

Final

Shengzhi Luo

06/05/2022

Data load and pre-process

```
# Load data, clean column names, eliminate indexes containing NA entries
heartattack = read_csv("heartattack.csv") %>%
  mutate(
    sex = factor(sex,ordered=TRUE),
    exng = factor(exng,ordered = TRUE),
    cp = factor(cp,ordered = TRUE),
    fbs = factor(fbs,ordered = TRUE),
    restecg = factor(restecg,ordered = TRUE),
    output = factor(case_when(output== 0 ~ "less",
                              output == 1 ~ "more")),
    output = fct_relevel(output, "less")
  )%>%
  na.omit()%>%
  dplyr::select(-thall, -slp)

knitr::opts_chunk$set(warning = FALSE, message = FALSE, fig.align = "center")
theme_set(theme_minimal() + theme(legend.position = "bottom"))
options(
  ggplot2.continuous.colour = "viridis",
  ggplot2.continuous.fill = "viridis"
)
scale_colour_discrete = scale_colour_viridis_d
scale_fill_discrete = scale_fill_viridis_d

set.seed(100)
# Partition data into training/test sets
indexTrain = createDataPartition(y = heartattack$output,
                                p = 0.7,
                                list = FALSE)

training_df = heartattack[indexTrain, ]
testing_df = heartattack[-indexTrain, ]
# Create matrices for future analysis
# Training data
x_train = model.matrix(output~.,training_df)[, -11]
y_train = training_df$output
# Testing data
x_test <- model.matrix(output~.,testing_df)[, -11]
y_test <- testing_df$output
x_train_df = data.frame(x_train)
```

```
summary(heartattack)
```

```
##      age      sex      cp      trtbps      chol      fbs
## Min.   :29.00  0: 96  0:143  Min.   : 94.0  Min.   :126.0  0:258
## 1st Qu.:47.50  1:207  1: 50  1st Qu.:120.0  1st Qu.:211.0  1: 45
## Median :55.00      2: 87  Median :130.0  Median :240.0
## Mean   :54.37      3: 23  Mean   :131.6  Mean   :246.3
## 3rd Qu.:61.00      3rd Qu.:140.0  3rd Qu.:274.5
## Max.   :77.00      Max.   :200.0  Max.   :564.0
## restecg  thalachh  exng  oldpeak  caa  output
## 0:147  Min.   : 71.0  0:204  Min.   :0.00  Min.   :0.0000  less:138
## 1:152  1st Qu.:133.5  1: 99  1st Qu.:0.00  1st Qu.:0.0000  more:165
## 2: 4  Median :153.0  Median :0.80  Median :0.0000
##      Mean   :149.6  Mean   :1.04  Mean   :0.7294
##      3rd Qu.:166.0  3rd Qu.:1.60  3rd Qu.:1.0000
##      Max.   :202.0  Max.   :6.20  Max.   :4.0000
```

```
skimr::skim_without_charts(heartattack)
```

Table 1: Data summary

Name	heartattack
Number of rows	303
Number of columns	12
Column type frequency:	
factor	6
numeric	6
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
sex	0	1	TRUE	2	1: 207, 0: 96
cp	0	1	TRUE	4	0: 143, 2: 87, 1: 50, 3: 23
fbs	0	1	TRUE	2	0: 258, 1: 45
restecg	0	1	TRUE	3	1: 152, 0: 147, 2: 4
exng	0	1	TRUE	2	0: 204, 1: 99
output	0	1	FALSE	2	mor: 165, les: 138

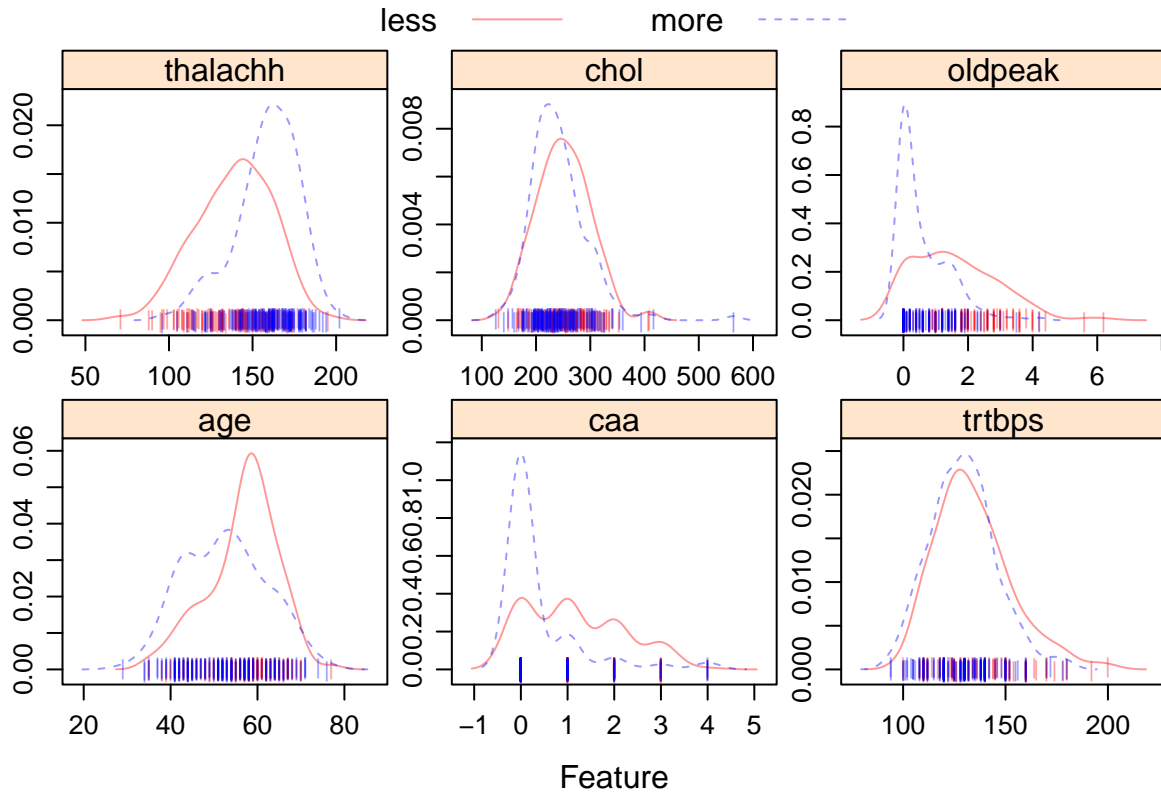
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
age	0	1	54.37	9.08	29	47.5	55.0	61.0	77.0
trtbps	0	1	131.62	17.54	94	120.0	130.0	140.0	200.0
chol	0	1	246.26	51.83	126	211.0	240.0	274.5	564.0
thalachh	0	1	149.65	22.91	71	133.5	153.0	166.0	202.0
oldpeak	0	1	1.04	1.16	0	0.0	0.8	1.6	6.2
caa	0	1	0.73	1.02	0	0.0	0.0	1.0	4.0

```

theme1 = transparentTheme(trans = 0.4)
trellis.par.set(theme1)
featurePlot(x = heartattack %>% dplyr::select(age, caa, trtbps, thalachh, chol, oldpeak),
            y = heartattack$output,
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free")),
            plot = "density", pch = "|",
            auto.key = list(columns = 2))

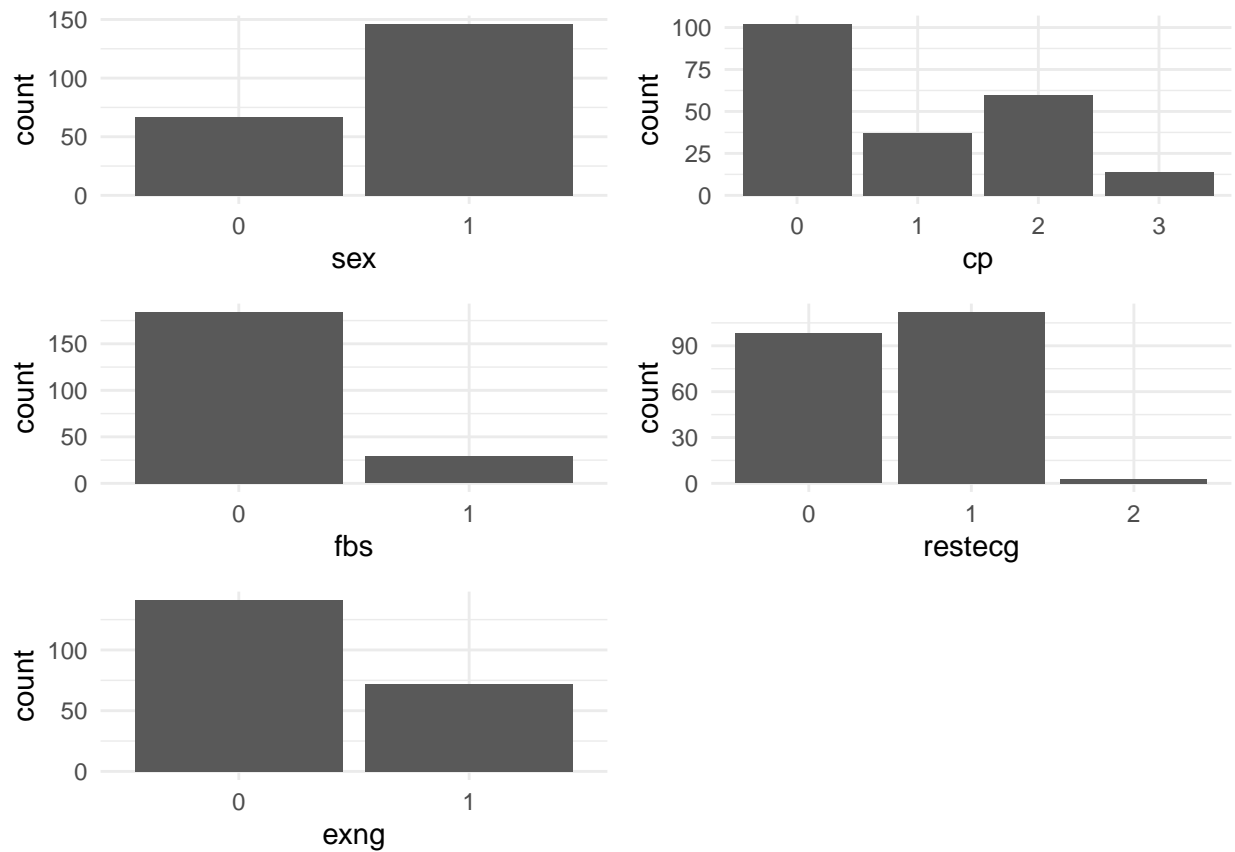
```



```

par(mfrow = c(2, 3))
fp1 <- ggplot(training_df) + geom_bar(aes(x = sex))
fp2 <- ggplot(training_df) + geom_bar(aes(x = cp))
fp3 <- ggplot(training_df) + geom_bar(aes(x = fbs))
fp4 <- ggplot(training_df) + geom_bar(aes(x = restecg))
fp5 <- ggplot(training_df) + geom_bar(aes(x = exng))
figure <- ggarrange(fp1, fp2, fp3, fp4, fp5,
                    ncol = 2, nrow = 3)
figure

```



glm

```
set.seed(100)
ctrl = trainControl(method = "cv",
                    summaryFunction = twoClassSummary,
                    classProbs = TRUE)

model.glm <- train(output ~ . , heartattack,
                  subset = indexTrain,
                  method = "glm",
                  metric = "ROC",
                  trControl = ctrl)
```

penalized glm

```
glmGrid <- expand.grid(.alpha = seq(0, 1, length = 11),
                     .lambda = exp(seq(2, -2, length = 35)))

set.seed(100)
model.glmn <- train(output ~ . , heartattack,
                  subset = indexTrain,
                  method = "glmnet",
                  tuneGrid = glmGrid,
                  metric = "ROC",
                  trControl = ctrl)
```

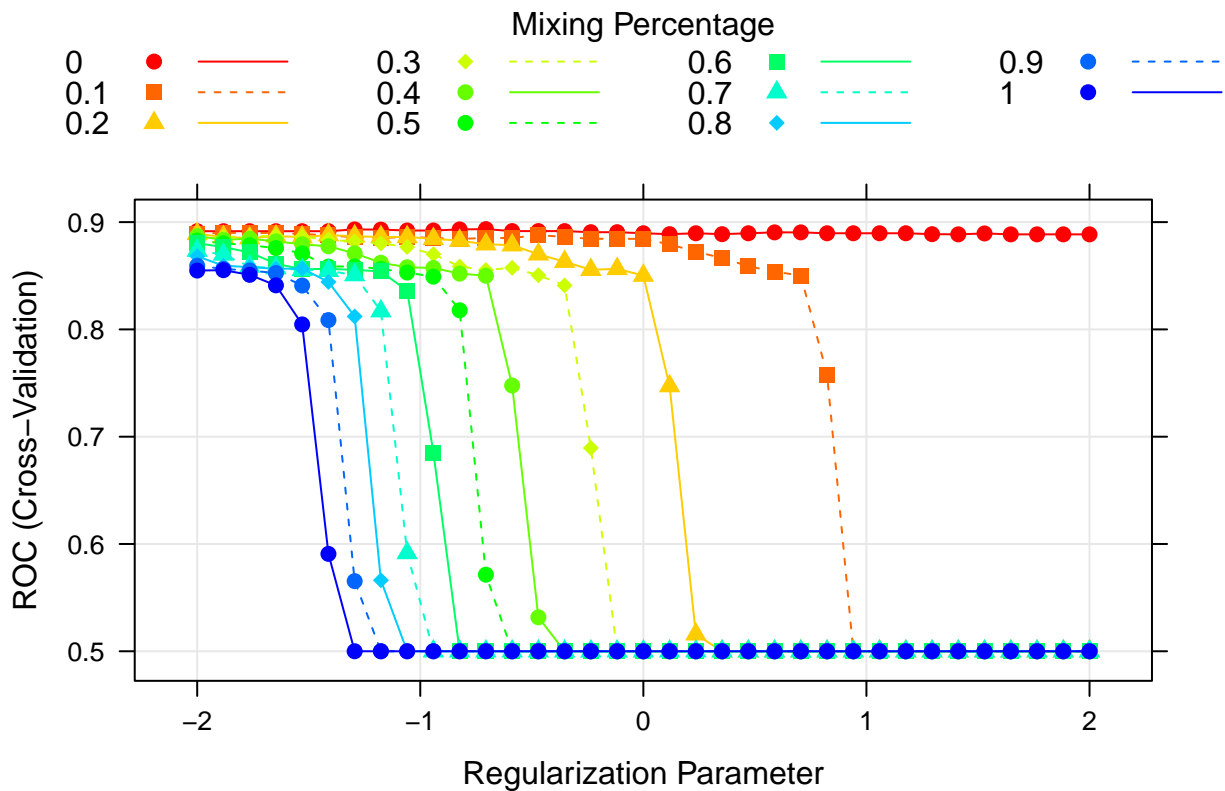
```
model.glmn$bestTune
```

```
##      alpha      lambda
## 12      0 0.4936728
```

```
myCol<- rainbow(15)
```

```
myPar <- list(superpose.symbol = list(col = myCol),
              superpose.line = list(col = myCol))
```

```
plot(model.glmn, par.settings = myPar, xTrans = function(x) log(x))
```

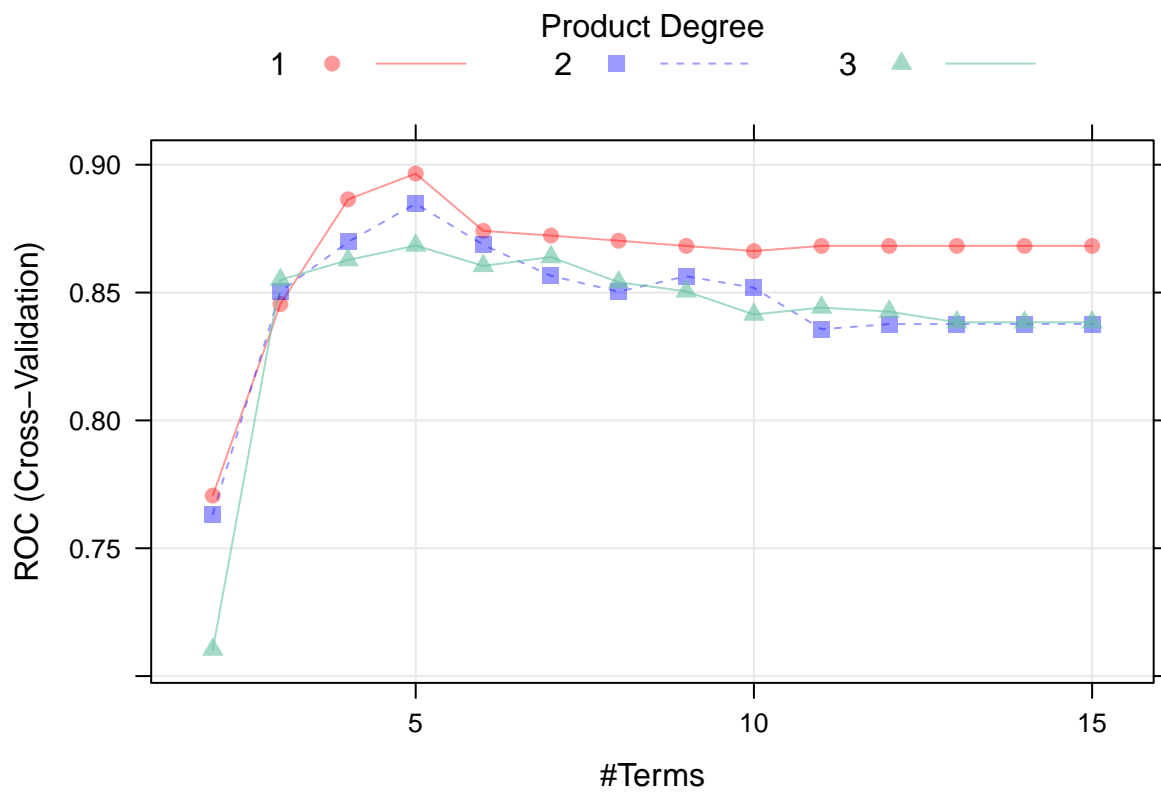


Mars

```
set.seed(100)
```

```
model.mars <- train(output ~ . , heartattack,
                    subset = indexTrain,
                    method = "earth",
                    tuneGrid = expand.grid(degree = 1:3,
                                          nprune = 2:15),
                    metric = "ROC",
                    trControl = ctrl)
```

```
plot(model.mars)
```



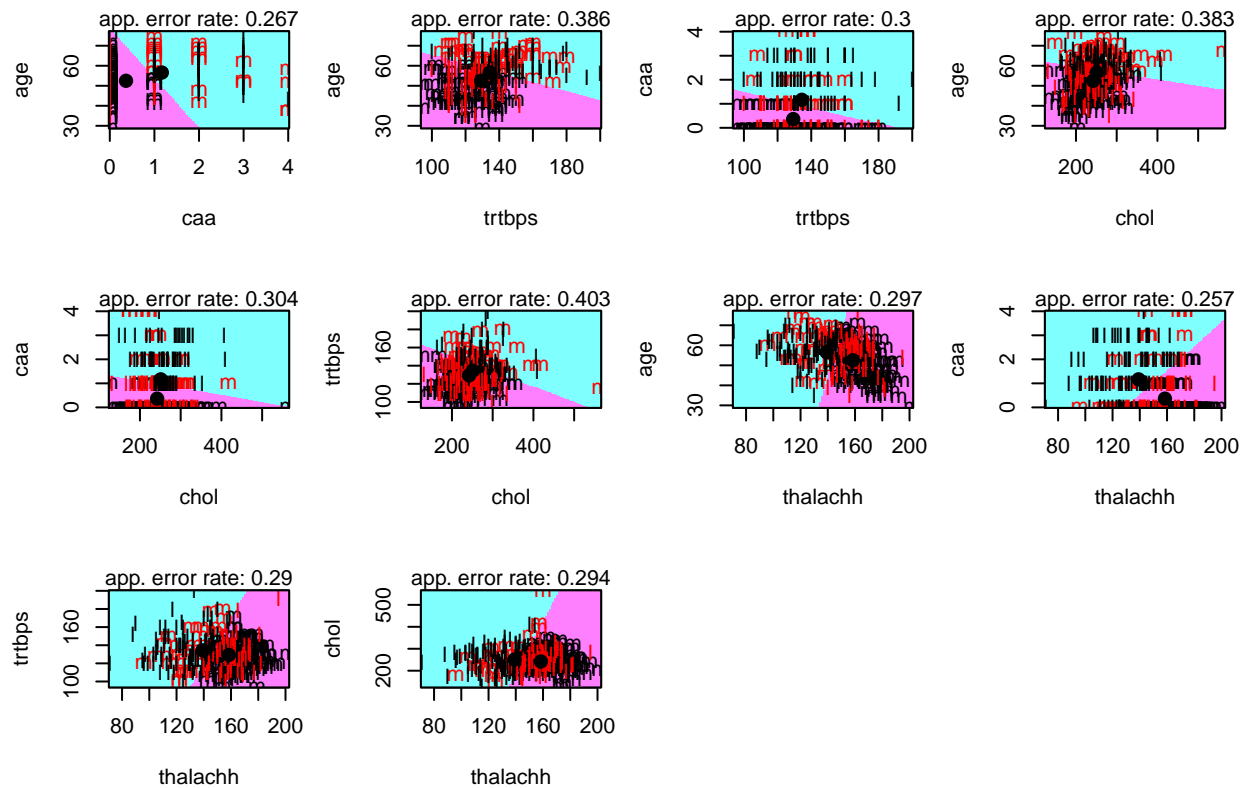
```
coef(model.mars$finalModel) %>% knitr::kable(col.names = "Coefficient")
```

	Coefficient
(Intercept)	-1.7741033
h(-0.223607-cp.L)	-5.1683707
h(3.2-oldpeak)	1.0065639
h(1-caa)	1.8692254
sex.L	-0.9708507

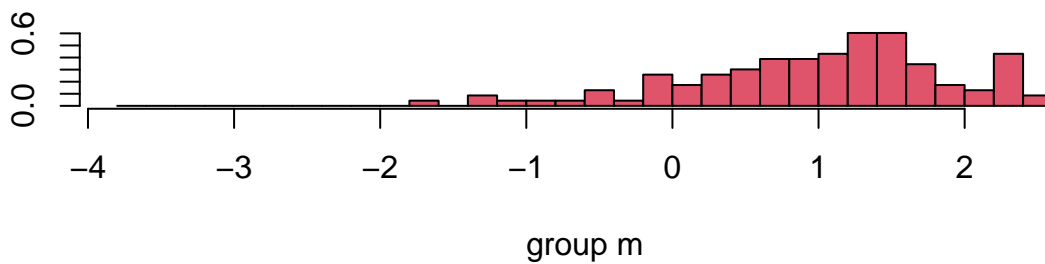
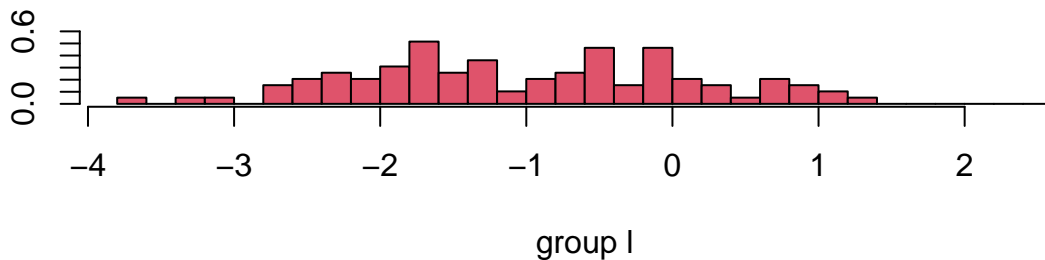
LDA

```
partimat(output~age+caa+trtbps+chol+thalachh, method = "lda", data = heartattack)
```

Partition Plot



```
lda.fit <- lda(output~., data = heartattack, subset = indexTrain)
plot(lda.fit, col = as.numeric(heartattack$output), abbrev = TRUE)
```



```
lda.fit$scaling
```

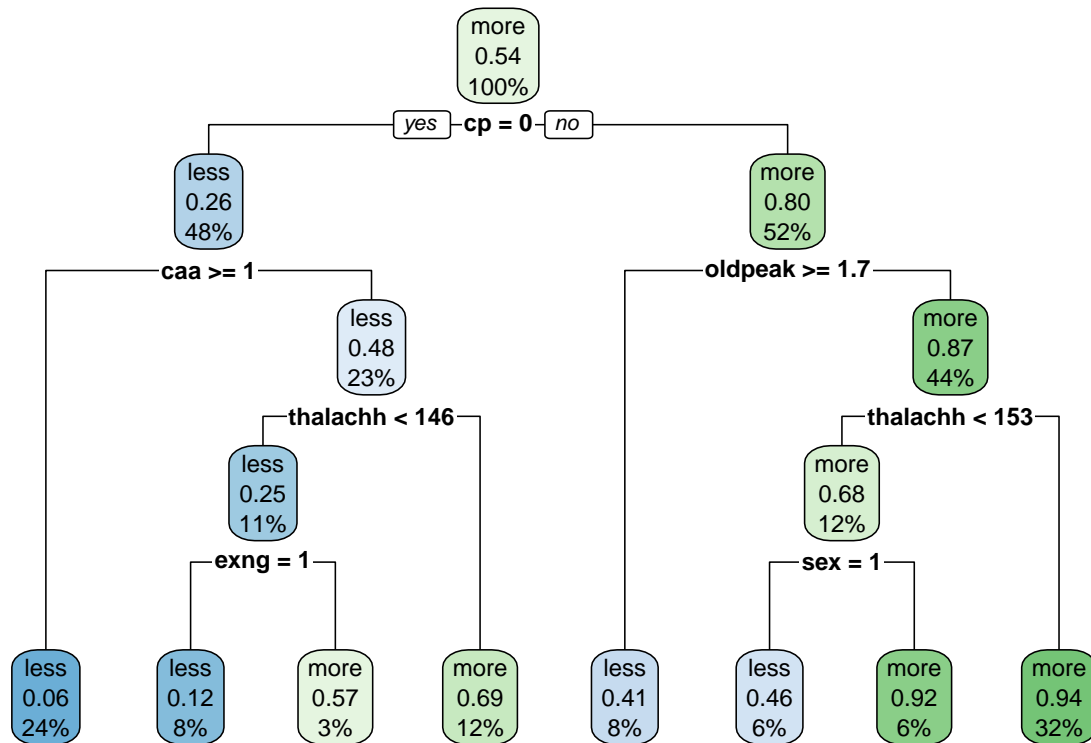
```
##                LD1
## age            0.0066864838
## sex.L          -0.6091628623
## cp.L           1.1073026498
## cp.Q           -0.2684478839
## cp.C           0.3307629074
## trtbps         -0.0063575621
## chol           -0.0005054975
## fbs.L          0.1732598970
## restecg.L      -0.1344302825
## restecg.Q      -0.1050512674
## thalachh       0.0153301665
## exng.L         -0.3222138032
## oldpeak        -0.4002979962
## caa            -0.3158398231
```

```
set.seed(100)
model.lda = train(output ~ . , heartattack,
                  subset = indexTrain,
                  method = "lda",
                  metric = "ROC",
                  trControl = ctrl)
```

CTREE

```
set.seed(100)
```

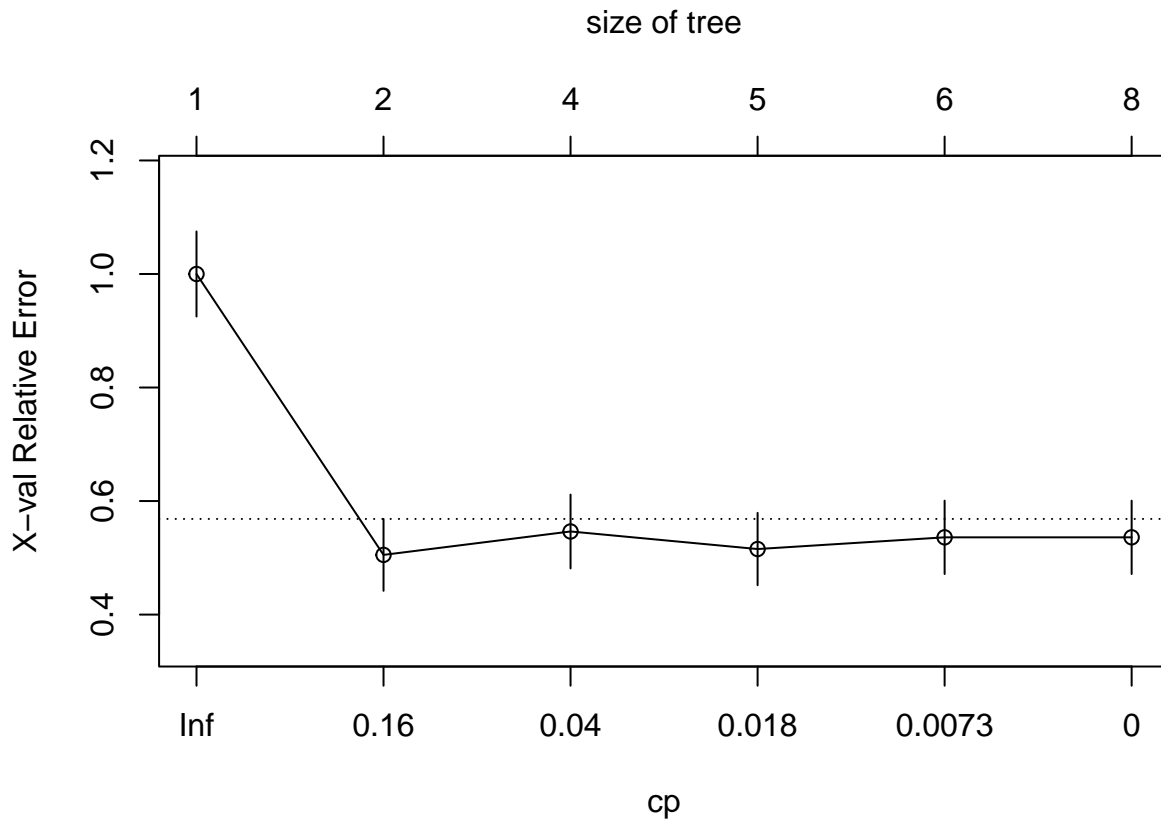
```
tree1 <- rpart(formula = output ~ . , data = heartattack, subset = indexTrain, control = rpart.control(c))
rpart.plot(tree1)
```

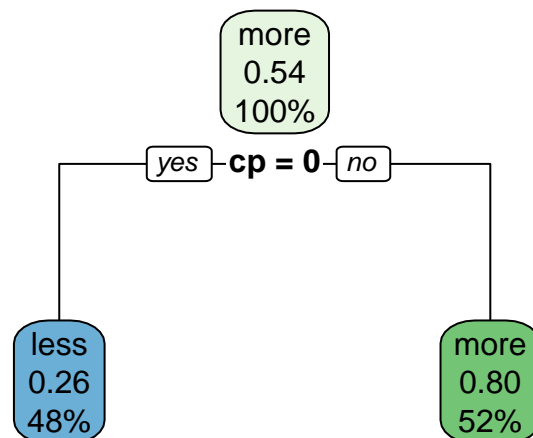
```
printcp(tree1)
```

```
##
## Classification tree:
## rpart(formula = output ~ ., data = heartattack, subset = indexTrain,
##       control = rpart.control(cp = 0))
##
## Variables actually used in tree construction:
## [1] caa      cp      exng    oldpeak sex    thalachh
##
## Root node error: 97/213 = 0.4554
##
## n= 213
##
##      CP nsplit rel error  xerror   xstd
## 1 0.4948454      0  1.00000 1.00000 0.074930
## 2 0.0515464      1  0.50515 0.50515 0.063323
## 3 0.0309278      3  0.40206 0.54639 0.065048
## 4 0.0103093      4  0.37113 0.51546 0.063770
## 5 0.0051546      5  0.36082 0.53608 0.064633
## 6 0.0000000      7  0.35052 0.53608 0.064633
```

```
cpTable <- tree1$cptable
plotcp(tree1)
```



```
minErr <- which.min(cpTable[,4])
tree2 <- prune(tree1, cp = cpTable[minErr,1])
rpart.plot(tree2)
```



```
set.seed(100)
model.ctree <- train(output ~ . , heartattack,
  subset = indexTrain,
  method = "ctree",
  tuneGrid = data.frame(mincriterion = 1-exp(seq(-2, -1, length = 50))),
  metric = "ROC",
  trControl = ctrl)
```

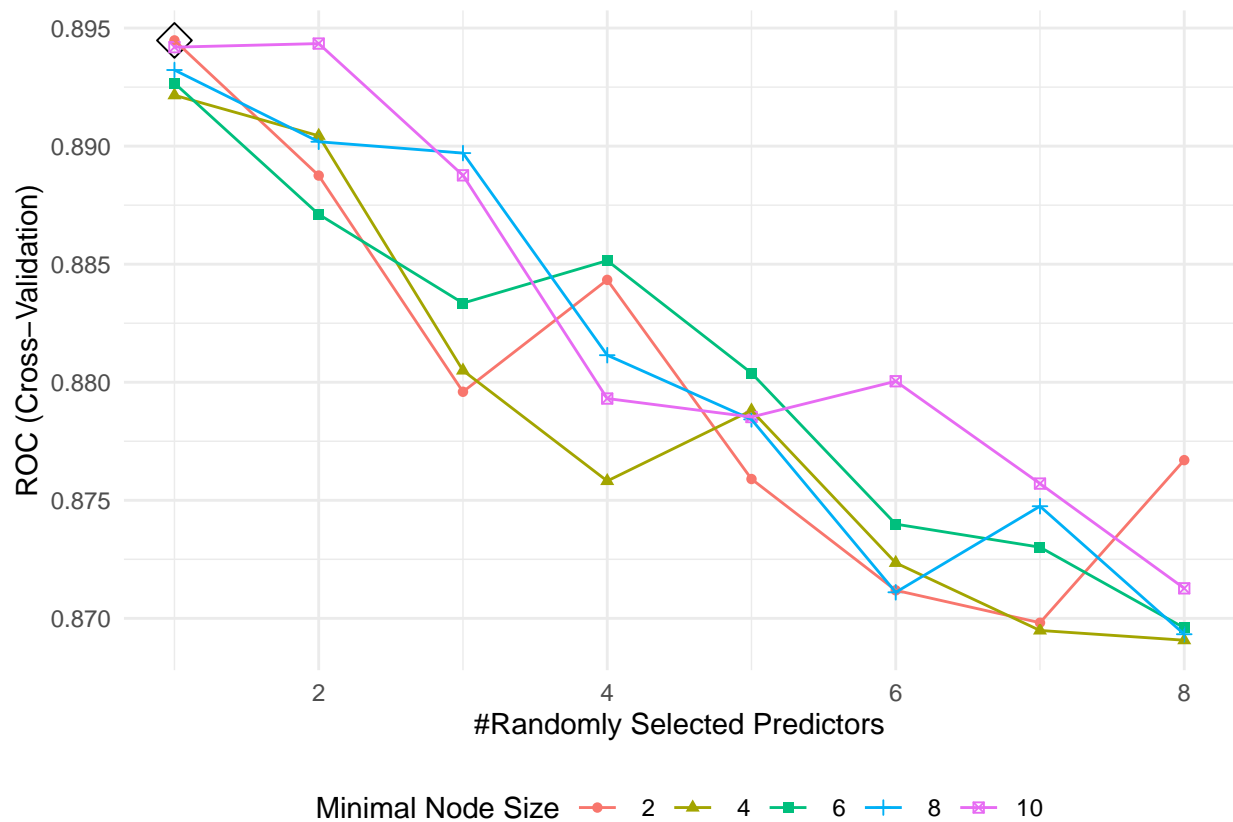
Random Tree

```
set.seed(100)

model.rpart <- train(output ~ . ,
                     heartattack,
                     subset = indexTrain,
                     method = "rpart",
                     tuneGrid = data.frame(cp = exp(seq(-6,-3, len = 50))),
                     trControl = ctrl,
                     metric = "ROC")
```

```
rf.grid <- expand.grid(mtry = 1:8,
                      splitrule = "gini",
                      min.node.size = seq(from = 2, to = 10, by = 2))

set.seed(1)
model.rf <- train(output ~ . ,
                  heartattack,
                  subset = indexTrain,
                  method = "ranger",
                  tuneGrid = rf.grid,
                  metric = "ROC",
                  trControl = ctrl)
ggplot(model.rf, highlight = TRUE)
```

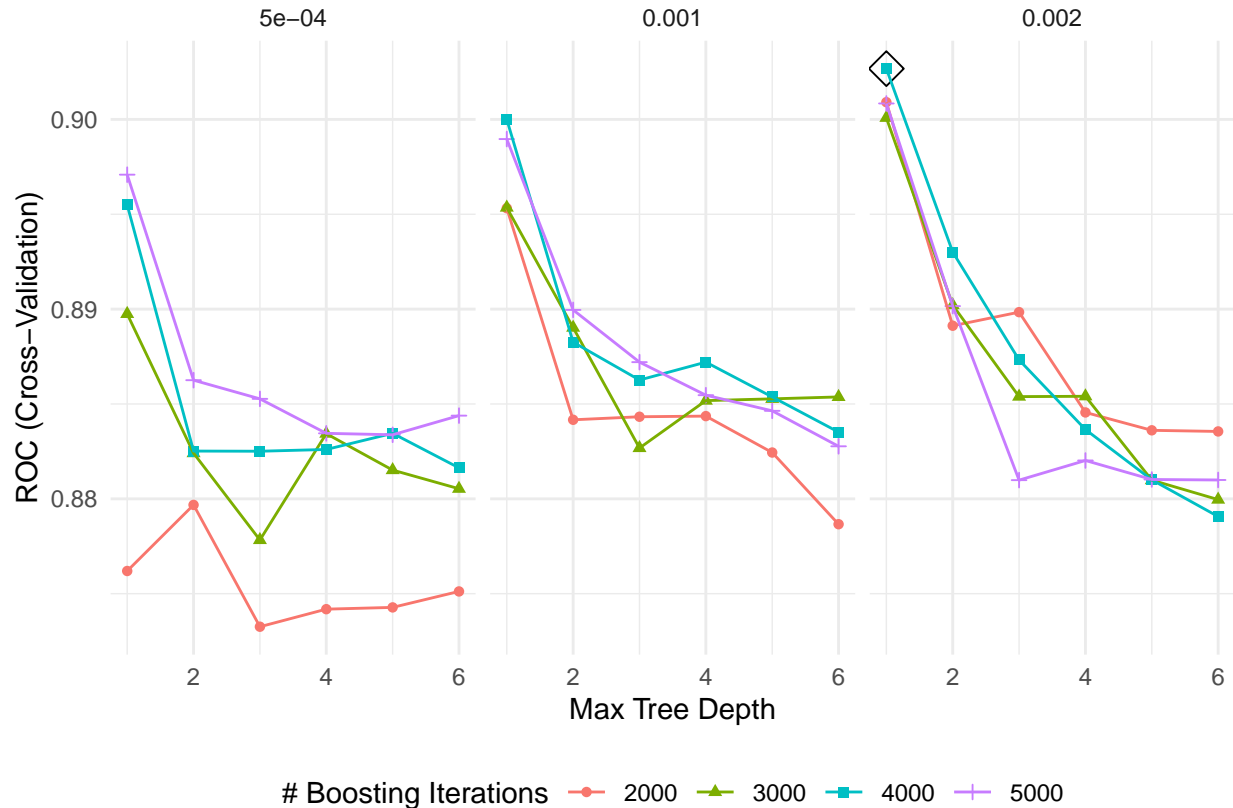


AdaBoost

```
gbmA.grid <- expand.grid(n.trees = c(2000,3000,4000,5000),
                        interaction.depth = 1:6,
                        shrinkage = c(0.0005,0.001,0.002),
                        n.minobsinnode = 1)

set.seed(1)
model.gbmA <- train(output ~ . ,
                    heartattack,
                    subset = indexTrain,
                    tuneGrid = gbmA.grid,
                    trControl = ctrl,
                    method = "gbm",
                    distribution = "adaboost",
                    metric = "ROC",
                    verbose = FALSE)

ggplot(model.gbmA, highlight = TRUE)
```



```
#gbmA.pred <- predict(gbmA.fit, newdata = dat[-rowTrain,], type = "prob")[,1]
```

GAM

```
set.seed(100)
model.gam <- train(output ~ . ,
                  heartattack,
                  subset = indexTrain,
```

```

        method="gam",
        metric = "ROC",
        tuneGrid = data.frame(method = "GCV.Cp",select=c(TRUE,FALSE)),
        trControl = ctrl)
model.gam$bestTune

##    select method
## 2    TRUE GCV.Cp
model.gam$results

##    method select      ROC      Sens      Spec      ROCSD      SensSD      SpecSD
## 1 GCV.Cp  FALSE 0.8461364 0.7433333 0.8106061 0.10954792 0.1330707 0.08681875
## 2 GCV.Cp   TRUE 0.8647222 0.7533333 0.8121212 0.07637024 0.1245899 0.11604078
model.gam$finalModel

##
## Family: binomial
## Link function: logit
##
## Formula:
## .outcome ~ sex.L + cp.Q + fbs.L + restecg.Q + exng.L + restecg.L +
##      cp.L + cp.C + caa + s(oldpeak) + s(age) + s(trtbps) + s(thalachh) +
##      s(chol)
##
## Estimated degrees of freedom:
## 0.911 0.000 0.251 0.826 0.000 total = 11.99
##
## UBRE score: -0.1567041

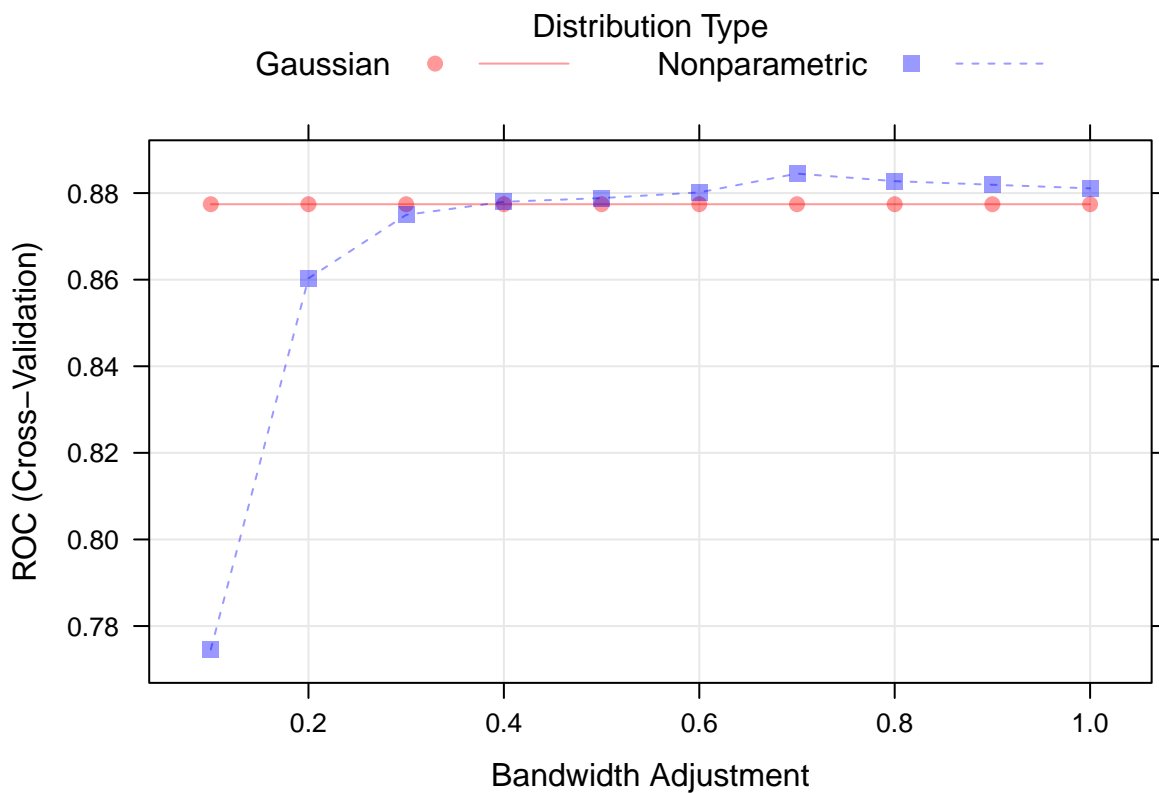
```

NB

```

nbGrid <- expand.grid(usekernel = c(FALSE,TRUE),
                     fL = 1,
                     adjust = seq(.1, 1, by = .1))
set.seed(100)
model.nb <- train(output ~ . ,
                  heartattack,
                  subset = indexTrain,
                  method = "nb",
                  tuneGrid = nbGrid,
                  metric = "ROC",
                  trControl = ctrl)
plot(model.nb)

```

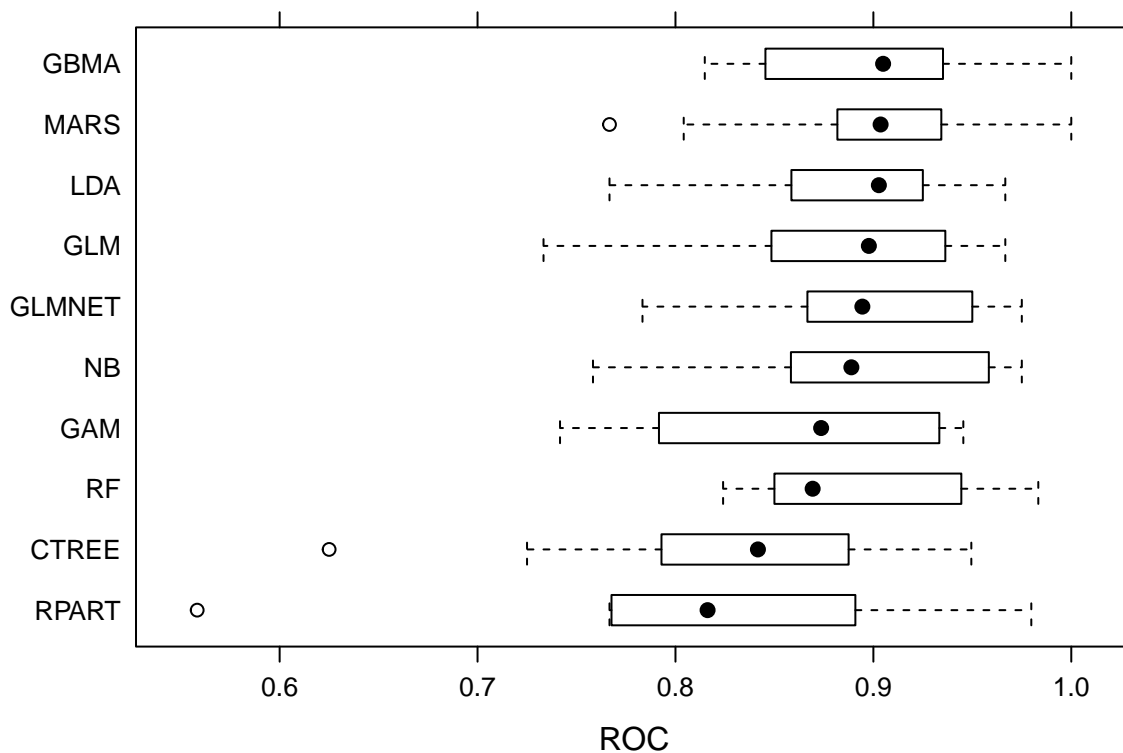


```
res <- resamples(list(GLM = model.glm, GLMNET = model.glmn, MARS = model.mars, CTREE = model.ctree, RPART = model.rpart))
trainROC <- bwplot(res, metric = "ROC")
summary(res)
```

```
##
## Call:
## summary.resamples(object = res)
##
## Models: GLM, GLMNET, MARS, CTREE, RPART, LDA, RF, GBMA, NB, GAM
## Number of resamples: 10
##
## ROC
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max. NA's
## GLM      0.7333333 0.8530303 0.8977273 0.8761364 0.9335227 0.9666667    0
## GLMNET    0.7833333 0.8671717 0.8944444 0.8934596 0.9397727 0.9750000    0
## MARS      0.7666667 0.8853220 0.9036616 0.8965152 0.9340909 1.0000000    0
## CTREE     0.6250000 0.8003788 0.8416667 0.8269318 0.8843750 0.9494949    0
## RPART     0.5583333 0.7767992 0.8162247 0.8137500 0.8765152 0.9797980    0
## LDA       0.7666667 0.8626894 0.9027778 0.8841919 0.9232955 0.9666667    0
## RF        0.8240741 0.8511364 0.8693182 0.8944781 0.9424242 0.9833333    0
## GBMA      0.8148148 0.8528409 0.9049242 0.9026768 0.9311869 1.0000000    0
## NB        0.7583333 0.8604167 0.8888889 0.8844949 0.9414773 0.9750000    0
## GAM       0.7416667 0.8058712 0.8736111 0.8647222 0.9333333 0.9454545    0
##
## Sens
##      Min.   1st Qu.   Median     Mean   3rd Qu. Max. NA's
## GLM      0.5000000 0.7777778 0.7888889 0.7533333 0.8000000 0.9    0
## GLMNET    0.5000000 0.6166667 0.6833333 0.6888889 0.8000000 0.8    0
## MARS      0.5000000 0.6750000 0.7388889 0.7444444 0.8000000 1.0    0
```

```
## CTREE 0.6000000 0.7000000 0.7388889 0.7566667 0.8000000 1.0 0
## RPART 0.5000000 0.6750000 0.7888889 0.7544444 0.8000000 1.0 0
## LDA 0.5000000 0.7000000 0.7388889 0.7233333 0.7777778 0.9 0
## RF 0.5555556 0.6750000 0.7500000 0.7477778 0.8000000 1.0 0
## GBMA 0.5555556 0.6166667 0.7388889 0.7500000 0.8000000 1.0 0
## NB 0.6666667 0.7194444 0.8000000 0.7944444 0.8000000 1.0 0
## GAM 0.5000000 0.7194444 0.7777778 0.7533333 0.8000000 0.9 0
##
## Spec
##      Min.    1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLM      0.6666667 0.7670455 0.8712121 0.8363636 0.9147727 0.9166667 0
## GLMNET 0.7500000 0.8181818 0.8712121 0.8621212 0.9147727 1.0000000 0
## MARS     0.6666667 0.8333333 0.8712121 0.8636364 0.9166667 1.0000000 0
## CTREE    0.5000000 0.7329545 0.7840909 0.7666667 0.8295455 0.9166667 0
## RPART    0.4166667 0.7329545 0.7840909 0.7681818 0.8333333 0.9166667 0
## LDA      0.5833333 0.7708333 0.9090909 0.8371212 0.9147727 0.9166667 0
## RF       0.6666667 0.7329545 0.9090909 0.8356061 0.9166667 0.9166667 0
## GBMA     0.6363636 0.8409091 0.9128788 0.8772727 0.9166667 1.0000000 0
## NB       0.5833333 0.6647727 0.8181818 0.7765152 0.8901515 0.9166667 0
## GAM      0.5833333 0.7537879 0.8333333 0.8121212 0.9090909 0.9166667 0
```

trainROC



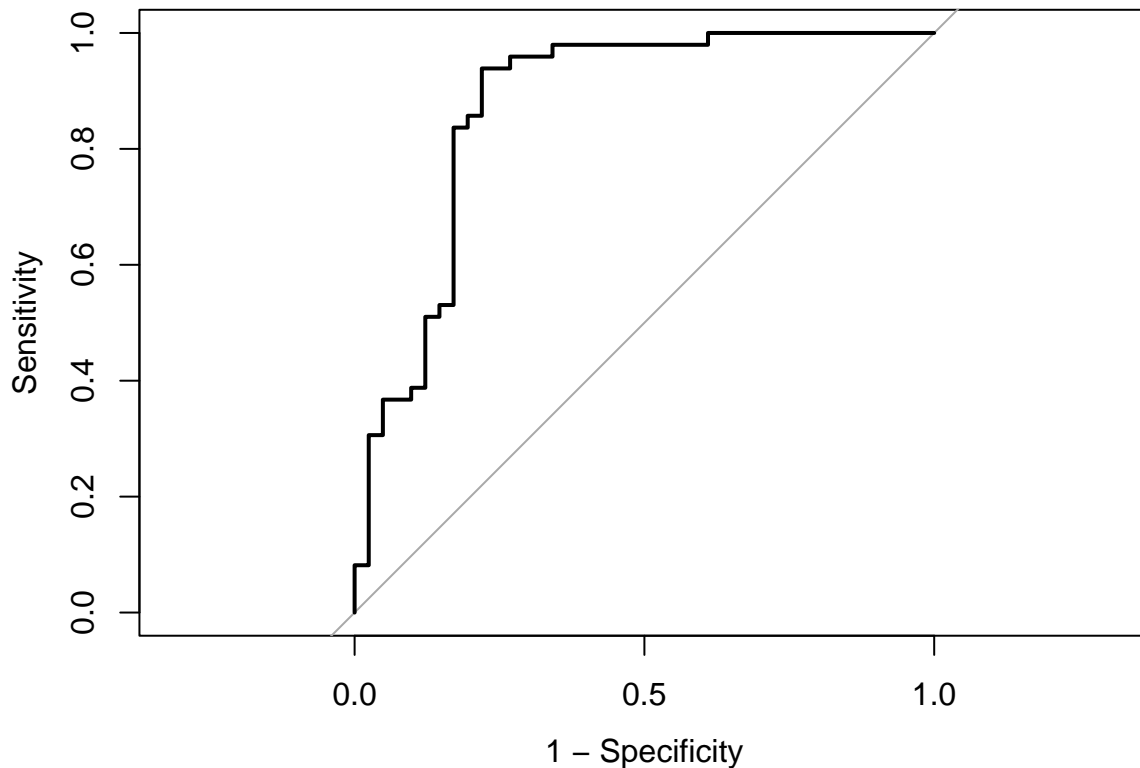
```
lda.pred <- predict(model.lda, newdata = heartattack[-indexTrain,], type = "prob")[,2]
rpart.pred <- predict(model.rpart, newdata = heartattack[-indexTrain,], type = "prob")[,2]
ctree.pred <- predict(model.ctree, newdata = heartattack[-indexTrain,], type = "prob")[,2]
glm.pred <- predict(model.glm, newdata = heartattack[-indexTrain,], type = "prob")[,2]
glmnet.pred <- predict(model.glmnet, newdata = heartattack[-indexTrain,], type = "prob")[,2]
mars.pred <- predict(model.mars, newdata = heartattack[-indexTrain,], type = "prob")[,2]
rf.pred <- predict(model.rf, newdata = heartattack[-indexTrain,], type = "prob")[,2]
gbma.pred <- predict(model.gbma, newdata = heartattack[-indexTrain,], type = "prob")[,2]
```

```
nb.pred <- predict(model.nb, newdata = heartattack[-indexTrain,], type = "prob")[,2]
gam.pred <- predict(model.gam, newdata = heartattack[-indexTrain,], type = "prob")[,2]
```

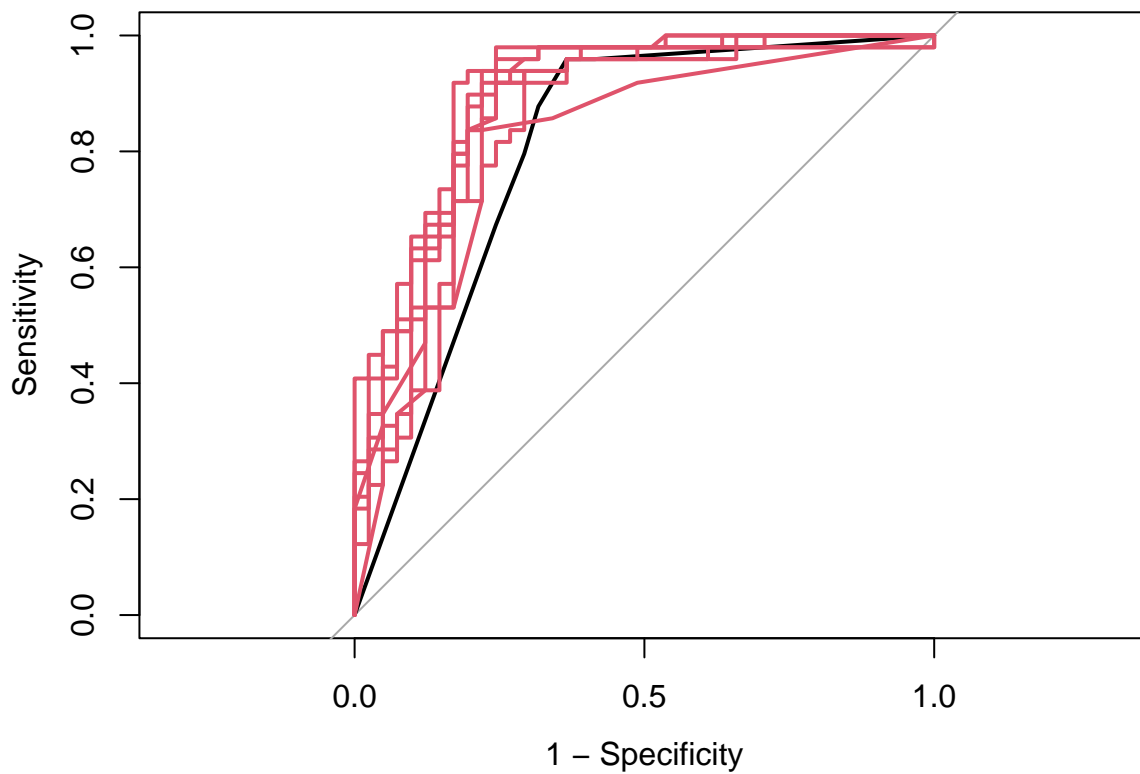
```
roc.lda <- roc(heartattack$output[-indexTrain], lda.pred)
roc.rpart <- roc(heartattack$output[-indexTrain], rpart.pred)
roc.ctree <- roc(heartattack$output[-indexTrain], ctree.pred)
roc.glm <- roc(heartattack$output[-indexTrain], glm.pred)
roc.glmn <- roc(heartattack$output[-indexTrain], glmn.pred)
roc.mars <- roc(heartattack$output[-indexTrain], mars.pred)
roc.rf <- roc(heartattack$output[-indexTrain], rf.pred)
roc.gbmA <- roc(heartattack$output[-indexTrain], gbmA.pred)
roc.nb <- roc(heartattack$output[-indexTrain], nb.pred)
roc.gam <- roc(heartattack$output[-indexTrain], gam.pred)
```

```
auc <- c(roc.lda$auc[1], roc.rpart$auc[1], roc.ctree$auc[1], roc.glm$auc[1], roc.glmn$auc[1], roc.rf$auc[1], roc.gbmA$auc[1], roc.nb$auc[1], roc.gam$auc[1])
```

```
plot(roc.lda, legacy.axes = TRUE)
```

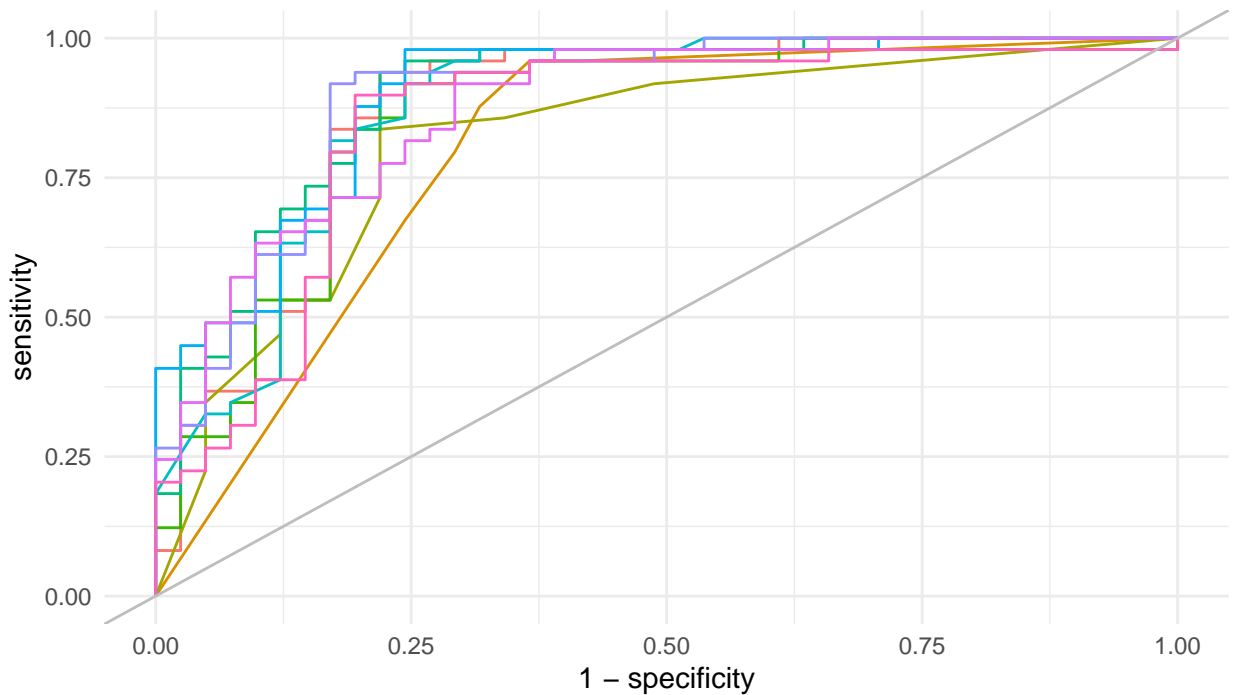


```
plot(roc.rpart, legacy.axes = TRUE)
plot(roc.ctree, col = 2, add = TRUE)
plot(roc.glm, col = 2, add = TRUE)
plot(roc.glmn, col = 2, add = TRUE)
plot(roc.mars, col = 2, add = TRUE)
plot(roc.rf, col = 2, add = TRUE)
plot(roc.gbmA, col = 2, add = TRUE)
plot(roc.nb, col = 2, add = TRUE)
plot(roc.gam, col = 2, add = TRUE)
```

```
modelNames <- c("lda", "rpart", "ctree", "glm", "glmnet", "mars", "rf", "gbm", "nb", "gam")

ggroc(list(roc.lda, roc.rpart, roc.ctree, roc.glm, roc.glmnet, roc.mars, roc.rf, roc.gbm, roc.nb, roc.gam),
  scale_color_discrete(labels = paste0(modelNames, " (", round(auc, 3), ")"),
  name = "Models (AUC)" +
  geom_abline(intercept = 0, slope = 1, color = "grey")
```

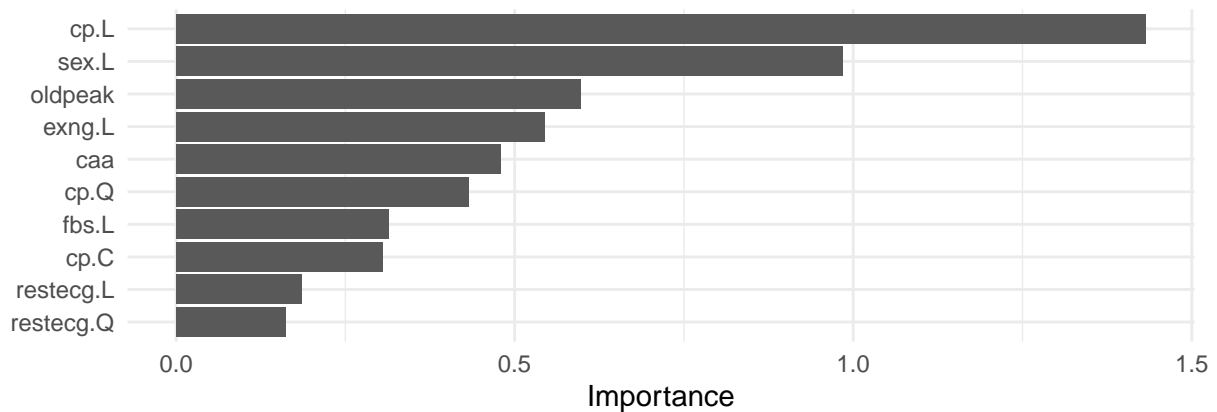
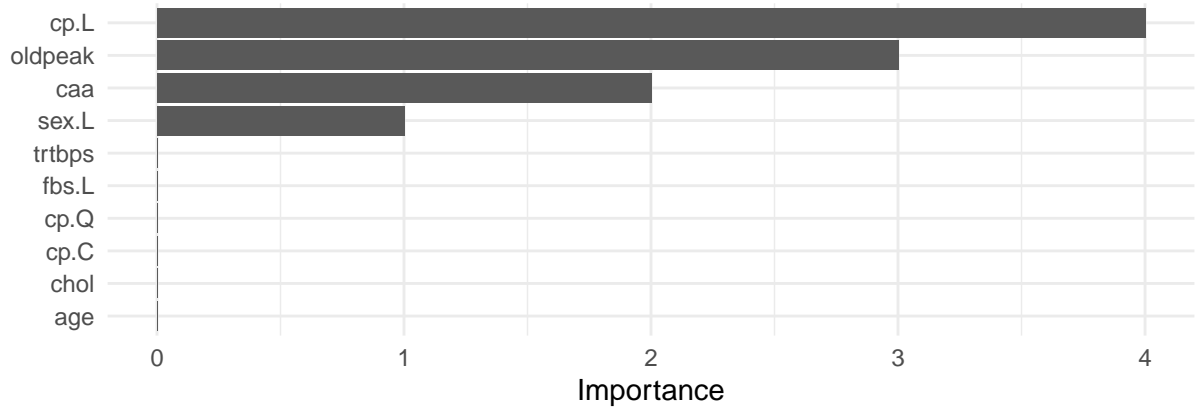


Models (AUC)

lda (0.873)	ctree (0.816)	glmn (0.897)	rf (0.879)	nb (0.87)
rpart (0.803)	glm (0.855)	mars (0.898)	gbml (0.893)	gam (0.898)

Variable Importance

```
glm.vip <- varImp(model.glm)$importance
glmn.vip <- varImp(model.glmn)$importance
mars.vip <- varImp(model.mars)
rpart.vip <- varImp(model.rpart)$importance
vip1 <- vip(model.mars$finalModel)
vip2 <- vip(model.glmn$finalModel)
vip <- grid.arrange(vip1,vip2)
```

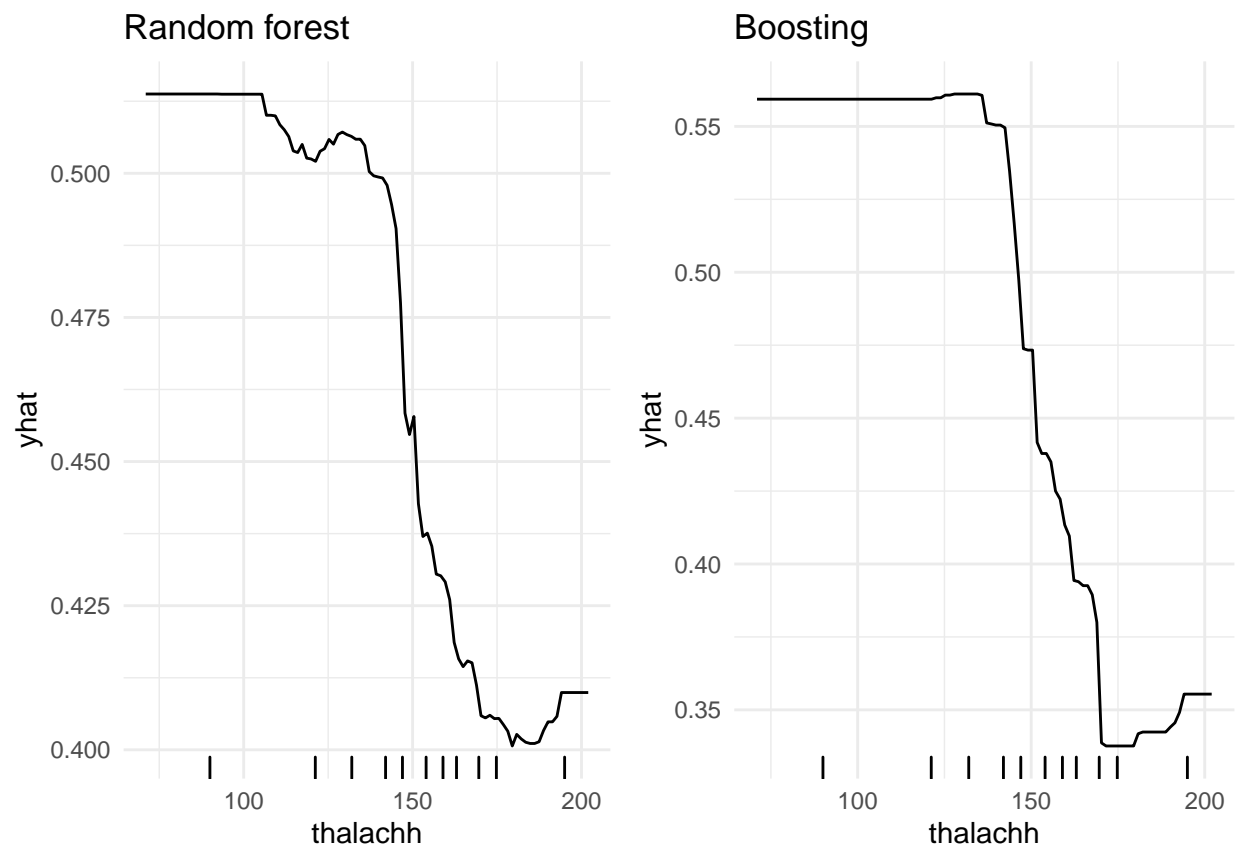


PDP

```
pdp.rf <- model.rf %>%
  partial(pred.var = "thalachh",
    grid.resolution = 100,
    prob = TRUE) %>%
  autoplot(rug = TRUE, train = training_df) +
  ggtitle("Random forest")

pdp.gbmA <- model.gbmA %>%
  partial(pred.var = "thalachh",
    grid.resolution = 100,
    prob = TRUE) %>%
  autoplot(rug = TRUE, train = training_df) +
  ggtitle("Boosting")

grid.arrange(pdp.rf, pdp.gbmA, nrow = 1)
```

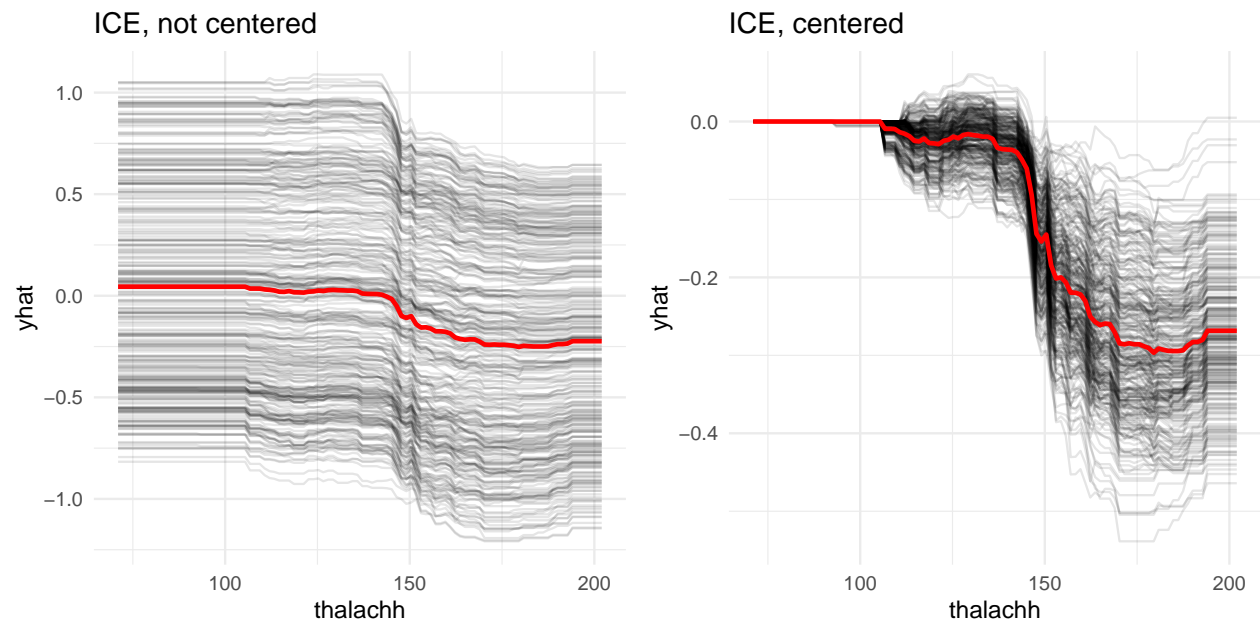


Individual conditional expectation (ICE) curves

```
ice1.rf <- model.rf %>%
  partial(pred.var = "thalachh",
    grid.resolution = 100,
    ice = TRUE) %>%
  autoplot(train = training_df, alpha = .1) +
  ggtitle("ICE, not centered")

ice2.rf <- model.rf %>%
  partial(pred.var = "thalachh",
    grid.resolution = 100,
    ice = TRUE) %>%
  autoplot(train = training_df, alpha = .1,
    center = TRUE) +
  ggtitle("ICE, centered")

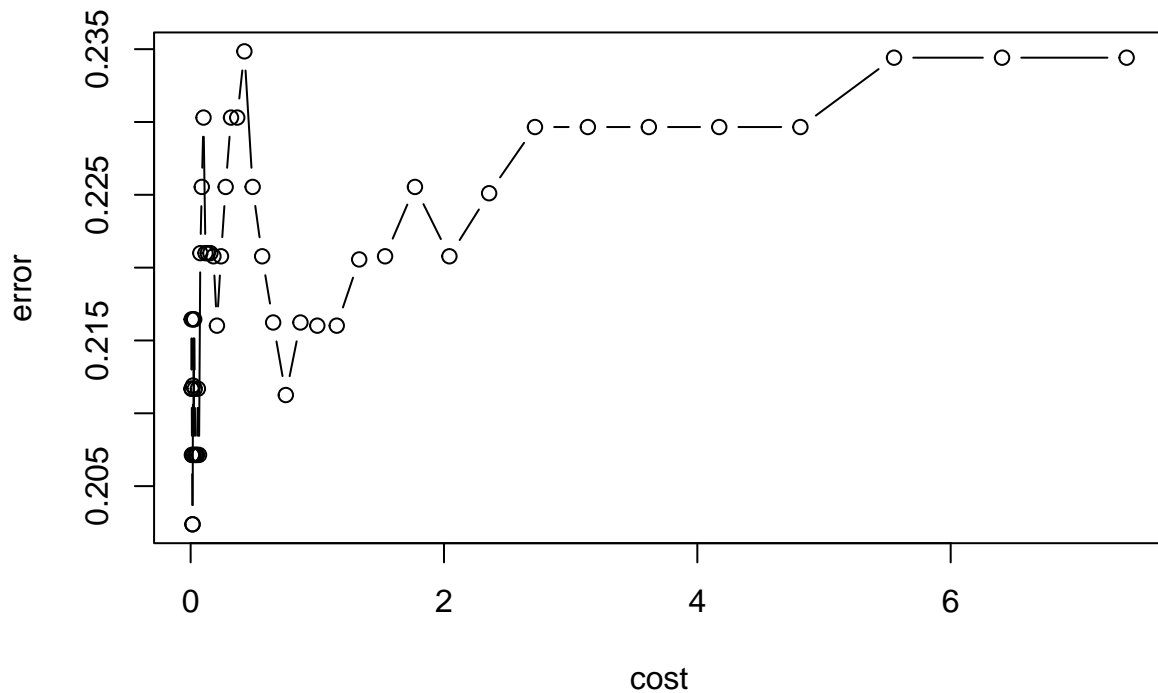
grid.arrange(ice1.rf, ice2.rf, nrow = 1)
```



Support Vector

```
set.seed(5009)
linear.tune <- tune.svm(output ~ . ,
  data = training_df,
  kernel = "linear",
  cost = exp(seq(-5,2,len=50)),
  scale = TRUE)
plot(linear.tune)
```

Performance of 'svm'



```
#summary(linear.tune)
```

```
linear.tune$best.parameters
```

```
##          cost
## 6 0.01376379
```

```
best.linear <- linear.tune$best.model
summary(best.linear)
```

```
##
## Call:
## best.svm(x = output ~ ., data = training_df, cost = exp(seq(-5, 2,
##   len = 50)), kernel = "linear", scale = TRUE)
##
##
## Parameters:
##   SVM-Type:  C-classification
##   SVM-Kernel: linear
##   cost:      0.01376379
##
## Number of Support Vectors: 144
##
## ( 72 72 )
##
##
## Number of Classes: 2
##
## Levels:
## less more
```

```

pred.linear <- predict(best.linear, newdata = testing_df)

confushion_test <- confusionMatrix(data = pred.linear,
                                   reference = heartattack$output[-indexTrain])
confushion_test

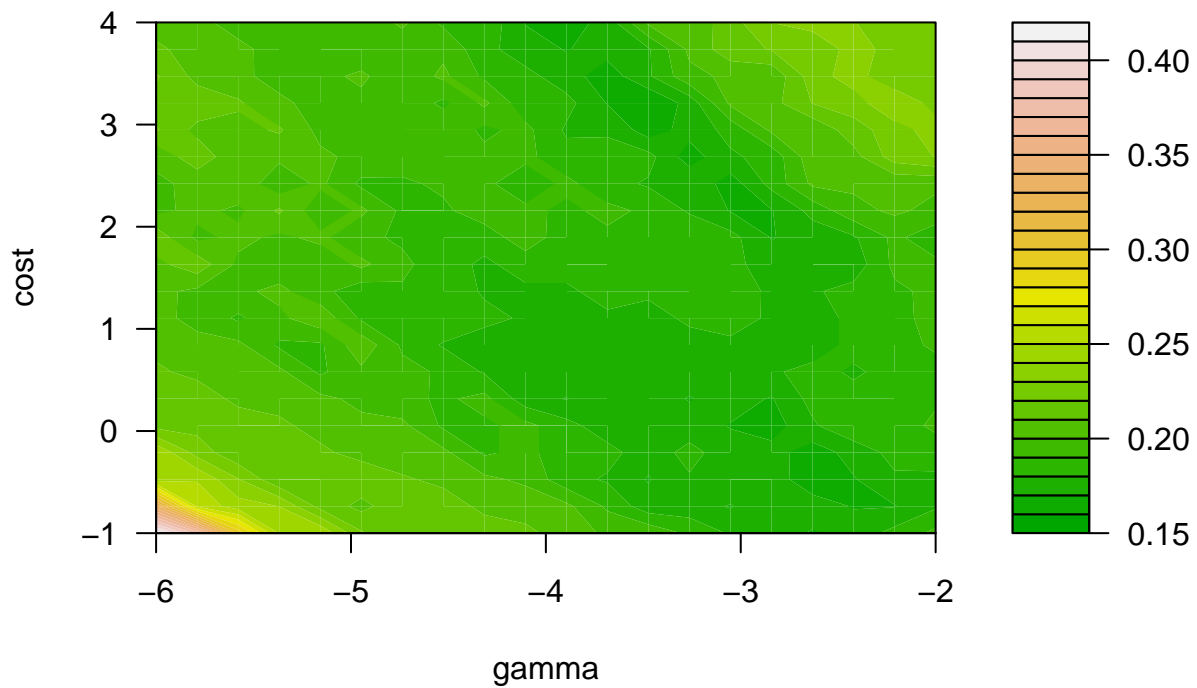
## Confusion Matrix and Statistics
##
##           Reference
## Prediction less more
##      less  28   3
##      more  13  46
##
##           Accuracy : 0.8222
##           95% CI : (0.7274, 0.8948)
##      No Information Rate : 0.5444
##      P-Value [Acc > NIR] : 2.84e-08
##
##           Kappa : 0.6343
##
##  McNemar's Test P-Value : 0.02445
##
##           Sensitivity : 0.6829
##           Specificity : 0.9388
##           Pos Pred Value : 0.9032
##           Neg Pred Value : 0.7797
##           Prevalence : 0.4556
##           Detection Rate : 0.3111
##      Detection Prevalence : 0.3444
##           Balanced Accuracy : 0.8109
##
##           'Positive' Class : less
##

set.seed(5009)
radial.tune <- tune.svm(output ~ . ,
                        data = training_df,
                        kernel = "radial",
                        cost = exp(seq(-1,4,len=20)),
                        gamma = exp(seq(-6,-2,len=20)))

plot(radial.tune, transform.y = log, transform.x = log,
     color.palette = terrain.colors)

```

Performance of 'svm'



```
# summary(radial.tune)
radial.tune$best.parameters
```

```
##      gamma      cost
## 333 0.0310026 24.79213
```

```
best.radial <- radial.tune$best.model
summary(best.radial)
```

```
##
## Call:
## best.svm(x = output ~ ., data = training_df, gamma = exp(seq(-6,
##      -2, len = 20)), cost = exp(seq(-1, 4, len = 20)), kernel = "radial")
##
##
## Parameters:
##   SVM-Type:  C-classification
##   SVM-Kernel: radial
##      cost:  24.79213
##
## Number of Support Vectors:  104
##
##   ( 53 51 )
##
##
## Number of Classes:  2
##
## Levels:
## less more
```



```

pred.radial <- predict(best.radial, newdata = heartattack[-indexTrain,])

confusionMatrix(data = pred.radial,
                 reference = heartattack$output[-indexTrain])

```

```

## Confusion Matrix and Statistics
##
##           Reference
## Prediction less more
##      less  32   7
##      more   9  42
##
##              Accuracy : 0.8222
##              95% CI : (0.7274, 0.8948)
##      No Information Rate : 0.5444
##      P-Value [Acc > NIR] : 2.84e-08
##
##              Kappa : 0.6402
##
##  Mcnemar's Test P-Value : 0.8026
##
##      Sensitivity : 0.7805
##      Specificity : 0.8571
##      Pos Pred Value : 0.8205
##      Neg Pred Value : 0.8235
##      Prevalence : 0.4556
##      Detection Rate : 0.3556
##      Detection Prevalence : 0.4333
##      Balanced Accuracy : 0.8188
##
##      'Positive' Class : less
##

```

```

heartattack_dat = heartattack %>% as.data.frame()
hc.complete <- hclust(dist(heartattack_dat), method = "complete")
fviz_dend(hc.complete, k = 3,
          cex = 0.3,
          palette = "jco",
          color_labels_by_k = TRUE,
          rect = TRUE, rect_fill = TRUE, rect_border = "jco",
          labels_track_height = 2.5)

```

Cluster Dendrogram

