

haoran

*Haoran Hu*

*2019-5-14*

```
heart_disease = read_csv("../data/heart.csv")
```

```
## Parsed with column specification:
## cols(
##   age = col_double(),
##   sex = col_double(),
##   cp = col_double(),
##   trestbps = col_double(),
##   chol = col_double(),
##   fbs = col_double(),
##   restecg = col_double(),
##   thalach = col_double(),
##   exang = col_double(),
##   oldpeak = col_double(),
##   slope = col_double(),
##   ca = col_double(),
##   thal = col_double(),
##   target = col_double()
## )
```

```
set.seed(1)
trRows = createDataPartition(heart_disease$target, p = .75, list = FALSE)
train = heart_disease[trRows,]
test = heart_disease[-trRows,]

train = train %>%
  mutate(cp=as.factor(cp),
         restecg=as.factor(restecg),
         slope=as.factor(slope),
         thal=as.factor(thal))
train.x <- model.matrix(target~.,train)[,-1]
train.y <- train$target

test = test %>%
  mutate(cp=as.factor(cp),
         restecg=as.factor(restecg),
         slope=as.factor(slope),
         thal=as.factor(thal))
test.x <- model.matrix(target~.,test)[,-1]
test.y <- test$target
```

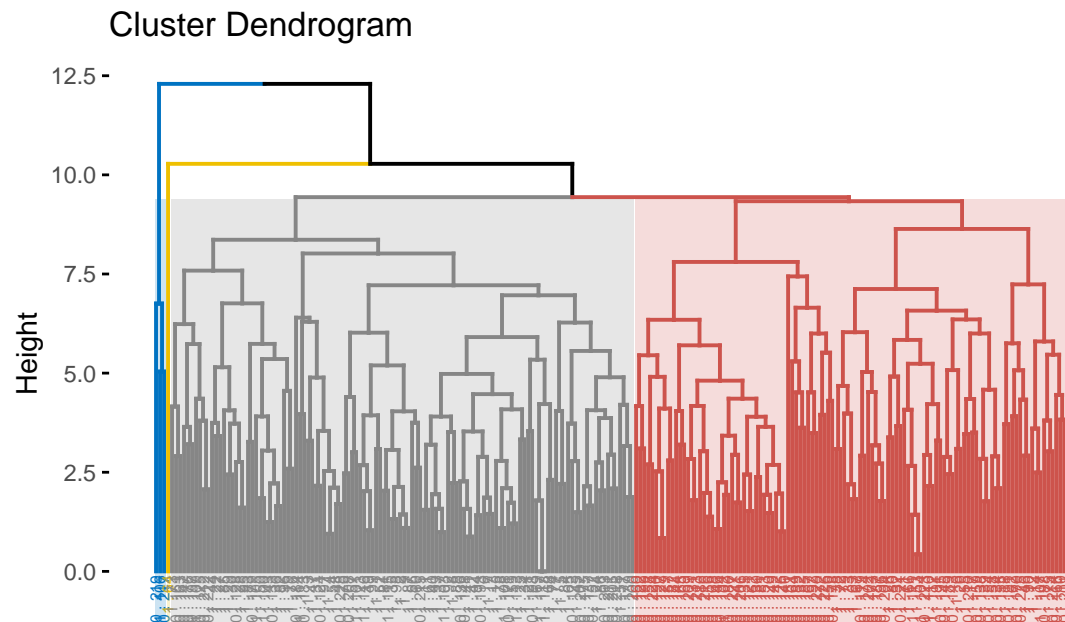
```
train.hc = train.x %>% as.data.frame() %>%
  mutate(target = as.character(train$target),
         num =as.character(1:228)) %>%
  mutate(name = paste(target,".", num)) %>%
  select(-num, -target) %>%
  column_to_rownames(var = "name") %>% scale()
```

```

hc.heart = hclust(dist(train.hc), method = "complete")

fviz_dend(hc.heart, k = 4,
  cex = 0.4,
  palette = "jco",
  color_labels_by_k = TRUE,
  rect = TRUE, rect_fill = TRUE, rect_border = "jco",
  labels_track_height = 0.8)

```

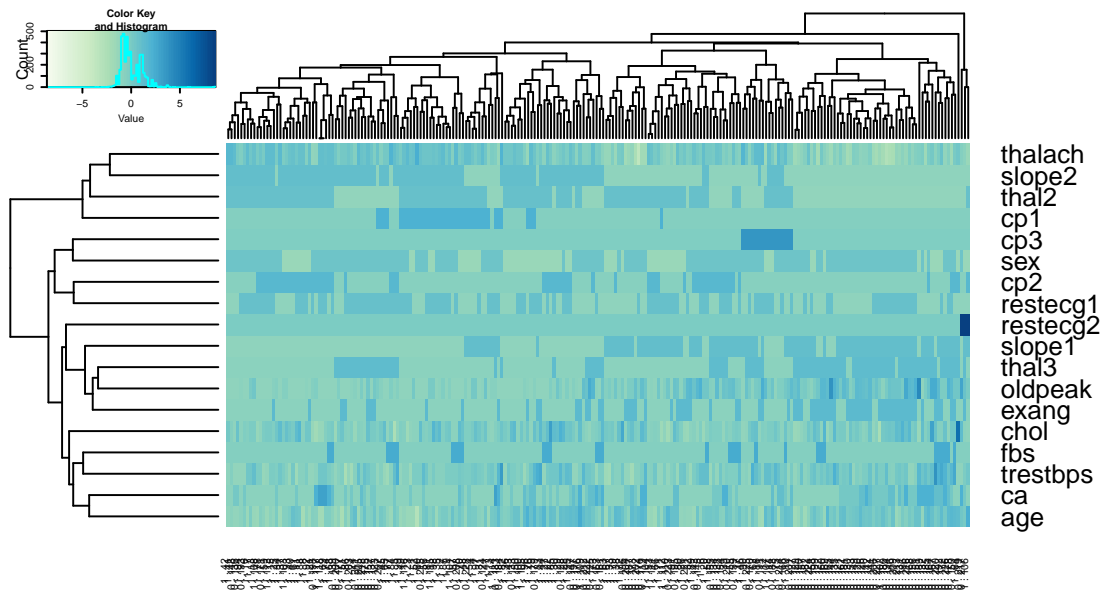


```

col1 <- colorRampPalette(brewer.pal(9, "GnBu"))(100)

heatmap.2(t(train.hc),
  col = col1, keysize = 1, key.par = list(cex = .3),
  cexRow = 1,
  #dendrogram = "col",
  trace = "none", key = TRUE, cexCol = 0.4,
  margins = c(5, 5))

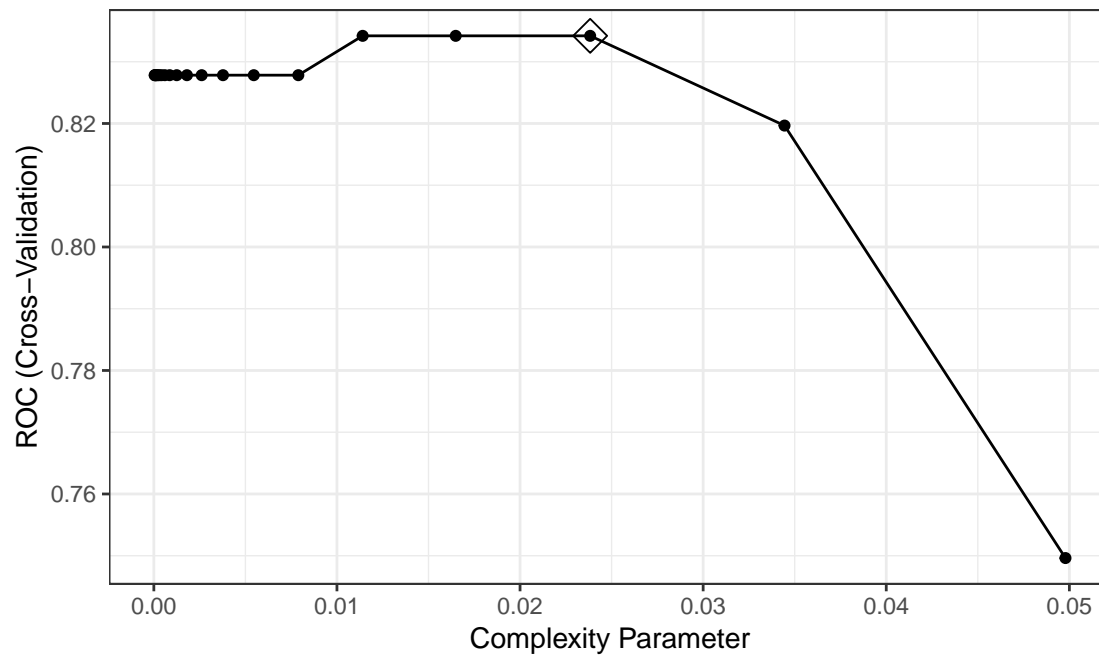
```



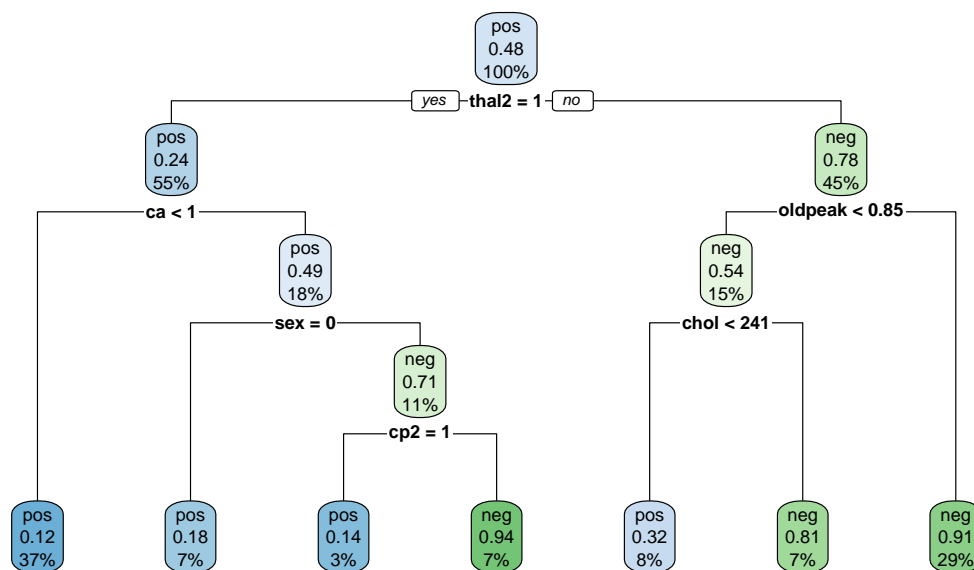
## Tree

```
ctrl = trainControl(method = "cv",
                    summaryFunction = twoClassSummary,
                    classProbs = TRUE)
train.y = as.factor(train.y) %>%
  recode_factor("1" = "pos", "0" = "neg")

set.seed(1)
tree.class <- train(train.x, train.y,
                   method = "rpart",
                   tuneGrid = data.frame(cp = exp(seq(-10, -3, len = 20))),
                   trControl = ctrl,
                   metric = "ROC")
ggplot(tree.class, highlight = TRUE)
```



```
rpart.plot(tree.class$finalModel)
```



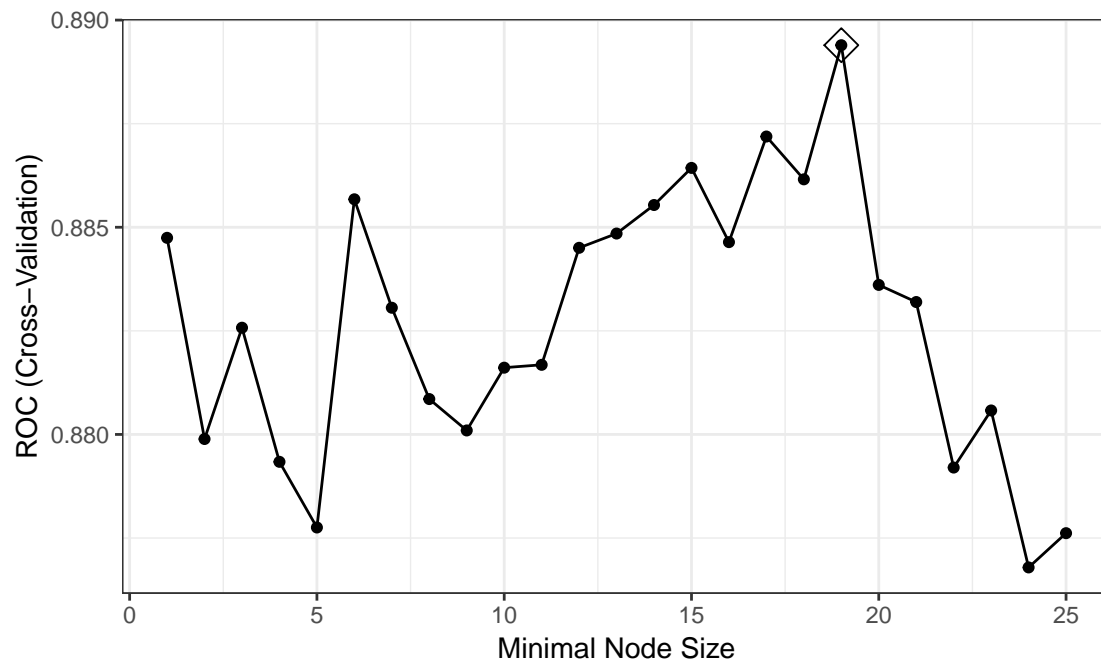
## Bagging

```
bagging.grid <- expand.grid(mtry = 18,  
  splitrule = "gini",  
  min.node.size = 1:25)
```

```

set.seed(1)
bagging.class <- train(train.x, train.y,
  method = "ranger",
  tuneGrid = bagging.grid,
  metric = "ROC",
  trControl = ctrl,
  importance = "impurity")
ggplot(bagging.class, highlight = TRUE)

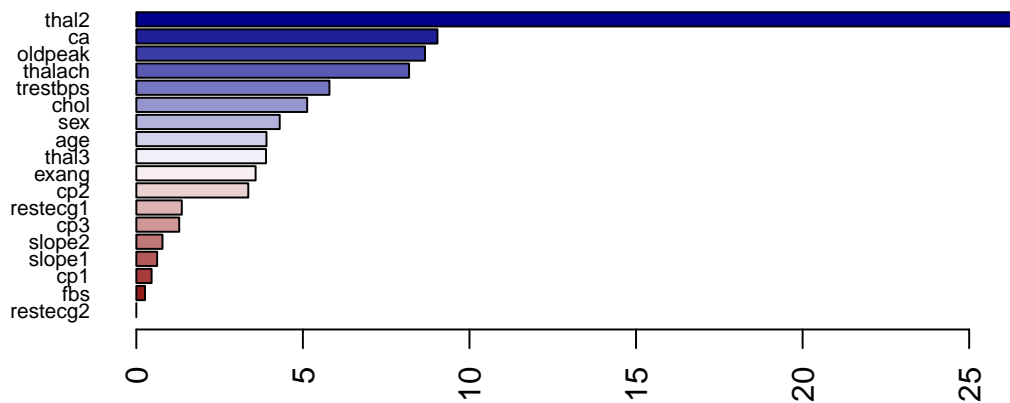
```



```

barplot(sort(ranger::importance(bagging.class$finalModel),
  decreasing = FALSE),
las = 2, horiz = TRUE, cex.names = 0.7,
col = colorRampPalette(colors = c("darkred", "white", "darkblue"))(18))

```



## Random Forest

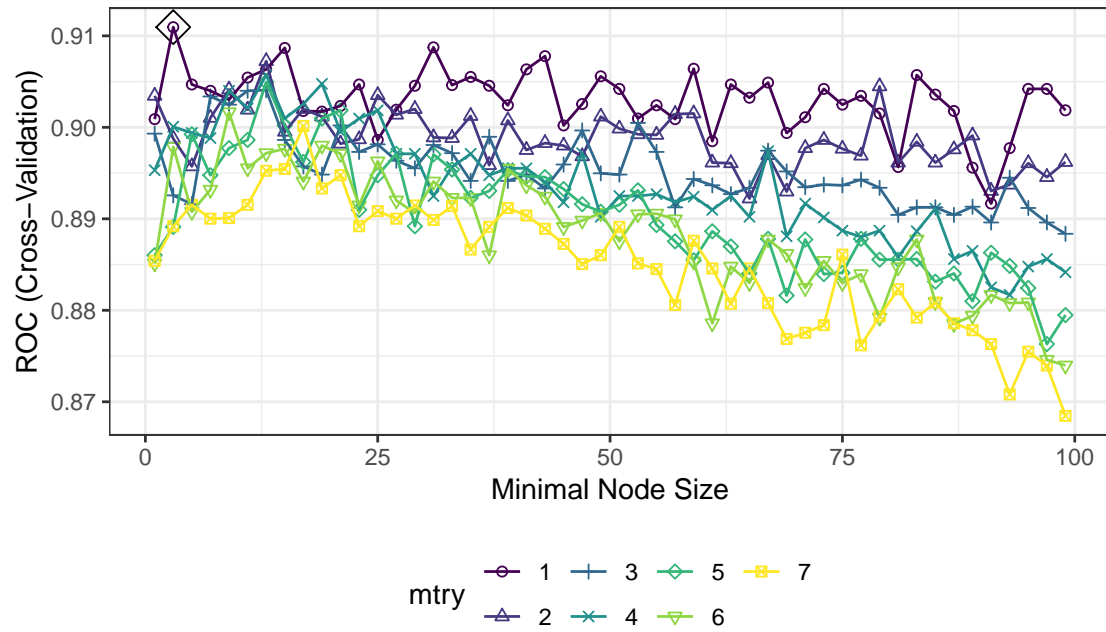
```
rf.grid <- expand.grid(mtry = 1:7,
                      splitrule = "gini",
                      min.node.size = seq(1,100, by = 2))
```

```
set.seed(1)
rf.class <- train(train.x, train.y,
                  method = "ranger",
                  tuneGrid = rf.grid,
                  metric = "ROC",
                  trControl = ctrl,
                  importance = "impurity")
```

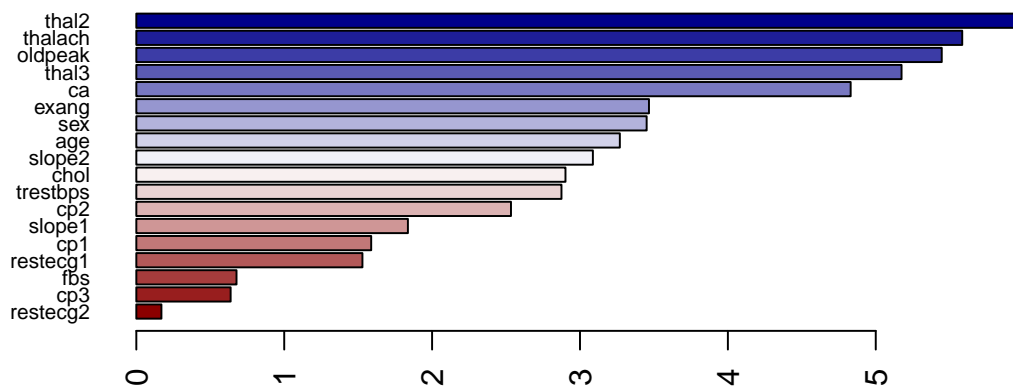
```
ggplot(rf.class, highlight = TRUE) +
  viridis::scale_color_viridis(
    discrete = TRUE
  ) + scale_shape_manual(values = seq(1,7))
```

```
## Scale for 'colour' is already present. Adding another scale for
## 'colour', which will replace the existing scale.
```

```
## Scale for 'shape' is already present. Adding another scale for 'shape',
## which will replace the existing scale.
```



```
barplot(sort(ranger::importance(rf.class$finalModel), decreasing = FALSE),
las = 2, horiz = TRUE, cex.names = 0.7,
col = colorRampPalette(colors = c("darkred", "white", "darkblue"))(18))
```



Boosting

```

boost.grid <- expand.grid(n.trees = seq(25, 1600, by = 25),
                        interaction.depth = 1:6,
                        shrinkage = c(0.019, 0.022, 0.025, 0.028,
                                      0.031, 0.034, 0.037, 0.040),
                        n.minobsinnode = 1)

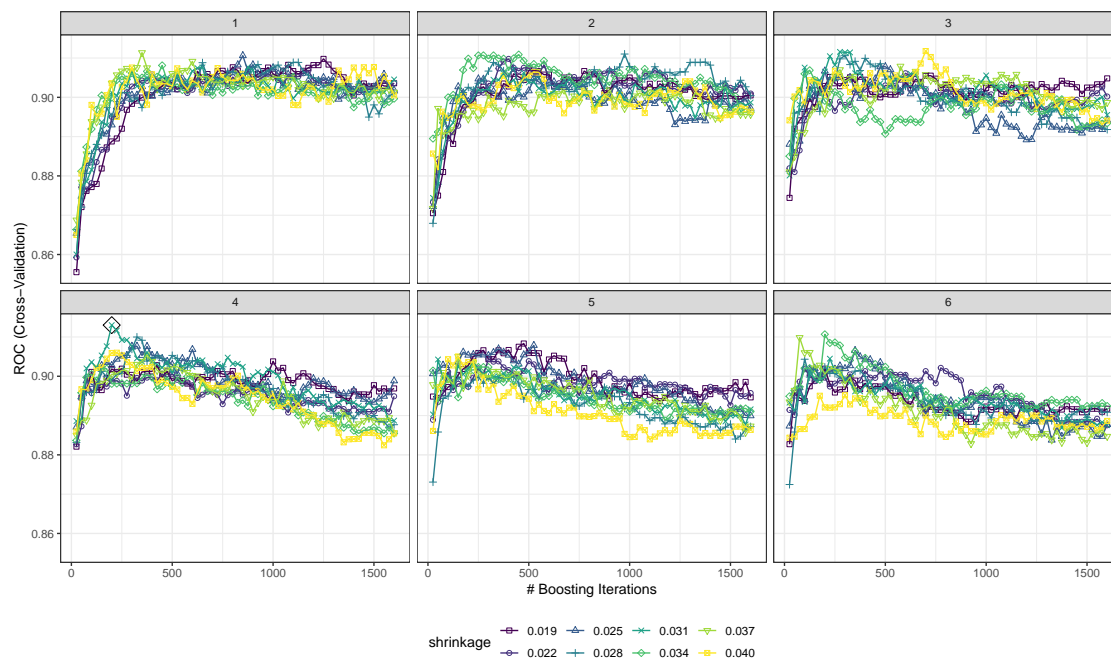
set.seed(1)
# Adaboost loss function
boost.class = train(train.x, train.y,
                    tuneGrid = boost.grid,
                    trControl = ctrl,
                    method = "gbm",
                    distribution = "adaboost",
                    metric = "ROC",
                    verbose = FALSE)

ggplot(boost.class, highlight = TRUE) +
  viridis::scale_color_viridis(
    discrete = TRUE
  ) + scale_shape_manual(values = seq(0,10))

```

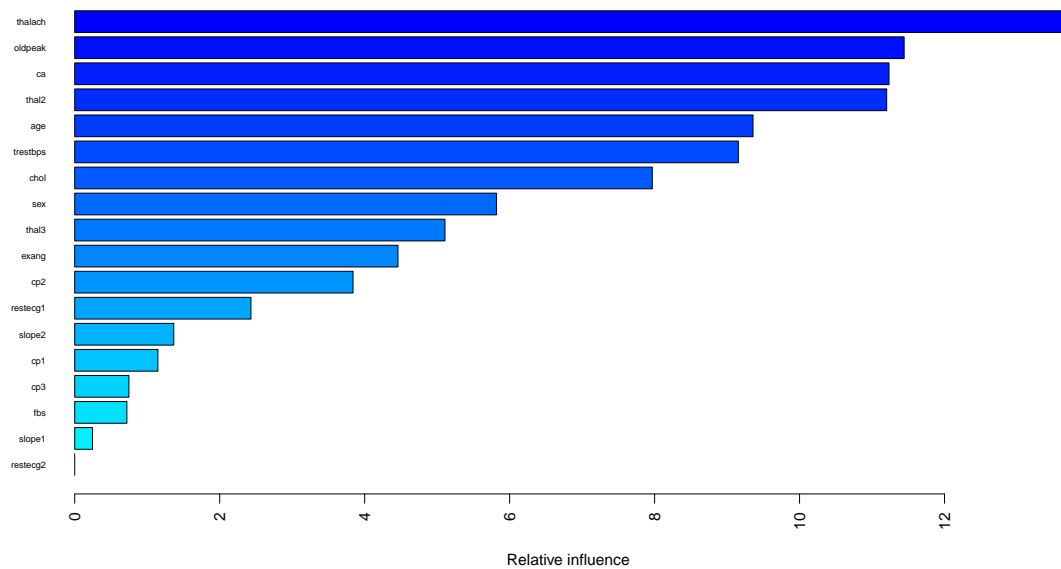
## Scale for 'colour' is already present. Adding another scale for  
## 'colour', which will replace the existing scale.

## Scale for 'shape' is already present. Adding another scale for 'shape',  
## which will replace the existing scale.



```
summary(boost.class$finalModel, las = 2, cBars = 19, cex.names = 0.6)
```





```
##           var    rel.inf
## thalach   thalach 13.7570379
## oldpeak   oldpeak 11.4426242
## ca        ca      11.2340461
## thal2     thal2   11.2018992
## age       age     9.3578463
## trestbps  trestbps 9.1552339
## chol      chol    7.9679361
## sex       sex     5.8183411
## thal3     thal3   5.1081799
## exang     exang   4.4596854
## cp2       cp2     3.8395550
## restecg1  restecg1 2.4312568
## slope2    slope2  1.3663795
## cp1       cp1     1.1477351
## cp3       cp3     0.7475945
## fbs       fbs     0.7196946
## slope1    slope1  0.2449543
## restecg2  restecg2 0.0000000
```

```
resamp = resamples(list(boost = boost.class, rf = rf.class,
                        bagging = bagging.class, tree = tree.class))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: boost, rf, bagging, tree
## Number of resamples: 10
##
```

```
## ROC
##           Min.   1st Qu.   Median     Mean   3rd Qu.     Max. NA's
## boost  0.8106061 0.8605372 0.9242424 0.9130165 0.9700413 1.0000000    0
## rf     0.8106061 0.8886019 0.9204545 0.9109504 0.9545455 0.9752066    0
## bagging 0.7954545 0.8522727 0.8942837 0.8893939 0.9318182 0.9696970    0
## tree   0.7537879 0.7926136 0.8162879 0.8341942 0.8747417 0.9583333    0
##
## Sens
##           Min.   1st Qu.   Median     Mean   3rd Qu.     Max. NA's
## boost  0.5833333 0.7500000 0.8257576 0.8143939 0.8901515 1.0000000    0
## rf     0.7500000 0.8219697 0.8712121 0.8643939 0.9166667 1.0000000    0
## bagging 0.6666667 0.7670455 0.8257576 0.8053030 0.8333333 0.9166667    0
## tree   0.6666667 0.7500000 0.8674242 0.8401515 0.9166667 1.0000000    0
##
## Spec
##           Min.   1st Qu.   Median     Mean   3rd Qu.     Max. NA's
## boost  0.5454545 0.7272727 0.8181818 0.7818182 0.8181818 0.9090909    0
## rf     0.5454545 0.7500000 0.8181818 0.7727273 0.8181818 0.9090909    0
## bagging 0.6363636 0.7500000 0.8181818 0.7909091 0.8181818 0.9090909    0
## tree   0.5454545 0.6590909 0.7727273 0.7454545 0.8181818 0.9090909    0
```

```
bwplot(resamp, metric = "ROC")
```

