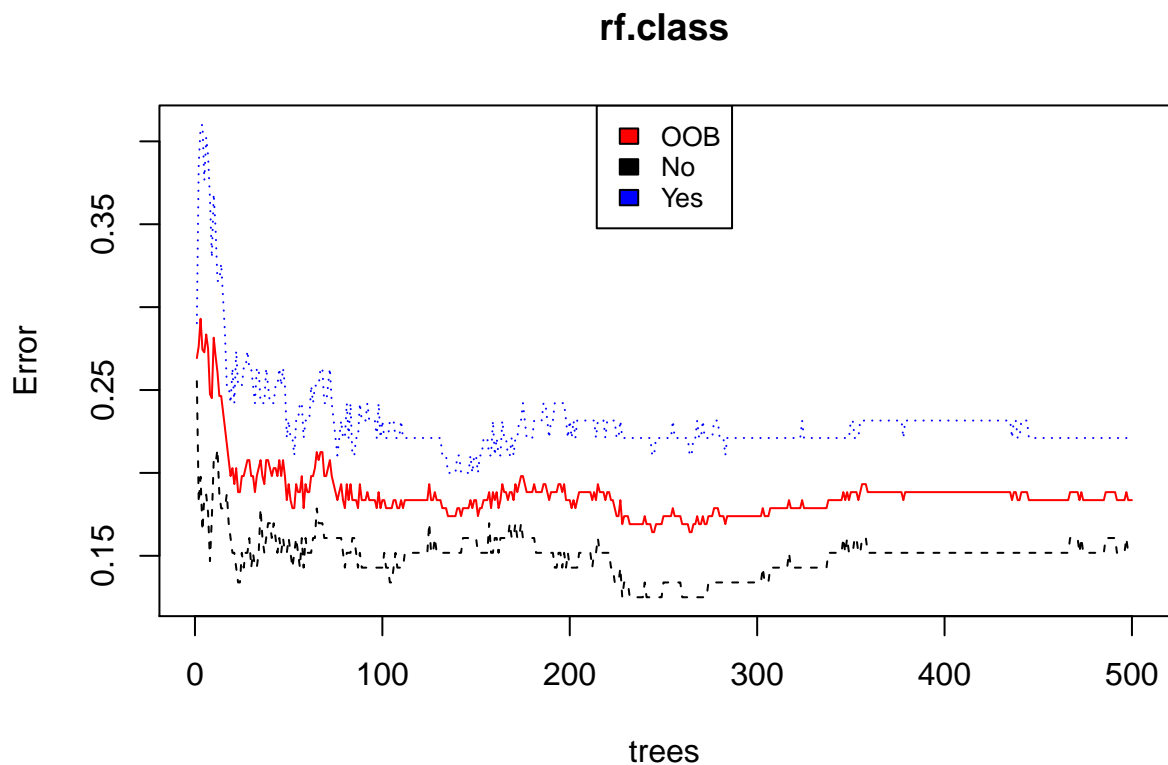
```
plot(rf.class, col=c("red", "black", "blue"))
legend("top", colnames(rf.class$err.rate) ,col=seq_len(3),cex=0.8,fill=c("red", "black", "blue"))
```



```
#Try choosing mtry by plotting the OOB error
p <- ncol(train) - 1
oob.error.class <- double(p) #initialize empty vector
set.seed(1)
for(m in 1:p) {
  fit <- randomForest(AHD ~ ., data=train, mtry=m, ntree=175)
  conf.mat <- fit$err.rate[175]
  oob.error.class[m] <- fit$err.rate[175, 'OOB']
}
matplot(1:p, oob.error.class, pch=19, col="red", type="b", ylab="Misclassification Error", xlab="mtry")
```



```
# setting mtry to sqrt(p) is a rule of thumb, this number can be set by  
# k fold CV as well  
rf.class <- randomForest(AHD~., data=train, mtry=3, importance=TRUE)  
plot(rf.class, col=c("red", "black", "blue"))  
legend("top", colnames(rf.class$err.rate) ,col=seq_len(3),cex=0.8,fill=c("red", "black", "blue"))
```



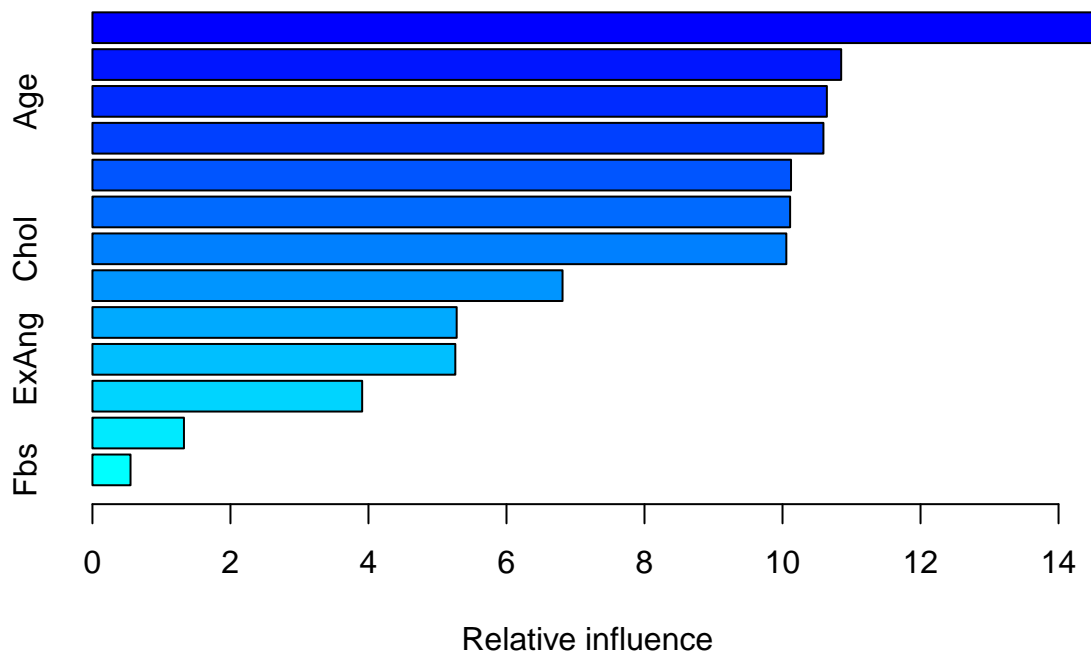
Boosting

```
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.0.4
```

```
## Loaded gbm 2.1.8
```

```
set.seed(1)
#format y for gbm, must be 0/1
train$ChestPain = as.numeric(factor(train$ChestPain))
train$Thal = as.numeric(factor(train$Thal))
AHD.0.1 <- ifelse(train$AHD == 'Yes', 1, 0)
class.boost = gbm(AHD.0.1 ~ . - AHD, data = train, n.trees = 5000, distribution = "adaboost", shrinkage = 0.05)
summary(class.boost)
```

```
##          var    rel.inf
## RestBP    RestBP 14.4899486
## Ca        Ca    10.8500863
## Age       Age    10.6427662
## Oldpeak   Oldpeak 10.5927425
## MaxHR     MaxHR  10.1234250
## Thal      Thal   10.1101401
## Chol      Chol   10.0551703
## ChestPain ChestPain 6.8123290
## Slope     Slope  5.2781342
## ExAng     ExAng  5.2586456
## Sex       Sex    3.9085508
## RestECG   RestECG 1.3262240
## Fbs       Fbs    0.5518371
```

Comparing Models

Citations