Final Project

Zhezheng Jin

Contents

```
library(caret)
library(MASS)
library(mlbench)
library(pROC)
library(klaR)
library(glmnet)
library(pdp)
library(vip)
library(AppliedPredictiveModeling)
library(tidyverse)
library(summarytools)
library(corrplot)
library(plotmo)
library(viridis)
library(gtsummary)
library(e1071)
library(tidymodels)
library(patchwork)
library(kernlab)
library(doParallel)
```

Data Import and Wrangling

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

skimr::skim(test_data)
```

Table 1: Data summary

Name	test_data
Number of rows	200
Number of columns	15
_____	
Column type frequency:	
factor	3
numeric	12
_____	
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
race	0	1	FALSE	4	1: 135, 3: 35, 4: 16, 2: 14
smoking	0	1	FALSE	3	0: 117, 1: 65, 2: 18
severity	0	1	FALSE	2	0: 135, 1: 65

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
id	0	1	453.05	279.98	5.0	230.25	441.00	676.25	1000.0	
age	0	1	60.24	4.18	49.0	58.00	60.00	63.00	71.0	
gender	0	1	0.44	0.50	0.0	0.00	0.00	1.00	1.0	
height	0	1	169.63	6.15	152.0	166.00	169.65	174.12	188.1	
weight	0	1	79.51	6.51	61.5	74.80	79.20	84.15	96.3	
bmi	0	1	27.72	2.72	20.4	26.05	27.50	29.70	35.3	
diabetes	0	1	0.12	0.33	0.0	0.00	0.00	0.00	1.0	
hypertension	0	1	0.48	0.50	0.0	0.00	0.00	1.00	1.0	
SBP	0	1	130.01	7.49	108.0	125.00	130.00	135.00	148.0	
LDL	0	1	111.30	18.45	70.0	98.75	111.50	124.00	165.0	
vaccine	0	1	0.64	0.48	0.0	0.00	1.00	1.00	1.0	
depression	0	1	6.72	2.21	2.0	5.00	7.00	8.00	12.0	

```
skimr::skim(training_data)
```

Table 4: Data summary

Name	training_data
Number of rows	800
Number of columns	15
Column type frequency:	
factor	3
numeric	12
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
race	0	1	FALSE	4	1: 521, 3: 149, 4: 80, 2: 50
smoking	0	1	FALSE	3	0: 467, 1: 248, 2: 85
severity	0	1	FALSE	2	0: 514, 1: 286

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
id	0	1	512.36	289.95	1.0	255.75	515.5	767.25	999.0	
age	0	1	60.03	4.30	46.0	57.00	60.0	63.00	72.0	
gender	0	1	0.49	0.50	0.0	0.00	0.0	1.00	1.0	
height	0	1	170.00	6.09	150.2	165.70	170.0	174.10	190.3	
weight	0	1	79.42	7.26	56.6	74.38	79.3	84.40	104.8	
bmi	0	1	27.54	2.74	19.6	25.78	27.6	29.10	37.4	
diabetes	0	1	0.15	0.36	0.0	0.00	0.0	0.00	1.0	
hypertension	0	1	0.46	0.50	0.0	0.00	0.0	1.00	1.0	
SBP	0	1	129.85	7.97	109.0	124.00	130.0	135.00	154.0	
LDL	0	1	110.25	20.05	41.0	98.00	111.0	123.00	174.0	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
vaccine	0	1	0.58	0.49	0.0	0.00	1.0	1.00	1.0	
depression	0	1	6.91	2.12	0.0	5.00	7.0	8.00	13.0	

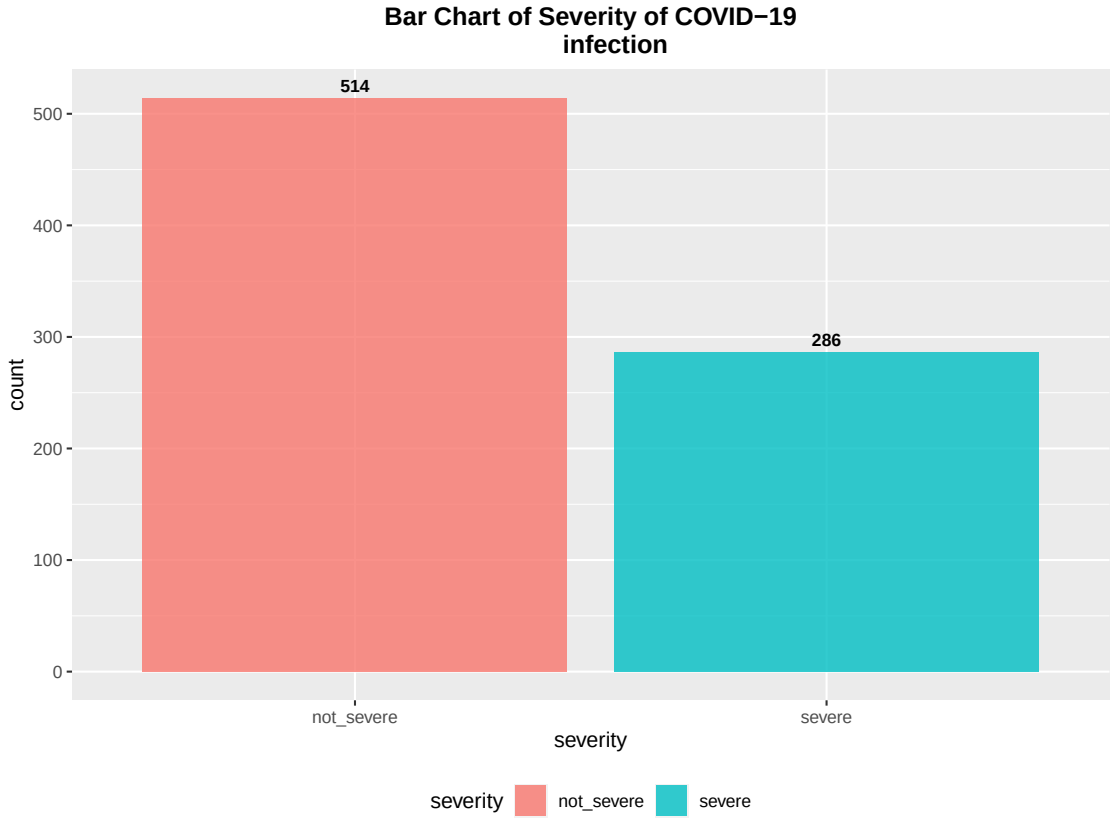
```
# In total, 7 factor variables in the data

train <- training_data %>%
  janitor::clean_names() %>%
  select(-id) %>%
  select(age,height,weight,bmi,sbp,ldl,everything()) %>%
  mutate(
    gender = factor(gender,levels = c("0","1"), labels = c("Female", "Male")),
    race = factor(race,levels = c("1","2","3","4"),
                  labels = c("White", "Asian","Black","Hispanic")),
    smoking = factor(smoking,levels = c("0","1","2"),
                     labels = c("Never_smoked", "Former_smoker","Current_smoker")),
    hypertension = factor(hypertension,levels = c("0", "1"),
                          labels = c("No", "Yes")),
    diabetes = factor(diabetes,levels = c("0", "1"),
                      labels = c("No", "Yes")),
    vaccine = factor(vaccine,levels = c("0", "1"),
                     labels = c("Not_vaccinated", "Vaccinated")),
    severity = factor(severity,levels = c("0", "1"),
                      labels = c("not_severe", "severe"))
      )
test <- test_data %>%
  janitor::clean_names() %>%
  select(-id) %>%
  mutate(
    gender = factor(gender,levels = c("0","1"), labels = c("Female", "Male")),
    race = factor(race,levels = c("1","2","3","4"),
                  labels = c("White", "Asian","Black","Hispanic")),
    smoking = factor(smoking,levels = c("0","1","2"),
                     labels = c("Never_smoked", "Former_smoker","Current_smoker")),
    hypertension = factor(hypertension,levels = c("0", "1"),
                          labels = c("No", "Yes")),
    diabetes = factor(diabetes,levels = c("0", "1"),
                      labels = c("No", "Yes")),
    vaccine = factor(vaccine,levels = c("0", "1"),
                     labels = c("Not_vaccinated", "Vaccinated")),
    severity = factor(severity,levels = c("0", "1"),
                      labels = c("not_severe", "severe"))
      )
```

EDA

We will use training data for EDA.

Outcome:severity

```
# Bar chart
y_bar <- train %>%
  ggplot(aes(x = severity, fill = severity)) +
  geom_bar(stat = "Count", position = "dodge", alpha = 0.8) +
  labs(x = "severity", fill = "severity",title = "Bar Chart of Severity of COVID-19
       infection") +
  geom_text(stat = "Count", aes(label = after_stat(count), group = severity),
            position = position_dodge(width = 0.9), vjust = -0.5,size = 3,fontface = "bold") +
  theme(
        legend.position = "bottom",
        plot.title = element_text(face = "bold",hjust = 0.5)
       )

y_bar
```



Numerical Predictors

```
skimr::skim(train)
```

Table 7: Data summary

Name	train
Number of rows	800
Number of columns	14
Column type frequency:	
factor	7
numeric	7
Group variables	None

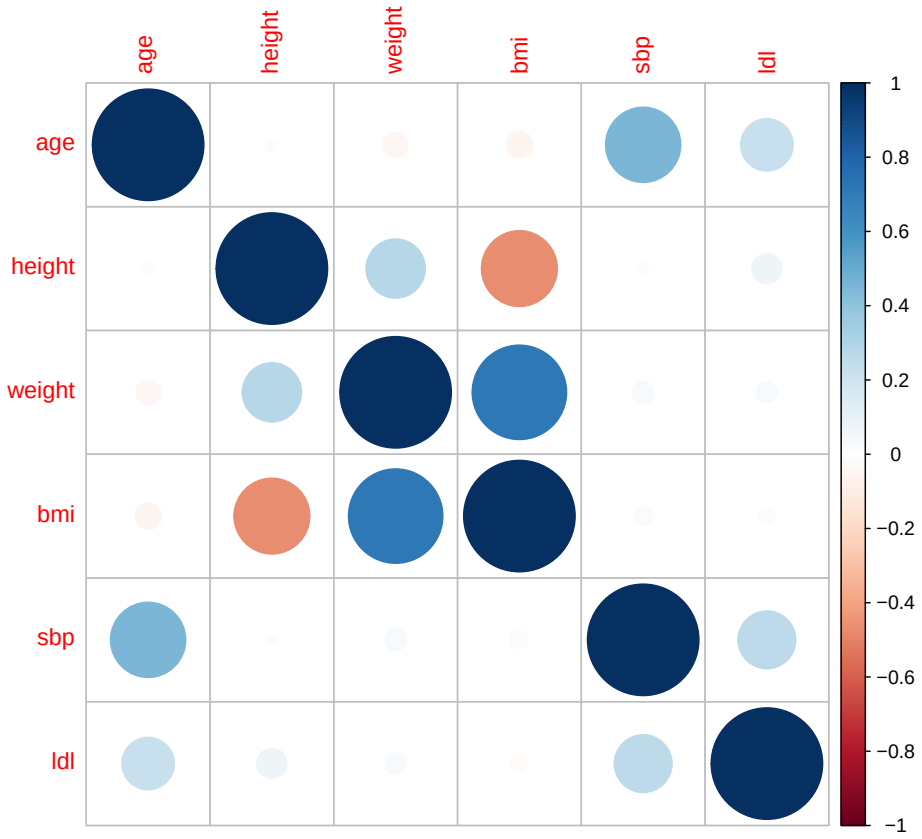
Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gender	0	1	FALSE	2	Fem: 410, Mal: 390
race	0	1	FALSE	4	Whi: 521, Bla: 149, His: 80, Asi: 50
smoking	0	1	FALSE	3	Nev: 467, For: 248, Cur: 85
diabetes	0	1	FALSE	2	No: 679, Yes: 121
hypertension	0	1	FALSE	2	No: 432, Yes: 368
vaccine	0	1	FALSE	2	Vac: 464, Not: 336
severity	0	1	FALSE	2	not: 514, sev: 286

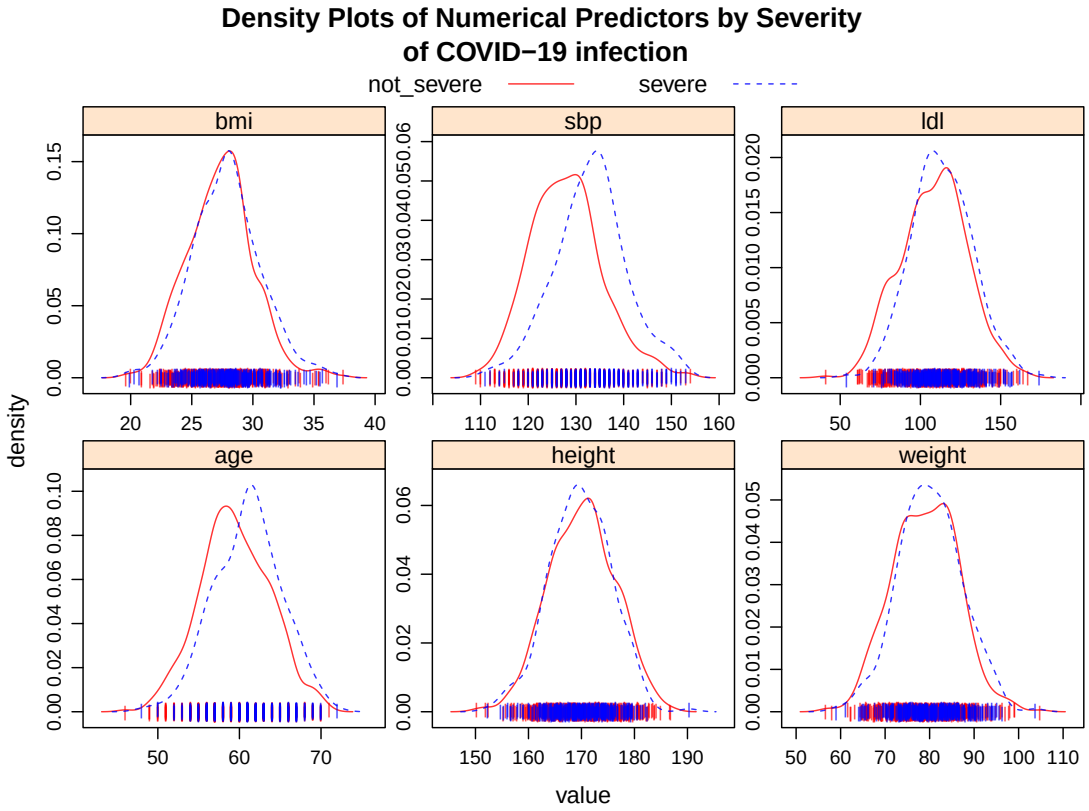
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
age	0	1	60.03	4.30	46.0	57.00	60.0	63.0	72.0	
height	0	1	170.00	6.09	150.2	165.70	170.0	174.1	190.3	
weight	0	1	79.42	7.26	56.6	74.38	79.3	84.4	104.8	
bmi	0	1	27.54	2.74	19.6	25.78	27.6	29.1	37.4	
sbp	0	1	129.85	7.97	109.0	124.00	130.0	135.0	154.0	
ldl	0	1	110.25	20.05	41.0	98.00	111.0	123.0	174.0	
depression	0	1	6.91	2.12	0.0	5.00	7.0	8.0	13.0	

```
# Multicollinearity

corrplot(cor(train[, 1:6]), method = "circle", type = "full")
```



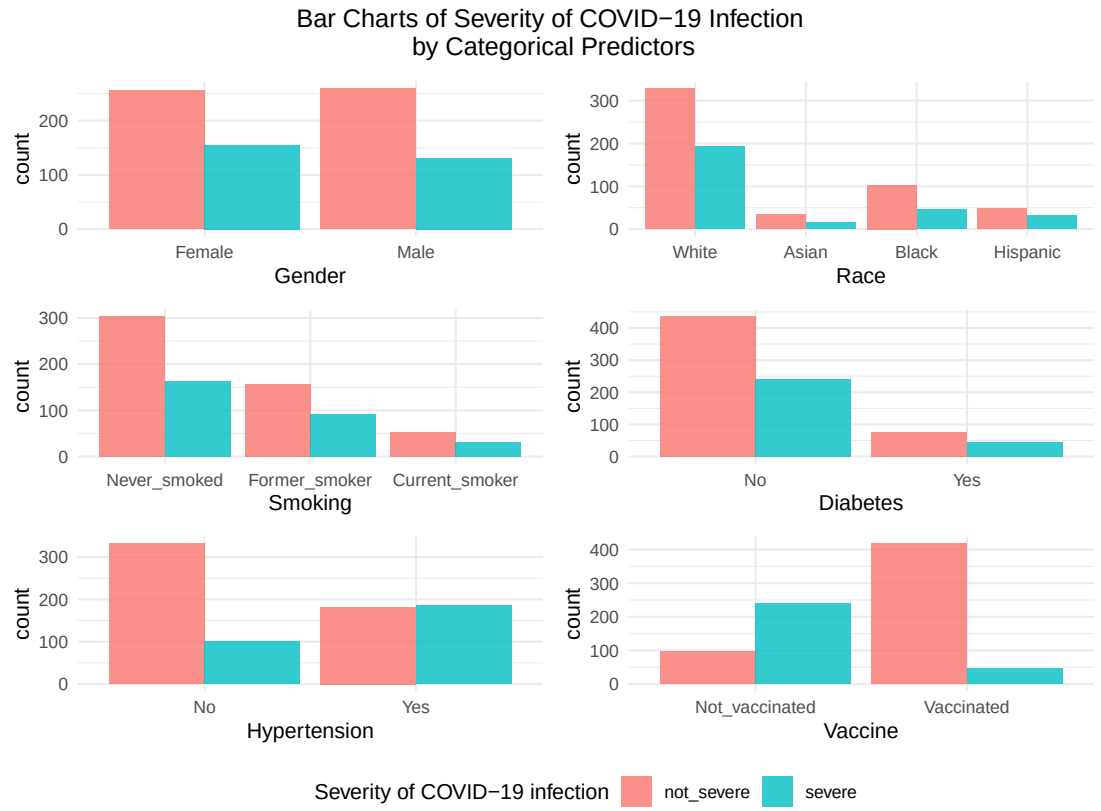

```
# Multicollinearity presents.

# Density Plots
theme1 <- transparentTheme(trans = .8)
trellis.par.set(theme1)

density <- featurePlot(x = train[, 1:6],
                       y = train$severity,
                       scales = list(x = list(relation = "free"),
                                     y = list(relation = "free")),
                       plot = "density", pch = "|",
                       auto.key = list(columns = 2),
                       labels = c("value","density"),
                       main = "Density Plots of Numerical Predictors by Severity
of COVID-19 infection")
density
```



Categorical Predictors

```
# Bar Chart
gender_bar <- train %>%
  ggplot(aes(x = gender, fill = severity)) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.8) +
  labs(x = "Gender", fill = "Severity of COVID-19 infection") +
  theme_minimal() +
  theme(legend.position = "bottom")

race_bar <- train %>%
  ggplot(aes(x = race, fill = severity)) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.8) +
  labs(x = "Race", fill = "Severity of COVID-19 infection") +
  theme_minimal() +
  theme(legend.position = "bottom")

smoking_bar <- train %>%
  ggplot(aes(x = smoking, fill = severity)) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.8) +
  labs(x = "Smoking", fill = "Severity of COVID-19 infection") +
  theme_minimal() +
  theme(legend.position = "bottom")

diabetes_bar <- train %>%
```

```
  ggplot(aes(x = diabetes, fill = severity)) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.8) +
  labs(x = "Diabetes", fill = "Severity of COVID-19 infection") +
  theme_minimal() +
  theme(legend.position = "bottom")

hypertension_bar <- train %>%
  ggplot(aes(x = hypertension, fill = severity)) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.8) +
  labs(x = "Hypertension", fill = "Severity of COVID-19 infection") +
  theme_minimal() +
  theme(legend.position = "bottom")

vaccine_bar <- train %>%
  ggplot(aes(x = vaccine, fill = severity)) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.8) +
  labs(x = "Vaccine", fill = "Severity of COVID-19 infection") +
  theme_minimal() +
  theme(legend.position = "bottom")

cate.bar <- gender_bar + race_bar + smoking_bar +
  diabetes_bar + hypertension_bar + vaccine_bar +
  plot_layout(ncol = 2) +
  plot_annotation(title = "Bar Charts of Severity of COVID-19 Infection
by Categorical Predictors")
cate.bar + plot_layout(guides = 'collect') &
  theme(legend.position = "bottom", plot.title = element_text(hjust = 0.5))
```



Model Training

Using caret

Penalized Logistic Regression

```
registerDoParallel(detectCores() - 1)
ctrl <- trainControl(method = "cv", number = 10,
                     allowParallel = TRUE)

glmnGrid <- expand.grid(.alpha = seq(0, 1, length = 21),
                        .lambda = exp(seq(-6, 3, length = 50)))

set.seed(2358)
glmn.fit <- train(severity ~ .,
                  data = train,
                  method = "glmnet",
                  tuneGrid = glmnGrid,
                  trControl = ctrl)
glmn.fit$bestTune # within the range
```

```
##     alpha     lambda
## 265  0.25 0.03243324
```

```r
# Performance Evaluation
glmn.pred.prob <- predict(glmn.fit, newdata = test,type = "prob") [,2]
glmn.pred <- rep("not_severe", length(glmn.pred.prob))
glmn.pred[glmn.pred.prob>0.5] <- "severe"

confusionMatrix(data = as.factor(glmn.pred),
                reference = test$severity,
                positive = "severe")
```

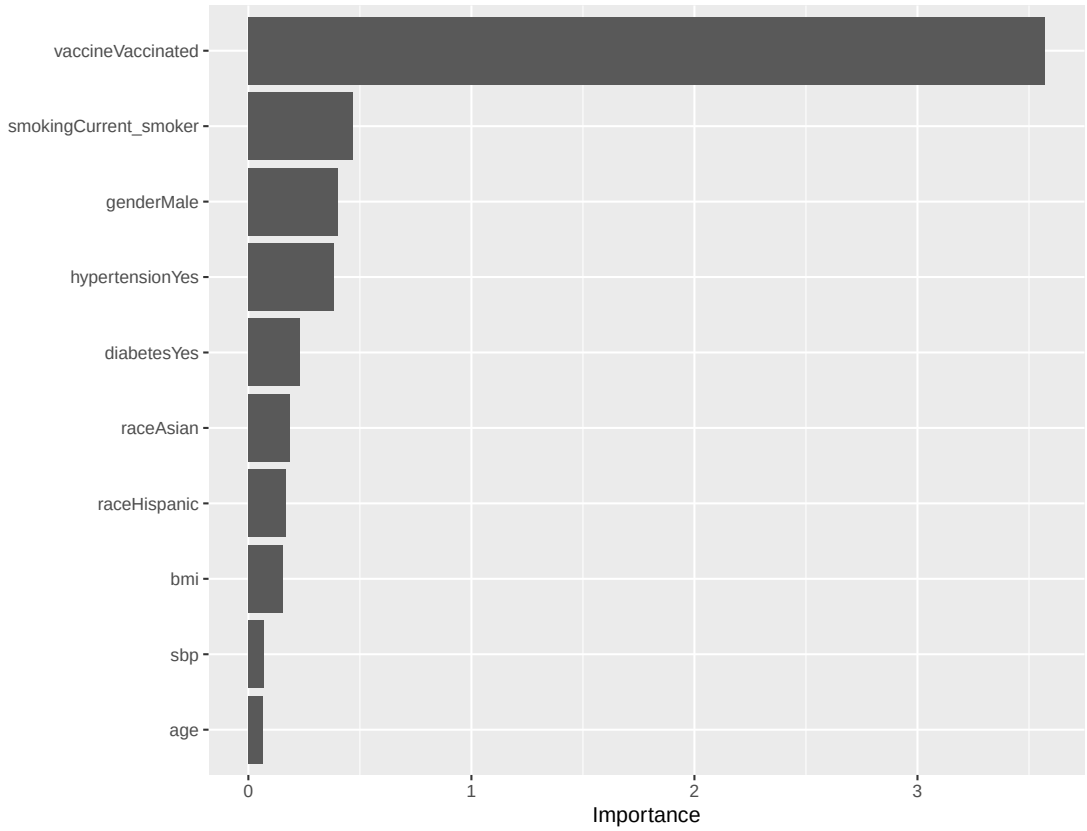
```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   not_severe severe
##   not_severe        122     15
##   severe             13     50
##
##                Accuracy : 0.86
##                  95% CI : (0.8041, 0.9049)
##     No Information Rate : 0.675
##     P-Value [Acc > NIR] : 1.698e-09
##
##                   Kappa : 0.6783
##
##  Mcnemar's Test P-Value : 0.8501
##
##             Sensitivity : 0.7692
##             Specificity : 0.9037
##          Pos Pred Value : 0.7937
##          Neg Pred Value : 0.8905
##              Prevalence : 0.3250
##          Detection Rate : 0.2500
##    Detection Prevalence : 0.3150
##       Balanced Accuracy : 0.8365
##
##        'Positive' Class : severe
##
```

```r
coef(glmn.fit$finalModel, s = glmn.fit$bestTune$lambda)
```

```
## 17 x 1 sparse Matrix of class "dgCMatrix"
##                                    s1
## (Intercept)           -10.445162599
## age                     0.043691132
## height                 -0.003841106
## weight                  .
## bmi                     0.092054943
## sbp                     0.046329891
## ldl                     0.004695970
## genderMale             -0.192816190
## raceAsian               .
## raceBlack               .
## raceHispanic            .
```

```
## smokingFormer_smoker     .
## smokingCurrent_smoker    0.129960885
## diabetesYes              .
## hypertensionYes          0.395198082
## vaccineVaccinated       -2.691020050
## depression               .
```

```
vip(glmn.fit$finalModel)
```

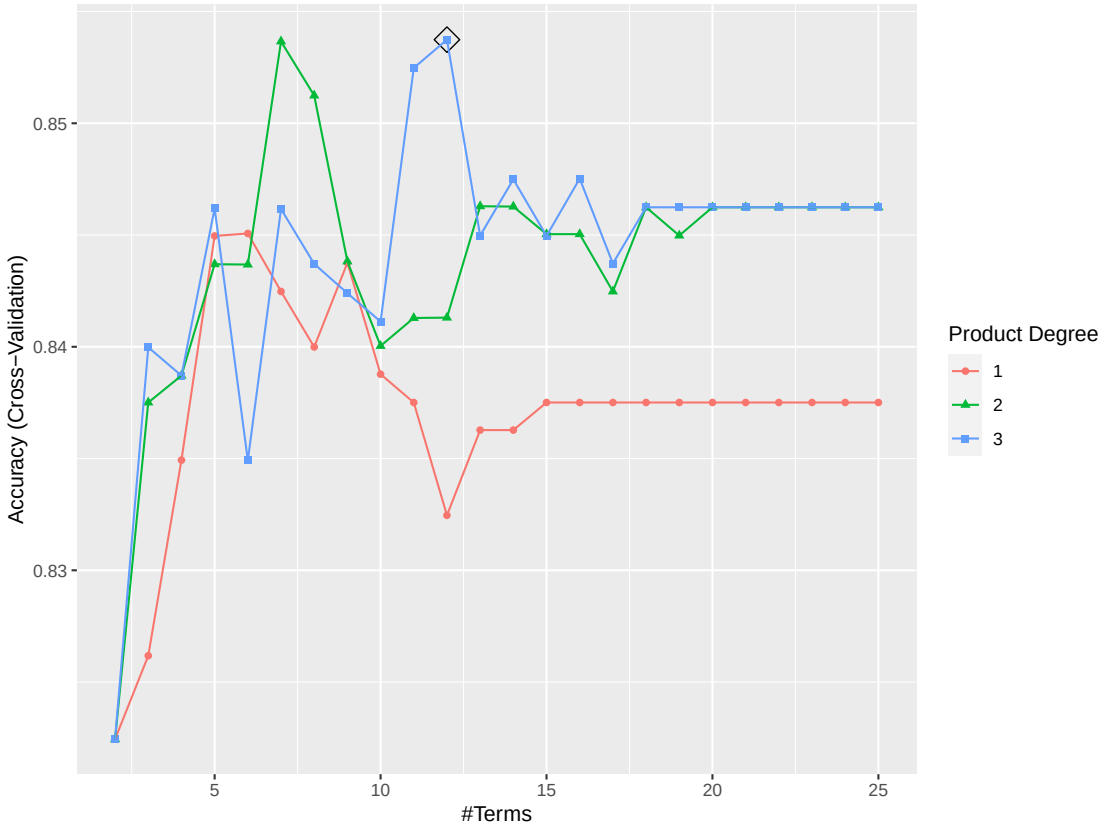


MARS

```
set.seed(2358)
mars.fit <- train(severity ~ .,
                  data = train,
                  method = "earth",
                  tuneGrid = expand.grid(degree = 1:3,
                                         nprune = 2:25),
                  trControl = ctrl)
```

```
## Loading required package: earth
```

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



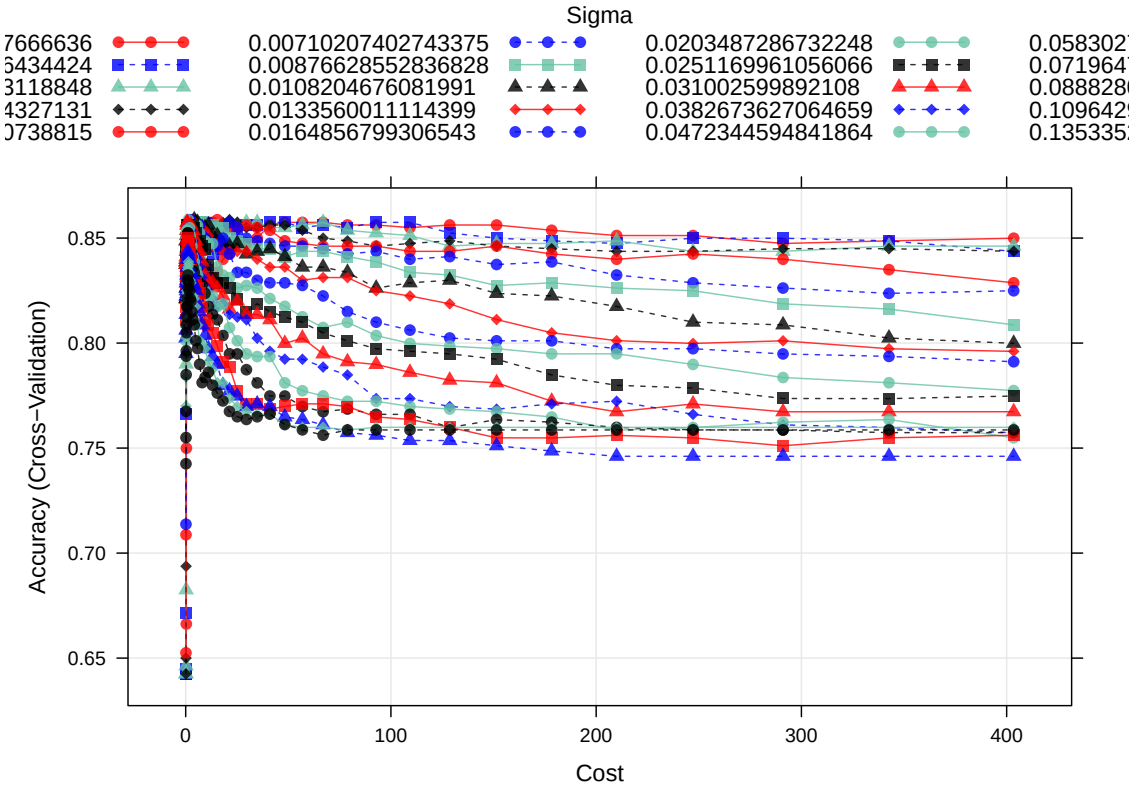

```
# Performance Evaluation
mars.pred <- predict(mars.fit, newdata = test)

confusionMatrix(data = as.factor(mars.pred),
                reference = test$severity,
                positive = "severe")
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   not_severe severe
##   not_severe        116     17
##   severe             19     48
##
##                Accuracy : 0.82
##                  95% CI : (0.7596, 0.8706)
##     No Information Rate : 0.675
##     P-Value [Acc > NIR] : 3.135e-06
##
##                   Kappa : 0.593
##
##  Mcnemar's Test P-Value : 0.8676
##
##             Sensitivity : 0.7385
##             Specificity : 0.8593
##          Pos Pred Value : 0.7164
```

```
##          Neg Pred Value : 0.8722
##              Prevalence : 0.3250
##          Detection Rate : 0.2400
##    Detection Prevalence : 0.3350
##       Balanced Accuracy : 0.7989
##
##        'Positive' Class : severe
##
```

SVM-Radial Kernel

```
svmr.grid <- expand.grid(C = exp(seq(-2,6,len=50)),
                         sigma = exp(seq(-6,-2,len=20)))
set.seed(2358)
svmr.fit <- train(severity ~ .,
                  data = train,
                  method = "svmRadialSigma",
                  tuneGrid = svmr.grid,
                  trControl = ctrl)

plot(svmr.fit, transform.y = log, transform.x = log,
     color.palette = terrain.colors)
```

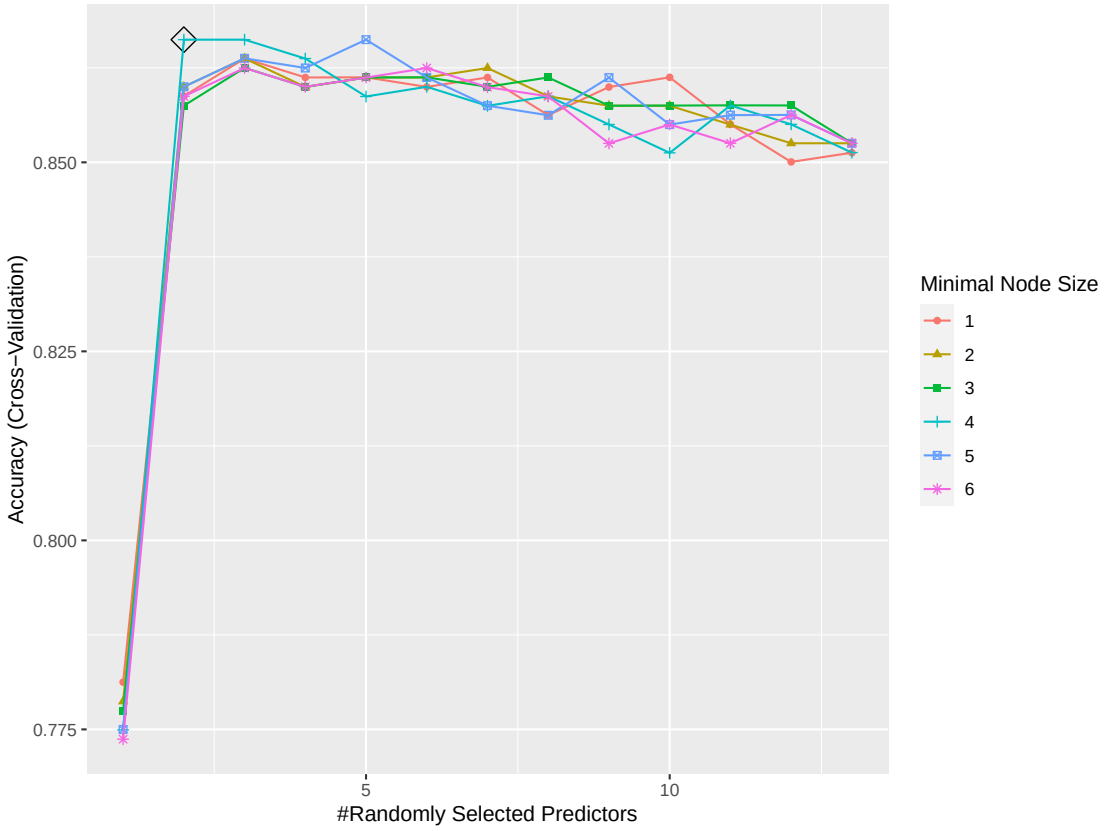


```r
# Performance Evaluation
svmr.pred <- predict(svmr.fit, newdata = test)

confusionMatrix(data = as.factor(svmr.pred),
                reference = test$severity,
                positive = "severe")
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   not_severe severe
##   not_severe        119     15
##   severe             16     50
##
##                Accuracy : 0.845
##                  95% CI : (0.7873, 0.8922)
##     No Information Rate : 0.675
##     P-Value [Acc > NIR] : 3.744e-08
##
##                   Kappa : 0.6481
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.7692
##             Specificity : 0.8815
##          Pos Pred Value : 0.7576
##          Neg Pred Value : 0.8881
##              Prevalence : 0.3250
##          Detection Rate : 0.2500
##    Detection Prevalence : 0.3300
##       Balanced Accuracy : 0.8254
##
##        'Positive' Class : severe
##
```

Random Forest

```r
rf.grid <- expand.grid(mtry = 1:13,
                       splitrule = "gini",
                       min.node.size = 1:6)
set.seed(2358)
rf.fit <- train(severity ~ .,
                data = train,
                method = "ranger",
                tuneGrid = rf.grid,
                trControl = ctrl)

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



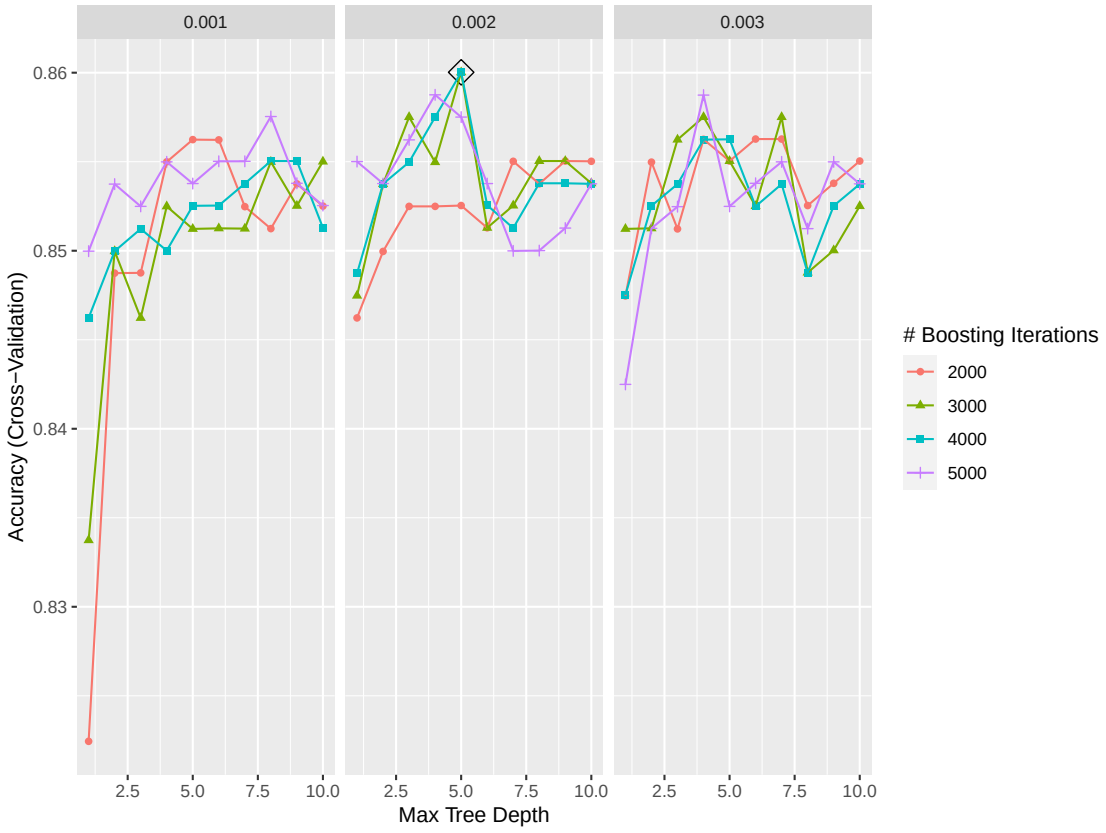

```
# Performance Evaluation
rf.pred <- predict(rf.fit, newdata = test)

confusionMatrix(data = as.factor(rf.pred),
                reference = test$severity,
                positive = "severe")
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   not_severe severe
##   not_severe        126     19
##   severe              9     46
##
##                Accuracy : 0.86
##                  95% CI : (0.8041, 0.9049)
##     No Information Rate : 0.675
##     P-Value [Acc > NIR] : 1.698e-09
##
##                   Kappa : 0.6677
##
##  Mcnemar's Test P-Value : 0.08897
##
##             Sensitivity : 0.7077
##             Specificity : 0.9333
##          Pos Pred Value : 0.8364
```

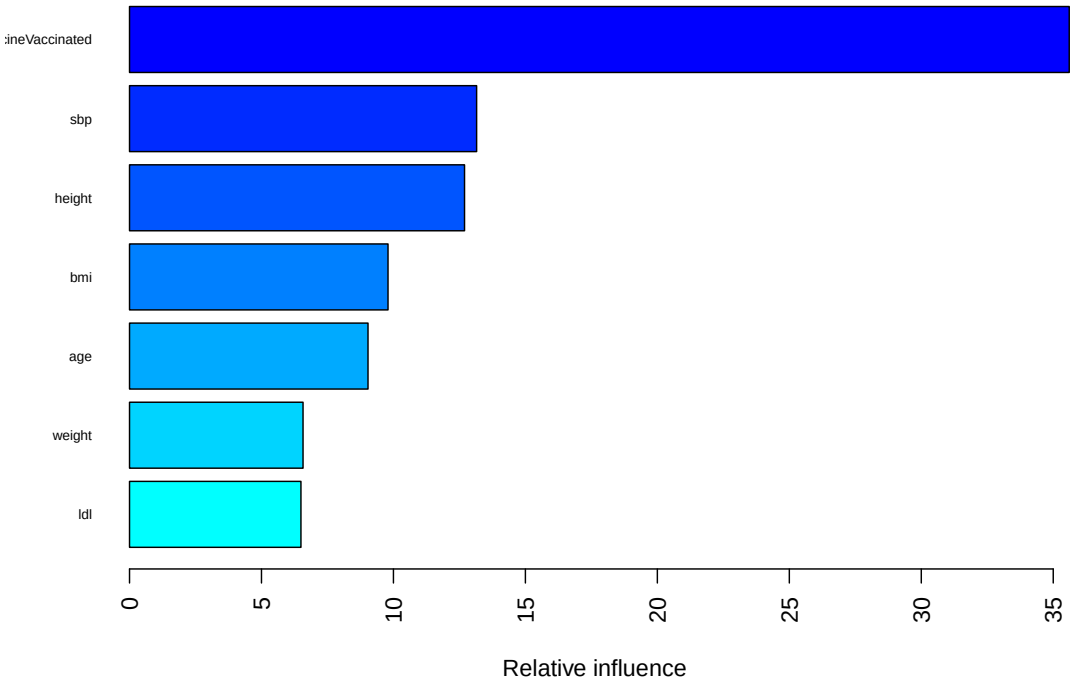
```
##          Neg Pred Value : 0.8690
##              Prevalence : 0.3250
##          Detection Rate : 0.2300
##    Detection Prevalence : 0.2750
##       Balanced Accuracy : 0.8205
##
##        'Positive' Class : severe
##
```

Classification Tree(Adaboost)

```
gbmA.grid <- expand.grid(n.trees = c(2000,3000,4000,5000),
                         interaction.depth = 1:10,
                         shrinkage = c(0.001,0.002,0.003),
                         n.minobsinnode = 1)
set.seed(2358)
gbmA.fit <- train(severity ~ .,
                  data = train,
                  method = "gbm",
                  tuneGrid = gbmA.grid,
                  trControl = ctrl,
                  distribution = "adaboost",
                  verbose = FALSE)

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



```r
# Variable importance
summary(gbmA.fit$finalModel, las = 2, cBars = 7, cex.names = 0.6)
```




```
##                                       var     rel.inf
## vaccineVaccinated         vaccineVaccinated 35.60604381
## sbp                                   sbp 13.15191007
## height                             height 12.69336688
## bmi                                   bmi  9.79417630
## age                                   age  9.03634720
## weight                             weight  6.57422671
## ldl                                   ldl  6.49548285
## depression                      depression  2.64439101
## smokingCurrent_smoker smokingCurrent_smoker  1.35288280
## genderMale                      genderMale  0.93432633
## smokingFormer_smoker   smokingFormer_smoker  0.49852423
## raceAsian                       raceAsian  0.37282520
## diabetesYes                    diabetesYes  0.27504987
## raceBlack                       raceBlack  0.27252134
## hypertensionYes             hypertensionYes  0.24327454
## raceHispanic                  raceHispanic  0.05465085
```

```r
# Performance Evaluation
gbmA.pred <- predict(gbmA.fit, newdata = test)

confusionMatrix(data = as.factor(gbmA.pred),
```

```
                reference = test$severity,
                positive = "severe")
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   not_severe severe
##   not_severe        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 : 0.7231
##             Specificity : 0.9185
##          Pos Pred Value : 0.8103
##          Neg Pred Value : 0.8732
##              Prevalence : 0.3250
##          Detection Rate : 0.2350
##    Detection Prevalence : 0.2900
##       Balanced Accuracy : 0.8208
##
##        'Positive' Class : severe
##
```

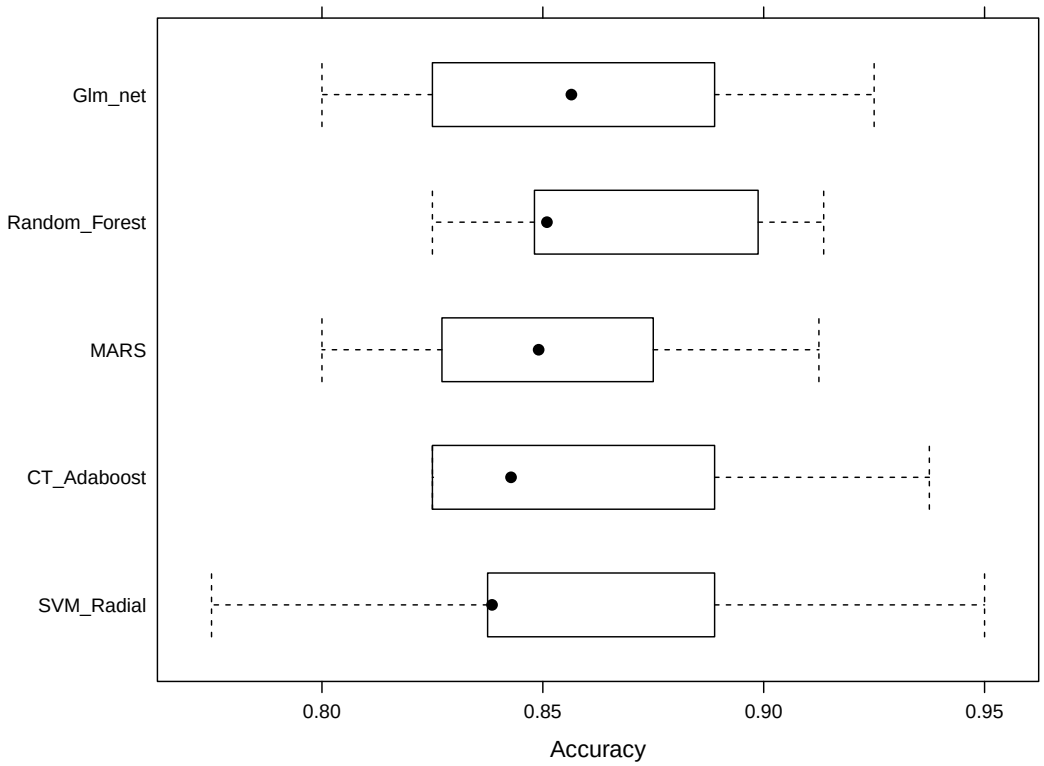
Comparison

```
res <- resamples(
  list(
    Glm_net = glmn.fit,
    MARS = mars.fit,
    SVM_Radial = svmr.fit,
    Random_Forest = rf.fit,
    CT_Adaboost = gbmA.fit
    ))
summary(res)
```

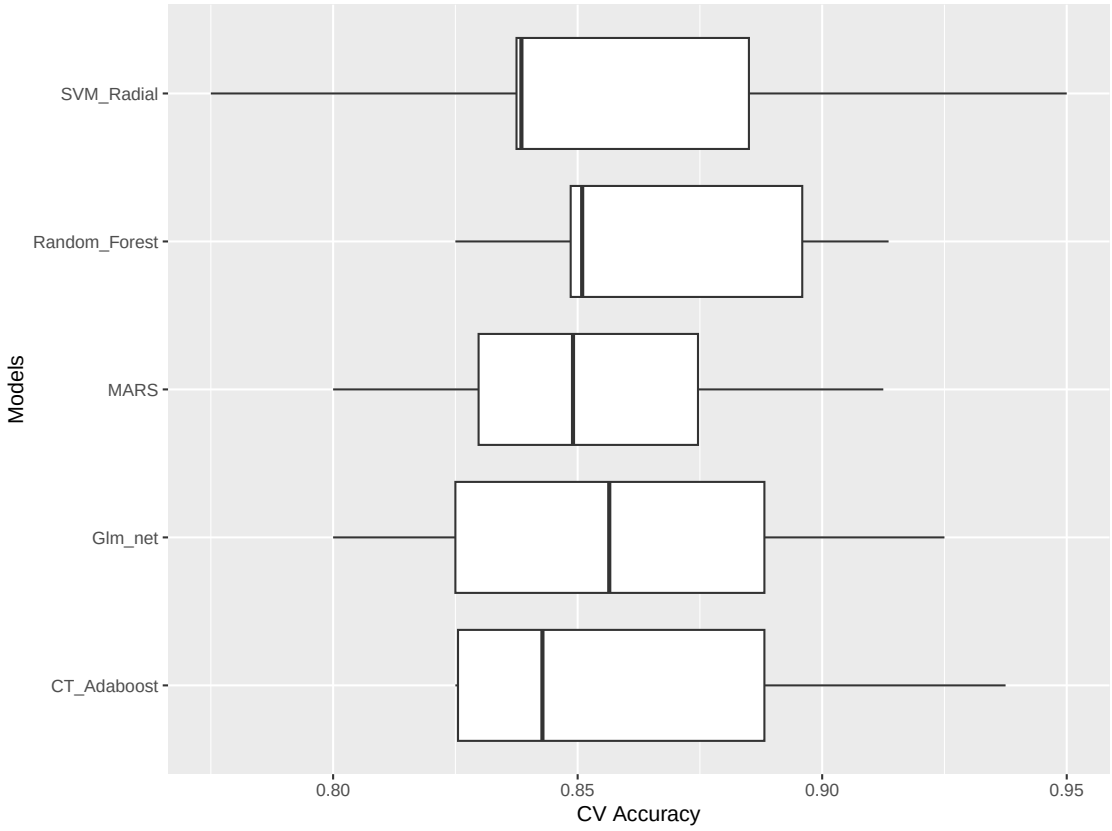
```
##
## Call:
## summary.resamples(object = res)
##
## Models: Glm_net, MARS, SVM_Radial, Random_Forest, CT_Adaboost
## Number of resamples: 10
##
## Accuracy
```

```
##                Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## Glm_net       0.800 0.8250000 0.8564619 0.8575389 0.8881857 0.9250000    0
## MARS          0.800 0.8297454 0.8490506 0.8537414 0.8746044 0.9125000    0
## SVM_Radial    0.775 0.8375000 0.8385031 0.8587256 0.8850211 0.9500000    0
## Random_Forest 0.825 0.8485759 0.8509259 0.8662268 0.8959256 0.9135802    0
## CT_Adaboost   0.825 0.8255401 0.8428006 0.8600227 0.8881857 0.9375000    0
##
## Kappa
##                    Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## Glm_net       0.5626822 0.6184653 0.6835201 0.6882100 0.7629431 0.8352780    0
## MARS          0.5736176 0.6237332 0.6690666 0.6779444 0.7121073 0.8033708    0
## SVM_Radial    0.5272489 0.6463772 0.6603063 0.6944816 0.7513171 0.8885017    0
## Random_Forest 0.5909423 0.6418413 0.6665123 0.6954834 0.7721945 0.8105580    0
## CT_Adaboost   0.5882353 0.6189778 0.6400205 0.6862958 0.7593196 0.8595506    0
```

```
bwplot(res, metric="Accuracy")
```




```
res$values %>%
  dplyr::select(1, ends_with("Accuracy")) %>%
  gather(model, Accuracy, -1) %>%
  mutate(model = sub("~Accuracy", "", model)) %>%
  ggplot() +
  geom_boxplot(aes(x = Accuracy, y = model)) +
  labs(x = "CV Accuracy", y = "Models")
```


SVM_Radial
Random_Forest
MARS
Glm_net
CT_Adaboost
Models
0.80
0.85
0.90
0.95
CV Accuracy