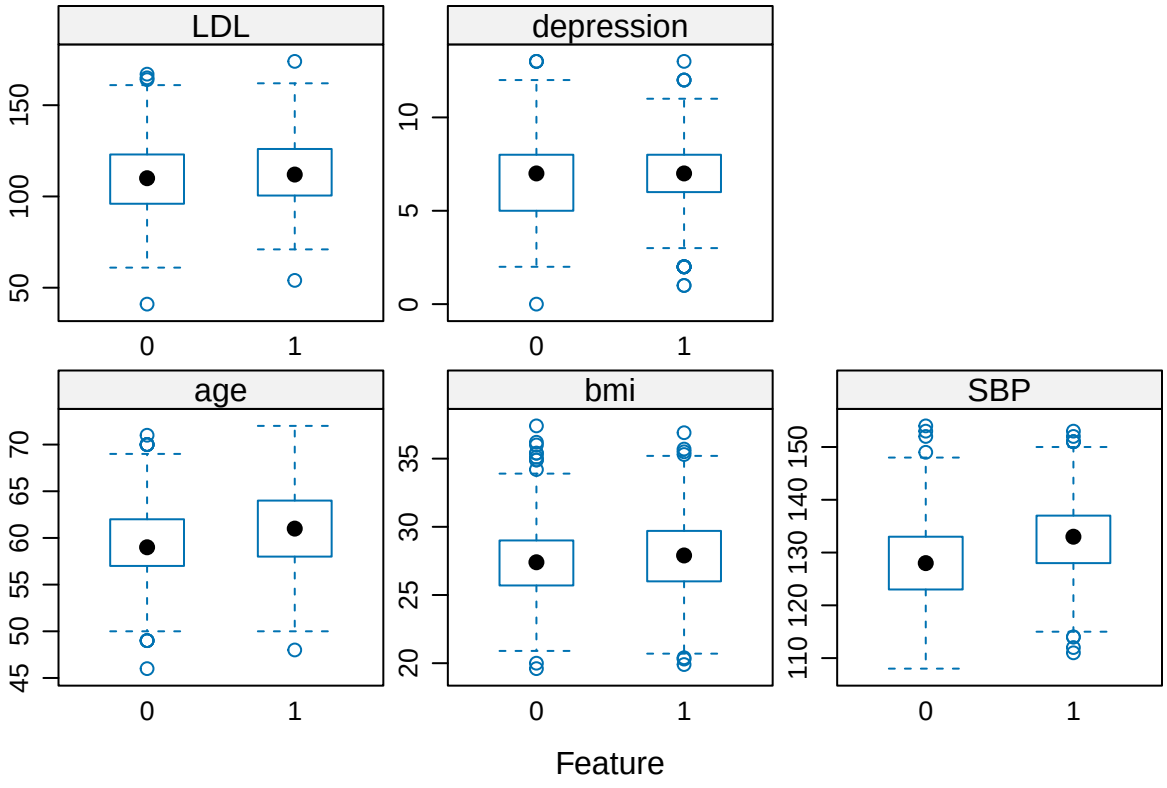
final_project

Yixiao Sun

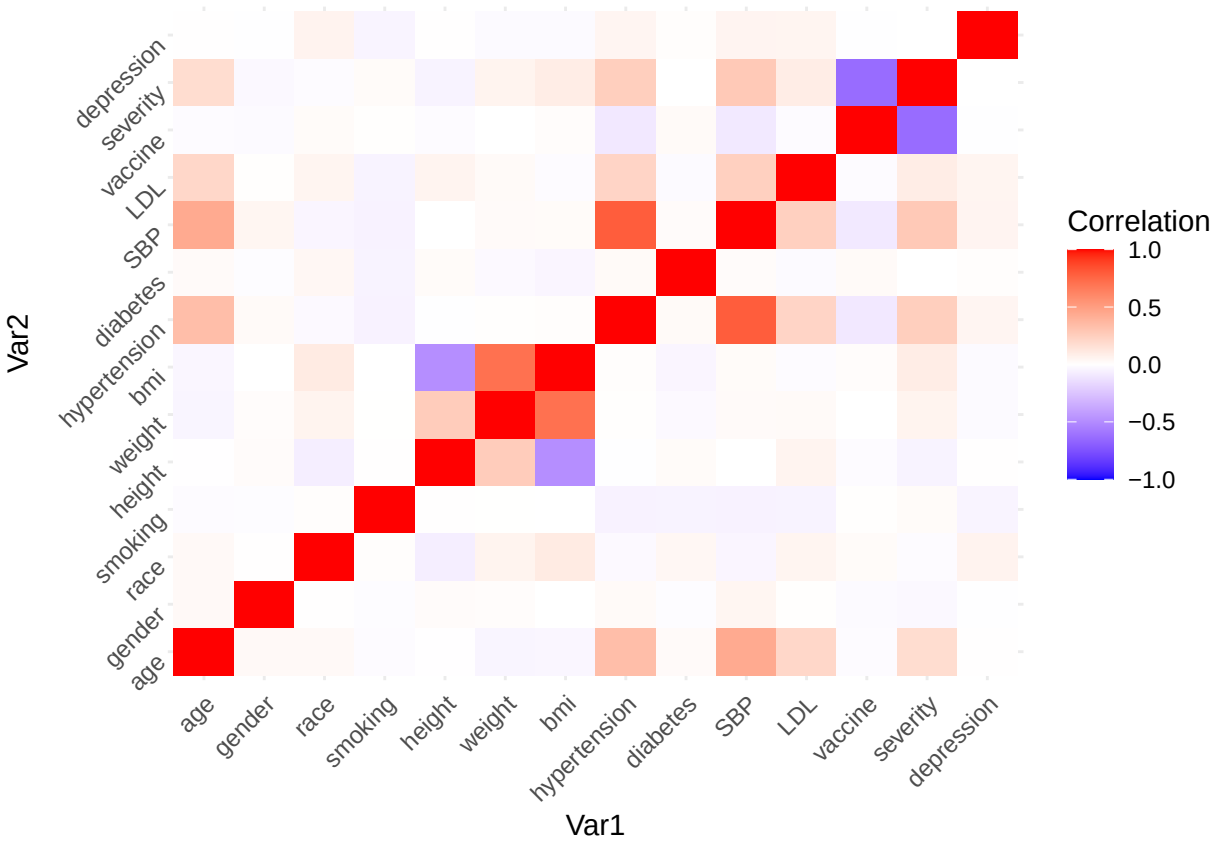
2024-05-04

```
library(tidyverse)
library(summarytools)
library(leaps)
library(corrplot)
library(dplyr)
library(ggplot2)
library(ISLR)
library(glmnet)
library(caret)
library(tidymodels)
library(plotmo)
library(earth)
library(pls)
library(rpart.plot)
library(gbm)
library(ranger)
```

```
load("severity_test.RData")
load("severity_training.RData")
test_data<-test_data[,-1]
training_data<-training_data[,-1]
data<-rbind(training_data,test_data)

data_vis = subset(data, select = -c(gender, race, smoking, hypertension, diabetes, vaccine,height,weigh

featurePlot(x = data_vis[, 1:5],
            y = data_vis$severity,
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free")),
            plot = "box")
```




```r
columns_to_convert <- c("gender", "race", "smoking", "hypertension", "diabetes", "vaccine", "severity")

data <- data %>% mutate(across(all_of(columns_to_convert), as.numeric))

numeric_data <- data[, c("age", "gender", "race", "smoking", "height", "weight", "bmi", "hypertension",

correlation_matrix <- cor(numeric_data)
correlation_df <- as.data.frame(as.table(correlation_matrix))
names(correlation_df) <- c("Var1", "Var2", "Correlation")

ggplot(correlation_df, aes(x = Var1, y = Var2, fill = Correlation)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       midpoint = 0, limits = c(-1, 1),
                       name = "Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
        axis.text.y = element_text(angle = 45, vjust = 1, hjust = 1))
```




```
train_data <- training_data %>%
  dplyr::select(-height, -weight, -hypertension) %>%
  dplyr::mutate(severity = ifelse(severity == 0, "Notsevere","Severe"))%>%
  dplyr::mutate(gender = as.factor(gender),
         race = as.factor(race),
         smoking = as.factor(smoking),
         diabetes = as.factor(diabetes),
         vaccine = as.factor(vaccine),
         severity = as.factor(severity))

test_data <- test_data %>%
  dplyr::select(-height, -weight, -hypertension)%>%
  dplyr::mutate(severity = ifelse(severity == 0, "Notsevere","Severe"))%>%
  dplyr::mutate(gender = as.factor(gender),
         race = as.factor(race),
         smoking = as.factor(smoking),
         diabetes = as.factor(diabetes),
         vaccine = as.factor(vaccine),
         severity = as.factor(severity))
```

```
ctrl1 <- trainControl(method = "cv", number = 10,summaryFunction = twoClassSummary,classProbs = TRUE)

set.seed(1)

enet.fit <- train(severity ~ .,
```

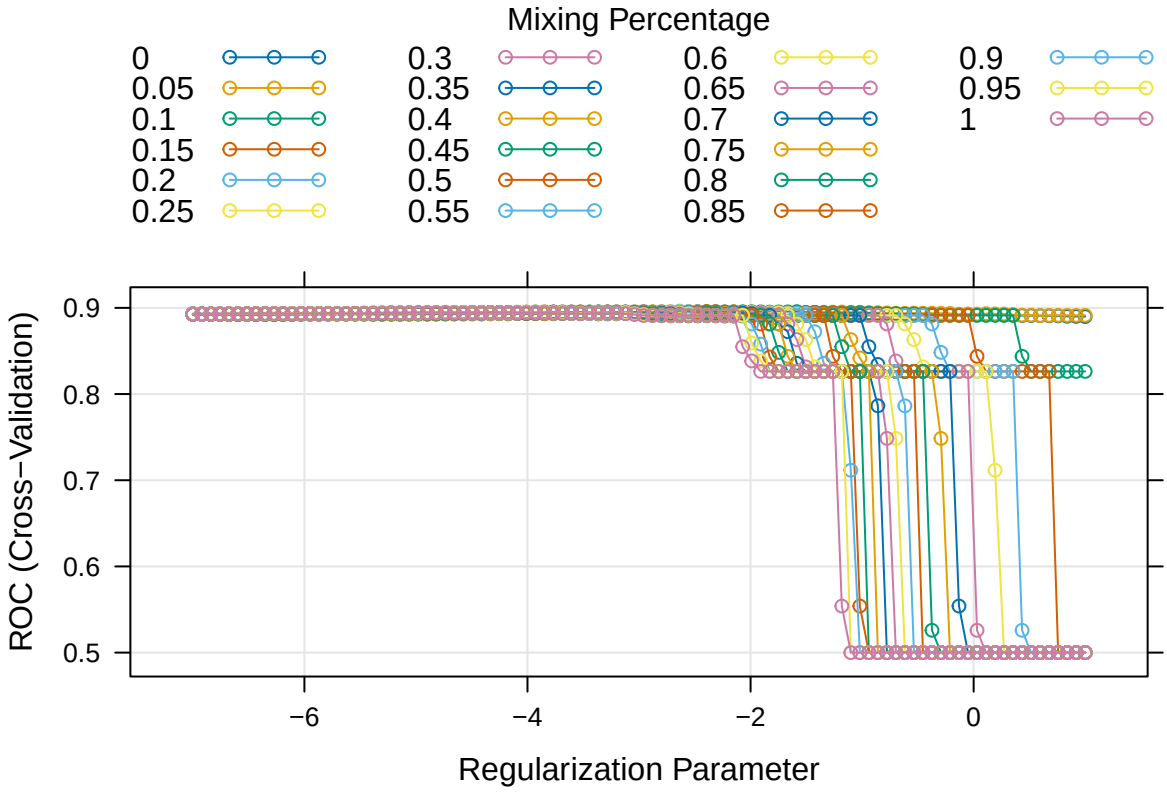
```
                  data = train_data,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = seq(0,1,length = 21),
                                         lambda = exp(seq(1, -7, length = 100))),
                  metric = "ROC",
                  trControl = ctrl1)
enet.fit$bestTune
```

```
##     alpha     lambda
## 455   0.2 0.07162124
```

```
print(coef(enet.fit$finalModel,enet.fit$bestTune$lambda))
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept) -9.815779556
## age          0.033203973
## gender1     -0.096764117
## race2        .
## race3        .
## race4        .
## smoking1     .
## smoking2     .
## bmi          0.059915336
## diabetes1    .
## SBP          0.048797462
## LDL          0.003387789
## vaccine1    -2.168752328
## depression   .
```

```
plot(enet.fit, xTrans = log)
```

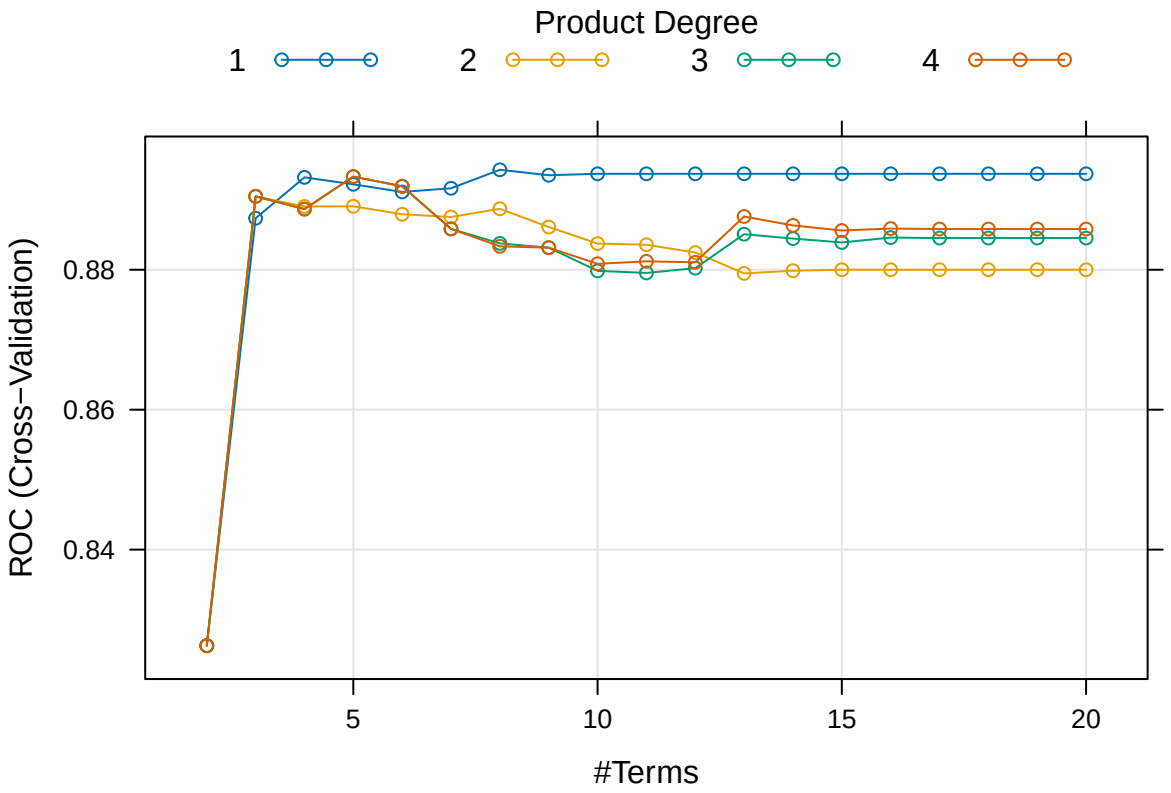



```
set.seed(1)
predict_prob <- predict(enet.fit, newdata = test_data)
confusionMatrix(data = predict_prob, reference = test_data$severity)
```

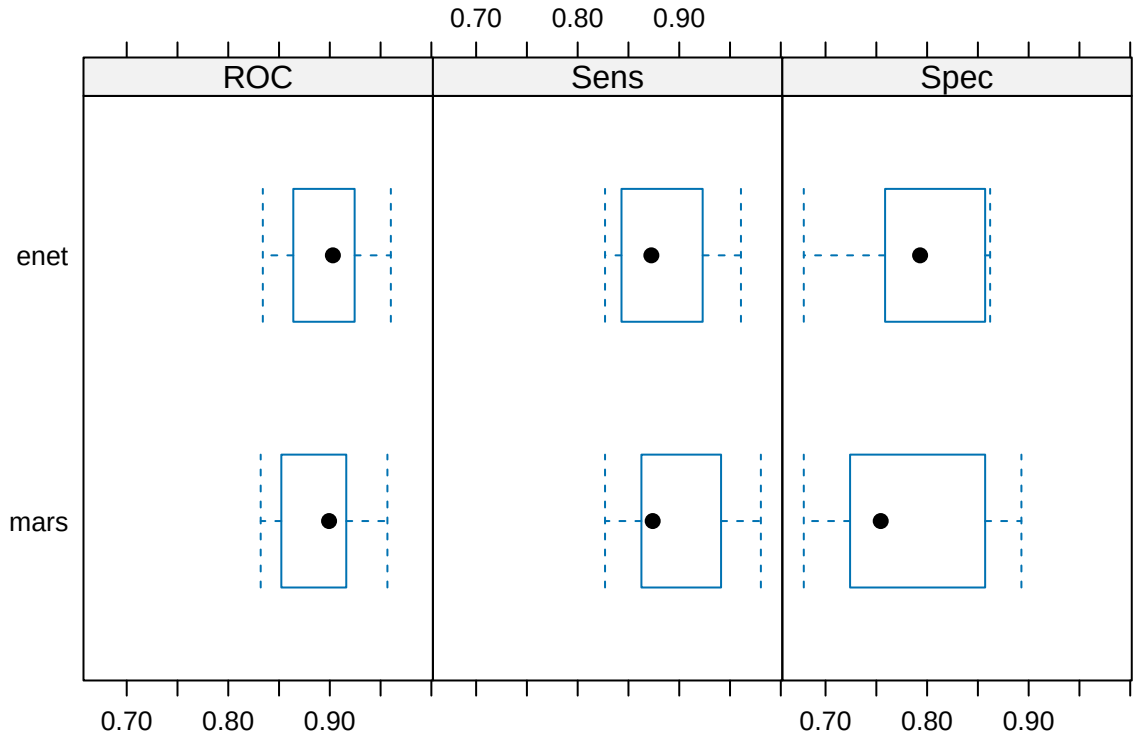
```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Notsevere Severe
##   Notsevere       120     14
##   Severe           15     51
##
##                Accuracy : 0.855
##                  95% CI : (0.7984, 0.9007)
##     No Information Rate : 0.675
##     P-Value [Acc > NIR] : 4.95e-09
##
##                   Kappa : 0.6708
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8889
##             Specificity : 0.7846
##          Pos Pred Value : 0.8955
##          Neg Pred Value : 0.7727
##              Prevalence : 0.6750
```

```
##          Detection Rate : 0.6000
##    Detection Prevalence : 0.6700
##       Balanced Accuracy : 0.8368
##
##        'Positive' Class : Notsevere
##
```

```r
set.seed(1)
mars.fit <- train(severity ~.,
                  data = train_data,
                  method = "earth",
                  tuneGrid = expand.grid(degree = 1:4,
                                         nprune = 2:20),
                  metric = "ROC",
                  trControl = ctrl1)
plot(mars.fit)
```




```r
bwplot(resamples(list(enet = enet.fit, mars = mars.fit)), matrix = "ROC")
```




```
predict_prob2 <- predict(mars.fit, newdata = test_data)
confusionMatrix(data = predict_prob2, reference = test_data$severity)
```

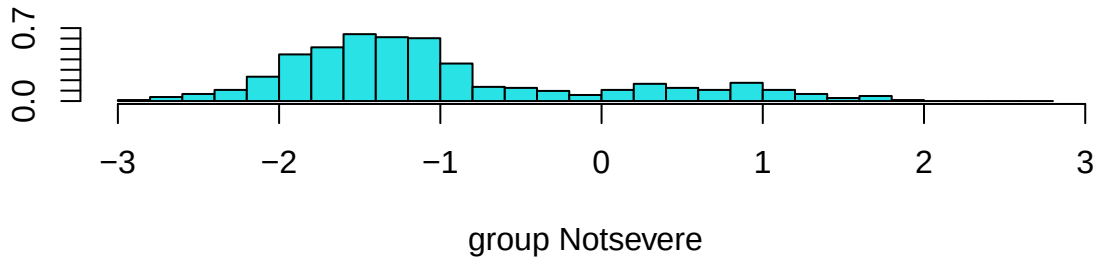
```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Notsevere Severe
##   Notsevere       121     15
##   Severe           14     50
##
##                Accuracy : 0.855
##                  95% CI : (0.7984, 0.9007)
##     No Information Rate : 0.675
##     P-Value [Acc > NIR] : 4.95e-09
##
##                   Kappa : 0.6682
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8963
##             Specificity : 0.7692
##          Pos Pred Value : 0.8897
##          Neg Pred Value : 0.7812
##              Prevalence : 0.6750
##          Detection Rate : 0.6050
```

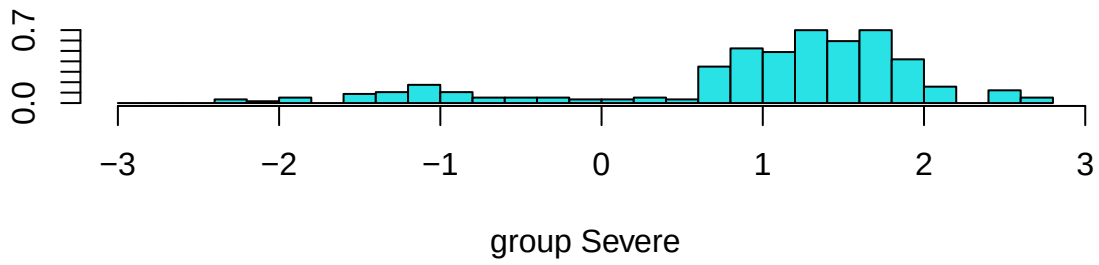
```
##    Detection Prevalence : 0.6800
##       Balanced Accuracy : 0.8328
##
##        'Positive' Class : Notsevere
##
```

```
set.seed(1)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
lda.fit <- train(severity ~ .,
                 data = train_data,
                 method = "lda",
                 metric = "ROC",
                 trControl = ctrl1)
lda<-lda(severity ~ ., data = train_data)
plot(lda)
```





```r
lda.model = lda(severity~., data = train_data)
lda.model$scaling
```

```
##                     LD1
## age          0.033776508
## gender1     -0.240800793
## race2       -0.142027982
## race3        0.014109166
## race4       -0.120544798
## smoking1     0.024587810
## smoking2     0.241597041
## bmi          0.084298067
## diabetes1    0.135449828
## SBP          0.046689082
## LDL          0.004678763
## vaccine1    -2.486523800
## depression -0.009677249
```

```r
head(predict(lda.model)$x)
```

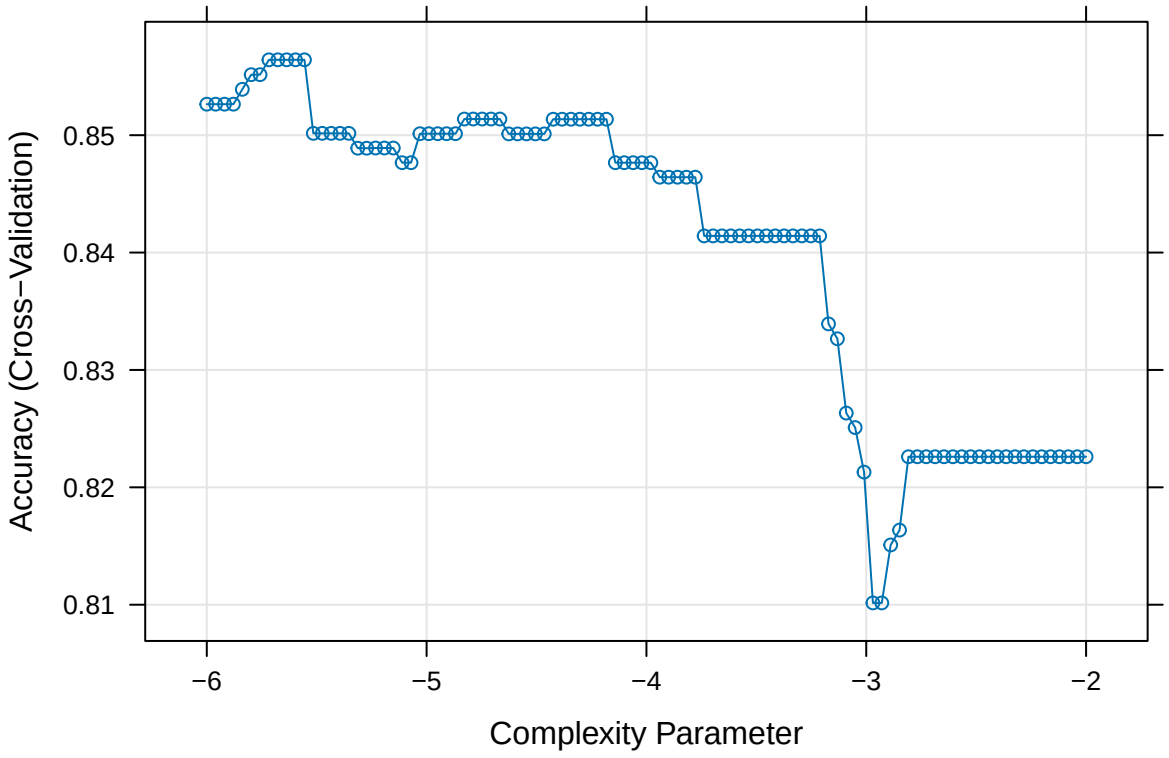
```
##          LD1
## 1 -1.6319419
## 2  1.1957678
## 3 -1.2937548
## 4 -1.4972707
## 6 -0.7666826
## 9  2.0220931
```

```r
predict_prob3 <- predict(lda.fit, newdata = test_data)
confusionMatrix(data = predict_prob3, reference = test_data$severity)
```

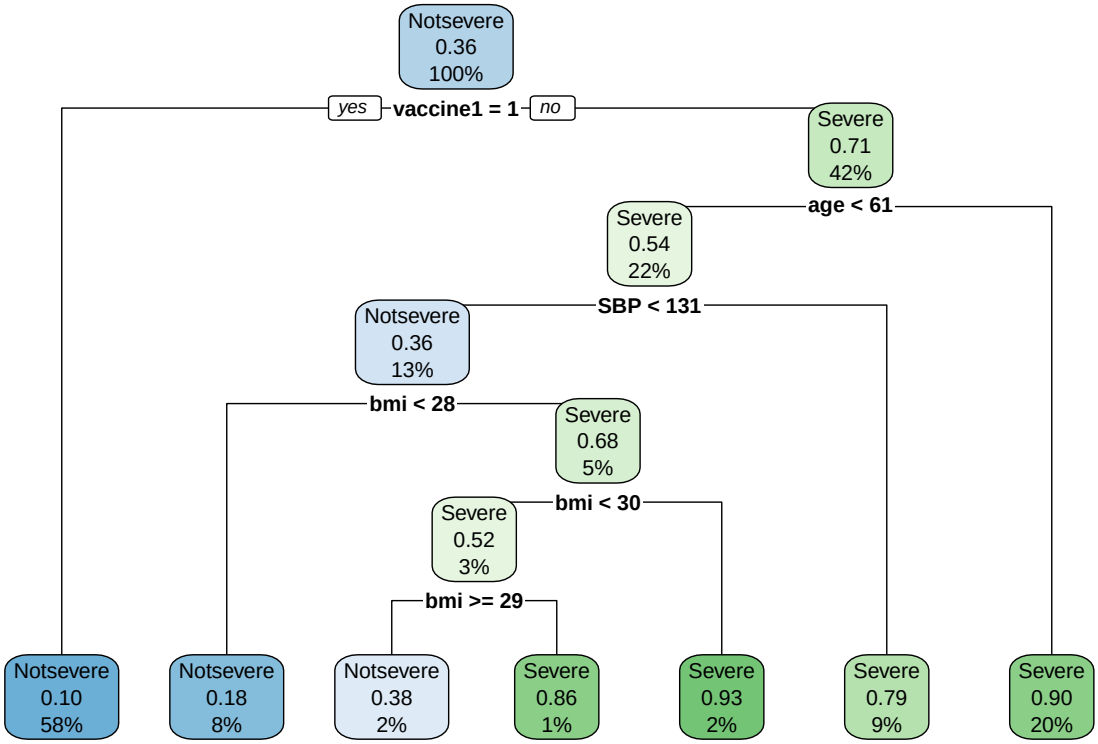
```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Notsevere Severe
##   Notsevere       115     12
##   Severe           20     53
##
##                Accuracy : 0.84
##                  95% CI : (0.7817, 0.8879)
##     No Information Rate : 0.675
##     P-Value [Acc > NIR] : 9.736e-08
##
##                   Kappa : 0.6466
##
##  Mcnemar's Test P-Value : 0.2159
##
##             Sensitivity : 0.8519
##             Specificity : 0.8154
##          Pos Pred Value : 0.9055
##          Neg Pred Value : 0.7260
```

```
##              Prevalence : 0.6750
##          Detection Rate : 0.5750
##    Detection Prevalence : 0.6350
##       Balanced Accuracy : 0.8336
##
##        'Positive' Class : Notsevere
##
```

```
ctrl <-trainControl(method = "cv")
set.seed(1)

rpart.fit <- train(severity ~ . ,
                   train_data,
                   method = "rpart",
                   tuneGrid = data.frame(cp = exp(seq(-6,-2, length = 100))),
                   trControl = ctrl)

plot(rpart.fit, xTrans = log)
```




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




```
predict_prob4 <- predict(rpart.fit, newdata = test_data)
confusionMatrix(data = predict_prob4, reference = test_data$severity)
```

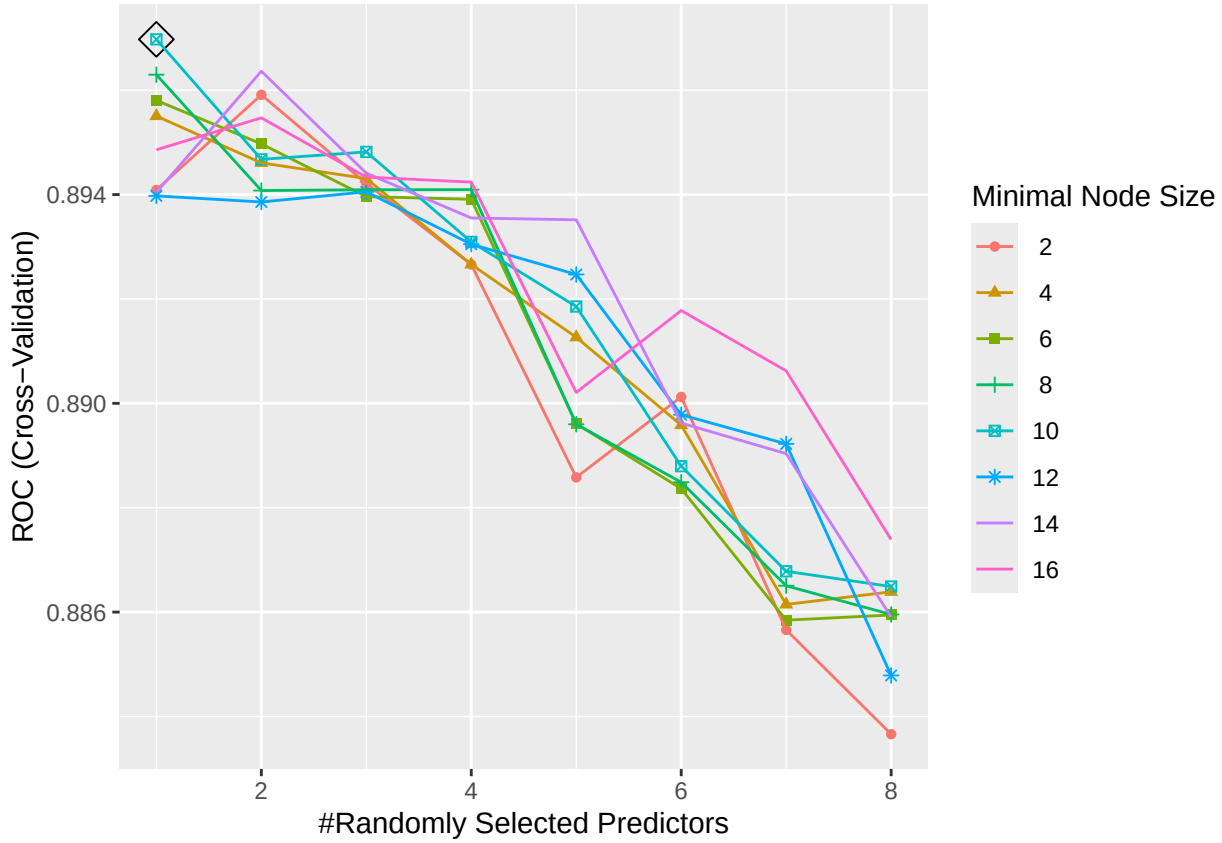
```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Notsevere Severe
##   Notsevere       125     20
##   Severe           10     45
##
##                Accuracy : 0.85
##                  95% CI : (0.7928, 0.8965)
##     No Information Rate : 0.675
##     P-Value [Acc > NIR] : 1.387e-08
##
##                   Kappa : 0.6439
##
##  Mcnemar's Test P-Value : 0.1003
##
##             Sensitivity : 0.9259
##             Specificity : 0.6923
##          Pos Pred Value : 0.8621
##          Neg Pred Value : 0.8182
##              Prevalence : 0.6750
##          Detection Rate : 0.6250
```

```
##    Detection Prevalence : 0.7250
##       Balanced Accuracy : 0.8091
##
##        'Positive' Class : Notsevere
##
```

```
set.seed(1)
ctrl2 <- trainControl(method = "cv",
                      classProbs = TRUE,
                      summaryFunction = twoClassSummary)

rf.grid <- expand.grid(mtry = 1:8,
                       splitrule = "gini",
                       min.node.size = seq(from = 2, to = 16, by = 2))

rf.fit <- train(severity ~ . ,
                train_data,
                method = "ranger",
                tuneGrid = rf.grid,
                metric = "ROC",
                trControl = ctrl2)
ggplot(rf.fit, highlight = TRUE)
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values because more
## than 6 becomes difficult to discriminate
## i you have requested 8 values. Consider specifying shapes manually if you need
##   that many have them.
```

```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

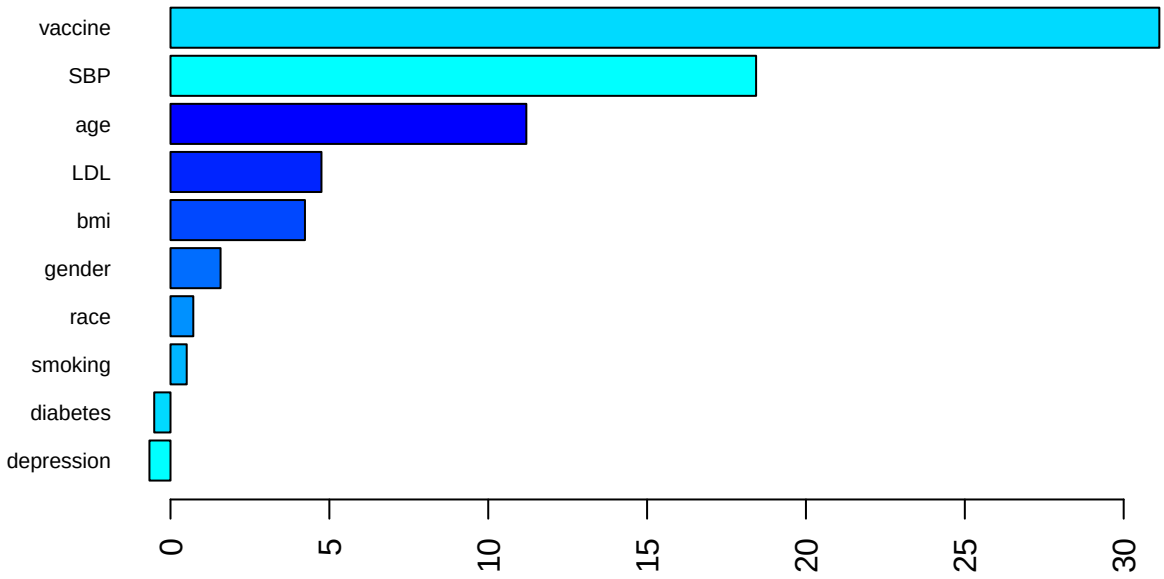



```
predict_prob5 <- predict(rf.fit, newdata = test_data)
confusionMatrix(data = predict_prob5, reference = test_data$severity)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Notsevere Severe
##   Notsevere       131     34
##   Severe            4     31
##
##                Accuracy : 0.81
##                  95% CI : (0.7487, 0.8619)
##     No Information Rate : 0.675
##     P-Value [Acc > NIR] : 1.459e-05
##
##                   Kappa : 0.5081
##
##  Mcnemar's Test P-Value : 2.546e-06
##
##             Sensitivity : 0.9704
##             Specificity : 0.4769
##          Pos Pred Value : 0.7939
##          Neg Pred Value : 0.8857
##              Prevalence : 0.6750
##          Detection Rate : 0.6550
```

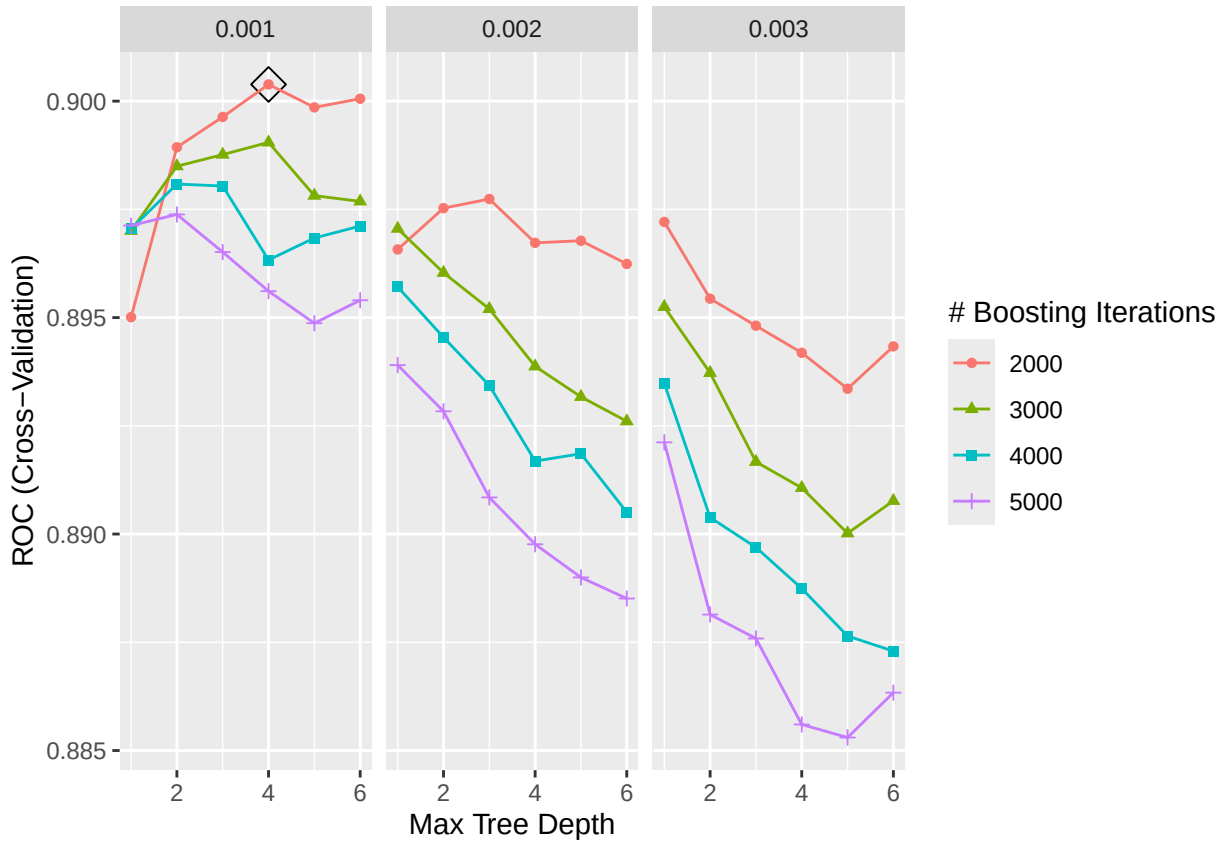
```
##    Detection Prevalence : 0.8250
##       Balanced Accuracy : 0.7236
##
##        'Positive' Class : Notsevere
##
```

```r
rf2.final.per <- ranger(severity ~ . ,
                        train_data,
                        mtry = rf.fit$bestTune[[1]],
                        min.node.size = rf.fit$bestTune[[3]],
                        splitrule = "gini",
                        importance = "permutation",
                        scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(rf2.final.per), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(8))
```

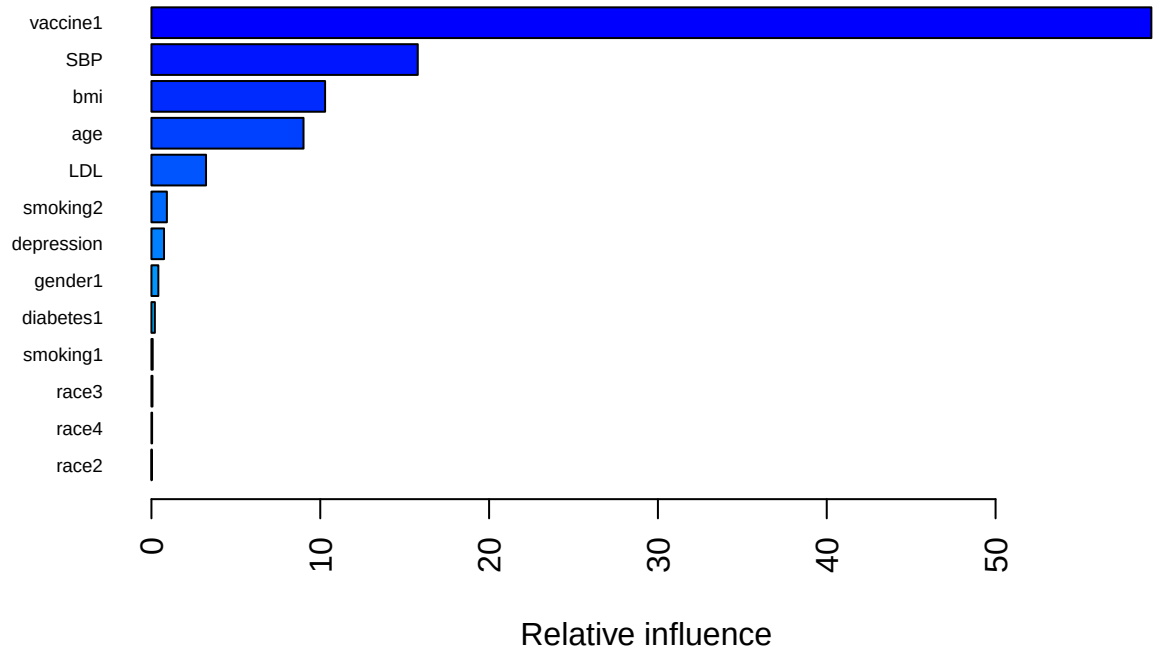



```r
set.seed(1)
gbmA.grid <- expand.grid(n.trees = c(2000,3000,4000,5000),
                         interaction.depth = 1:6,
                         shrinkage = c(0.001, 0.002, 0.003),
                         n.minobsinnode = 1)

gbmA.fit <- train(severity ~ . ,
```

```
                  train_data,
                  tuneGrid = gbmA.grid,
                  trControl = ctrl2,
                  method = "gbm",
                  distribution = "adaboost",
                  metric = "ROC",
                  verbose = FALSE)

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




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




```
##                   var     rel.inf
## vaccine1     vaccine1 59.22797659
## SBP               SBP 15.77557131
## bmi               bmi 10.28628377
## age               age  9.00819425
## LDL               LDL  3.23279545
## smoking2     smoking2  0.91443038
## depression depression  0.74527515
## gender1       gender1  0.40966689
## diabetes1   diabetes1  0.20260166
## smoking1     smoking1  0.07431990
## race3           race3  0.06156638
## race4           race4  0.04001045
## race2           race2  0.02130782
```

```
predict_prob6 <- predict(gbmA.fit, newdata = test_data)
confusionMatrix(data = predict_prob6, reference = test_data$severity)
```

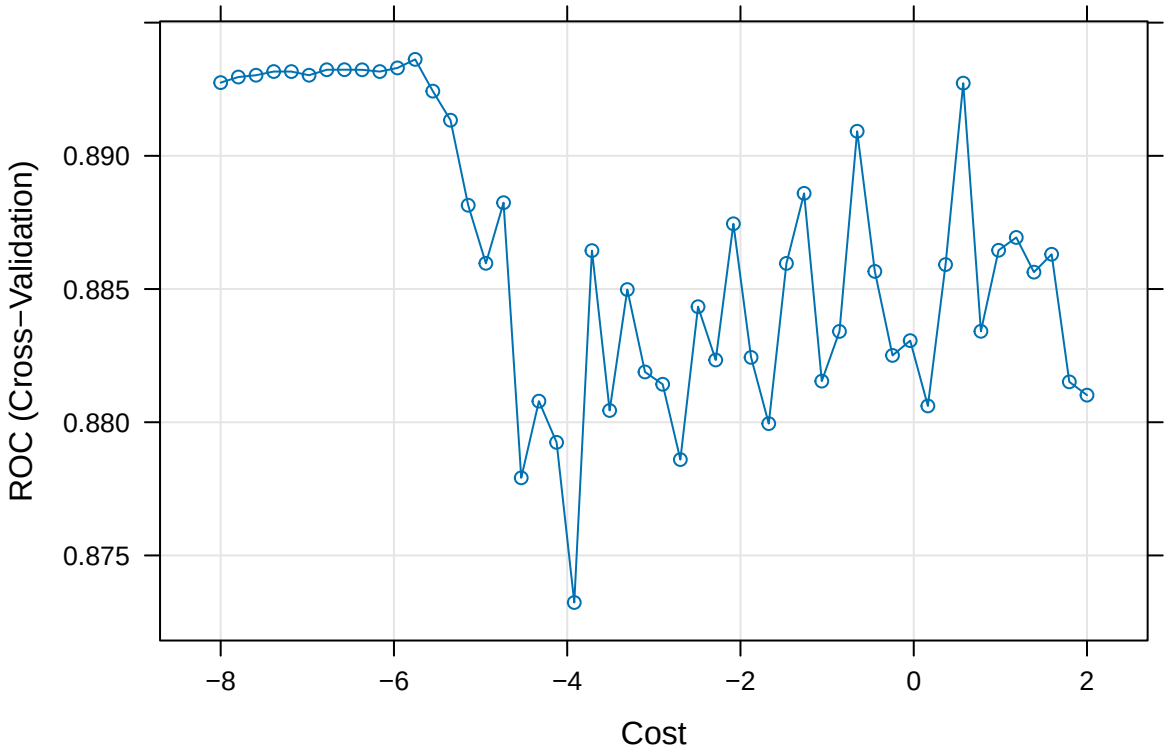
```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Notsevere Severe
##   Notsevere       124     18
##   Severe           11     47
##
```

```
##                Accuracy : 0.855
##                  95% CI : (0.7984, 0.9007)
##     No Information Rate : 0.675
##     P-Value [Acc > NIR] : 4.95e-09
##
##                   Kappa : 0.66
##
##  Mcnemar's Test P-Value : 0.2652
##
##             Sensitivity : 0.9185
##             Specificity : 0.7231
##          Pos Pred Value : 0.8732
##          Neg Pred Value : 0.8103
##              Prevalence : 0.6750
##          Detection Rate : 0.6200
##    Detection Prevalence : 0.7100
##       Balanced Accuracy : 0.8208
##
##        'Positive' Class : Notsevere
##
```

```r
set.seed(1)
svml.fit <- train(severity ~.,
                  data = train_data,
                  method = "svmLinear",
                  tuneGrid = data.frame(C = exp(seq(-8, 2, len = 50))), trControl = ctrl2)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

```r
plot(svml.fit, highlight = TRUE, xTrans = log)
```




```
svml.fit$bestTune
```

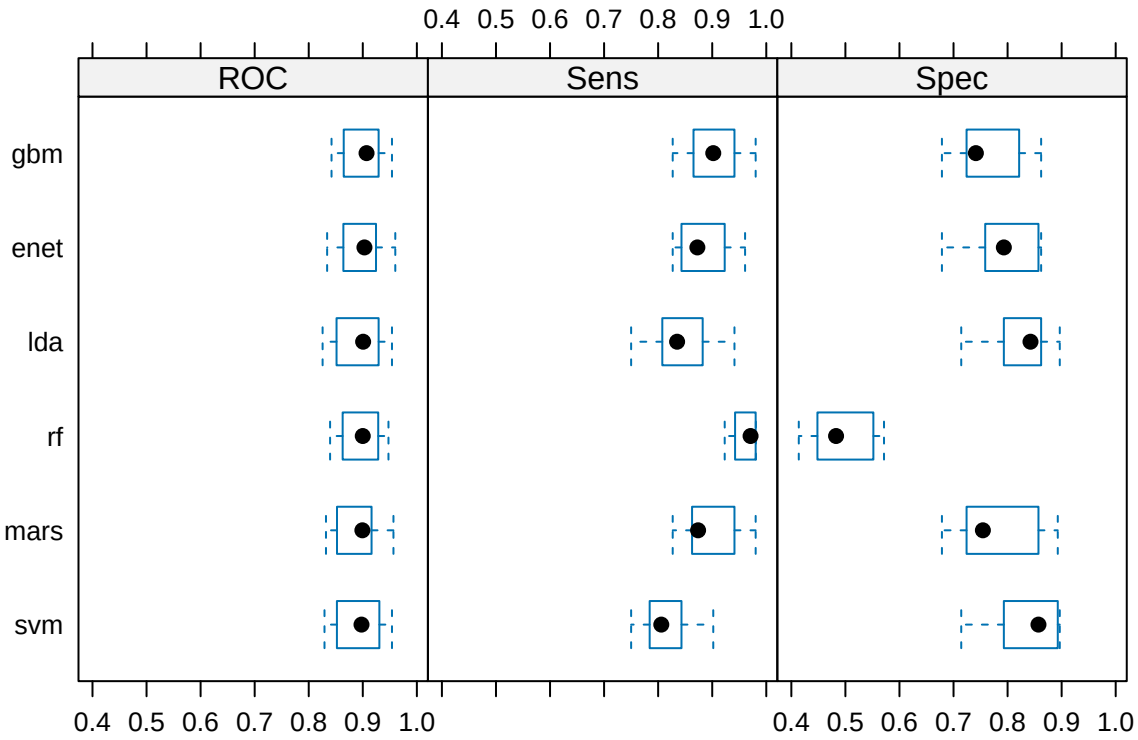
```
##              C
## 12 0.003166583
```

```
predict_prob7 <- predict(svml.fit, newdata = test_data)
confusionMatrix(data = predict_prob7, reference = test_data$severity)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Notsevere Severe
##   Notsevere       115     12
##   Severe           20     53
##
##                Accuracy : 0.84
##                  95% CI : (0.7817, 0.8879)
##     No Information Rate : 0.675
##     P-Value [Acc > NIR] : 9.736e-08
##
##                   Kappa : 0.6466
##
##  Mcnemar's Test P-Value : 0.2159
##
##             Sensitivity : 0.8519
```

```
##             Specificity : 0.8154
##          Pos Pred Value : 0.9055
##          Neg Pred Value : 0.7260
##              Prevalence : 0.6750
##          Detection Rate : 0.5750
##    Detection Prevalence : 0.6350
##       Balanced Accuracy : 0.8336
##
##        'Positive' Class : Notsevere
##
```

```
bwplot(resamples(list(enet = enet.fit, mars = mars.fit, lda = lda.fit, rf = rf.fit,gbm = gbmA.fit, svm =
```




```
resamp<-resamples(list(enet = enet.fit, mars = mars.fit, lda = lda.fit, rf = rf.fit, gbm = gbmA.fit, svm
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: enet, mars, lda, rf, gbm, svm
## Number of resamples: 10
##
## ROC
##           Min.    1st Qu.     Median       Mean    3rd Qu.       Max. NA's
```

```
## enet 0.8340336 0.8650823 0.9030270 0.8961996 0.9239051 0.9601082    0
## mars 0.8319328 0.8615131 0.8992628 0.8942837 0.9156230 0.9567275    0
## lda  0.8256303 0.8541114 0.9006345 0.8930392 0.9289216 0.9540230    0
## rf   0.8395225 0.8637365 0.8999519 0.8969747 0.9275210 0.9474790    0
## gbm  0.8421751 0.8679461 0.9069278 0.9003858 0.9265211 0.9540230    0
## svm  0.8291317 0.8546088 0.8975984 0.8936210 0.9291168 0.9540230    0
##
## Sens
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## enet 0.8269231 0.8438914 0.8725490 0.8815234 0.9177979 0.9607843    0
## mars 0.8269231 0.8627451 0.8738688 0.8892534 0.9264706 0.9803922    0
## lda  0.7500000 0.8125000 0.8350302 0.8446833 0.8774510 0.9411765    0
## rf   0.9230769 0.9469268 0.9709653 0.9631222 0.9803922 0.9807692    0
## gbm  0.8269231 0.8745287 0.9019608 0.9068627 0.9411765 0.9803922    0
## svm  0.7500000 0.7853507 0.8058069 0.8134992 0.8382353 0.9019608    0
##
## Spec
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## enet 0.6785714 0.7653941 0.7931034 0.8006158 0.8571429 0.8620690    0
## mars 0.6785714 0.7241379 0.7543103 0.7764778 0.8497537 0.8928571    0
## lda  0.7142857 0.8017241 0.8423645 0.8317734 0.8608374 0.8965517    0
## rf   0.4137931 0.4522783 0.4827586 0.4900246 0.5387931 0.5714286    0
## gbm  0.6785714 0.7241379 0.7413793 0.7551724 0.8057266 0.8620690    0
## svm  0.7142857 0.8017241 0.8571429 0.8390394 0.8851601 0.8965517    0
```

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