Final Secondary Analysis

Tianshu Liu, Lincole Jiang, Jiong Ma

Contents

```
library(tidyverse)
library(summarytools)
library(corrplot)
library(caret)
library(vip)
library(rpart.plot)
library(ranger)
```

1 Model Training

1.1 Secondary Analysis

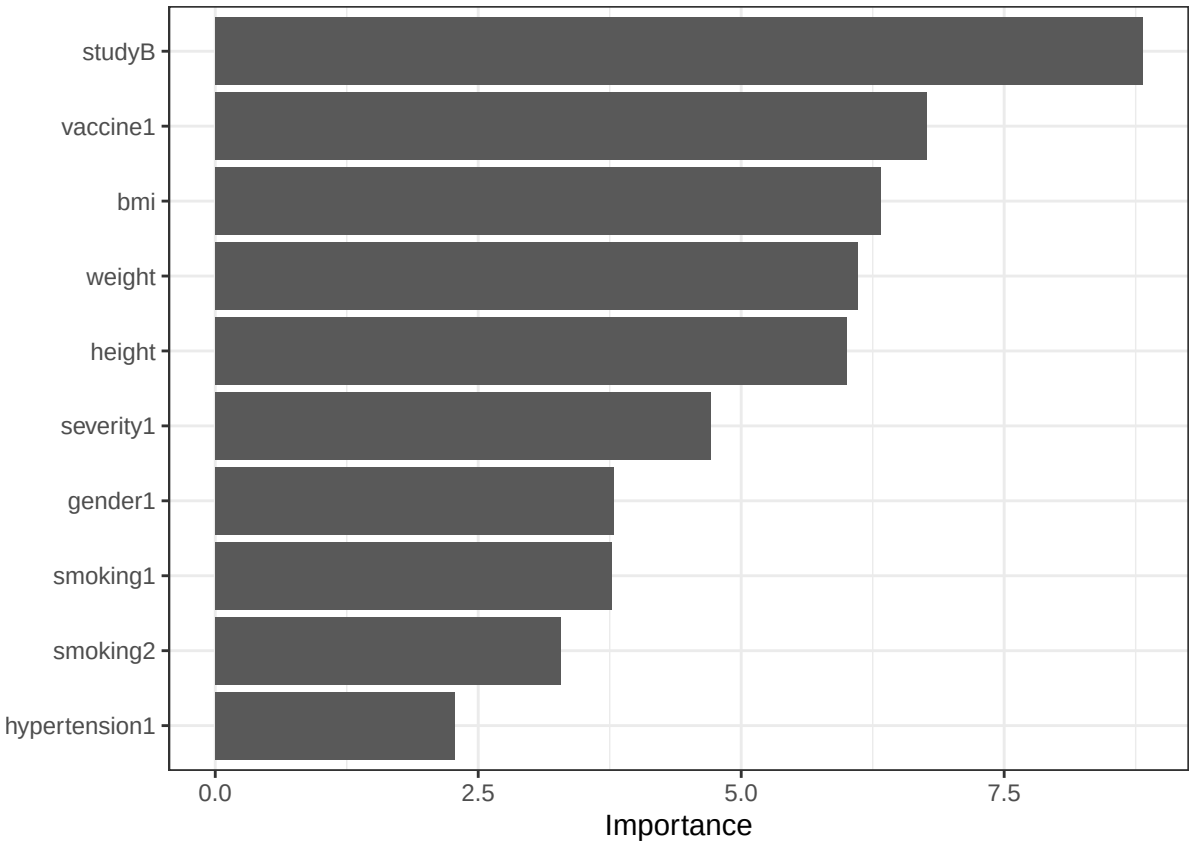
```
ctrl1 <- trainControl(method = "cv", number = 5)
```

1.1.1 Logistic Regression

```
set.seed(1)
glm.fit <- train(x = train.x,
                 y = train.bin.y,
                 method = 'glm',
                 trControl = ctrl1)
coef(glm.fit$finalModel)
```

```
##   (Intercept)           age       gender1         race2         race3
## -85.313351100   0.014271893  -0.323467202  -0.104376256  -0.039351201
##         race4      smoking1      smoking2        height        weight
##   0.003550318   0.367190652   0.502782618   0.502637208  -0.545510559
##           bmi hypertension1     diabetes1           SBP           LDL
##   1.634689792   0.325697328  -0.070567897  -0.005832901  -0.001829020
##      vaccine1     severity1        studyB        studyC
##  -0.600151829   0.761039467  -1.066825060  -0.031460504
```

```
vip(glm.fit$finalModel) + theme_bw()
```

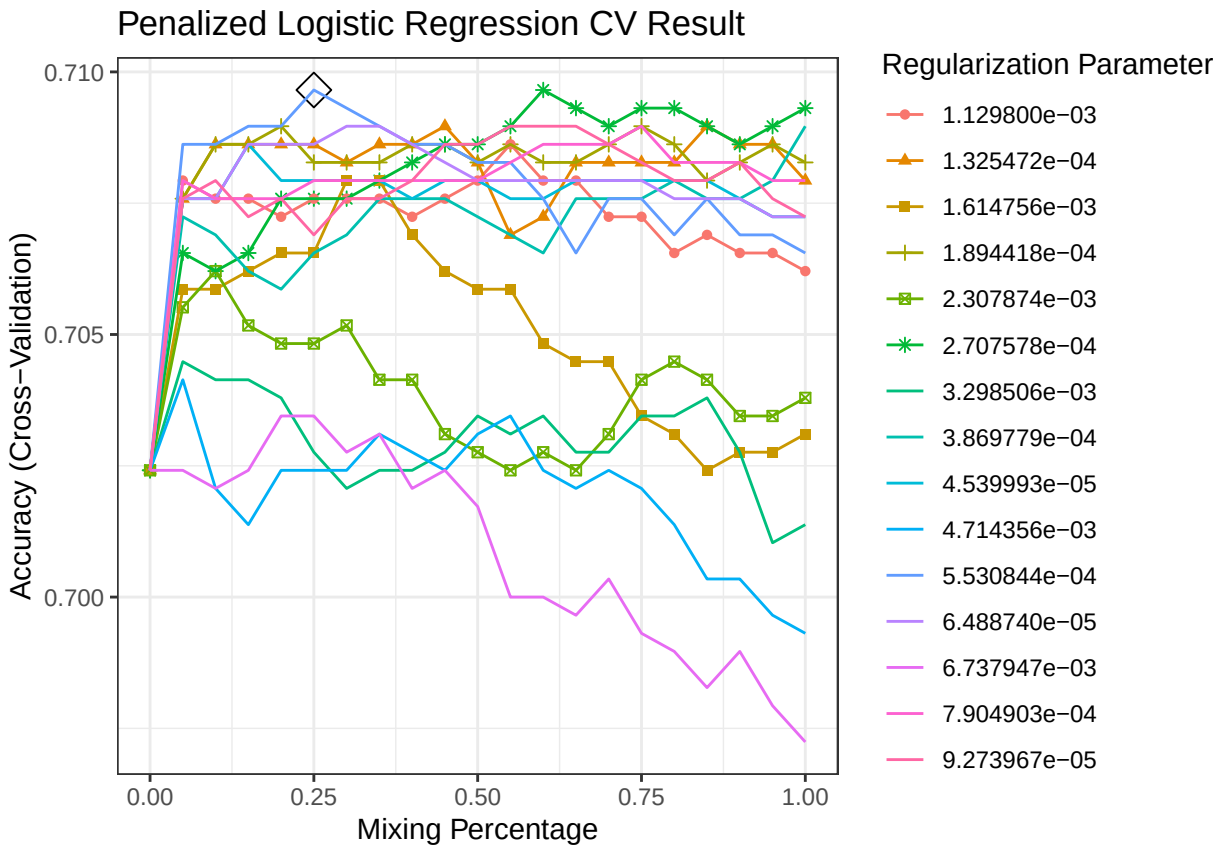


1.1.2 Penalized Logistic Regression

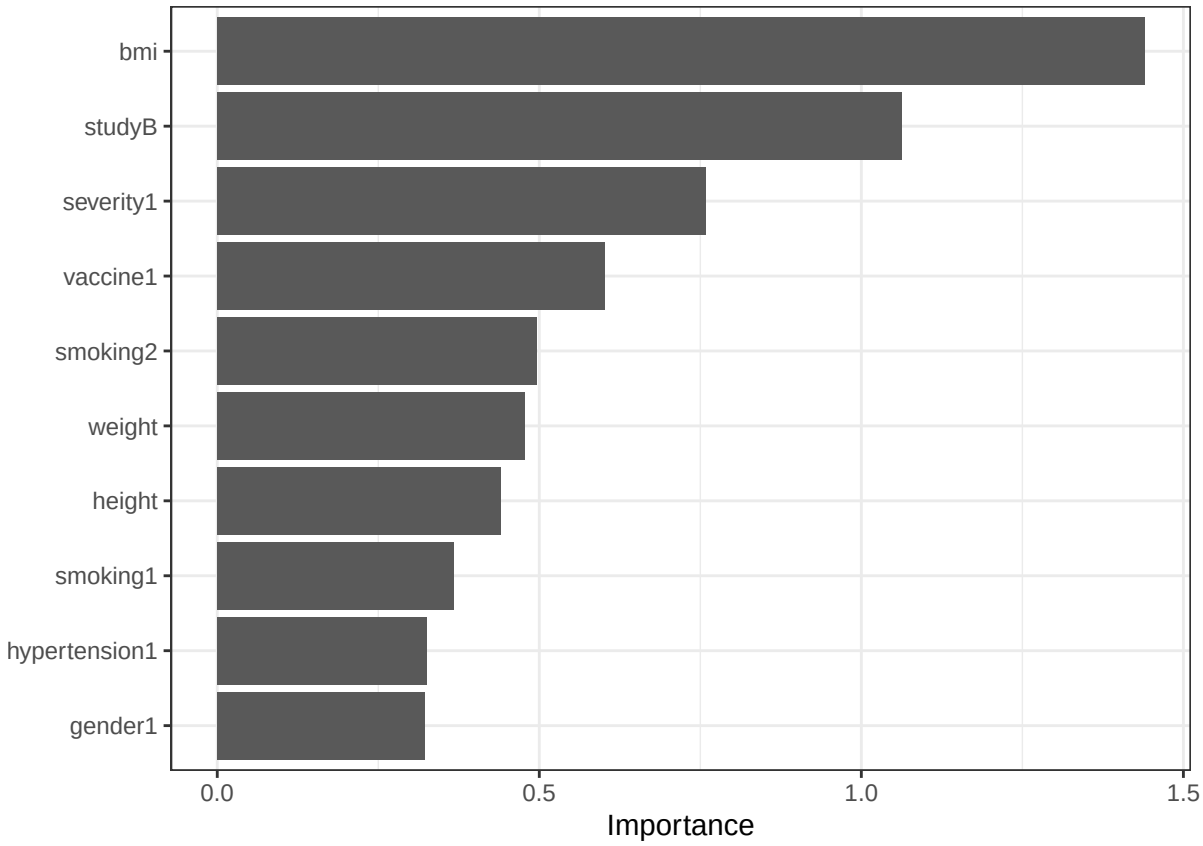
```
glmnGrid <- expand.grid(.alpha = seq(0, 1, length = 21),
                        .lambda = exp(seq(-10, -5, length = 15)))
set.seed(1)
glmn.fit <- train(train.x,
                  train.bin.y,
                  method = 'glmnet',
                  tuneGrid = glmnGrid,
                  trControl = ctrl1)

glmn.fit$bestTune
```

```
##    alpha       lambda
## 83  0.25 0.0005530844
```

```
myCol<- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
              superpose.line = list(col = myCol))

ggplot(glmn.fit, highlight = TRUE) +
  labs(title="Penalized Logistic Regression CV Result") +
  theme_bw()
```

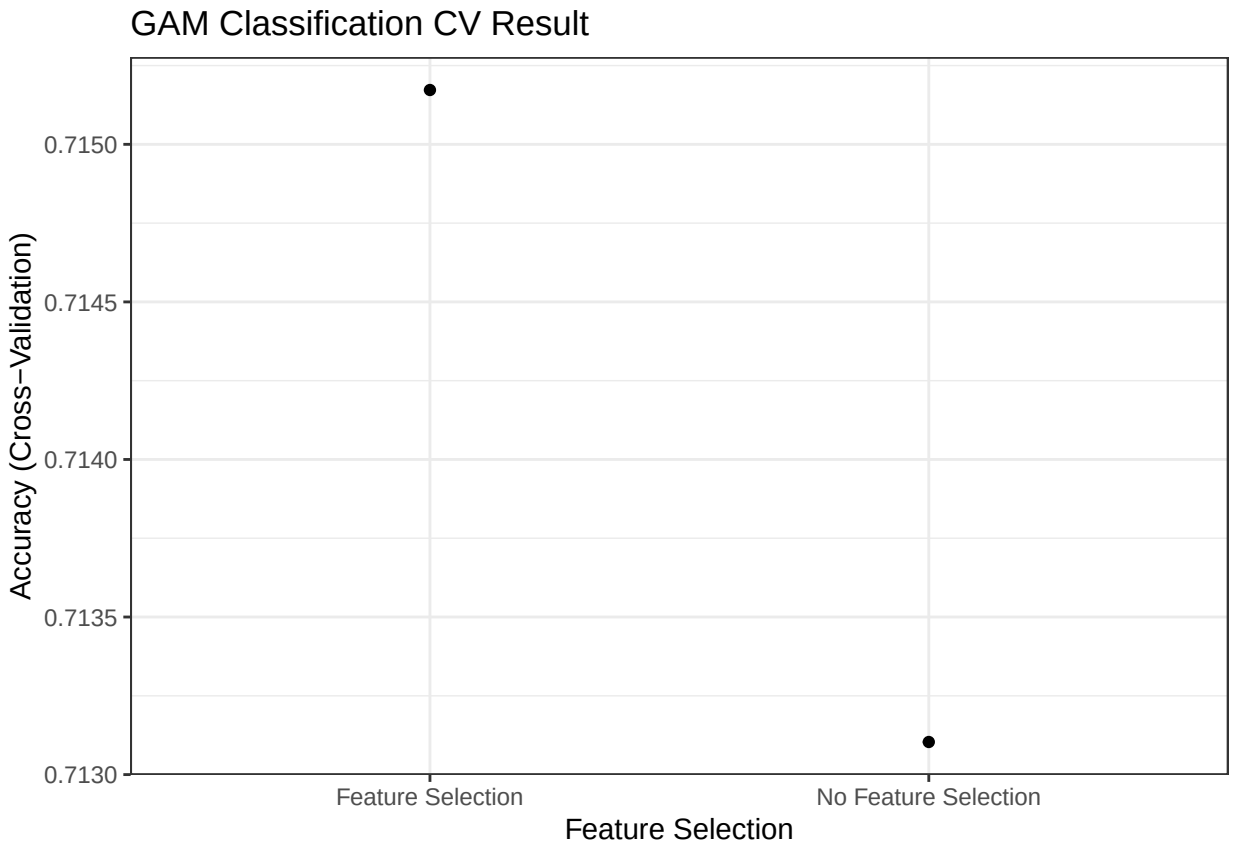



```
ggsave("./figure/penal_logi_cv.jpeg", dpi = 500)

#coef(glmn.fit$finalModel)
vip(glmn.fit$finalModel) + theme_bw()
```



1.1.3 Generalized Additive Model (GAM) for classification

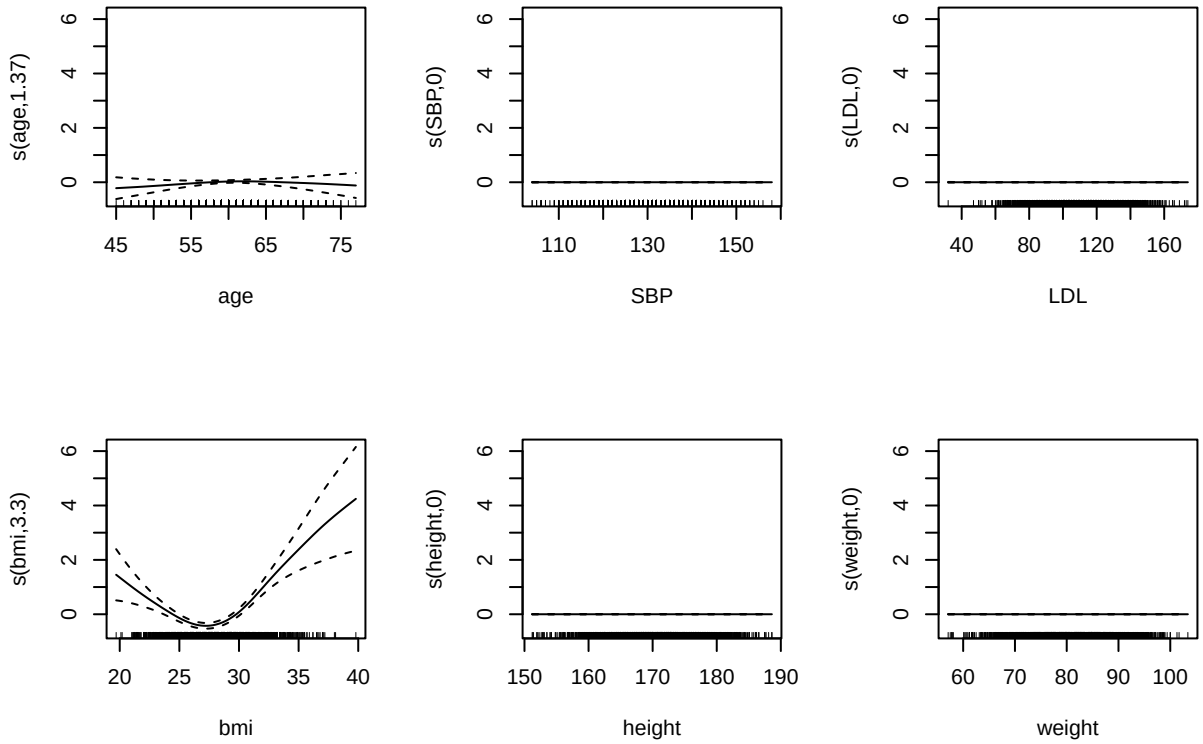
```
set.seed(1)
gam.bin.fit <- train(train.x,
                     train.bin.y,
                     method = "gam",
                     trControl = ctrl1)

ggplot(gam.bin.fit) +
  labs(title = "GAM Classification CV Result") +
  theme_bw()
```




```
ggsave("./figure/gam_binned_cv.jpeg", dpi = 500)

gam.bin.fit$bestTune
```

```
##   select method
## 2   TRUE GCV.Cp
```

```
par(mfrow=c(2, 3))
plot(gam.bin.fit$finalModel)
```



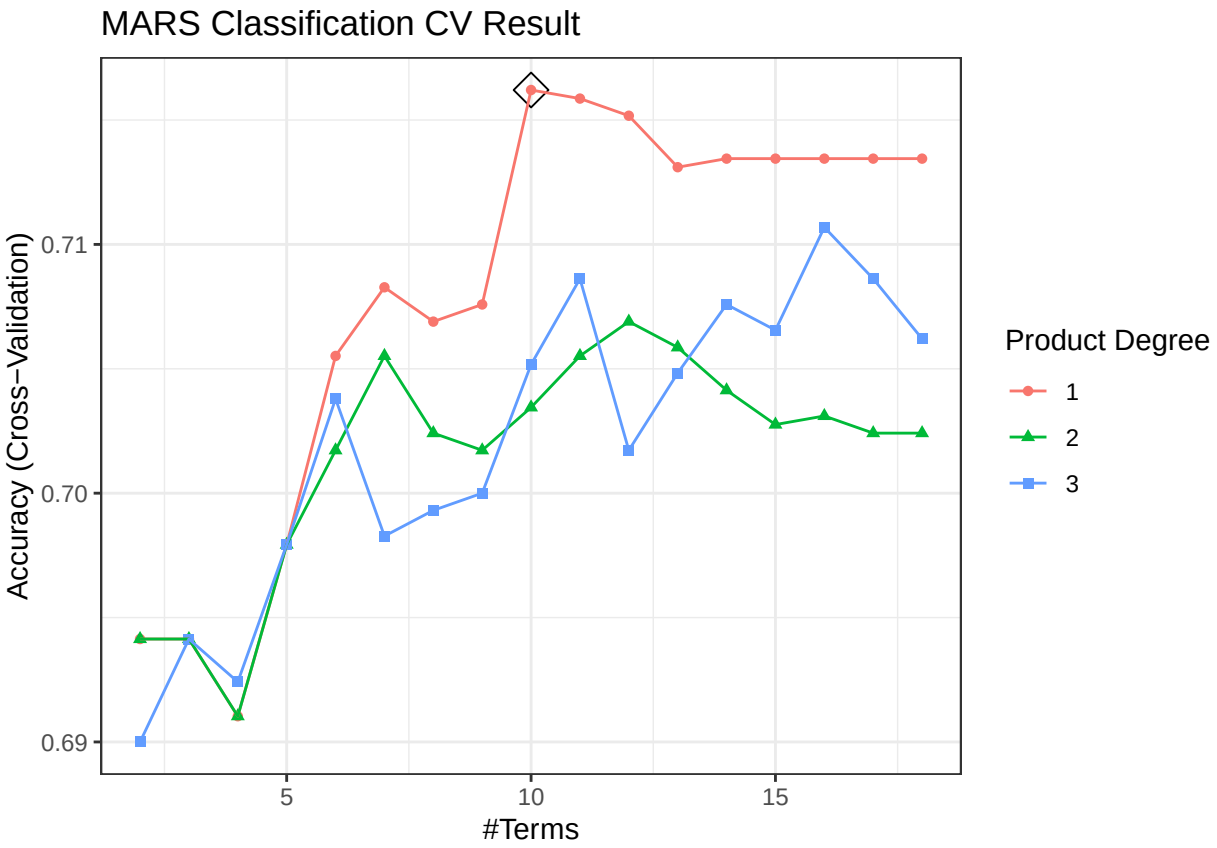

```
par(mfrow=c(1, 1))
```

1.1.4 Multivariate Adaptive Regression Splines (MARS) for classification

```
set.seed(1)
mars.bin.fit <- train(train.x,
                      train.bin.y,
                      method = "earth",
                      tuneGrid = expand.grid(degree = 1:3,
                                             nprune = 2:ncol(train.x)),
                      trControl = ctrl1)

mars.bin.fit$bestTune
```

```
##   nprune degree
## 9     10      1
```

```
ggplot(mars.bin.fit, highlight = TRUE) +
  labs(title  ="MARS Classification CV Result") +
  theme_bw()
```




```
ggsave("./figure/mars_binned_cv.jpeg", dpi = 500)

mars.bin.fit$bestTune
```

```
##   nprune degree
## 9     10      1
```

```
coef(mars.bin.fit$finalModel) %>%
  broom::tidy() %>%
  knitr::kable()
```

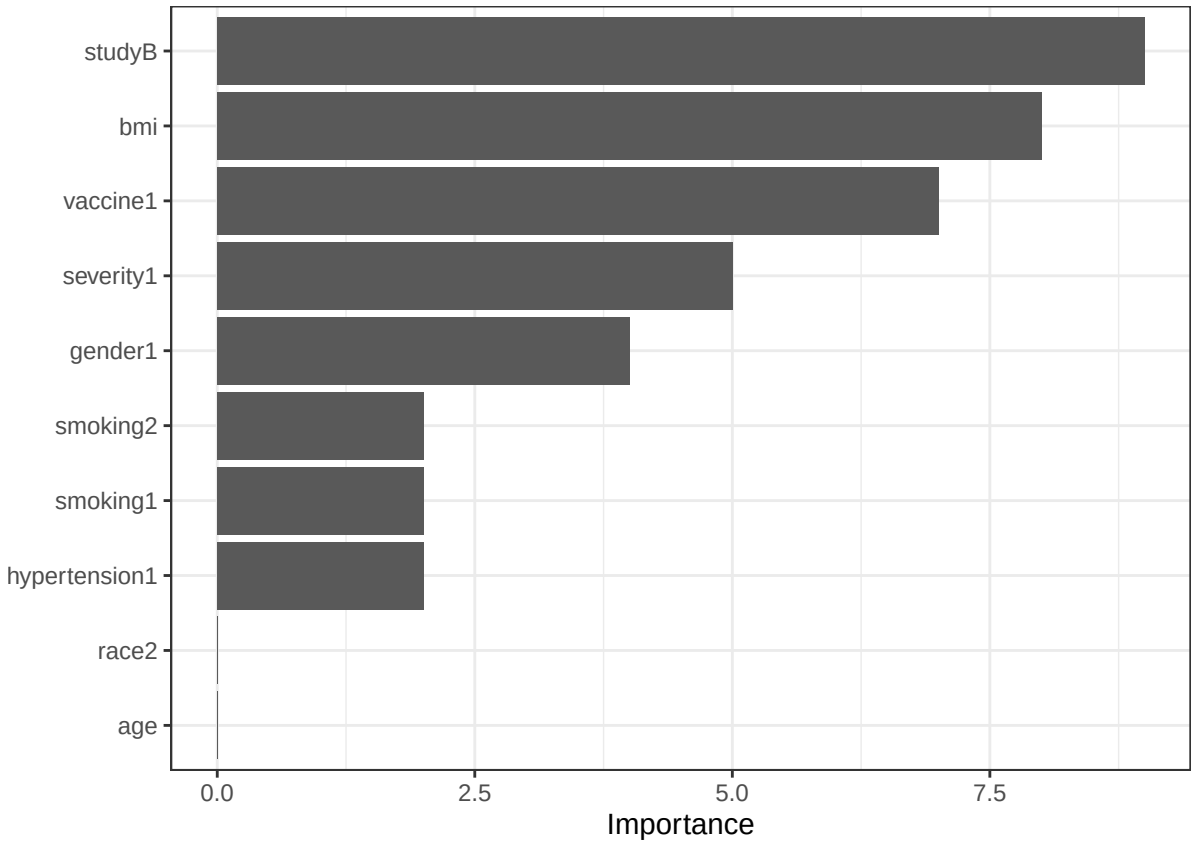
names	x
(Intercept)	1.1021073
studyB	-1.0784308
h(bmi-26.9)	0.2906422
h(26.9-bmi)	0.2906009
vaccine1	-0.6182904
severity1	0.8024101
gender1	-0.3318841
hypertension1	0.3034198
smoking1	0.3860960
smoking2	0.5281300

```
summary(mars.bin.fit$finalModel)
```

```
## Call: earth(x=matrix[2900,18], y=factor.object, keepxy=TRUE,
##             glm=list(family=function.object, maxit=100), degree=1, nprune=10)
```

```
##
## GLM coefficients
##                      gt30
## (Intercept)     1.1021073
## gender1        -0.3318841
## smoking1        0.3860960
## smoking2        0.5281300
## hypertension1   0.3034198
## vaccine1       -0.6182904
## severity1       0.8024101
## studyB         -1.0784308
## h(26.9-bmi)     0.2906009
## h(bmi-26.9)     0.2906422
##
## GLM (family binomial, link logit):
##  nulldev   df       dev   df   devratio     AIC iters converged
##  3571.35 2899   3209.82 2890      0.101    3230     4         1
##
## Earth selected 10 of 14 terms, and 8 of 18 predictors (nprune=10)
## Termination condition: RSq changed by less than 0.001 at 14 terms
## Importance: studyB, bmi, vaccine1, severity1, gender1, smoking1, smoking2, ...
## Number of terms at each degree of interaction: 1 9 (additive model)
## Earth GCV 0.1908237    RSS 546.1611    GRSq 0.1018244    RSq 0.1129434
```

```
vip(mars.bin.fit$finalModel) + theme_bw()
```



1.1.5 Linear Discriminant Analysis (LDA)

```
set.seed(1)
lda.fit <- train(train.x,
                 train.bin.y,
                 method = "lda",
                 trControl = ctrl1)
```

1.1.6 Quadratic Discriminant Analysis (QDA)

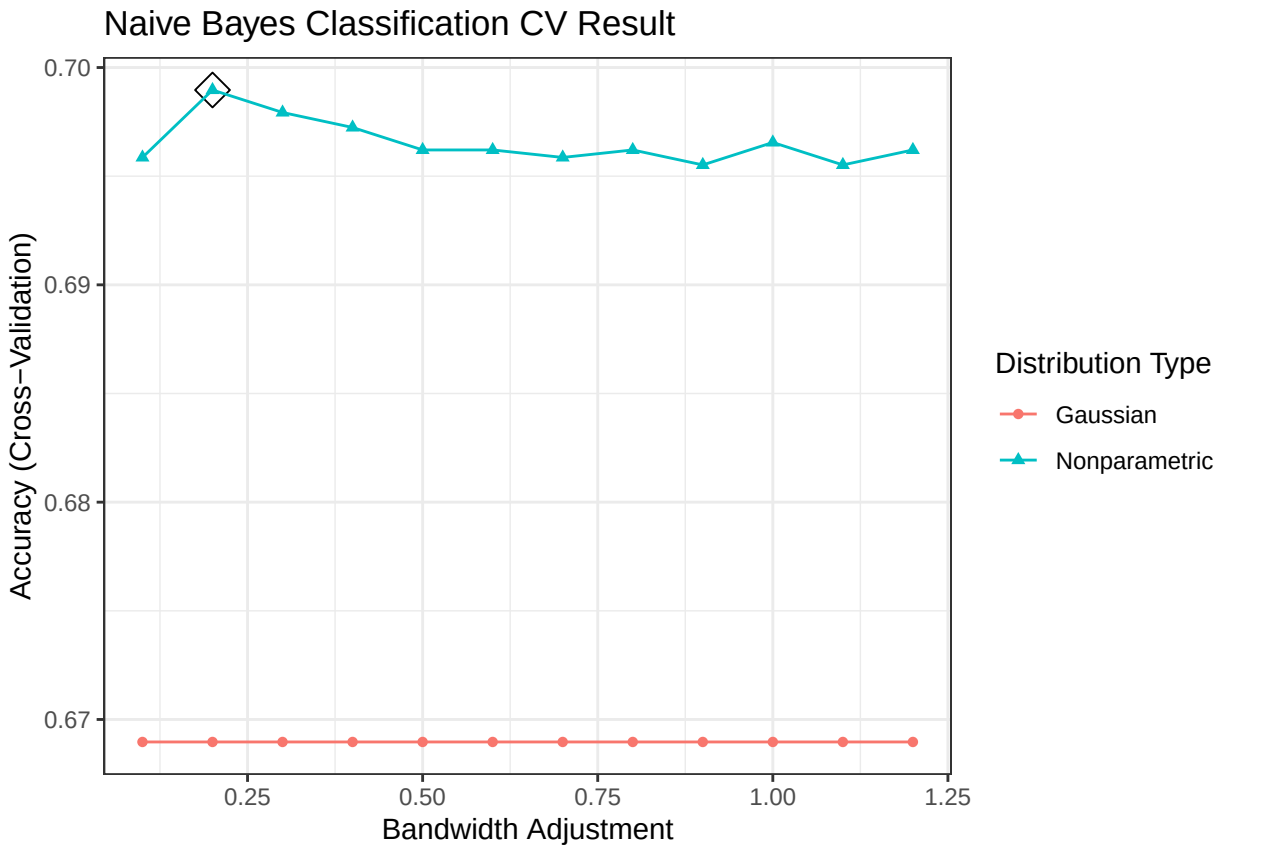
```
set.seed(1)
qda.fit <- train(train.x,
                  train.bin.y,
                  method = "qda",
                  trControl = ctrl1)
```

1.1.7 Naive Bayes (NB)

```
nbGrid <- expand.grid(usekernel = c(FALSE,TRUE),
                      fL = 1,
                      adjust = seq(0.1, 1.2, by = .1))
set.seed(1)
nb.fit <- train(train.x,
                train.bin.y,
                method = "nb",
                tuneGrid = nbGrid,
                trControl = ctrl1)
nb.fit$bestTune
```

```
##    fL usekernel adjust
## 14  1      TRUE    0.2
```

```
ggplot(nb.fit, highlight = TRUE) +
  labs(title  ="Naive Bayes Classification CV Result") +
  theme_bw()
```



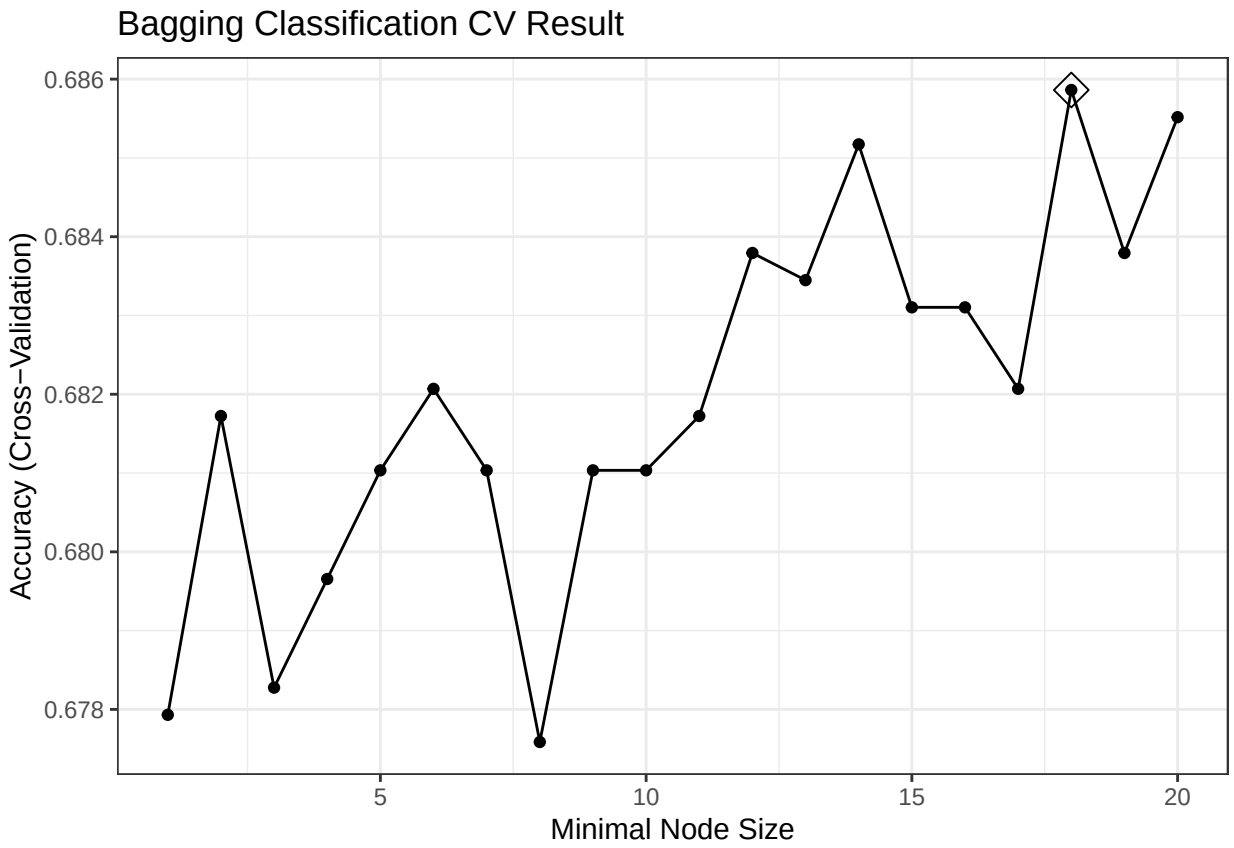

```
ggsave("./figure/nb_cv.jpeg", dpi = 500)
```

1.1.8 Bagging

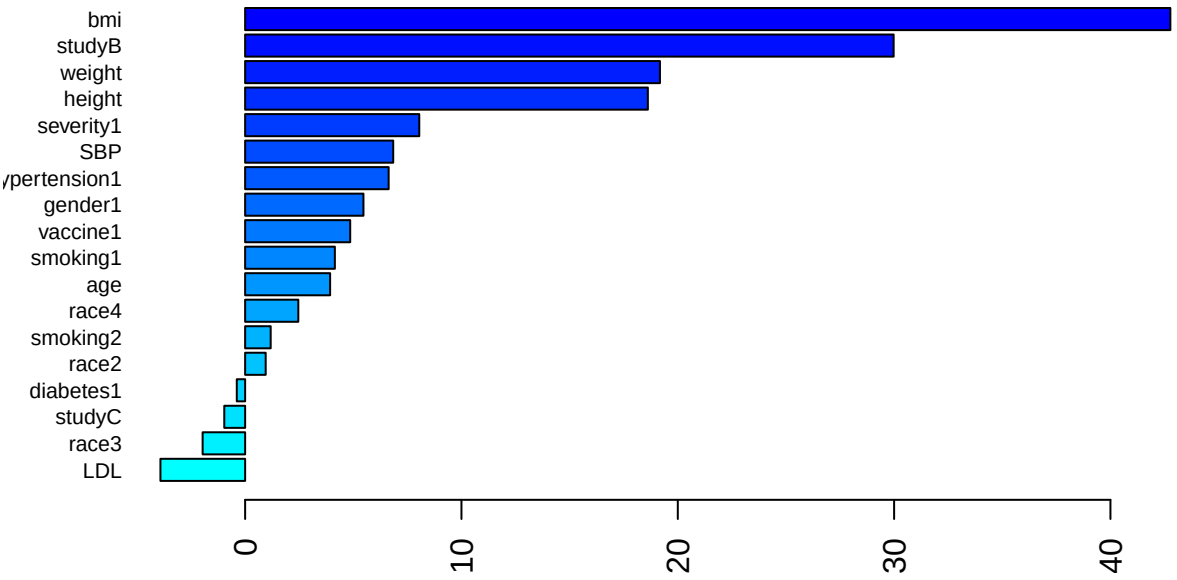
```
bag.grid2 <- expand.grid(mtry = ncol(train.x),
                         splitrule = "gini",
                         min.node.size = 1:20)
set.seed(1)
bag.fit2 <- train(train.x,
                  train.bin.y,
                  method = "ranger",
                  tuneGrid = bag.grid2,
                  trControl = ctrl1)

bag.fit2$bestTune
```

```
##    mtry splitrule min.node.size
## 18   18      gini            18
```

```
ggplot(bag.fit2, highlight = TRUE) +
  labs(title = "Bagging Classification CV Result") +
  theme_bw()
```




```
ggsave("./figure/bagging_classification_cv.jpeg", dpi = 500)

bag.final.per2 <- ranger(recovery_time ~ .,
                         data = train.bin.dat.matrix,
                         mtry = ncol(train.x),
                        splitrule = "gini",
                        min.node.size = bag.fit2$bestTune[[3]],
                        importance = "permutation",
                        scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(bag.final.per2),
             decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(ncol(train.x)))
```

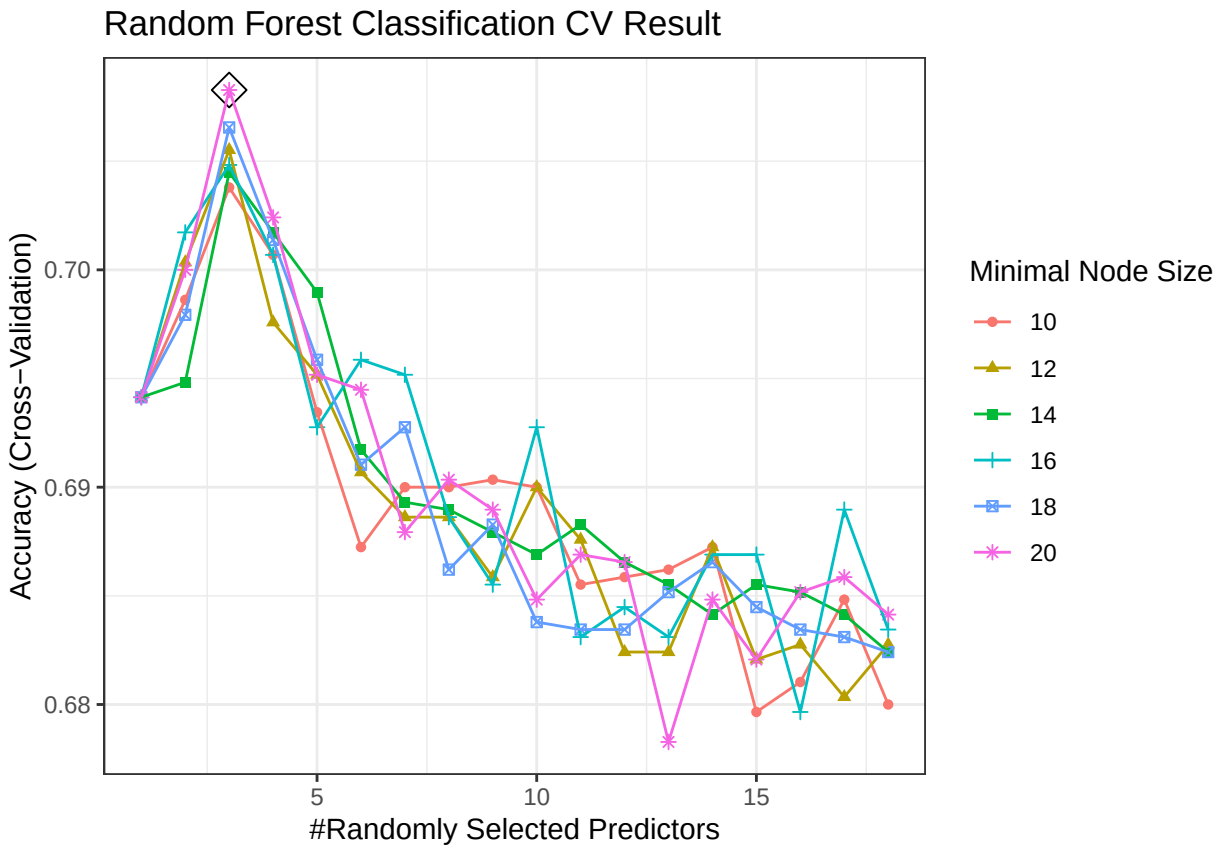


1.1.9 Random Forest

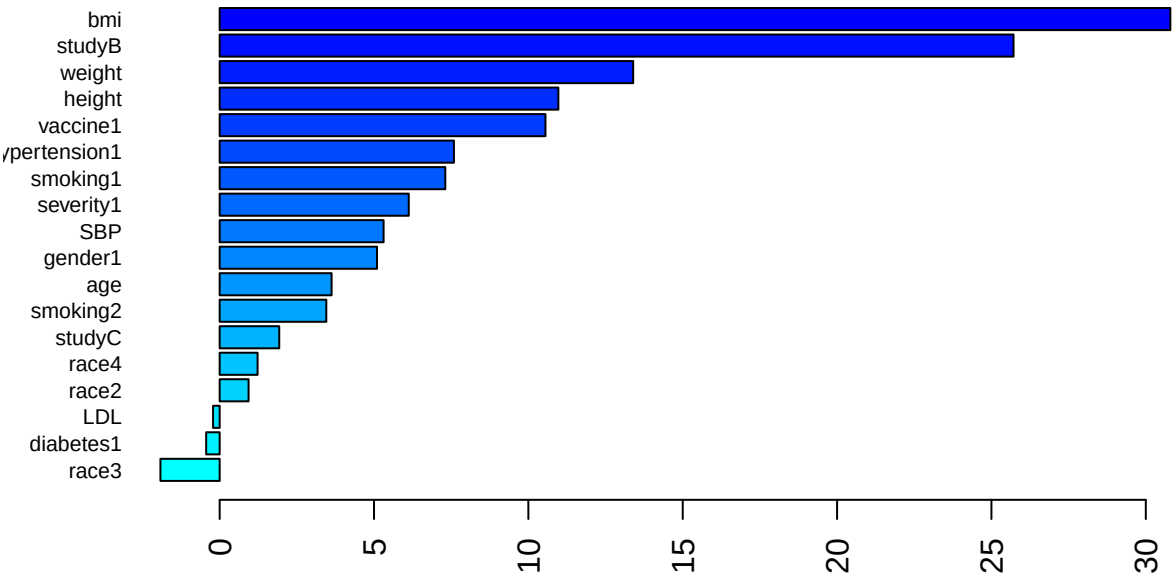
```
rf.grid2 <- expand.grid(mtry = 1:ncol(train.x),
                        splitrule = "gini",
                        min.node.size = seq(10, 20, by=2))
set.seed(1)
rf.fit2 <- train(train.x,
                 train.bin.y,
                 method = "ranger",
                 tuneGrid = rf.grid2,
                 trControl = ctrl1)

rf.fit2$bestTune
```

```
##    mtry splitrule min.node.size
## 18    3      gini            20
```

```
ggplot(rf.fit2, highlight = TRUE) +
  labs(title = "Random Forest Classification CV Result") +
  theme_bw()
```




```
ggsave("./figure/rf_classification_cv.jpeg", dpi = 500)

rf.final.per2 <- ranger(recovery_time ~ .,
                        data = train.bin.dat.matrix,
                        mtry = rf.fit2$bestTune[[1]],
                        splitrule = "gini",
                        min.node.size = rf.fit2$bestTune[[3]],
                        importance = "permutation",
                        scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(rf.final.per2), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(ncol(train.x)))
```

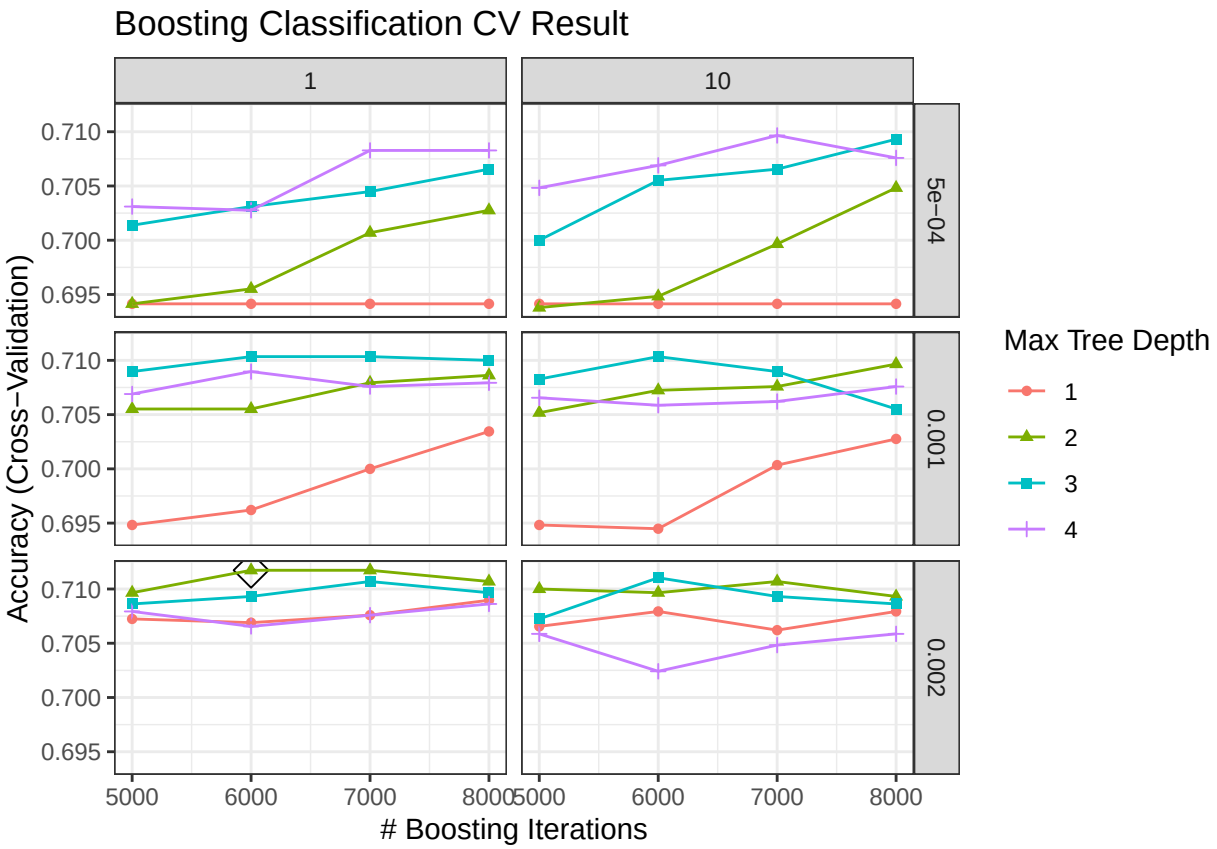


1.1.10 Boosting

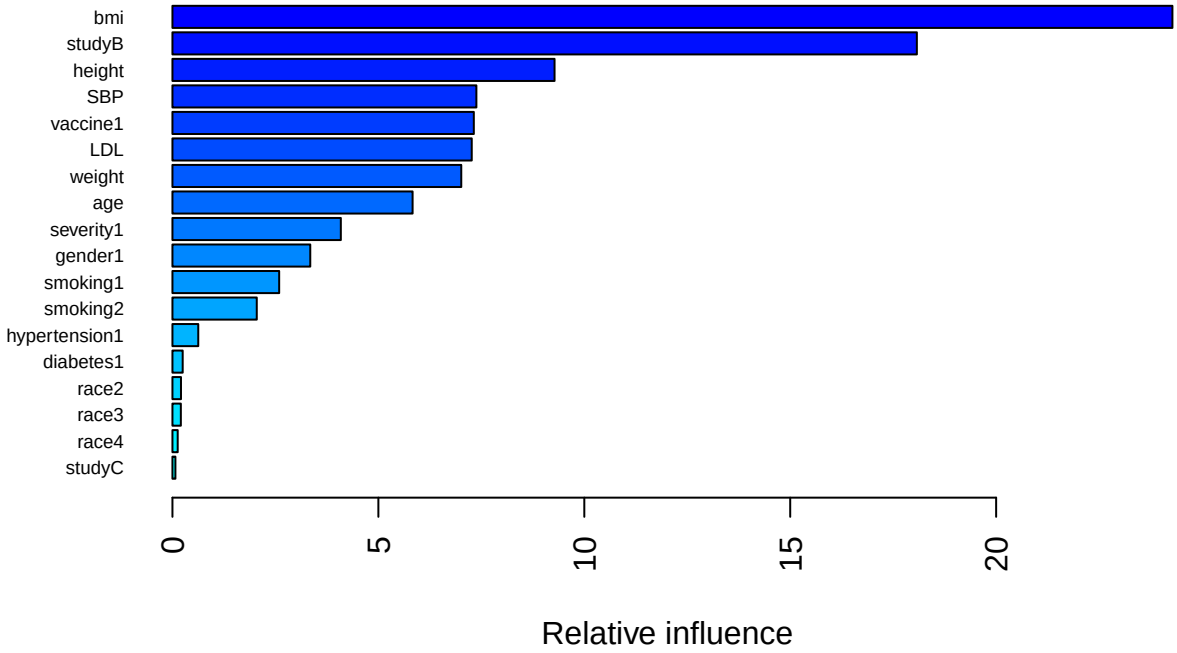
```
set.seed(1)
bst.grid2 <- expand.grid(n.trees = c(5000, 6000, 7000, 8000),
                         interaction.depth = 1:4,
                         shrinkage = c(0.0005, 0.001, 0.002),
                         n.minobsinnode = c(1,10))

bst.fit2 <- train(train.x,
                  train.bin.y,
                  method = "gbm",
                  tuneGrid = bst.grid2,
                  trControl = ctrl1,
                  verbose = FALSE)

bst.fit2$bestTune
```

```
##    n.trees interaction.depth shrinkage n.minobsinnode
## 74    6000                 2     0.002              1
```

```
ggplot(bst.fit2, highlight = TRUE) +
  labs(title = "Boosting Classification CV Result") +
  theme_bw()
```




```
ggsave("./figure/boosting_classification_cv.jpeg", dpi = 500)

# Variable Importance
summary(bst.fit2$finalModel, las = 2, cBars = ncol(train.x), cex.names = 0.6)
```




```
##                        var     rel.inf
```

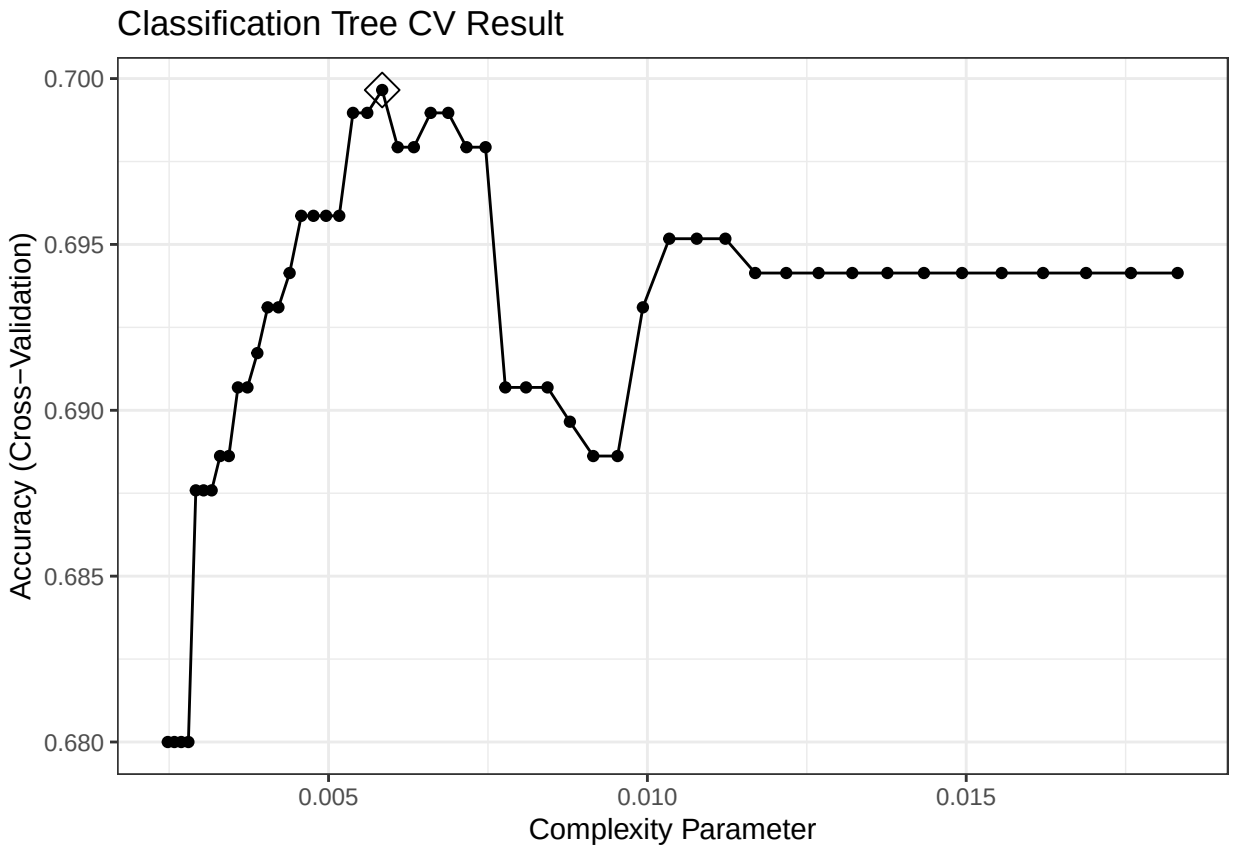
```
## bmi                    bmi 24.27996796
## studyB              studyB 18.07482490
## height              height  9.27741014
## SBP                    SBP  7.37997245
## vaccine1          vaccine1  7.31766352
## LDL                    LDL  7.26693577
## weight              weight  7.01275278
## age                    age  5.83057254
## severity1        severity1  4.08846656
## gender1            gender1  3.34607451
## smoking1          smoking1  2.59108804
## smoking2          smoking2  2.04775125
## hypertension1 hypertension1  0.62685044
## diabetes1        diabetes1  0.24890993
## race2                race2  0.20737303
## race3                race3  0.20408588
## race4                race4  0.12758383
## studyC              studyC  0.07171648
```

1.1.11 Classification Trees

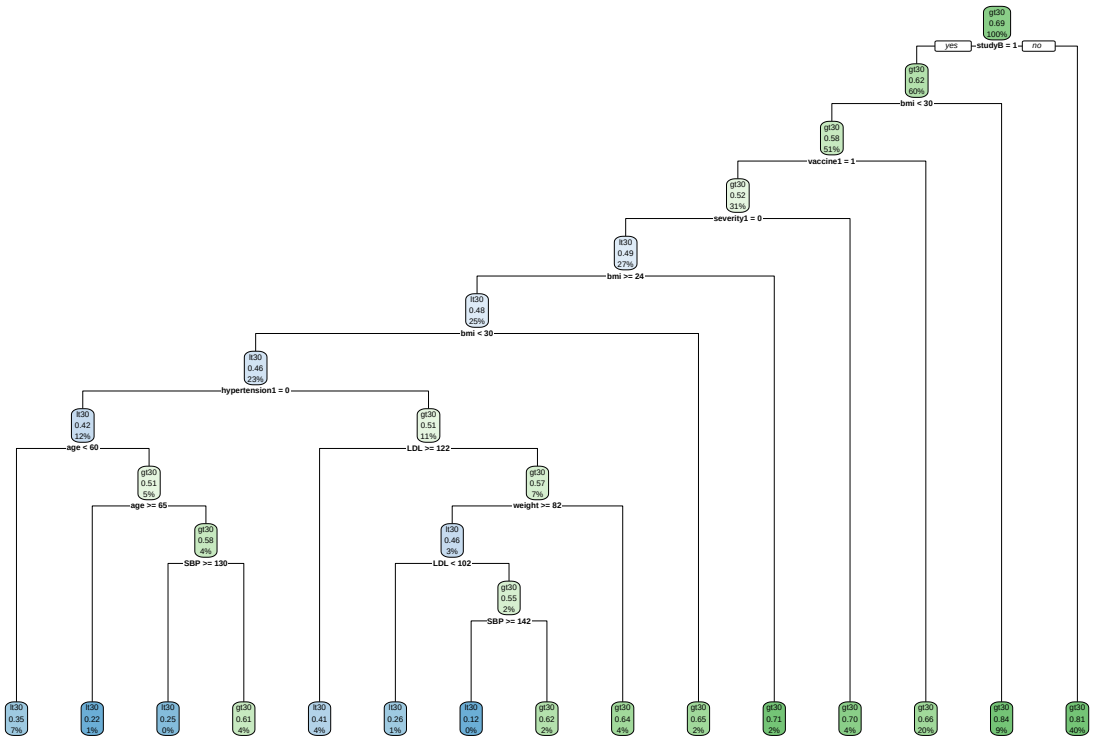
```
rpart.grid = expand.grid(cp = exp(seq(-6,-4, len = 50)))
set.seed(1)
rpart.fit2 <- train(train.x,
                    train.bin.y,
                    method = "rpart",
                    tuneGrid = rpart.grid,
                    trControl = ctrl1)

rpart.fit2$bestTune
```

```
##            cp
## 22 0.005840977
```

```
ggplot(rpart.fit2, highlight = TRUE) +
  labs(title = "Classification Tree CV Result") +
  theme_bw()
```




```
ggsave("./figure/rpart2_cv.jpeg", dpi = 500)

rpart.plot(rpart.fit2$finalModel)
```

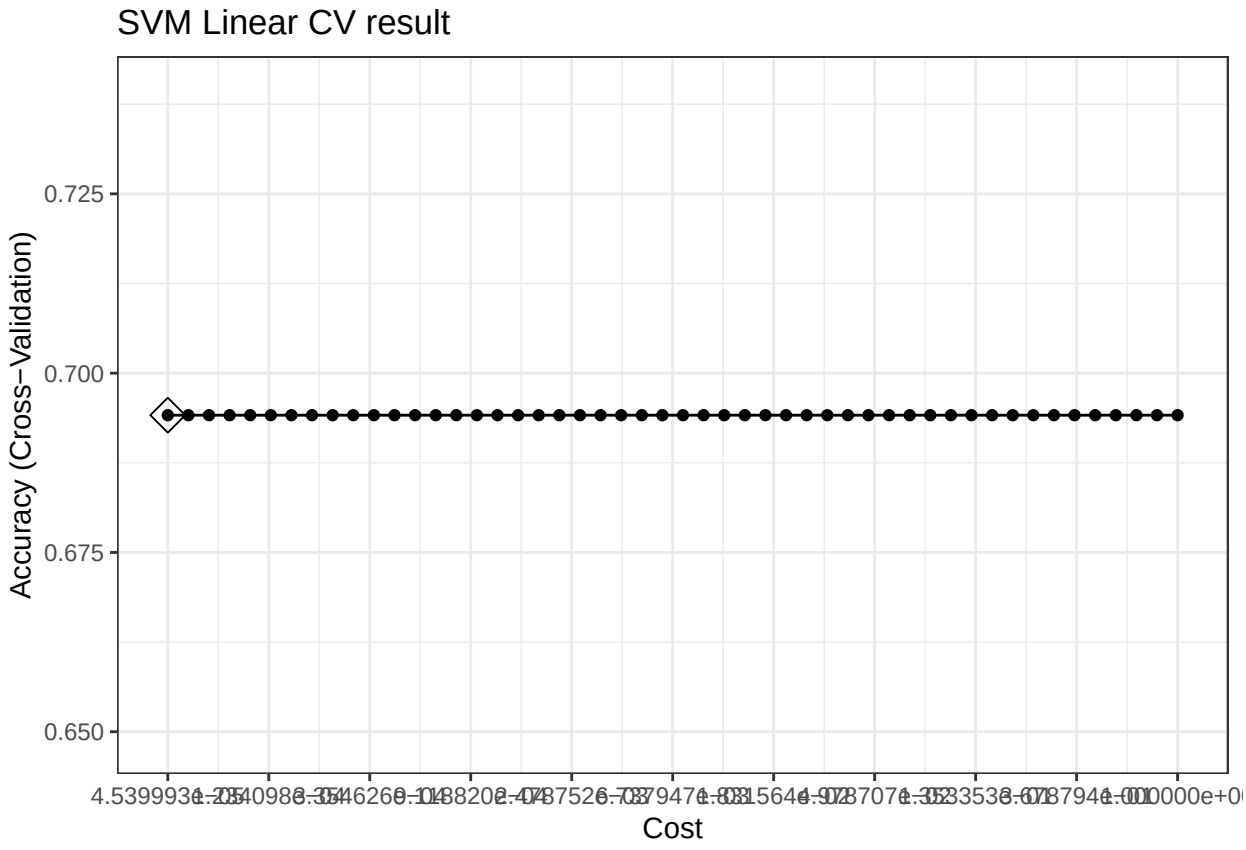


```
jpeg("./figure/rpart2.jpeg", width = 8, height = 6, units="in", res=500)
rpart.plot(rpart.fit2$finalModel)
dev.off()
```

```
## pdf
##   2
```

1.1.12 Support Vector Machine (SVM)

```
set.seed(1)
svml.fit <- train(train.x,
                  train.bin.y,
                  method = "svmLinear",
                  tuneGrid = data.frame(C = exp(seq(-10, 0,len=50))),
                  trControl = ctrl1)
ggplot(svml.fit, highlight = TRUE) +
  scale_x_continuous(trans='log',n.breaks = 10) +
  labs(title = "SVM Linear CV result") +
  theme_bw()
```

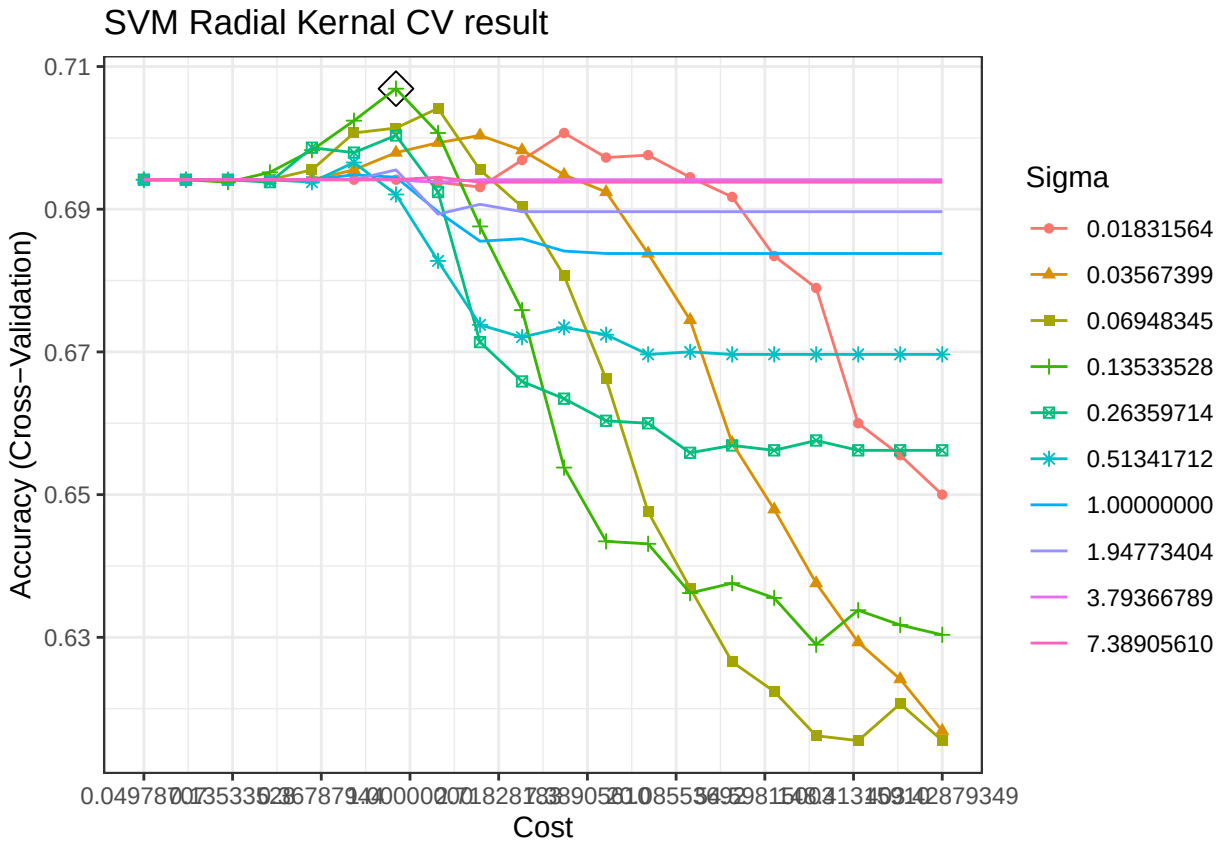



```
svmr.grid <- expand.grid(C = exp(seq(-3, 6, len = 20)),
                         sigma = exp(seq(-4, 2, len = 10)))

set.seed(1)
svmr.fit <- train(train.x,
                  train.bin.y,
                  method = "svmRadialSigma",
```

```
                  tuneGrid = svmr.grid,
                  trControl = ctrl1)

svmr.fit$bestTune
```

```
##        sigma         C
## 64 0.1353353 0.8539397
```

```
myCol<- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
              superpose.line = list(col = myCol))
ggplot(svmr.fit, highlight = TRUE, par.settings = myPar) +
  scale_x_continuous(trans='log',n.breaks = 10) +
  labs(title = "SVM Radial Kernal CV result") +
  theme_bw()
```




```
ggsave("./figure/svmr_cv.jpeg", dpi = 500)

confusionMatrix(svmr.fit)
```

```
## Cross-Validated (5 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##            Reference
## Prediction lt30 gt30
##       lt30  7.0  5.8
##       gt30 23.6 63.7
```

```
##
##  Accuracy (average) : 0.7069
```

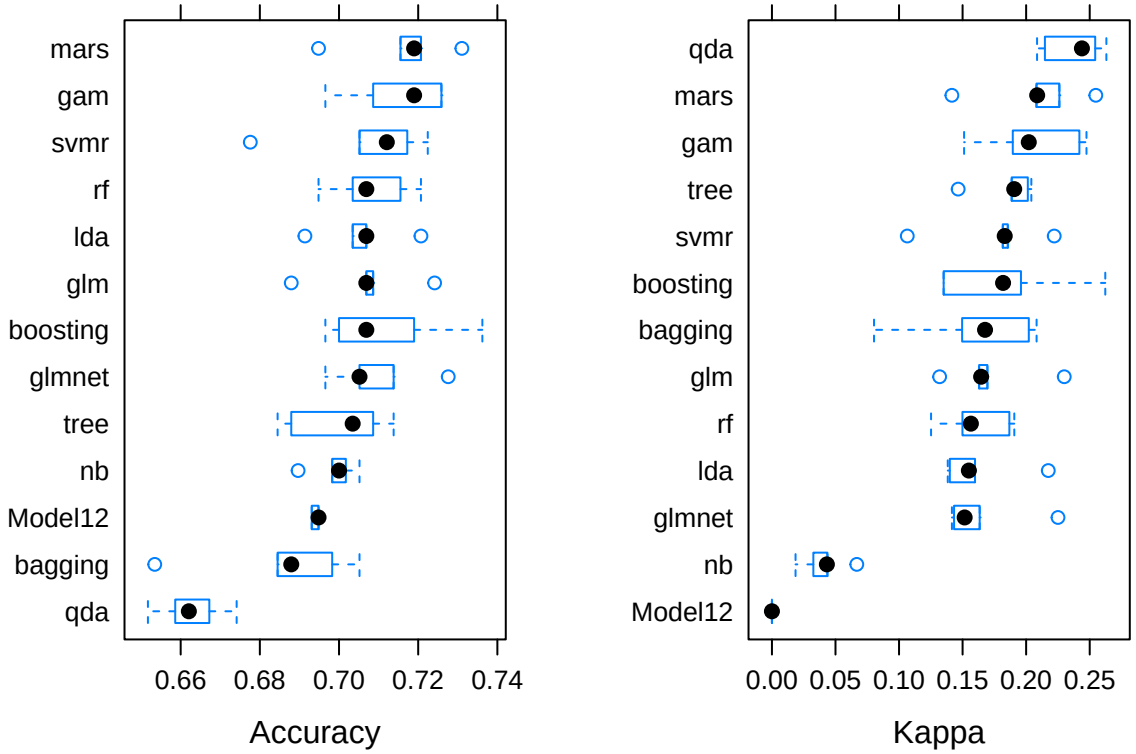
1.2 Model Selection

```
set.seed(1)
resamp <- resamples(list(glm = glm.fit,
                         glmnet = glmn.fit,
                         gam = gam.bin.fit,
                         mars = mars.bin.fit,
                         lda = lda.fit,
                         qda = qda.fit,
                         nb = nb.fit,
                         bagging = bag.fit2,
                         rf = rf.fit2,
                         boosting = bst.fit2,
                         tree = rpart.fit2,
                         svml <- svml.fit,
                         svmr = svmr.fit))

summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: glm, glmnet, gam, mars, lda, qda, nb, bagging, rf, boosting, tree, Model12, svmr
## Number of resamples: 5
##
## Accuracy
##                Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## glm      0.6879310 0.7068966 0.7068966 0.7068966 0.7086207 0.7241379    0
## glmnet   0.6965517 0.7051724 0.7051724 0.7096552 0.7137931 0.7275862    0
## gam      0.6965517 0.7086207 0.7189655 0.7151724 0.7258621 0.7258621    0
## mars     0.6948276 0.7155172 0.7189655 0.7162069 0.7206897 0.7310345    0
## lda      0.6913793 0.7034483 0.7068966 0.7058621 0.7068966 0.7206897    0
## qda      0.6517241 0.6586207 0.6620690 0.6627586 0.6672414 0.6741379    0
## nb       0.6896552 0.6982759 0.7000000 0.6989655 0.7017241 0.7051724    0
## bagging  0.6534483 0.6844828 0.6879310 0.6858621 0.6982759 0.7051724    0
## rf       0.6948276 0.7034483 0.7068966 0.7082759 0.7155172 0.7206897    0
## boosting 0.6965517 0.7000000 0.7068966 0.7117241 0.7189655 0.7362069    0
## tree     0.6844828 0.6879310 0.7034483 0.6996552 0.7086207 0.7137931    0
## Model12  0.6931034 0.6931034 0.6948276 0.6941379 0.6948276 0.6948276    0
## svmr     0.6775862 0.7051724 0.7120690 0.7068966 0.7172414 0.7224138    0
##
## Kappa
##                 Min.    1st Qu.     Median       Mean    3rd Qu.       Max. NA's
## glm      0.13193755 0.16300233 0.16457670 0.17180013 0.16951910 0.22996498    0
## glmnet   0.14160780 0.14328657 0.15161158 0.16500024 0.16343731 0.22505793    0
## gam      0.15120069 0.18948866 0.20205942 0.20642269 0.24188616 0.24747854    0
## mars     0.14158138 0.20810159 0.20867236 0.20786109 0.22614439 0.25480571    0
## lda      0.13833743 0.13995551 0.15495372 0.16208589 0.15977844 0.21740434    0
## qda      0.20859227 0.21475849 0.24387746 0.23693725 0.25432987 0.26312818    0
## nb       0.01857562 0.03259626 0.04323094 0.04096494 0.04365231 0.06676954    0
```

```
## bagging  0.08030925 0.14962901 0.16765997 0.16157987 0.20209421 0.20820693    0
## rf       0.12519599 0.14989348 0.15652962 0.16181846 0.18678922 0.19068399    0
## boosting 0.13507028 0.13509117 0.18183779 0.18203883 0.19592434 0.26227055    0
## tree     0.14650407 0.18862646 0.19061127 0.18622778 0.20132325 0.20407387    0
## Model12  0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000    0
## svmr     0.10647202 0.18162915 0.18314068 0.17578391 0.18558504 0.22209264    0
```

```
p1=bwplot(resamp, metric = "Accuracy")
p2=bwplot(resamp, metric = "Kappa")
grid.arrange(p1, p2 ,ncol=2)
```




```
jpeg("./figure/resample2.jpeg", width = 8, height=6, units="in", res=500)
p1=bwplot(resamp, metric = "Accuracy")
p2=bwplot(resamp, metric = "Kappa")
grid.arrange(p1, p2, ncol=2)
dev.off()
```

```
## pdf
##   2
```

1.3 Training / Testing Error