P8106 Data Science II Final Project R Code

Huanyu Chen

```r
library(dplyr)
library(ggplot2)
library(caret)
library(rpart.plot)
library(pROC)
library(randomForest)
library(glmnet)
library(MASS)
library(gbm)
library(pdp)
library(gridExtra)
```

```r
load("severity_test.RData")
load("severity_training.RData")

test_data <- test_data %>%
  dplyr::select(-id) %>%
  mutate(
    gender = case_when(gender == 0 ~ "Female",
                     gender == 1 ~"Male"),
    race = case_when(race == 1 ~ "White",
                     race == 2 ~ "Asian",
                     race == 3 ~ "Black",
                     race == 4 ~ "Hispanic"),
  ) %>%
  mutate(
    gender = as.factor(gender),
    diabetes = as.factor(diabetes),
    hypertension = as.factor(hypertension),
    vaccine = as.factor(vaccine),
    severity = factor(severity, levels = c(1, 0), labels = c("Severe", "Not Severe"))
  )

training_data <- training_data %>%
  dplyr::select(-id) %>%
  mutate(
    gender = case_when(gender == 0 ~ "Female",
                     gender == 1 ~"Male"),
    race = case_when(race == 1 ~ "White",
                     race == 2 ~ "Asian",
                     race == 3 ~ "Black",
                     race == 4 ~ "Hispanic"),
  ) %>%
  mutate(
    gender = as.factor(gender),
    diabetes = as.factor(diabetes),
    hypertension = as.factor(hypertension),
    vaccine = as.factor(vaccine),
    severity = factor(severity, levels = c(1, 0), labels = c("Severe", "Not Severe"))
  )
```

1 Exploratory Analysis and Data Visualization

1.1 Data Summary

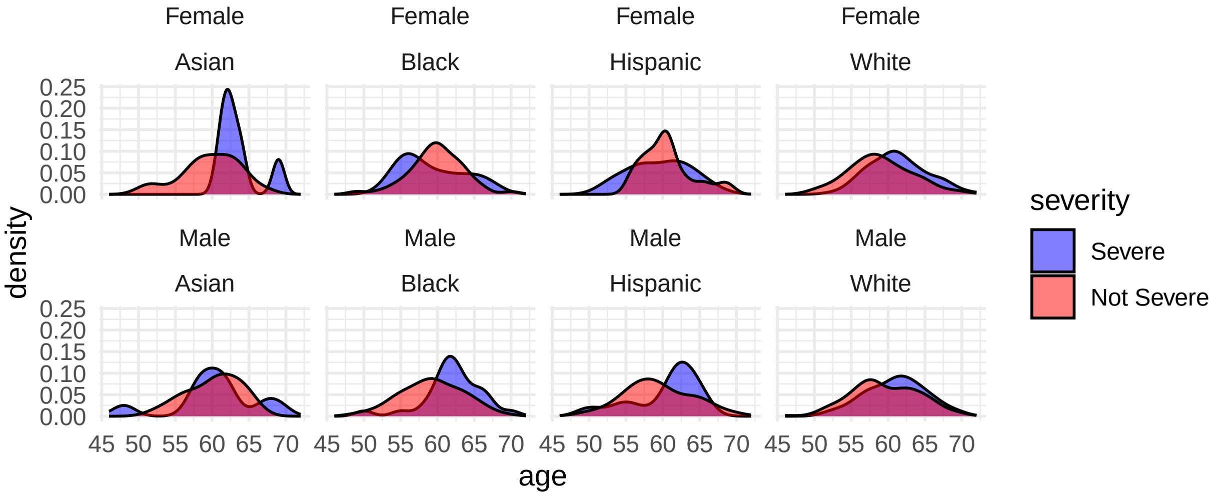
There are 13 potential predictors in this study: 7 of them are numeric variables (including `age`, `height`, `weight`, `bmi`, `SBP`, `LDL`, and `depression`), and the remaining 6 are categorical variables (including `gender`, `race`, `smoking`, `diabetes`, `hypertension` and `vaccine`). The response variable `severity` has two values: 1 stands for severe status (286 observations in this study) and 0 stands for non-severe status (514 observations).

1.2 Multivariate Density Plot of Age by Severity, Gender, and Race

From **Figure 1**, it is evident that severity of COVID-19 tends to be higher among older individuals overall. However, specific trends vary across different demographic groups. Notably, the severity appears less pronounced among Female Black, Female Hispanic, and Male Asian populations. This suggests that factors beyond age, such as gender and race, may play a role in determining the severity of COVID-19 symptoms.

```
ggplot(training_data, aes(x = age, fill = severity)) +
    geom_density(alpha = 0.5) +
    scale_fill_manual(values = c("blue", "red")) +
    labs(title = "Figure 1: Multivariate Density Plot of Age by Severity, Gender, and Race") +
    facet_wrap(~ gender + race, ncol = 4) +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme_minimal()
```

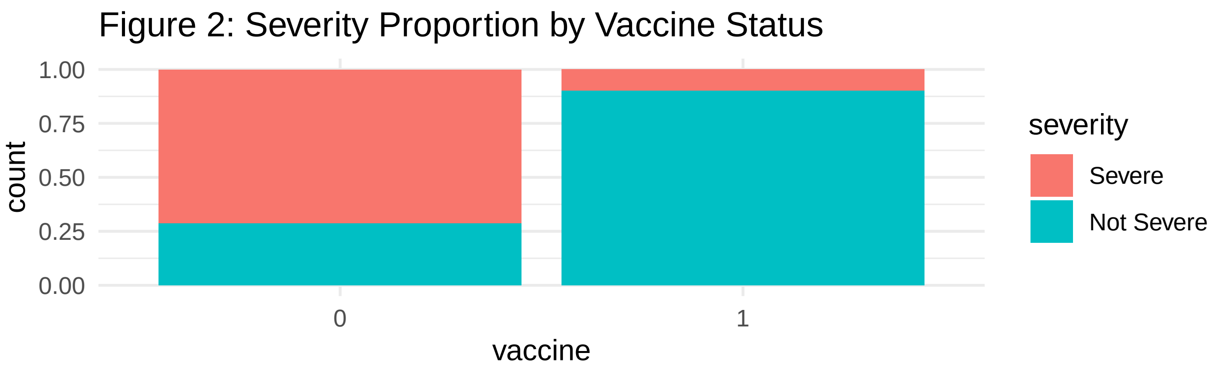
Figure 1: Multivariate Density Plot of Age by Severity, Gender, and Race



1.3 Severity Proportion by Vaccine Status

From **Figure 2**, it is apparent that the severity of COVID-19 is lower among individuals who have received the vaccine. From our data, it can be reasonably concluded that vaccination is targeted at mitigating the symptoms of COVID-19.

```r
ggplot(training_data, aes(x = vaccine, fill = severity)) +
  geom_bar(position = "fill") +
  labs(title = "Figure 2: Severity Proportion by Vaccine Status") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_minimal()
```



2 Model Training

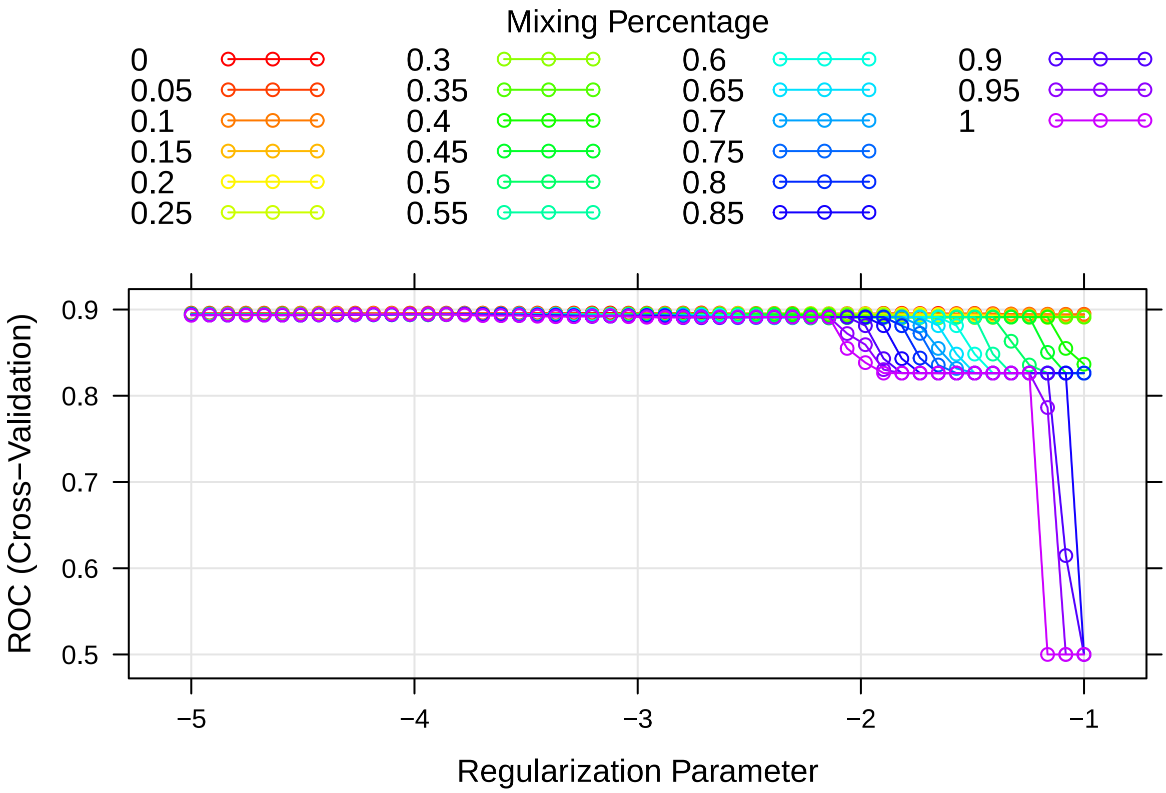
2.1 Penalized Logistic Regression

```r
training_data$severity <- make.names(training_data$severity)
ctrl <- trainControl(method = "cv", number = 10,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)

glmnGrid <- expand.grid(.alpha = seq(0, 1, length = 21),
                        .lambda = exp(seq(-5, -1, length = 50)))

set.seed(1)
model.glmn <- train(x = training_data[1:13],
                    y = training_data$severity,
                    method = "glmnet",
                    tuneGrid = glmnGrid,
                    metric = "ROC",
                    trControl = ctrl)
```

```r
model.glmn$bestTune
```

```
##    alpha     lambda
## 29     0 0.06625226
```

```r
myCol <- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
              superpose.line = list(col = myCol))
plot(model.glmn, par.settings = myPar, xTrans = function(x) log(x))
```



```
coef(model.glmn$finalModel, model.glmn$bestTune$lambda)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)  -7.622906745
## age           0.041829349
## gender        .
## race          .
## smoking       0.093362349
## height       -0.014775893
## weight        0.011547469
## bmi           0.056161552
## diabetes      0.094897688
## hypertension  0.431063809
## SBP           0.037607718
## LDL           0.005585451
## vaccine      -2.245823360
## depression   -0.014046490
```

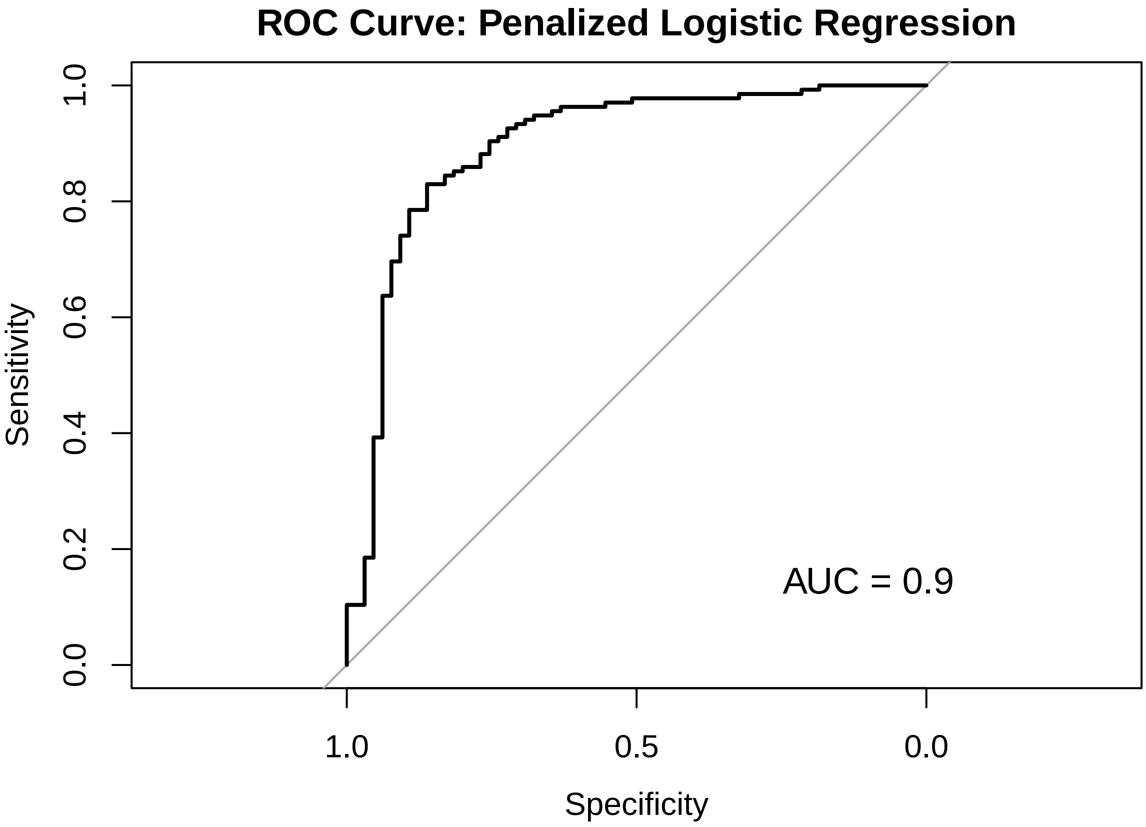
```
predictions.glmn <- predict(model.glmn, newdata = test_data, type = "prob")
```

```
## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion
```

```
predicted_probabilities.glmn <- predictions.glmn[, "Severe"]
roc_curve.glmn <- roc(test_data$severity, predicted_probabilities.glmn)
```

```
## Setting levels: control = Severe, case = Not Severe
## Setting direction: controls > cases
```

```
plot(roc_curve.glmn, main = "ROC Curve: Penalized Logistic Regression")
text(0.1, 0.1, paste("AUC =", round(auc(roc_curve.glmn), 2)), adj = c(0.5, -0.5), cex = 1.2)
```

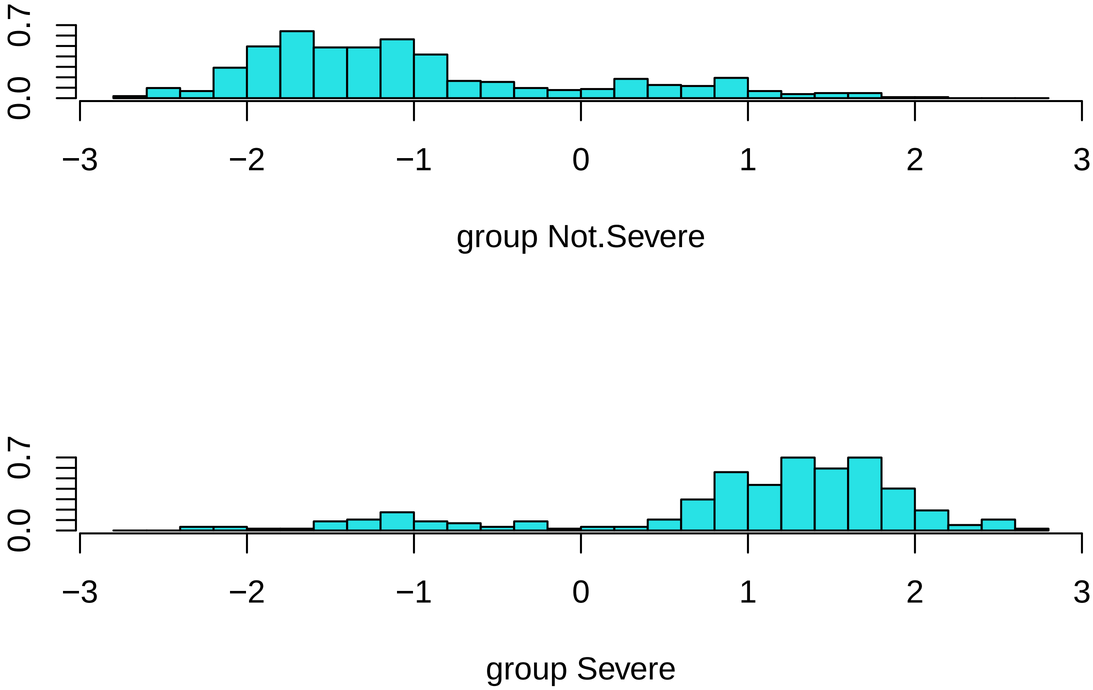


```
auc(roc_curve.glmn)
```

```
## Area under the curve: 0.8953
```

2.2 Linear Discriminant Analysis

```
lda.fit <- lda(severity~., data = training_data)
plot(lda.fit)
```

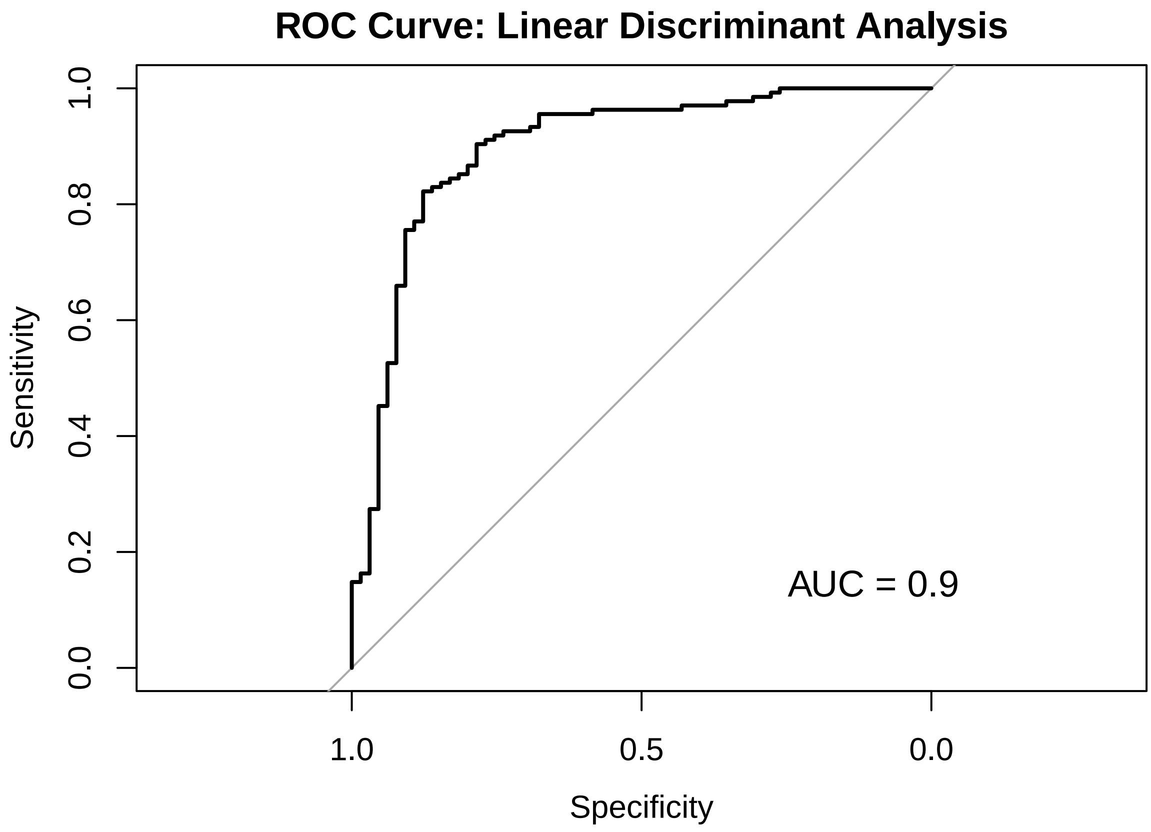


```r
set.seed(1)
lda_fit = train(x = model.matrix(severity ~ ., data = training_data)[, -1],
                y = training_data$severity, method = "lda",
                metric = "ROC",
                trControl = ctrl)
```

```r
lda.pred <- predict(lda.fit, newdata = test_data)
lda.probs <- lda.pred$posterior[, "Severe"]
roc_curve.lda <- roc(test_data$severity, lda.probs)
```

```
## Setting levels: control = Severe, case = Not Severe

## Setting direction: controls > cases
```

```r
plot(roc_curve.lda, main = "ROC Curve: Linear Discriminant Analysis")
text(0.1, 0.1, paste("AUC =", round(auc(roc_curve.lda), 2)), adj = c(0.5, -0.5), cex = 1.2)
```



```
auc(roc_curve.lda)
```

```
## Area under the curve: 0.8978
```

2.3 Quadratic Discriminant Analysis

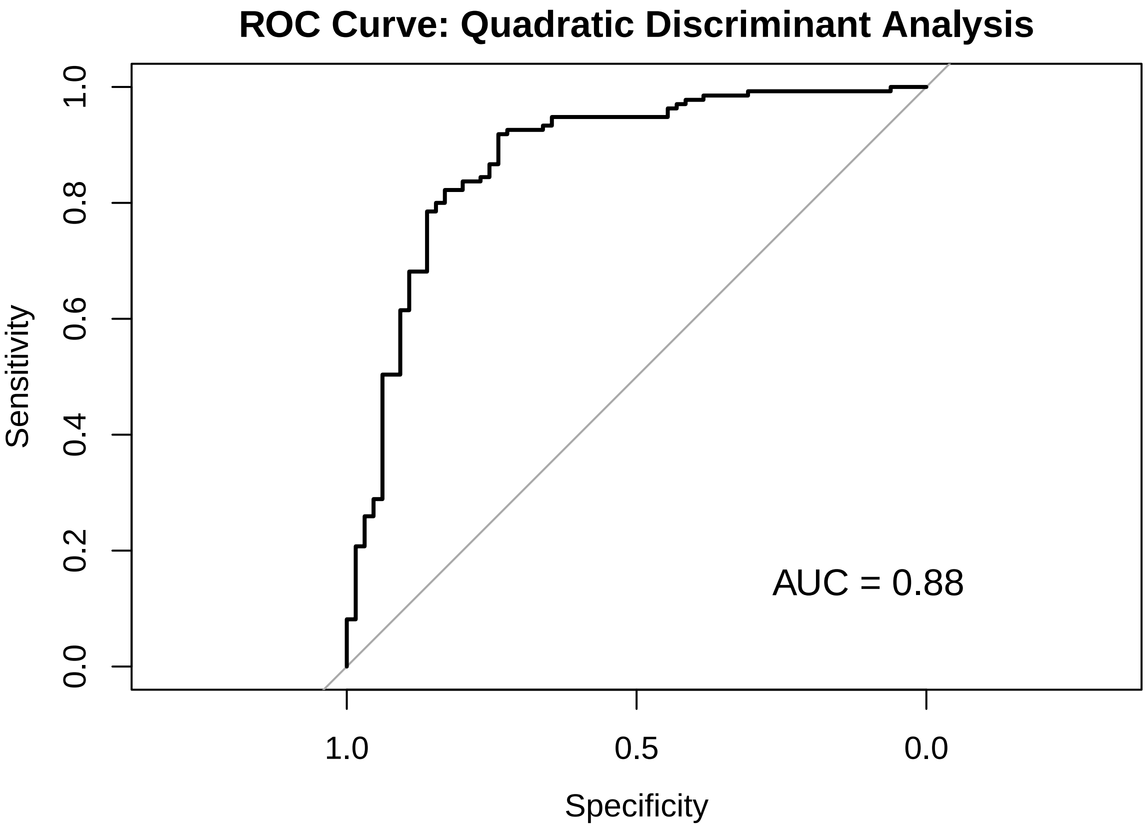
```
qda.fit <- qda(severity~., data = training_data)

set.seed(1)
qda_fit = train(x = model.matrix(severity ~ ., data = training_data)[, -1],
                y = training_data$severity, method = "qda",
                metric = "ROC",
                trControl = ctrl)
```

```
qda.pred <- predict(qda.fit, newdata = test_data)
qda.probs <- qda.pred$posterior[, "Severe"]
roc_curve.qda <- roc(test_data$severity, qda.probs)
```

```
## Setting levels: control = Severe, case = Not Severe
```

```
## Setting direction: controls > cases
```

```r
plot(roc_curve.qda, main = "ROC Curve: Quadratic Discriminant Analysis")
text(0.1, 0.1, paste("AUC =", round(auc(roc_curve.qda), 2)), adj = c(0.5, -0.5), cex = 1.2)
```

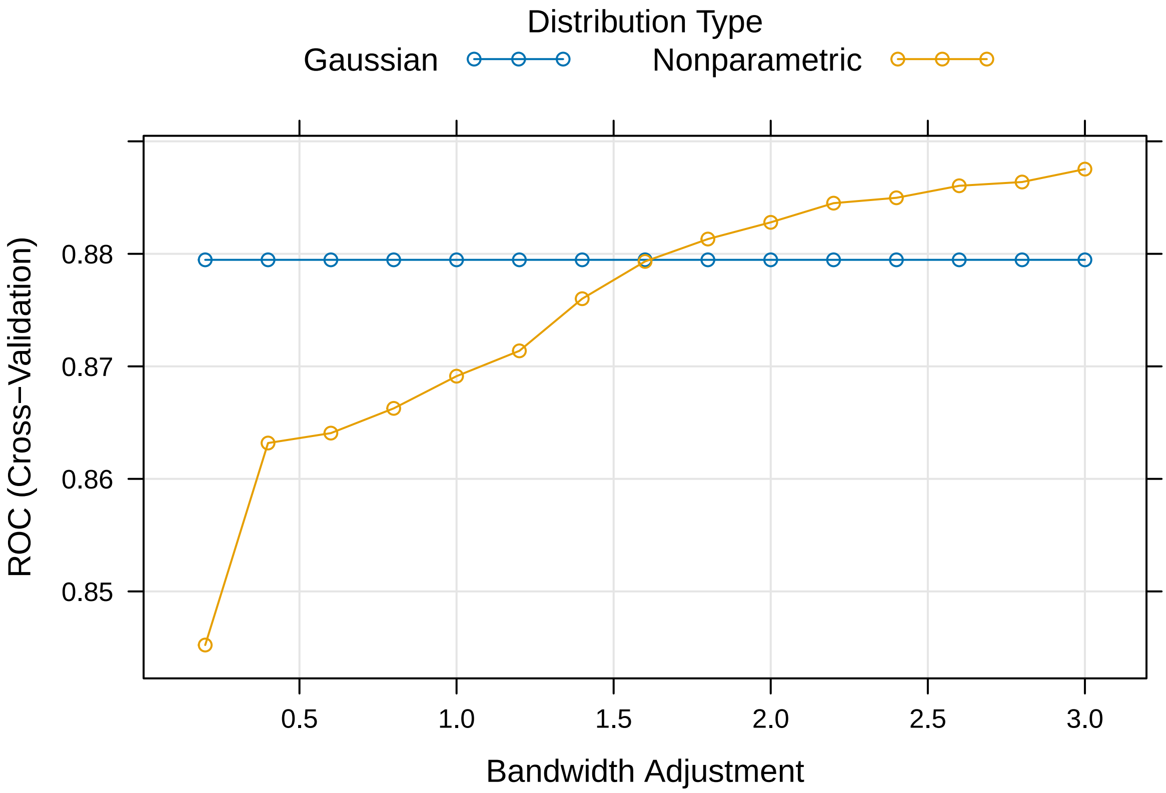


```r
auc(roc_curve.qda)
```

```
## Area under the curve: 0.8772
```

2.4 Naive Bayes

```r
nbGrid <- expand.grid(usekernel = c(FALSE, TRUE),
                      fL = 1,
                      adjust = seq(.2, 3, by = .2))

set.seed(1)
model.nb <- train(x = training_data[, 1:13],
                  y = training_data$severity,
                  method = "nb",
                  tuneGrid = nbGrid,
                  metric = "ROC",
                  trControl = ctrl)
plot(model.nb)
```

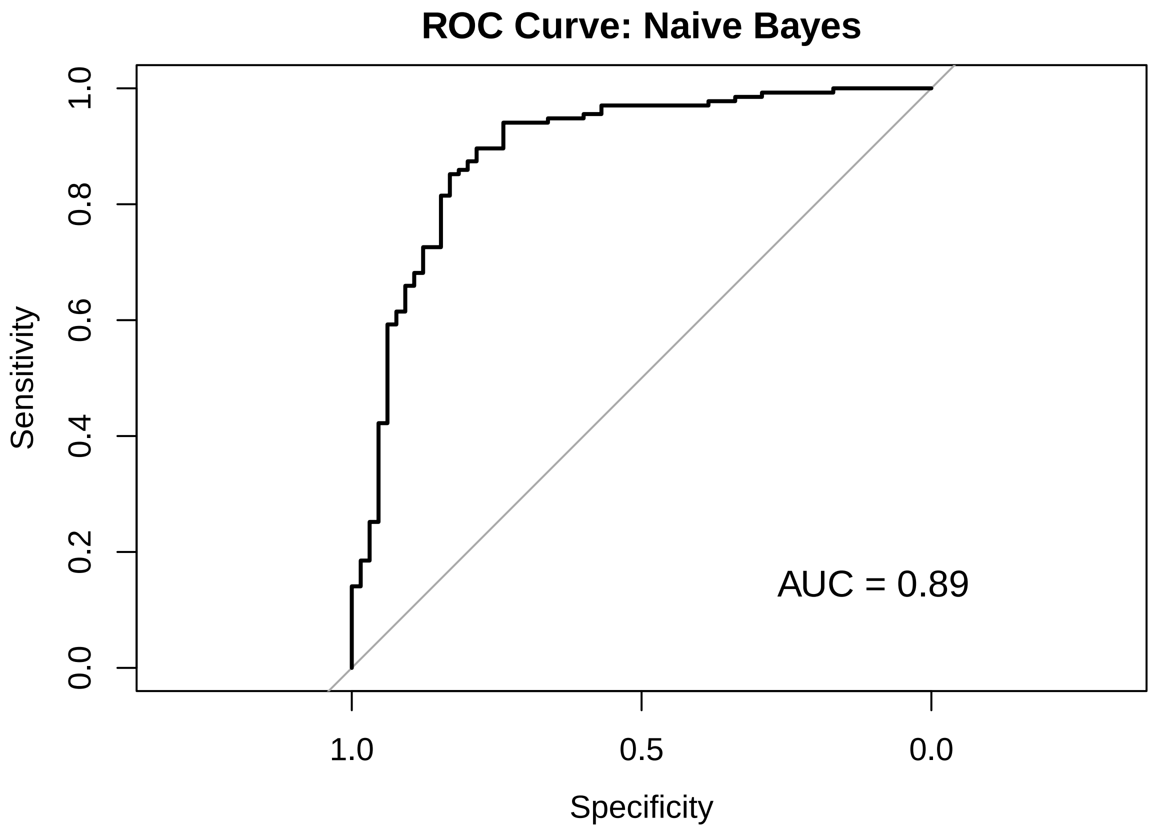


```
predictions.nb <- predict(model.nb, newdata = test_data, type = "prob")
predicted_probabilities.nb <- predictions.nb[, "Severe"]
roc_curve.nb <- roc(test_data$severity, predicted_probabilities.nb)
```

```
## Setting levels: control = Severe, case = Not Severe
```

```
## Setting direction: controls > cases
```

```
plot(roc_curve.nb, main = "ROC Curve: Naive Bayes")
text(0.1, 0.1, paste("AUC =", round(auc(roc_curve.nb), 2)), adj = c(0.5, -0.5), cex = 1.2)
```



```
auc(roc_curve.nb)
```

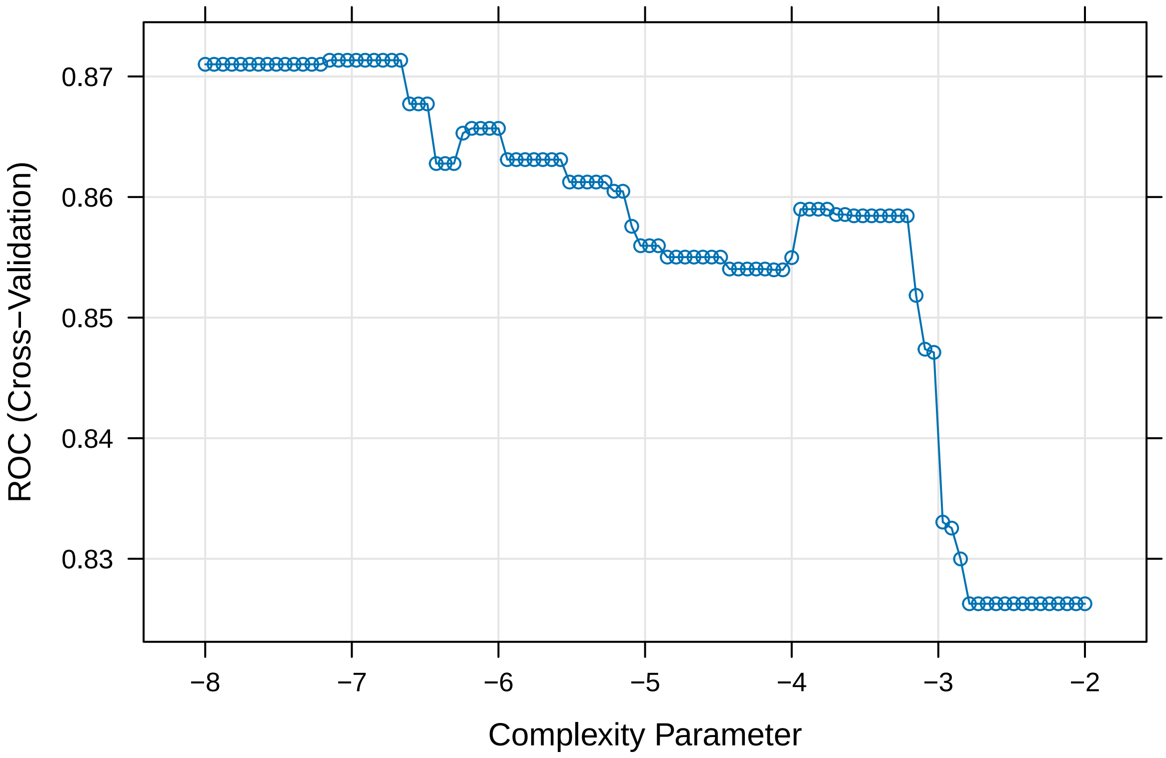
```
## Area under the curve: 0.8919
```

2.5 Classification Trees

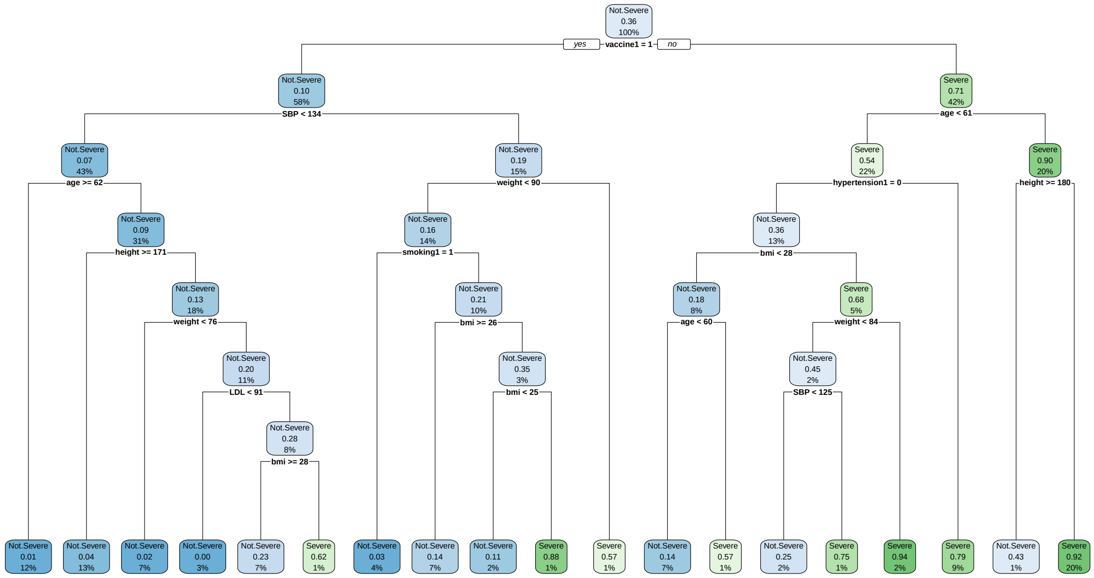
```
ctrl <- trainControl(method = "cv",
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)
set.seed(1)
rpart.fit <- train(severity ~ . ,
                   training_data,
                   method = "rpart",
                   tuneGrid = data.frame(cp = exp(seq(-8,-2, length = 100))),
                   trControl = ctrl)
```

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

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



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

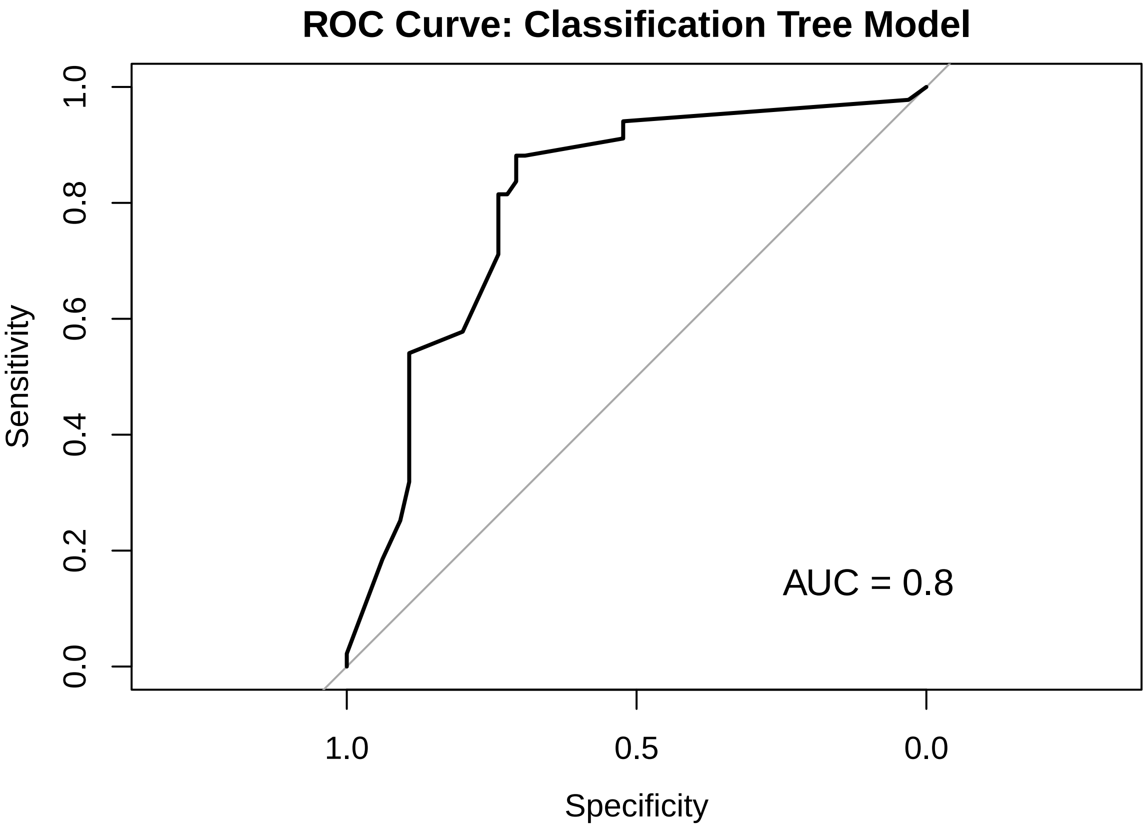


```
predictions.rpart <- predict(rpart.fit, newdata = test_data, type = "prob")
predicted_probabilities.rpart <- predictions.rpart[, "Severe"]
roc_curve.rpart <- roc(test_data$severity, predicted_probabilities.rpart)
```

```
## Setting levels: control = Severe, case = Not Severe
```

```
## Setting direction: controls > cases
```

```
plot(roc_curve.rpart, main = "ROC Curve: Classification Tree Model")
text(0.1, 0.1, paste("AUC =", round(auc(roc_curve.rpart), 2)), adj = c(0.5, -0.5), cex = 1.2)
```

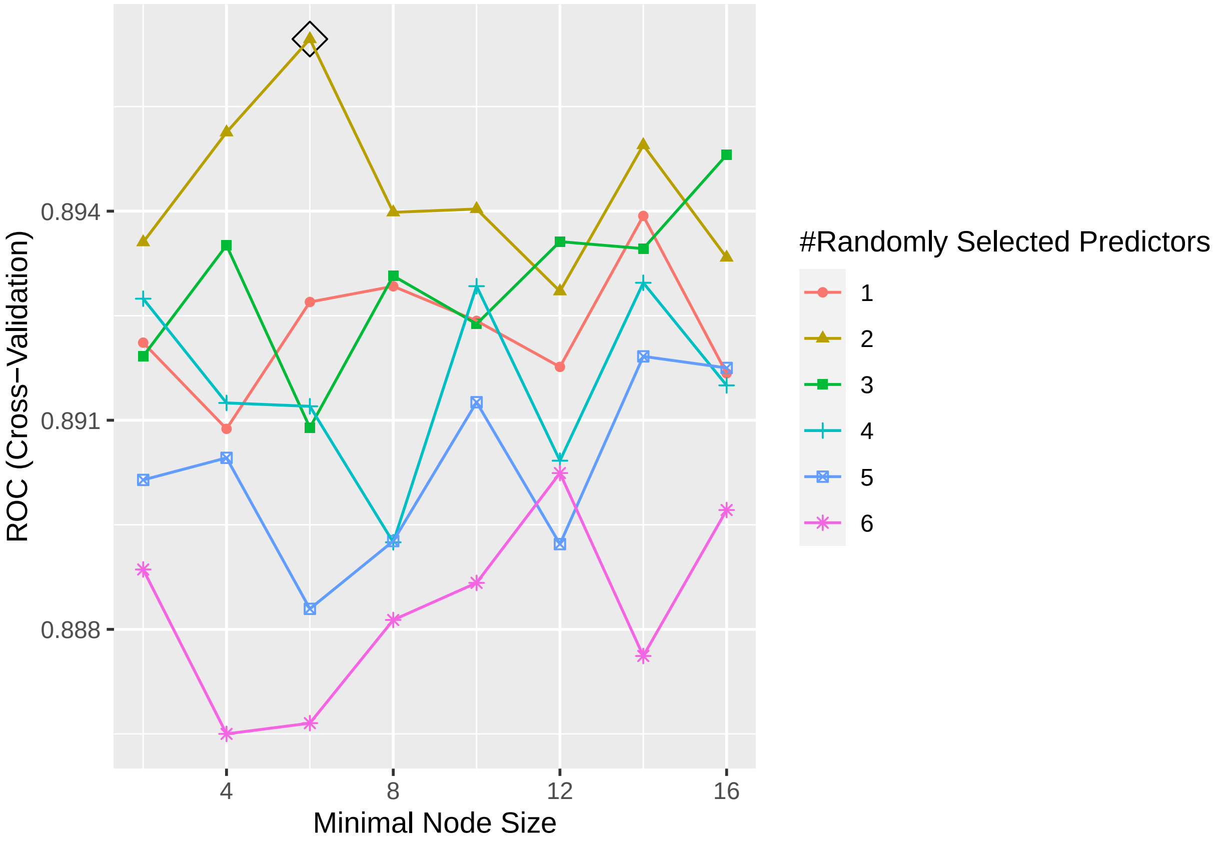


```
auc(roc_curve.rpart)
```

```
## Area under the curve: 0.8019
```

2.6 Random Forests

```
training_data$severity <- make.names(training_data$severity)
ctrl <- trainControl(method = "cv",
                     classProbs = TRUE,
                     summaryFunction = twoClassSummary)

rf.grid <- expand.grid(mtry = 1:6,
                       splitrule = "gini",
                       min.node.size = seq(from = 2, to = 16, by = 2))

set.seed(1)
rf.fit <- train(severity ~ . ,
                training_data,
                method = "ranger",
                tuneGrid = rf.grid,
                metric = "ROC",
                 trControl = ctrl)

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

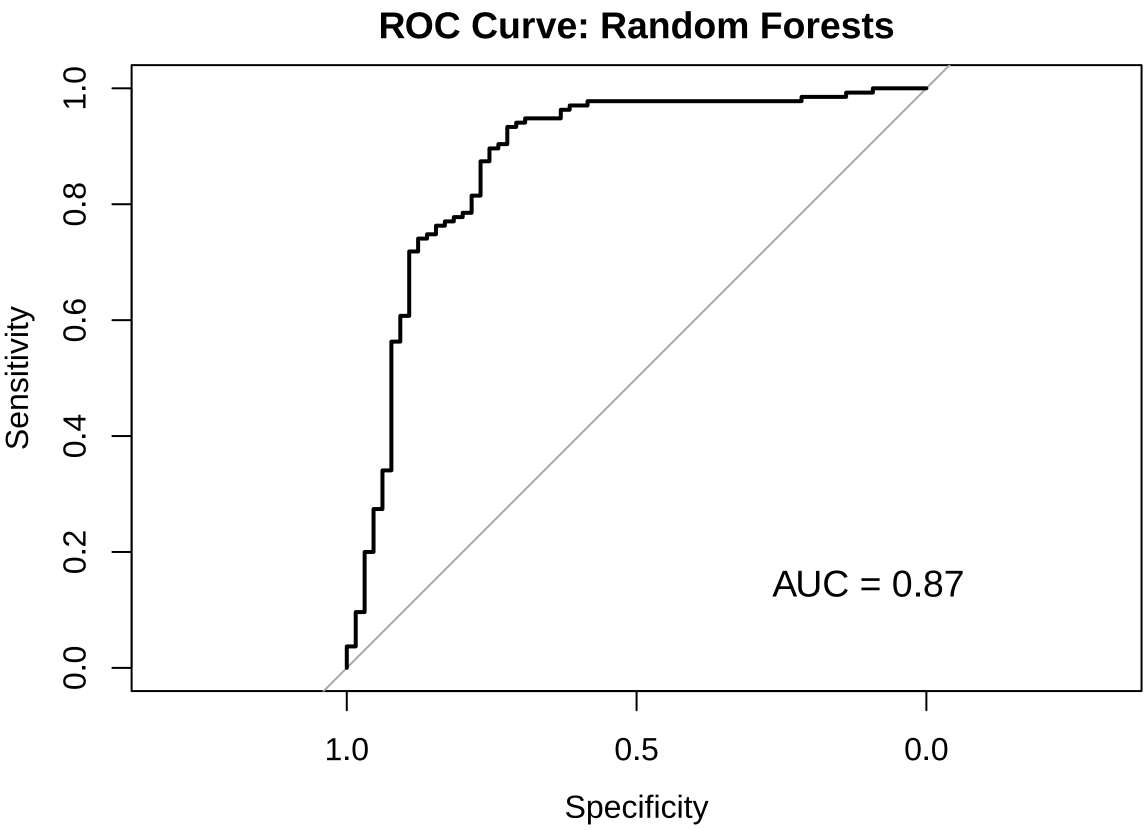


```
rf.pred <- predict(rf.fit, newdata = test_data, type = "prob")[,1]
roc_curve.rf <- roc(test_data$severity, rf.pred)
```

```
## Setting levels: control = Severe, case = Not Severe
```

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

```
plot(roc_curve.rf, main = "ROC Curve: Random Forests")
text(0.1, 0.1, paste("AUC =", round(auc(roc_curve.rf), 2)), adj = c(0.5, -0.5), cex = 1.2)
```

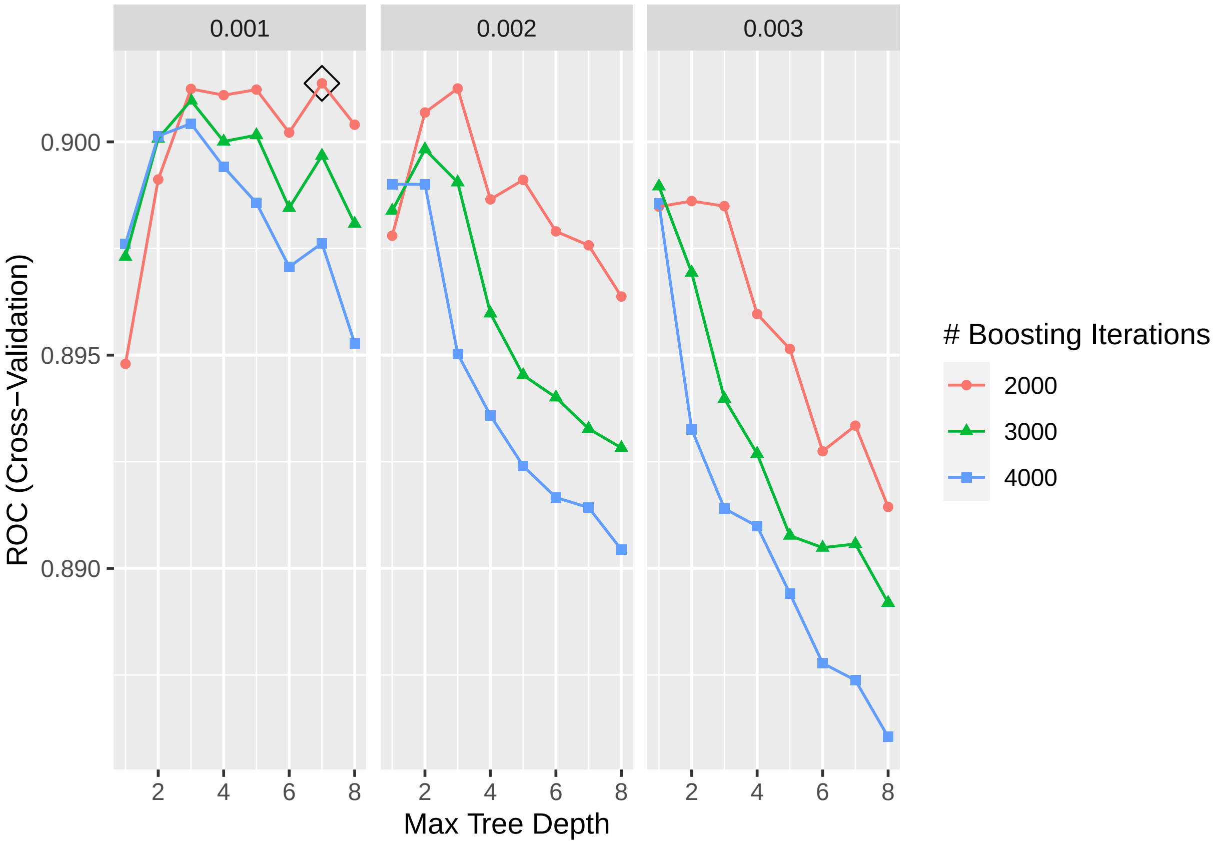


```r
auc(roc_curve.rf)
```

```
## Area under the curve: 0.8746
```

2.7 AdaBoost

```r
set.seed(1)
gbmA.grid = expand.grid(n.trees = c(2000,3000,4000),
                        interaction.depth = 1:8,
                        shrinkage = c(0.001,0.002, 0.003),
                        n.minobsinnode = 1)

gbmA.fit <- train(severity ~ . ,
                  training_data,
                  tuneGrid = gbmA.grid,
                  trControl = ctrl,
                  method = "gbm",
                  distribution = "adaboost",
                  metric = "ROC",
                  verbose = FALSE)

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

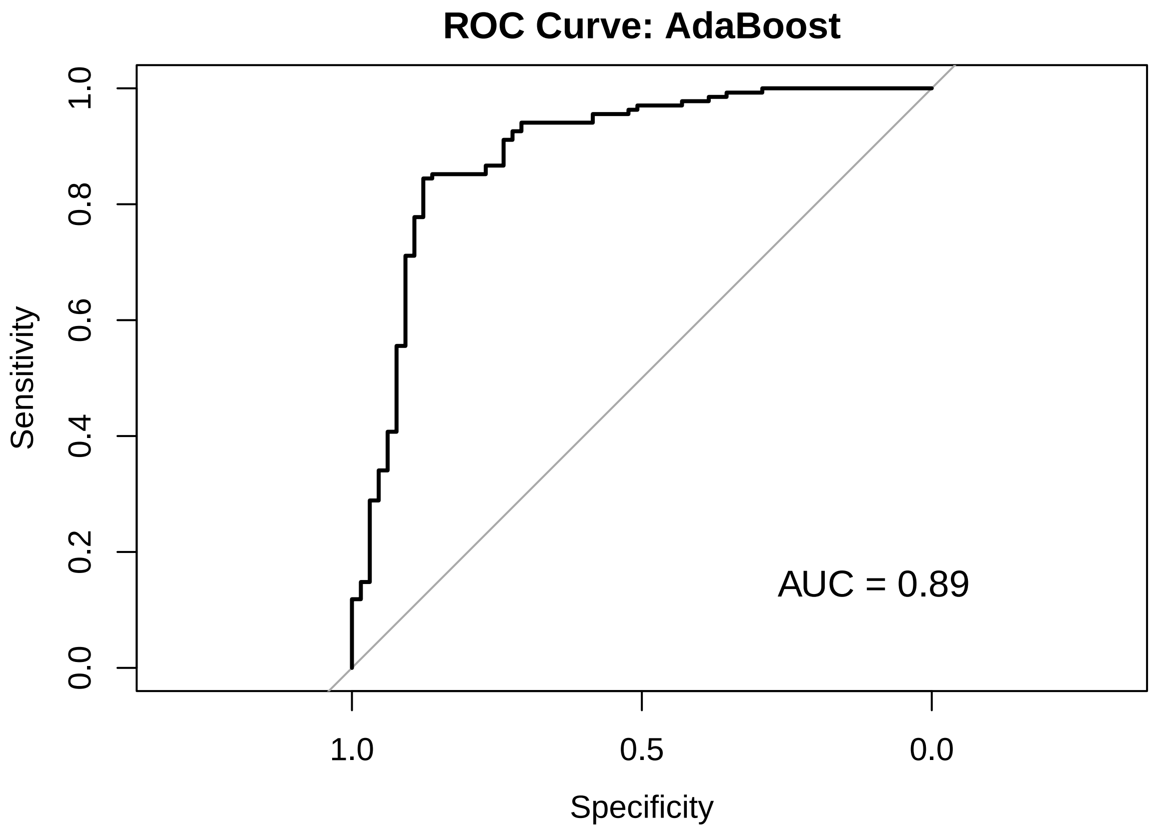


```
gbm.pred <- predict(gbmA.fit, newdata = test_data, type = "prob")[,1]
roc_curve.gbm <- roc(test_data$severity, gbm.pred)
```

```
## Setting levels: control = Severe, case = Not Severe
```

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

```
plot(roc_curve.gbm, main = "ROC Curve: AdaBoost")
text(0.1, 0.1, paste("AUC =", round(auc(roc_curve.gbm), 2)), adj = c(0.5, -0.5), cex = 1.2)
```



```
auc(roc_curve.gbm)
```

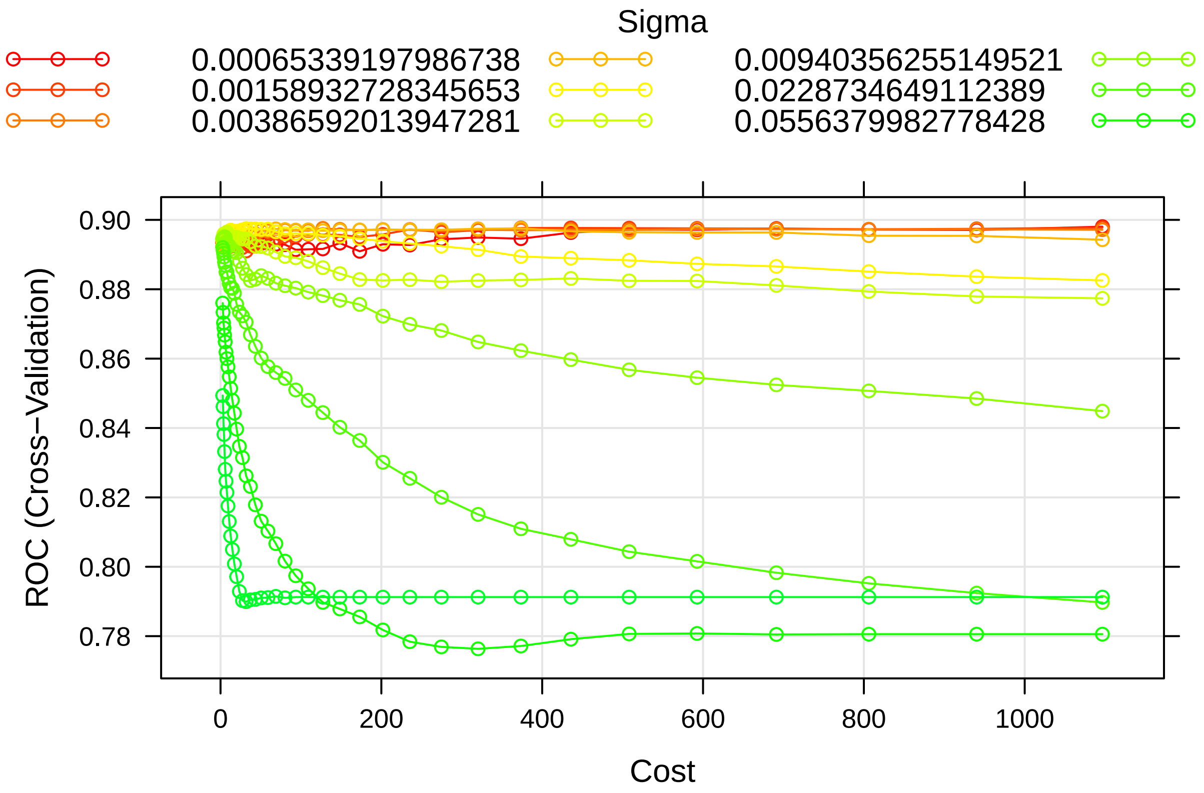
```
## Area under the curve: 0.8909
```

2.8 Support Vector Machine

```
svmr.grid <- expand.grid(C = exp(seq(1, 7, len = 40)),
                         sigma = exp(seq(-10, -2, len = 10)))

set.seed(1)
svmr.fit <- train(severity ~ . , data = training_data,
                  method = "svmRadialSigma",
                  tuneGrid = svmr.grid,
                  trControl = ctrl)
```

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

```
myCol <- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
superpose.line = list(col = myCol))
plot(svmr.fit, highlight = TRUE, par.settings = myPar)
```



```
set.seed(1)
svmr.fit2 <- train(severity ~ . , data = training_data,
                   method = "svmRadialCost",
                   tuneGrid = data.frame(C = exp(seq(-3, 3, len = 20))),
                   trControl = ctrl)
```

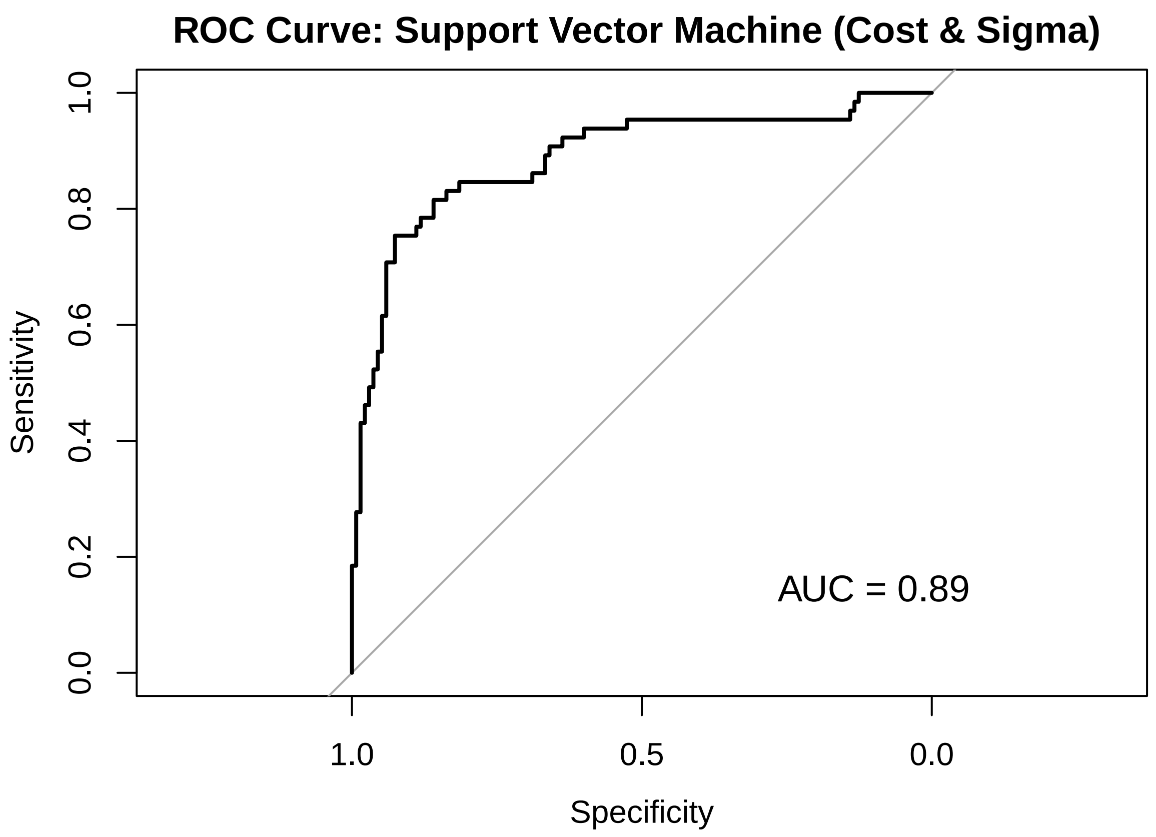
```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

```
test_data$severity <- make.names(test_data$severity)
svmr.pred <- predict(svmr.fit, newdata = test_data, type = "prob")[,1]
roc_curve.svmr <- roc(test_data$severity, svmr.pred)
```

```
## Setting levels: control = Not.Severe, case = Severe
```

```
## Setting direction: controls > cases
```

```
plot(roc_curve.svmr, main = "ROC Curve: Support Vector Machine (Cost & Sigma) ")
text(0.1, 0.1, paste("AUC =", round(auc(roc_curve.svmr), 2)), adj = c(0.5, -0.5), cex = 1.2)
```



```
auc(roc_curve.svmr)
```

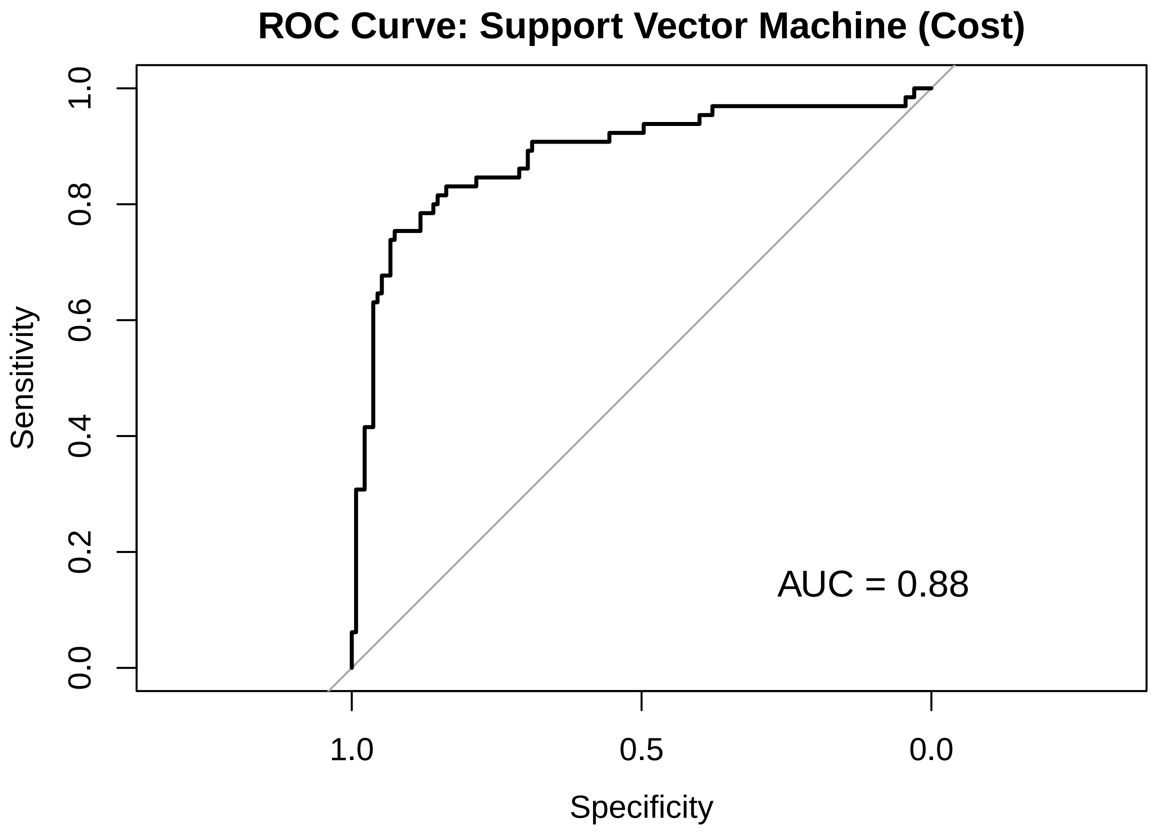
```
## Area under the curve: 0.8883
```

```
svmr2.pred <- predict(svmr.fit2, newdata = test_data, type = "prob")[,1]
roc_curve.svmr2 <- roc(test_data$severity, svmr2.pred)
```

```
## Setting levels: control = Not.Severe, case = Severe
```

```
## Setting direction: controls > cases
```

```
plot(roc_curve.svmr2, main = "ROC Curve: Support Vector Machine (Cost)")
text(0.1, 0.1, paste("AUC =", round(auc(roc_curve.svmr2), 2)), adj = c(0.5, -0.5), cex = 1.2)
```



```
auc(roc_curve.svmr2)
```

```
## Area under the curve: 0.8848
```

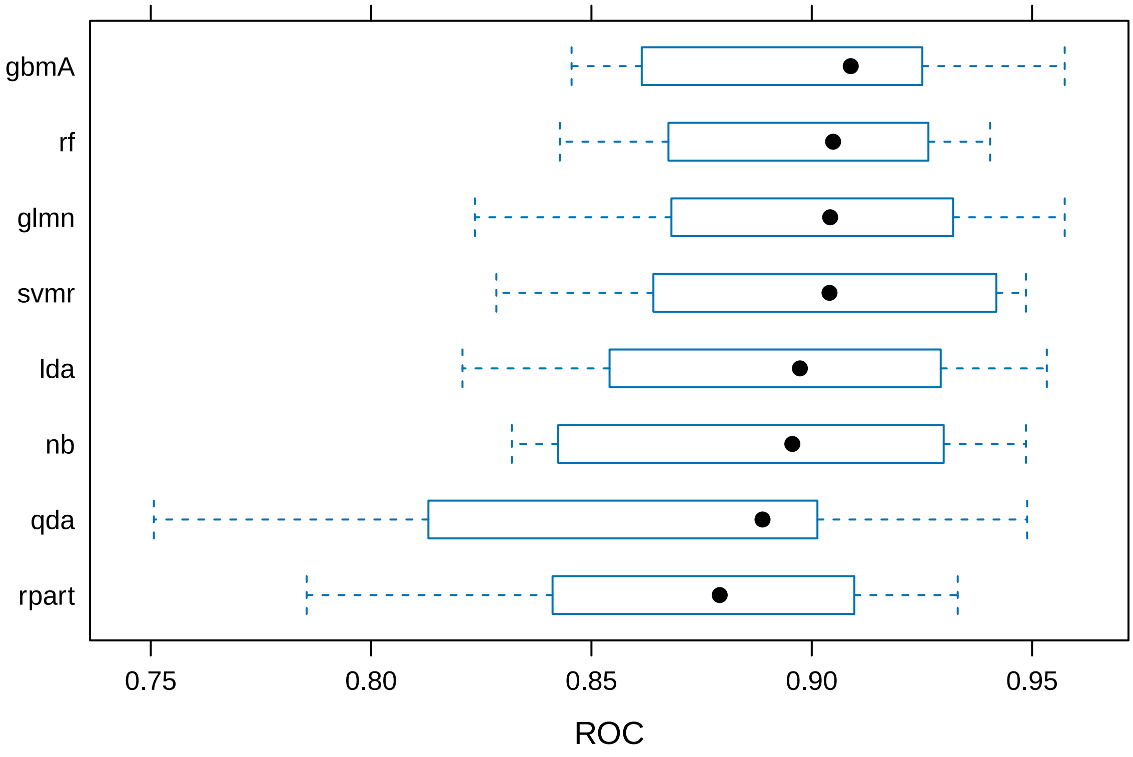
3 Results

3.1 Model Comparasion

```
resamp <- resamples(list(glmn = model.glmn, lda = lda_fit, qda = qda_fit,
                         nb = model.nb, rpart = rpart.fit, rf = rf.fit,
                         gbmA = gbmA.fit, svmr = svmr.fit))
summary(resamp)
```

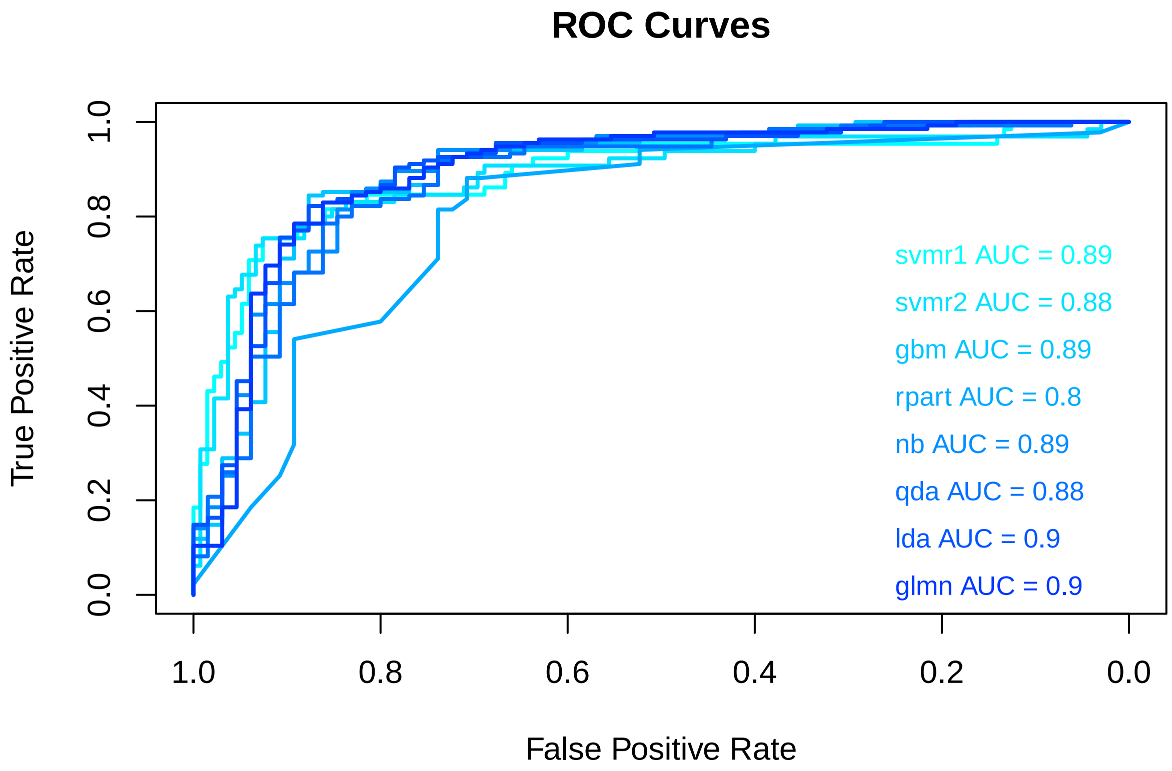
```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: glmn, lda, qda, nb, rpart, rf, gbmA, svmr
## Number of resamples: 10
##
## ROC
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## glmn  0.8235294 0.8687880 0.9041777 0.8960090 0.9301229 0.9574037    0
## lda   0.8207283 0.8569460 0.8973124 0.8924337 0.9280664 0.9533469    0
## qda   0.7507003 0.8227298 0.8888282 0.8646240 0.9012382 0.9488796    0
## nb    0.8319328 0.8432185 0.8955895 0.8875306 0.9275008 0.9486139    0
## rpart 0.7853641 0.8420077 0.8791150 0.8713441 0.9058787 0.9331232    0
## rf    0.8428382 0.8678704 0.9048408 0.8964693 0.9239051 0.9404762    0
## gbmA  0.8454907 0.8663045 0.9088456 0.9013714 0.9238849 0.9574037    0
## svmr  0.8284314 0.8677868 0.9040282 0.8980434 0.9385504 0.9486139    0
##
## Sens
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## glmn  0.8653846 0.8696267 0.9019608 0.9126320 0.9515460 0.9803922    0
## lda   0.7500000 0.8076923 0.8350302 0.8407994 0.8725490 0.9411765    0
## qda   0.7884615 0.8438914 0.8640649 0.8680618 0.8823529 0.9803922    0
## nb    0.8653846 0.9019608 0.9127074 0.9222474 0.9513575 0.9803922    0
## rpart 0.8269231 0.8552036 0.9019608 0.8951735 0.9362745 0.9423077    0
## rf    0.8653846 0.9082768 0.9411765 0.9360106 0.9754902 0.9807692    0
## gbmA  0.8269231 0.8889517 0.9117647 0.9185143 0.9607843 0.9807692    0
## svmr  0.7500000 0.7853507 0.8058069 0.8134992 0.8382353 0.9019608    0
##
## Spec
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## glmn  0.6428571 0.7327586 0.7721675 0.7621921 0.7912562 0.8620690    0
## lda   0.7142857 0.8017241 0.8571429 0.8352217 0.8620690 0.8965517    0
## qda   0.6428571 0.7306034 0.7715517 0.7724138 0.8275862 0.8620690    0
## nb    0.5714286 0.6293103 0.6841133 0.6887931 0.7500000 0.7931034    0
## rpart 0.5714286 0.6896552 0.7019704 0.7200739 0.7564655 0.8620690    0
## rf    0.5714286 0.6813424 0.7241379 0.7201970 0.7789409 0.8275862    0
## gbmA  0.6785714 0.7241379 0.7586207 0.7550493 0.7789409 0.8620690    0
## svmr  0.7142857 0.8017241 0.8571429 0.8390394 0.8851601 0.8965517    0
```

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



3.2 Model Performance

```
roc_curves <- list(svmr1 = roc_curve.svmr,
                   svmr2 = roc_curve.svmr2,
                   gbm = roc_curve.gbm,
                   rpart = roc_curve.rpart,
                   nb = roc_curve.nb,
                   qda = roc_curve.qda,
                   lda = roc_curve.lda,
                   glmn = roc_curve.glmn)

plot(0, 0, type = "n", xlim = c(1, 0), ylim = c(0, 1),
     xlab = "False Positive Rate", ylab = "True Positive Rate",
     main = "ROC Curves")

colors <- colorRampPalette(colors = c("cyan","blue"))(10)

for (i in seq_along(roc_curves)) {
  perf <- roc_curves[[i]]
  auc_val <- round(auc(perf), 2)
  col <- colors[i]
  lines(perf, col = col, lwd = 2)
  text(0.25, 0.8 - 0.1 * i, paste(names(roc_curves)[i], "AUC =", auc_val),
       adj = c(0, 0), col = col, cex = 0.8)
}
```

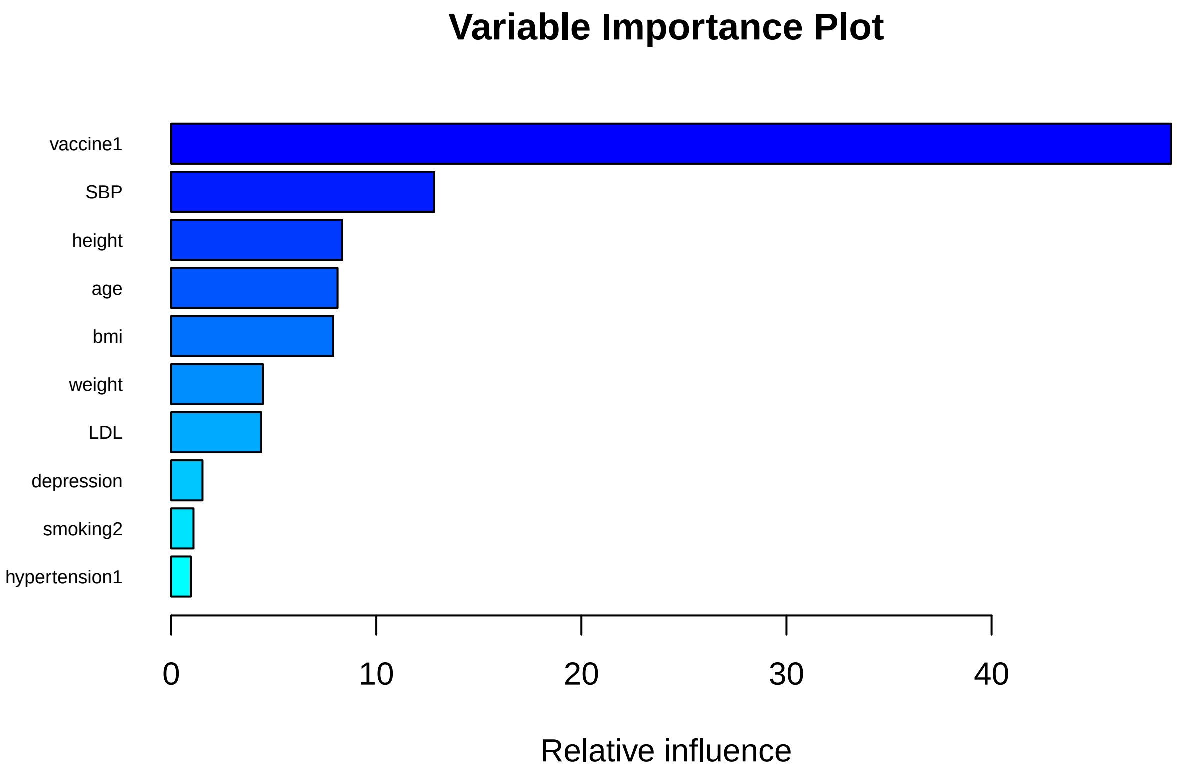


Through model comparasion using the resampling method and evaluating model performance with ROC curves, we have found that the Boosting model demonstrates superior performance. Consequently, we will proceed with utilizing the Boosting model for further analysis and predictions.

4 Conclusion

4.1 Variable Importance

```
plot_gbm <- summary(gbmA.fit$finalModel, las = 1, cBars = 10, cex.names = 0.6)
title("Variable Importance Plot")
```

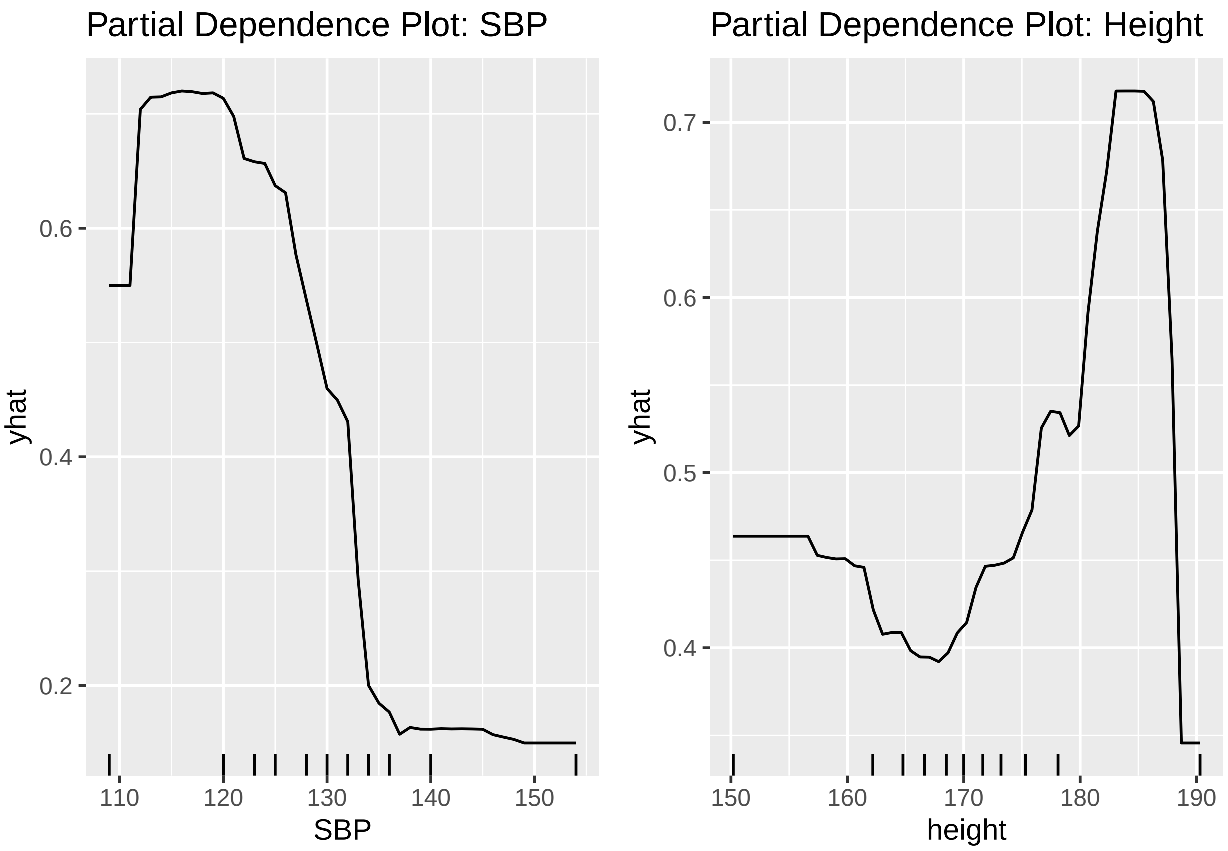


```
plot_gbm
```

```
##                         var     rel.inf
## vaccine1           vaccine1 48.75618901
## SBP                     SBP 12.82385743
## height               height  8.33939471
## age                     age  8.11102940
## bmi                     bmi  7.90197990
## weight               weight  4.46719701
## LDL                     LDL  4.38939762
## depression       depression  1.52517407
## smoking2           smoking2  1.08675091
## hypertension1 hypertension1  0.95455197
## genderMale       genderMale  0.75343279
## diabetes1         diabetes1  0.32022594
## smoking1           smoking1  0.23950743
## raceWhite         raceWhite  0.15636359
## raceBlack         raceBlack  0.13146902
## raceHispanic   raceHispanic  0.04347919
```

4.2 Partial Dependence Plot

```
p1 <- partial(gbmA.fit, pred.var = "SBP",
        plot = TRUE, rug = TRUE,
        plot.engine = "ggplot") + ggtitle("Partial Dependence Plot: SBP")
p2 <- partial(gbmA.fit, pred.var = "height",
        plot = TRUE, rug = TRUE,
        plot.engine = "ggplot") + ggtitle("Partial Dependence Plot: Height")

gridExtra::grid.arrange(p1, p2, nrow = 1)
```


Partial Dependence Plot: SBP
yhat
SBP
Partial Dependence Plot: Height
yhat
height