

class33.R

2022-02-25

```
Load data #
tox<-read.csv("C:/Users/m48/Desktop/tox21.txt", header=TRUE, sep = ',', stringsAsFactors = FALSE )

#Scanning for NAs
colSums(is.na(tox))
```

```
##          smiles          NR.AR          NR.AR.LBD          NR.AhR  NR.Aromatase
##           0           566           1073           1282           2010
##          NR.ER          NR.ER.LBD  NR.PPAR.gamma          SR.ARE          SR.ATAD5
##         1638           876           1381           1999           759
##          SR.HSE          SR.MMP          SR.p53
##         1364           2021           1057
```

```
# preprocessing
#removes the first variable(nama) and second variable(IC50) from the data set
x<-tox[,-c(1,2)]

# load IC50
IC50<-tox[,c(2)]

## x+IC50_org
tox2<-cbind(x, IC50)

selected_columns <- c(1:12)
tox2[selected_columns] <- lapply(tox2[selected_columns],
                                function(x)
                                replace(x, is.na(x), min(x, na.rm = T)
                                ))

#Scanning for NAs again
colSums(is.na(tox2))
```

```
##          NR.AR.LBD          NR.AhR  NR.Aromatase          NR.ER          NR.ER.LBD
##           0           0           0           0           0
##  NR.PPAR.gamma          SR.ARE          SR.ATAD5          SR.HSE          SR.MMP
##           0           0           0           0           0
##          SR.p53          IC50
##           0           0
```

```
table(tox2$IC50)
```

```
##
##      0      1
## 7522  309
```

```
# install.packages('caTools')
library(caTools)
#The purpose of this classification exercise is to create a supervised train
ing model that is able to predict FDA_APPROVED based on a CT_TOX
#split the data into three sets: 80% train, 10% valid and 10% test
set.seed(263)
train_ind <- sample(1:nrow(tox2), 0.8 * (nrow(tox2)))
training_set <- tox2[train_ind, ]

data2 <-tox2[-train_ind,]
test_ind <- sample(1:nrow(data2), 0.5 * (nrow(data2)))
test_set <- data2[test_ind, ]
valid_set <- data2[-test_ind, ]

#greate matrix for show "auc" result models
auc_result=matrix(0, nrow =8, ncol =1)

# It is also possible to change names
colnames(auc_result) <- c("auc")
rownames(auc_result) <- c("DT","LR","SVM","RF","NN","KNN","Bag.tree","Boost,
reg")
##### Decision Tree#####
#####

#install.packages("rpart.plot")
library(rpart)

fit <- rpart(IC50~., data = training_set, method = 'class')
#rpart.plot(fit)

predict_unseen <-predict(fit, test_set, type = 'class')

table_mat <- table(test_set$IC50, predict_unseen)
table_mat
```

```
##      predict_unseen
##           0      1
## 0 767      0
## 1   9      7
```

```
#Accuracy = (TP + TN)/(TN + FP + FN + TP)
accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)*100
accuracy_Test
```

```
## [1] 98.85057
```

```
library("ROCR")
Pred.cart = predict(fit, newdata = valid_set, type = "prob")[,2]
Pred2 = prediction(Pred.cart, valid_set$IC50)

x.ct.perf <- (performance(Pred2, "tpr", "fpr"))
# add=TRUE draws on the existing chart
plot(x.ct.perf, col=4, main="ROC curves of different machine learning classi-
fier")

# Draw a legend.
legend(0.7, 0.7, c('DT', 'LR','svm','RF', 'NN'), 4:8)

library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##      cov, smooth, var
```

```
predict2<-predict(fit, valid_set, type = 'class')
roc <- roc(valid_set$IC50, as.numeric(predict2))
```

```
## Setting levels: control = 0, case = 1
```

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

```
auc(roc)
```

```
## Area under the curve: 0.6798
```

```
auc_result[1,1]<- auc(roc)
#####logistic regression #####
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse
1.3.1 --
```

```
## v ggplot2 3.3.5      v purrr  0.3.4
## v tibble  3.1.6      v dplyr  1.0.8
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.1
```

```
## -- Conflicts ----- tidyverse_conflic
ts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
## lift
```

```
#Implementation of Logistic Regression to predict the binary outcome
# Fit the model
logistic_model <- glm( IC50 ~., data = training_set, family = binomial)
# Summarize the model
summary(logistic_model)
```

```
##
## Call:
## glm(formula = IC50 ~ ., family = binomial, data = training_set)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2664  -0.1878  -0.1878  -0.1878   3.6357
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.02894    0.10840 -37.166 < 2e-16 ***
## NR.AR.LBD      4.90835    0.23168  21.186 < 2e-16 ***
## NR.AhR         0.46270    0.24704   1.873 0.061075 .
## NR.Aromatase  -0.73715    0.46204  -1.595 0.110613
## NR.ER          1.24753    0.21206   5.883 4.03e-09 ***
## NR.ER.LBD      0.36148    0.29160   1.240 0.215108
## NR.PPAR.gamma -2.02604    0.62845  -3.224 0.001265 **
## SR.ARE         0.02990    0.24040   0.124 0.901003
## SR.ATAD5       0.33854    0.33097   1.023 0.306376
## SR.HSE        -1.14745    0.50753  -2.261 0.023769 *
## SR.MMP        -0.07506    0.24172  -0.311 0.756152
## SR.p53        -1.46144    0.39784  -3.673 0.000239 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2163.6  on 6263  degrees of freedom
## Residual deviance: 1382.9  on 6252  degrees of freedom
## AIC: 1406.9
##
## Number of Fisher Scoring iterations: 7
```

```
# Predicting in the test dataset
pred_prob <- predict(logistic_model, test_set, type = "response")
A<- as.matrix(pred_prob)

##Evaluating Logistic Regression Model
# Converting from probability to actual output
pred_class <- ifelse(pred_prob >= 0.018, "1", "0")

# Generating the classification table
ctab_test <- table(test_set$IC50, pred_class)
ctab_test
```

```
##      pred_class
##      0      1
## 0 629 138
## 1   3   13
```

```
#Accuracy = (TP + TN)/(TN + FP + FN + TP)
# Accuracy in Test dataset
accuracy_test <- sum(diag(ctab_test))/sum(ctab_test)*100
accuracy_test
```

```
## [1] 81.99234
```

```
p <- predict(logistic_model, newdata=valid_set, type="response")
pr <- prediction(p, valid_set$IC50)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf,col=5, add=TRUE)

auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.7800912
```

```
auc_result[2,1]<- auc
##### SVM #####
require(e1071)
```

```
## Loading required package: e1071
```

```
# svm requires tuning
set.seed(549)
x.svm.tune <- tune(svm, IC50~., data = training_set,
                  ranges = list(gamma = 2^(-8:1), cost = 2^(0:4)),
                  tunecontrol = tune.control(sampling = "fix"))
# display the tuning results (in text format)
x.svm.tune
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: fixed training/validation set
##
## - best parameters:
##   gamma cost
## 0.015625    8
##
## - best performance: 0.02749499
```

```

# I manually copied the cost and gamma from console messages above to parameters below.
x.svm <- svm(IC50~., data = training_set, cost=8, gamma=0.0156, probability = TRUE)

# Predicting in the test dataset
pred_prob1 <- predict(x.svm, test_set, type = "response")
#A<- as.matrix(pred_prob)

##Evaluating Logistic Regression Model
# Converting from probability to actual output
pred_class1 <- ifelse(pred_prob1 >= 0.02, "1", "0")

# Generating the classification table
ctab_test1 <- table(test_set$IC50, pred_class1)
ctab_test1

```

```

##      pred_class1
##           0     1
##    0 734    33
##    1   6    10

```

```

#Accuracy = (TP + TN) / (TN + FP + FN + TP)
# Accuracy in Test dataset
accuracy_test1 <- sum(diag(ctab_test1)) / sum(ctab_test1) * 100
accuracy_test1

```

```
## [1] 95.01916
```

```

library(ROCR)
p1 <- predict(x.svm, newdata=valid_set, type="response")
pr1 <- prediction(p1, valid_set$IC50)
prf1 <- performance(pr1, measure = "tpr", x.measure = "fpr")
plot(prf1,col=6, add=TRUE)

auc <- performance(pr1, measure = "auc")
auc <- auc@y.values[[1]]
auc

```

```
## [1] 0.6983013
```

```

auc_result[3,1]<- auc
##### Random forest#####
##An implementation of the random forest and bagging ensemble algorithms utilizing conditional inference trees as base learners (from party package)
require(party)

```

```
## Loading required package: party
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##  
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':  
##  
##   as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
##  
## Attaching package: 'strucchange'
```

```
## The following object is masked from 'package:stringr':  
##  
##   boundary
```



```
x.cf <- cforest(IC50 ~ ., data=training_set, control = cforest_unbiased(mtry
= 10))

# Predicting in the test dataset
x.cf.pred <- predict(x.cf, newdata=test_set)
#pred_prob2 <- predict(x.cf, test_set, type = "response")
#A<- as.matrix(x.cf.pred)

##Evaluating Logistic Regression Model
# Converting from probability to actual output
pred_class2 <- ifelse(x.cf.pred >= 0.05, "1", "0")

# Generating the classification table
ctab_test2 <- table(test_set$IC50, pred_class2)
ctab_test2
```

```
##      pred_class2
##      0      1
## 0 729   38
## 1    9    7
```

```
#Accuracy = (TP + TN)/(TN + FP + FN + TP)
# Accuracy in Test dataset
accuracy_test2 <- sum(diag(ctab_test2))/sum(ctab_test2)*100
accuracy_test2
```

```
## [1] 93.99745
```

```
library(ROCR)
p2 <- predict(x.cf , newdata=valid_set, type="response")
pr2 <- prediction(p2, valid_set$IC50)
prf2 <- performance(pr2, measure = "tpr", x.measure = "fpr")
plot(prf2,col=7, add=TRUE)

auc <- performance(pr2, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.7300972
```

```
auc_result[4,1]<- auc
##### neural network #####
#####
library(caret)
library(nnet)

IC502 <- factor(training_set$IC50, levels = c(0,1), labels = c("YES", "NO"))
length(IC502 )
```

```
## [1] 6264
```

```
training_set_nn<-training_set[,-12]
training_set_nn$IC50<- IC502

model_nn <- train(
  IC50 ~ ., training_set_nn,
  method = "nnet",
  metric = "AUC",
  trControl = trainControl(
    method = "cv",
    number = 5,
    verboseIter = TRUE,
    classProbs = TRUE,
    summaryFunction = twoClassSummary
  )
)
```

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

```
## + Fold1: size=1, decay=0e+00
## # weights: 14
## initial value 3507.935221
## iter 10 value 704.417127
## iter 20 value 616.385023
## iter 30 value 585.453737
## iter 40 value 585.374019
## iter 50 value 585.328416
## iter 60 value 585.243349
## iter 70 value 585.181827
## iter 80 value 585.031231
## iter 90 value 584.929331
## iter 100 value 578.713631
## final value 578.713631
## stopped after 100 iterations
## - Fold1: size=1, decay=0e+00
## + Fold1: size=3, decay=0e+00
## # weights: 40
## initial value 1506.146440
## iter 10 value 602.502006
## iter 20 value 557.266397
## iter 30 value 531.936709
## iter 40 value 529.055380
## iter 50 value 527.373616
## iter 60 value 525.624928
## iter 70 value 524.446273
## iter 80 value 523.788722
## iter 90 value 523.735181
## iter 100 value 523.728545
## final value 523.728545
## stopped after 100 iterations
## - Fold1: size=3, decay=0e+00
## + Fold1: size=5, decay=0e+00
## # weights: 66
## initial value 2331.873018
## iter 10 value 865.204830
## iter 20 value 741.398802
## iter 30 value 591.754018
## iter 40 value 548.812545
## iter 50 value 540.742847
## iter 60 value 540.724220
## iter 70 value 540.335933
## iter 80 value 539.071925
## iter 90 value 532.894643
## iter 100 value 530.679933
## final value 530.679933
## stopped after 100 iterations
## - Fold1: size=5, decay=0e+00
## + Fold1: size=1, decay=1e-01
## # weights: 14
## initial value 2249.603594
## iter 10 value 673.608060
```

```
## iter 20 value 588.595550
## iter 30 value 564.586019
## iter 40 value 556.087224
## iter 50 value 554.331522
## final value 554.331203
## converged
## - Fold1: size=1, decay=1e-01
## + Fold1: size=3, decay=1e-01
## # weights: 40
## initial value 5646.277874
## iter 10 value 605.901616
## iter 20 value 558.584231
## iter 30 value 547.009647
## iter 40 value 538.969509
## iter 50 value 536.560268
## iter 60 value 535.210844
## iter 70 value 534.169824
## iter 80 value 533.949212
## iter 90 value 533.748380
## iter 100 value 533.671239
## final value 533.671239
## stopped after 100 iterations
## - Fold1: size=3, decay=1e-01
## + Fold1: size=5, decay=1e-01
## # weights: 66
## initial value 2435.514261
## iter 10 value 757.368881
## iter 20 value 584.476208
## iter 30 value 548.607869
## iter 40 value 541.315434
## iter 50 value 537.421254
## iter 60 value 536.051854
## iter 70 value 535.183339
## iter 80 value 534.501418
## iter 90 value 534.092852
## iter 100 value 533.820390
## final value 533.820390
## stopped after 100 iterations
## - Fold1: size=5, decay=1e-01
## + Fold1: size=1, decay=1e-04
## # weights: 14
## initial value 2956.509488
## iter 10 value 693.839284
## iter 20 value 614.315847
## iter 30 value 570.501686
## iter 40 value 568.453485
## iter 50 value 568.356074
## iter 60 value 568.349964
## iter 70 value 568.107399
## iter 80 value 551.430657
## iter 90 value 545.643780
## iter 100 value 540.167954
```

```
## final value 540.167954
## stopped after 100 iterations
## - Fold1: size=1, decay=1e-04
## + Fold1: size=3, decay=1e-04
## # weights: 40
## initial value 4661.229073
## iter 10 value 865.706462
## iter 20 value 864.842323
## iter 30 value 555.573625
## iter 40 value 541.066840
## iter 50 value 540.566495
## iter 60 value 532.667842
## iter 70 value 529.831932
## iter 80 value 529.326642
## iter 90 value 529.226668
## iter 100 value 525.130696
## final value 525.130696
## stopped after 100 iterations
## - Fold1: size=3, decay=1e-04
## + Fold1: size=5, decay=1e-04
## # weights: 66
## initial value 5795.418277
## iter 10 value 865.887307
## iter 20 value 764.230893
## iter 30 value 605.165620
## iter 40 value 554.990174
## iter 50 value 542.044007
## iter 60 value 540.961175
## iter 70 value 539.937345
## iter 80 value 539.906912
## iter 90 value 539.802413
## iter 100 value 539.590290
## final value 539.590290
## stopped after 100 iterations
## - Fold1: size=5, decay=1e-04
## + Fold2: size=1, decay=0e+00
## # weights: 14
## initial value 2289.977737
## iter 10 value 658.144160
## iter 20 value 598.125724
## iter 30 value 592.110969
## iter 40 value 591.009090
## iter 50 value 590.339254
## iter 60 value 589.409879
## iter 70 value 572.518815
## iter 80 value 571.580705
## iter 90 value 571.273874
## iter 100 value 571.242744
## final value 571.242744
## stopped after 100 iterations
## - Fold2: size=1, decay=0e+00
## + Fold2: size=3, decay=0e+00
```

```
## # weights: 40
## initial value 4657.241512
## final value 865.489840
## converged
## - Fold2: size=3, decay=0e+00
## + Fold2: size=5, decay=0e+00
## # weights: 66
## initial value 1432.598355
## iter 10 value 722.915347
## iter 20 value 546.527177
## iter 30 value 527.269226
## iter 40 value 521.272768
## iter 50 value 518.156242
## iter 60 value 515.694583
## iter 70 value 511.668135
## iter 80 value 508.414714
## iter 90 value 507.328776
## iter 100 value 506.672119
## final value 506.672119
## stopped after 100 iterations
## - Fold2: size=5, decay=0e+00
## + Fold2: size=1, decay=1e-01
## # weights: 14
## initial value 3539.362243
## iter 10 value 707.269515
## iter 20 value 617.586760
## iter 30 value 595.683119
## iter 40 value 572.683325
## iter 50 value 563.444369
## iter 60 value 560.496836
## final value 560.492153
## converged
## - Fold2: size=1, decay=1e-01
## + Fold2: size=3, decay=1e-01
## # weights: 40
## initial value 4832.474233
## iter 10 value 925.364247
## iter 20 value 615.197762
## iter 30 value 569.474503
## iter 40 value 554.510148
## iter 50 value 551.566777
## iter 60 value 546.440588
## iter 70 value 544.849796
## iter 80 value 544.271170
## iter 90 value 543.227844
## iter 100 value 543.143726
## final value 543.143726
## stopped after 100 iterations
## - Fold2: size=3, decay=1e-01
## + Fold2: size=5, decay=1e-01
## # weights: 66
## initial value 2322.691270
```

```
## iter 10 value 606.830121
## iter 20 value 564.628849
## iter 30 value 545.209060
## iter 40 value 540.225758
## iter 50 value 535.231679
## iter 60 value 532.845022
## iter 70 value 532.098501
## iter 80 value 530.715240
## iter 90 value 529.560671
## iter 100 value 529.128601
## final value 529.128601
## stopped after 100 iterations
## - Fold2: size=5, decay=1e-01
## + Fold2: size=1, decay=1e-04
## # weights: 14
## initial value 2692.308115
## final value 865.522502
## converged
## - Fold2: size=1, decay=1e-04
## + Fold2: size=3, decay=1e-04
## # weights: 40
## initial value 2266.041453
## iter 10 value 694.414275
## iter 20 value 544.587437
## iter 30 value 528.961213
## iter 40 value 523.464598
## iter 50 value 519.866861
## iter 60 value 519.158258
## iter 70 value 514.803244
## iter 80 value 509.027078
## iter 90 value 504.599805
## iter 100 value 503.996776
## final value 503.996776
## stopped after 100 iterations
## - Fold2: size=3, decay=1e-04
## + Fold2: size=5, decay=1e-04
## # weights: 66
## initial value 7571.273723
## iter 10 value 593.006421
## iter 20 value 546.361700
## iter 30 value 522.622504
## iter 40 value 512.626226
## iter 50 value 510.870279
## iter 60 value 507.394293
## iter 70 value 506.083126
## iter 80 value 505.398386
## iter 90 value 504.828478
## iter 100 value 504.050433
## final value 504.050433
## stopped after 100 iterations
## - Fold2: size=5, decay=1e-04
## + Fold3: size=1, decay=0e+00
```

```
## # weights: 14
## initial value 2904.518565
## iter 10 value 589.538444
## iter 20 value 573.154571
## iter 30 value 569.617924
## iter 40 value 566.622834
## iter 50 value 565.356524
## iter 60 value 556.657274
## iter 70 value 552.761853
## iter 80 value 552.726071
## iter 90 value 552.583826
## iter 100 value 552.135604
## final value 552.135604
## stopped after 100 iterations
## - Fold3: size=1, decay=0e+00
## + Fold3: size=3, decay=0e+00
## # weights: 40
## initial value 2926.749757
## final value 865.447457
## converged
## - Fold3: size=3, decay=0e+00
## + Fold3: size=5, decay=0e+00
## # weights: 66
## initial value 3548.464859
## iter 10 value 640.520223
## iter 20 value 519.536355
## iter 30 value 489.865066
## iter 40 value 481.570026
## iter 50 value 478.712372
## iter 60 value 475.301201
## iter 70 value 473.405434
## iter 80 value 472.538064
## iter 90 value 472.287496
## iter 100 value 472.230722
## final value 472.230722
## stopped after 100 iterations
## - Fold3: size=5, decay=0e+00
## + Fold3: size=1, decay=1e-01
## # weights: 14
## initial value 3234.491432
## iter 10 value 675.768693
## iter 20 value 562.376656
## iter 30 value 553.422130
## iter 40 value 553.408421
## final value 553.408391
## converged
## - Fold3: size=1, decay=1e-01
## + Fold3: size=3, decay=1e-01
## # weights: 40
## initial value 3326.064872
## iter 10 value 886.931564
## iter 20 value 673.511089
```



```
## iter 30 value 524.852202
## iter 40 value 520.919828
## iter 50 value 519.002691
## iter 60 value 518.620787
## iter 70 value 518.558651
## iter 80 value 518.543510
## final value 518.543180
## converged
## - Fold3: size=3, decay=1e-01
## + Fold3: size=5, decay=1e-01
## # weights: 66
## initial value 2784.833392
## iter 10 value 918.182783
## iter 20 value 625.107569
## iter 30 value 557.075056
## iter 40 value 526.810044
## iter 50 value 522.010834
## iter 60 value 521.161875
## iter 70 value 519.674622
## iter 80 value 519.148942
## iter 90 value 519.001930
## iter 100 value 518.848878
## final value 518.848878
## stopped after 100 iterations
## - Fold3: size=5, decay=1e-01
## + Fold3: size=1, decay=1e-04
## # weights: 14
## initial value 5591.365922
## final value 865.469980
## converged
## - Fold3: size=1, decay=1e-04
## + Fold3: size=3, decay=1e-04
## # weights: 40
## initial value 6715.843627
## iter 10 value 865.879726
## iter 20 value 822.572620
## iter 30 value 589.927140
## iter 40 value 567.374700
## iter 50 value 541.240393
## iter 60 value 540.503690
## iter 70 value 539.767787
## iter 80 value 539.698757
## iter 90 value 539.416617
## iter 100 value 534.348461
## final value 534.348461
## stopped after 100 iterations
## - Fold3: size=3, decay=1e-04
## + Fold3: size=5, decay=1e-04
## # weights: 66
## initial value 4477.094584
## iter 10 value 865.815424
## iter 20 value 865.478649
```

```
## iter 30 value 589.227255
## iter 40 value 539.827559
## iter 50 value 518.548794
## iter 60 value 513.130318
## iter 70 value 510.074806
## iter 80 value 505.491625
## iter 90 value 503.043985
## iter 100 value 497.200251
## final value 497.200251
## stopped after 100 iterations
## - Fold3: size=5, decay=1e-04
## + Fold4: size=1, decay=0e+00
## # weights: 14
## initial value 2208.210125
## iter 10 value 594.410536
## iter 20 value 572.241519
## iter 30 value 572.169930
## iter 40 value 572.165238
## iter 50 value 572.153866
## final value 572.149494
## converged
## - Fold4: size=1, decay=0e+00
## + Fold4: size=3, decay=0e+00
## # weights: 40
## initial value 2435.435507
## iter 10 value 606.238622
## iter 20 value 517.537292
## iter 30 value 506.685575
## iter 40 value 502.542602
## iter 50 value 498.443210
## iter 60 value 496.325381
## iter 70 value 495.497363
## iter 80 value 495.481749
## final value 495.481725
## converged
## - Fold4: size=3, decay=0e+00
## + Fold4: size=5, decay=0e+00
## # weights: 66
## initial value 2645.102864
## iter 10 value 863.551912
## iter 20 value 562.575375
## iter 30 value 538.767890
## iter 40 value 534.989002
## iter 50 value 533.527667
## iter 60 value 511.391037
## iter 70 value 505.882843
## iter 80 value 505.115517
## iter 90 value 504.858189
## iter 100 value 504.645095
## final value 504.645095
## stopped after 100 iterations
## - Fold4: size=5, decay=0e+00
```

```
## + Fold4: size=1, decay=1e-01
## # weights: 14
## initial value 4198.918553
## iter 10 value 697.269817
## iter 20 value 565.350519
## iter 30 value 549.779215
## iter 40 value 548.737847
## iter 50 value 548.631222
## final value 548.631105
## converged
## - Fold4: size=1, decay=1e-01
## + Fold4: size=3, decay=1e-01
## # weights: 40
## initial value 3251.482463
## iter 10 value 898.032588
## iter 20 value 752.424035
## iter 30 value 545.574569
## iter 40 value 527.559342
## iter 50 value 525.494855
## iter 60 value 518.448911
## iter 70 value 514.476467
## iter 80 value 513.511865
## iter 90 value 511.721692
## iter 100 value 511.006251
## final value 511.006251
## stopped after 100 iterations
## - Fold4: size=3, decay=1e-01
## + Fold4: size=5, decay=1e-01
## # weights: 66
## initial value 4851.714185
## iter 10 value 739.148643
## iter 20 value 550.766105
## iter 30 value 525.994739
## iter 40 value 517.168853
## iter 50 value 515.249037
## iter 60 value 513.547917
## iter 70 value 510.753793
## iter 80 value 509.140150
## iter 90 value 508.279346
## iter 100 value 507.752194
## final value 507.752194
## stopped after 100 iterations
## - Fold4: size=5, decay=1e-01
## + Fold4: size=1, decay=1e-04
## # weights: 14
## initial value 4604.153695
## final value 865.500562
## converged
## - Fold4: size=1, decay=1e-04
## + Fold4: size=3, decay=1e-04
## # weights: 40
## initial value 4875.614995
```

```
## iter 10 value 865.597041
## iter 20 value 855.167292
## iter 30 value 548.198988
## iter 40 value 537.683734
## iter 50 value 526.414329
## iter 60 value 510.378048
## iter 70 value 507.495389
## iter 80 value 506.459642
## iter 90 value 505.915556
## iter 100 value 505.266689
## final value 505.266689
## stopped after 100 iterations
## - Fold4: size=3, decay=1e-04
## + Fold4: size=5, decay=1e-04
## # weights: 66
## initial value 2970.475681
## iter 10 value 865.596300
## iter 20 value 553.409199
## iter 30 value 540.432085
## iter 40 value 539.108834
## iter 50 value 535.837708
## iter 60 value 534.829017
## iter 70 value 534.569776
## iter 80 value 512.010613
## iter 90 value 508.961659
## iter 100 value 507.607066
## final value 507.607066
## stopped after 100 iterations
## - Fold4: size=5, decay=1e-04
## + Fold5: size=1, decay=0e+00
## # weights: 14
## initial value 4856.591857
## final value 865.447457
## converged
## - Fold5: size=1, decay=0e+00
## + Fold5: size=3, decay=0e+00
## # weights: 40
## initial value 3950.065338
## final value 865.447457
## converged
## - Fold5: size=3, decay=0e+00
## + Fold5: size=5, decay=0e+00
## # weights: 66
## initial value 6738.061783
## iter 10 value 534.577424
## iter 20 value 503.755877
## iter 30 value 488.244769
## iter 40 value 479.087502
## iter 50 value 471.824141
## iter 60 value 467.839077
## iter 70 value 465.884295
## iter 80 value 465.344097
```

```
## iter 90 value 464.832178
## iter 100 value 464.428472
## final value 464.428472
## stopped after 100 iterations
## - Fold5: size=5, decay=0e+00
## + Fold5: size=1, decay=1e-01
## # weights: 14
## initial value 2900.928887
## iter 10 value 666.428869
## iter 20 value 583.745268
## iter 30 value 565.147286
## iter 40 value 541.755610
## iter 50 value 534.679816
## final value 534.147167
## converged
## - Fold5: size=1, decay=1e-01
## + Fold5: size=3, decay=1e-01
## # weights: 40
## initial value 3881.438252
## iter 10 value 656.229939
## iter 20 value 528.751782
## iter 30 value 511.825295
## iter 40 value 507.398156
## iter 50 value 505.573979
## iter 60 value 504.042011
## iter 70 value 503.622929
## iter 80 value 502.327746
## iter 90 value 502.092235
## iter 100 value 502.085388
## final value 502.085388
## stopped after 100 iterations
## - Fold5: size=3, decay=1e-01
## + Fold5: size=5, decay=1e-01
## # weights: 66
## initial value 2123.119032
## iter 10 value 637.060566
## iter 20 value 542.289541
## iter 30 value 513.031751
## iter 40 value 507.662071
## iter 50 value 503.994935
## iter 60 value 498.999214
## iter 70 value 495.430377
## iter 80 value 494.726033
## iter 90 value 494.389528
## iter 100 value 494.262783
## final value 494.262783
## stopped after 100 iterations
## - Fold5: size=5, decay=1e-01
## + Fold5: size=1, decay=1e-04
## # weights: 14
## initial value 4128.290547
## iter 10 value 711.083685
```

```
## iter 20 value 612.096527
## iter 30 value 604.123398
## iter 40 value 583.492779
## iter 50 value 540.567506
## iter 60 value 533.220607
## iter 70 value 533.043700
## iter 80 value 533.042107
## final value 533.031247
## converged
## - Fold5: size=1, decay=1e-04
## + Fold5: size=3, decay=1e-04
## # weights: 40
## initial value 3250.410041
## iter 10 value 865.557329
## iter 20 value 863.273679
## iter 30 value 557.767656
## iter 40 value 522.836950
## iter 50 value 521.717198
## iter 60 value 520.719559
## iter 70 value 520.488254
## iter 80 value 520.463259
## iter 90 value 520.442734
## iter 100 value 520.428808
## final value 520.428808
## stopped after 100 iterations
## - Fold5: size=3, decay=1e-04
## + Fold5: size=5, decay=1e-04
## # weights: 66
## initial value 6077.359187
## iter 10 value 877.111104
## iter 20 value 865.497840
## iter 30 value 829.542642
## iter 40 value 513.963085
## iter 50 value 499.709309
## iter 60 value 495.646600
## iter 70 value 487.222173
## iter 80 value 486.531787
## iter 90 value 485.952746
## iter 100 value 484.727154
## final value 484.727154
## stopped after 100 iterations
## - Fold5: size=5, decay=1e-04
## Aggregating results
## Selecting tuning parameters
## Fitting size = 5, decay = 0.1 on full training set
## # weights: 66
## initial value 4799.099913
## iter 10 value 1107.227293
## iter 20 value 829.691188
## iter 30 value 710.924780
## iter 40 value 674.265694
## iter 50 value 662.809712
```

```
## iter 60 value 660.334233
## iter 70 value 656.847226
## iter 80 value 652.366330
## iter 90 value 650.268718
## iter 100 value 646.538393
## final value 646.538393
## stopped after 100 iterations
```

```
model_nn
```

```
## Neural Network
##
## 6264 samples
## 11 predictor
## 2 classes: 'YES', 'NO'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 5011, 5012, 5011, 5011, 5011
## Resampling results across tuning parameters:
##
## size decay ROC Sens Spec
## 1 0e+00 0.7337533 0.9921725 0.4038462
## 1 1e-04 0.7203773 0.9976686 0.1884615
## 1 1e-01 0.7550847 0.9965025 0.4730769
## 3 0e+00 0.6646860 0.9980017 0.2000000
## 3 1e-04 0.7976115 0.9961694 0.4692308
## 3 1e-01 0.7832002 0.9971686 0.4846154
## 5 0e+00 0.7684545 0.9961693 0.4692308
## 5 1e-04 0.7948068 0.9960028 0.4807692
## 5 1e-01 0.7978331 0.9971686 0.4846154
##
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were size = 5 and decay = 0.1.
```

```
nnprediction <- predict(model_nn, test_set)
IC503 <- factor(test_set$IC50, levels = c(0,1), labels = c("YES", "NO"))

length(IC503 )
```

```
## [1] 783
```

```
test_set$IC50<- IC503
cmnn <-confusionMatrix(nnprediction,test_set$IC50)
print(cmnn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction YES  NO
##           YES 766   9
##           NO   1   7
##
##           Accuracy : 0.9872
##           95% CI : (0.9766, 0.9939)
##           No Information Rate : 0.9796
##           P-Value [Acc > NIR] : 0.07530
##
##           Kappa : 0.5776
##
##  McNemar's Test P-Value : 0.02686
##
##           Sensitivity : 0.9987
##           Specificity : 0.4375
##           Pos Pred Value : 0.9884
##           Neg Pred Value : 0.8750
##           Prevalence : 0.9796
##           Detection Rate : 0.9783
##           Detection Prevalence : 0.9898
##           Balanced Accuracy : 0.7181
##
##           'Positive' Class : YES
##
```

```
# Predicting in the test dataset
x.nn.pred <- predict(model_nn, newdata=test_set)
#pred_prob2 <- predict(x.cf, test_set, type = "response")

# Generating the classification table
ctab_test3 <- table(test_set$IC50, x.nn.pred)
ctab_test3
```

```
##           x.nn.pred
##           YES  NO
## YES 766   1
## NO   9   7
```

```
#Accuracy = (TP + TN)/(TN + FP + FN + TP)
# Accuracy in Test dataset
accuracy_test3 <- sum(diag(ctab_test3))/sum(ctab_test3)*100
accuracy_test3
```

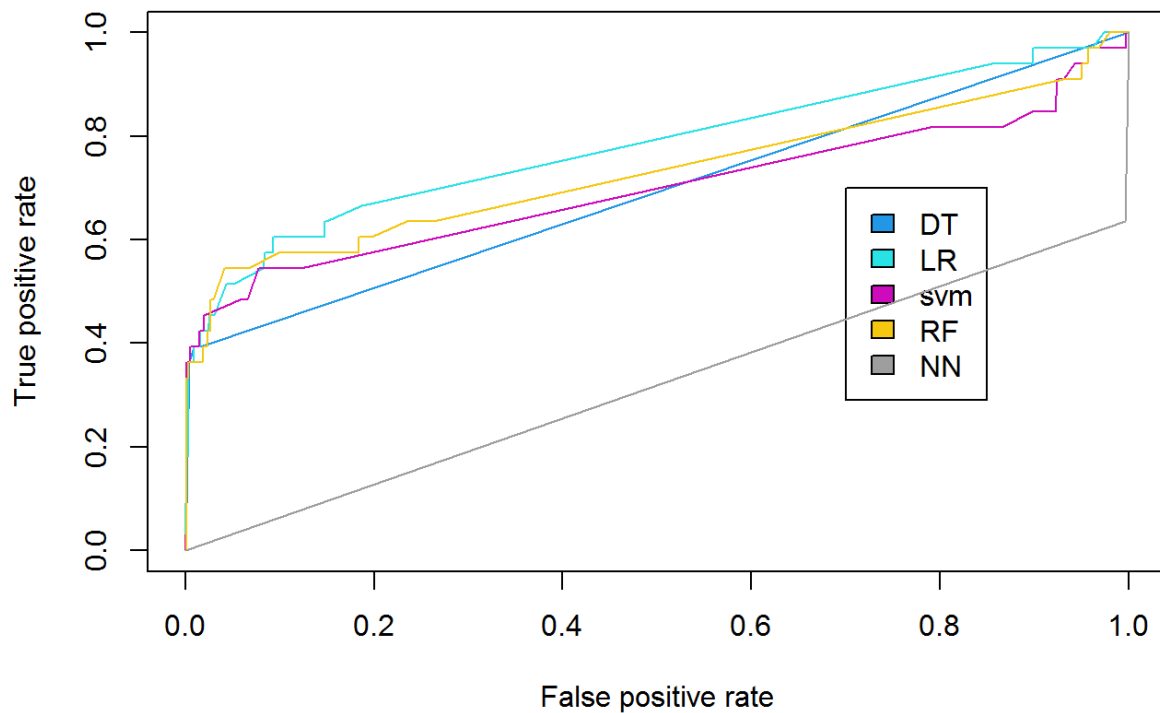
```
## [1] 98.72286
```



```
library(ROCR)
p3 <- predict(model_nn , newdata=valid_set)
p3 <- ifelse(p3=="YES",1,0)
pr3 <- prediction(p3, valid_set$IC50)

prf3 <- performance(pr3, measure = "tpr", x.measure = "fpr")
plot(prf3,col=8, add=TRUE)
```

ROC curves of different machine learning classifier



```
auc <- performance(pr3, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.3201792
```

```

auc_result[5,1]<- auc
##### KNN #####
##
training_set_knn<-training_set[,-12]
random.stuff <- matrix(runif(prod(dim(training_set_knn))), min = -0.00001, max = 0.0001), nrow = nrow(training_set_knn))

training_set_knn<-random.stuff + training_set_knn
training_set_knn<-cbind(training_set_knn,training_set$IC50)
names(training_set_knn)[12] <- 'IC50'

test_set_knn<-test_set[,-12]
random.stuff <- matrix(runif(prod(dim(test_set_knn))), min = -0.00001, max = 0.0001), nrow = nrow(test_set_knn))

test_set_knn<-random.stuff + test_set_knn
test_set_knn<-cbind(test_set_knn,test_set$IC50)
names(test_set_knn)[12] <- 'IC50'

valid_set_knn<-valid_set[,-12]
random.stuff <- matrix(runif(prod(dim(valid_set_knn))), min = -0.00001, max = 0.0001), nrow = nrow(valid_set_knn))

valid_set_knn<-random.stuff + valid_set_knn
valid_set_knn<-cbind(valid_set_knn,valid_set$IC50)
names(valid_set_knn)[12] <- 'IC50'

set.seed(123)
modelknn <- train(IC50~., data=training_set_knn,
                  method="knn",
                  tuneGrid=expand.grid(k=1:30))

```

```

## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to
do
## classification? If so, use a 2 level factor as your outcome column.

```

```
modelknn
```

```
## k-Nearest Neighbors
##
## 6264 samples
## 11 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 6264, 6264, 6264, 6264, 6264, 6264, ...
## Resampling results across tuning parameters:
##
##  k    RMSE      Rsquared    MAE
##  1  0.2123263  0.1908009  0.04597478
##  2  0.1979781  0.2242915  0.04602851
##  3  0.1877339  0.2531949  0.04606426
##  4  0.1808015  0.2759863  0.04612398
##  5  0.1760679  0.2939133  0.04612556
##  6  0.1729592  0.3057748  0.04617150
##  7  0.1703744  0.3169711  0.04614797
##  8  0.1689529  0.3224507  0.04625231
##  9  0.1676121  0.3283719  0.04632976
## 10  0.1664854  0.3338939  0.04632333
## 11  0.1658753  0.3362048  0.04640047
## 12  0.1654871  0.3375264  0.04655303
## 13  0.1648802  0.3410174  0.04650804
## 14  0.1646065  0.3425345  0.04659687
## 15  0.1642068  0.3448990  0.04660300
## 16  0.1640006  0.3458292  0.04669407
## 17  0.1637608  0.3473168  0.04671182
## 18  0.1635508  0.3484860  0.04677429
## 19  0.1635137  0.3478012  0.04695727
## 20  0.1634356  0.3480605  0.04702303
## 21  0.1634752  0.3472799  0.04714666
## 22  0.1634696  0.3469736  0.04729082
## 23  0.1634888  0.3465941  0.04739699
## 24  0.1635068  0.3463631  0.04751104
## 25  0.1635573  0.3458982  0.04763453
## 26  0.1636228  0.3454226  0.04769881
## 27  0.1636741  0.3450847  0.04779409
## 28  0.1637305  0.3449132  0.04789769
## 29  0.1638726  0.3442399  0.04807756
## 30  0.1639728  0.3437315  0.04823217
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 20.
```

```
#####

x.knn.pred <- predict(modelknn, newdata=test_set_knn)
#pred_prob2 <- predict(x.cf, test_set, type = "response")
#A<- as.matrix(x.knn.pred)

##Evaluating Logistic Regression Model
# Converting from probability to actual output
pred_class4 <- ifelse(x.knn.pred >= 0.05, "1", "0")

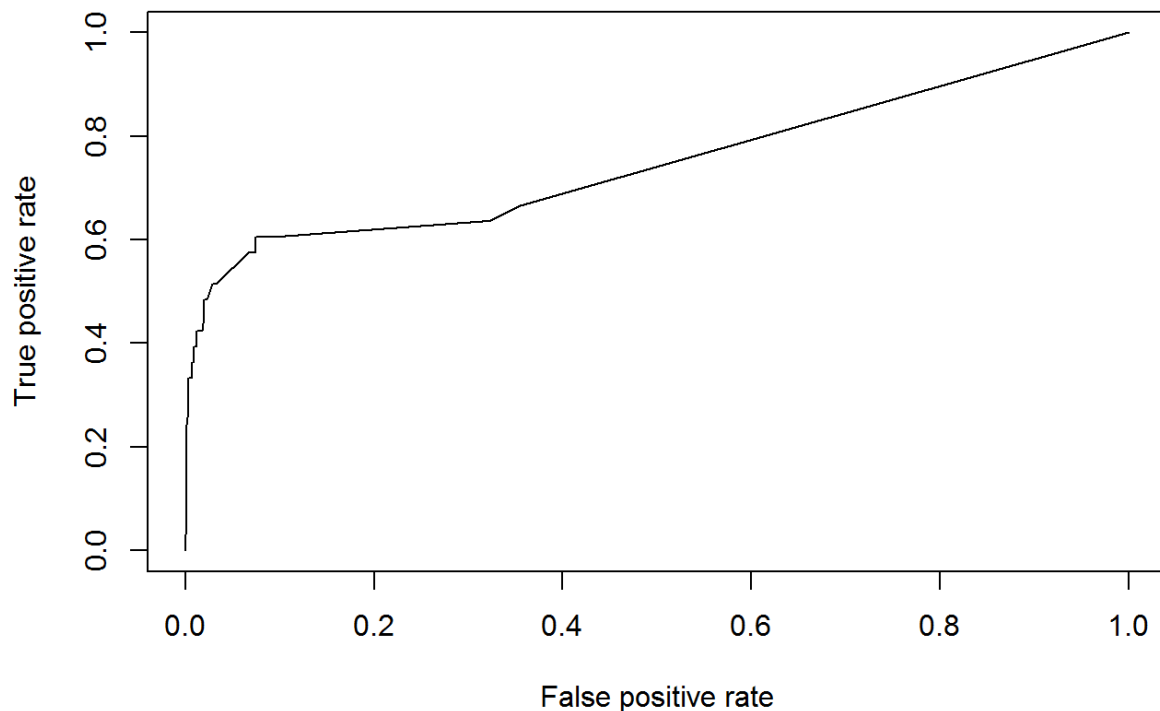
# Generating the classification table
ctab_test4 <- table(test_set_knn$IC50, pred_class4)
ctab_test4
```

```
##      pred_class4
##      0      1
## YES 534 233
## NO   5   11
```

```
#Accuracy = (TP + TN)/(TN + FP + FN + TP)
# Accuracy in Test dataset
accuracy_test4 <- sum(diag(ctab_test4))/sum(ctab_test4)*100
accuracy_test4
```

```
## [1] 69.60409
```

```
library(ROCR)
p4 <- predict(modelknn , newdata=valid_set_knn)
pr4 <- prediction(p4, valid_set_knn$IC50)
prf4 <- performance(pr4, measure = "tpr", x.measure = "fpr")
plot(prf4)
```



```
auc <- performance(pr4, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.7493847
```

```
auc_result[6,1]<- auc
##### Bagged #####
control = trainControl(method = 'repeatedcv',
                        number = 10,
                        repeats = 3,
                        verbose=TRUE)

# Build a Bagged CART model
```

```
set.seed(7)
mod_Treebag = train(IC50 ~., data=training_set, method='treebag', trControl
=control)
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying t
o do
## classification? If so, use a 2 level factor as your outcome column.
```

```
## + Fold01.Rep1: parameter=none
## - Fold01.Rep1: parameter=none
## + Fold02.Rep1: parameter=none
## - Fold02.Rep1: parameter=none
## + Fold03.Rep1: parameter=none
## - Fold03.Rep1: parameter=none
## + Fold04.Rep1: parameter=none
## - Fold04.Rep1: parameter=none
## + Fold05.Rep1: parameter=none
## - Fold05.Rep1: parameter=none
## + Fold06.Rep1: parameter=none
## - Fold06.Rep1: parameter=none
## + Fold07.Rep1: parameter=none
## - Fold07.Rep1: parameter=none
## + Fold08.Rep1: parameter=none
## - Fold08.Rep1: parameter=none
## + Fold09.Rep1: parameter=none
## - Fold09.Rep1: parameter=none
## + Fold10.Rep1: parameter=none
## - Fold10.Rep1: parameter=none
## + Fold01.Rep2: parameter=none
## - Fold01.Rep2: parameter=none
## + Fold02.Rep2: parameter=none
## - Fold02.Rep2: parameter=none
## + Fold03.Rep2: parameter=none
## - Fold03.Rep2: parameter=none
## + Fold04.Rep2: parameter=none
## - Fold04.Rep2: parameter=none
## + Fold05.Rep2: parameter=none
## - Fold05.Rep2: parameter=none
## + Fold06.Rep2: parameter=none
## - Fold06.Rep2: parameter=none
## + Fold07.Rep2: parameter=none
## - Fold07.Rep2: parameter=none
## + Fold08.Rep2: parameter=none
## - Fold08.Rep2: parameter=none
## + Fold09.Rep2: parameter=none
## - Fold09.Rep2: parameter=none
## + Fold10.Rep2: parameter=none
## - Fold10.Rep2: parameter=none
## + Fold01.Rep3: parameter=none
## - Fold01.Rep3: parameter=none
## + Fold02.Rep3: parameter=none
## - Fold02.Rep3: parameter=none
## + Fold03.Rep3: parameter=none
## - Fold03.Rep3: parameter=none
## + Fold04.Rep3: parameter=none
## - Fold04.Rep3: parameter=none
## + Fold05.Rep3: parameter=none
## - Fold05.Rep3: parameter=none
## + Fold06.Rep3: parameter=none
## - Fold06.Rep3: parameter=none
```

```
## + Fold07.Rep3: parameter=none
## - Fold07.Rep3: parameter=none
## + Fold08.Rep3: parameter=none
## - Fold08.Rep3: parameter=none
## + Fold09.Rep3: parameter=none
## - Fold09.Rep3: parameter=none
## + Fold10.Rep3: parameter=none
## - Fold10.Rep3: parameter=none
## Aggregating results
## Fitting final model on full training set
```

```
# Predicting in the test dataset
x.bag.pred <- predict(mod_Treebag, newdata=test_set)
#A<-as.matrix(x.bag.pred)
#pred_prob2 <- predict(x.cf, test_set, type = "response")

##Evaluating Logistic Regression Model
# Converting from probability to actual output
pred_class5 <- ifelse(x.bag.pred >= 0.03, "1", "0")

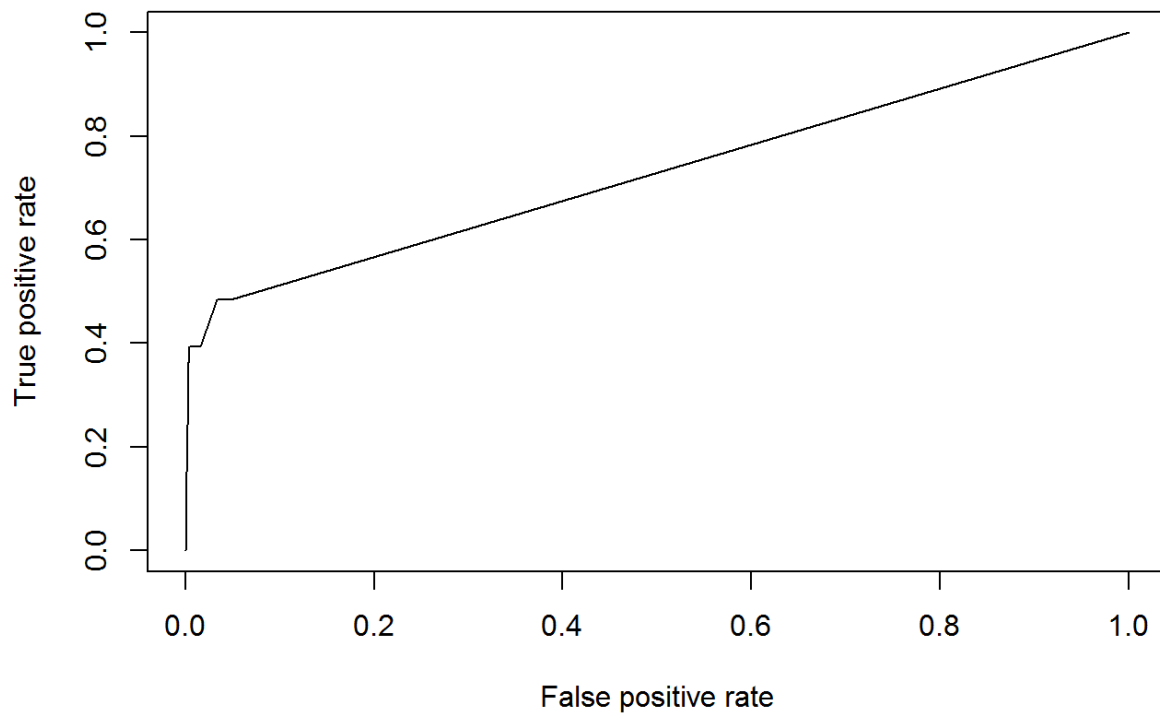
# Generating the classification table
ctab_test5 <- table(test_set$IC50, pred_class5)
ctab_test5
```

```
##      pred_class5
##      0      1
## YES 740    27
## NO   9      7
```

```
#Accuracy = (TP + TN)/(TN + FP + FN + TP)
# Accuracy in Test dataset
accuracy_test5 <- sum(diag(ctab_test5))/sum(ctab_test5)*100
accuracy_test5
```

```
## [1] 95.4023
```

```
library(ROCR)
p5 <- predict(mod_Treebag, newdata=valid_set_knn)
pr5 <- prediction(p5, valid_set$IC50)
prf5 <- performance(pr5, measure = "tpr", x.measure = "fpr")
plot(prf5)
```



```
auc <- performance(pr4, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.7493847
```

```
auc_result[7,1]<- auc
##### Boost #####

set.seed(89)
mod_GBM = train(IC50 ~., data=training_set, method='gbm', trControl=control)
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying t
o do
## classification? If so, use a 2 level factor as your outcome column.
```



```
## + Fold01.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0378	nan	0.1000	0.0023
## 2	0.0358	nan	0.1000	0.0020
## 3	0.0342	nan	0.1000	0.0014
## 4	0.0328	nan	0.1000	0.0010
## 5	0.0317	nan	0.1000	0.0010
## 6	0.0309	nan	0.1000	0.0008
## 7	0.0302	nan	0.1000	0.0006
## 8	0.0297	nan	0.1000	0.0005
## 9	0.0293	nan	0.1000	0.0004
## 10	0.0289	nan	0.1000	0.0003
## 20	0.0277	nan	0.1000	0.0001
## 40	0.0269	nan	0.1000	0.0000
## 60	0.0267	nan	0.1000	-0.0000
## 80	0.0266	nan	0.1000	-0.0000
## 100	0.0266	nan	0.1000	-0.0000
## 120	0.0266	nan	0.1000	-0.0000
## 140	0.0266	nan	0.1000	-0.0000
## 150	0.0266	nan	0.1000	-0.0000

```
##
## - Fold01.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
```

```
## + Fold01.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0374	nan	0.1000	0.0030
## 2	0.0351	nan	0.1000	0.0023
## 3	0.0332	nan	0.1000	0.0019
## 4	0.0316	nan	0.1000	0.0015
## 5	0.0303	nan	0.1000	0.0012
## 6	0.0292	nan	0.1000	0.0010
## 7	0.0283	nan	0.1000	0.0009
## 8	0.0275	nan	0.1000	0.0007
## 9	0.0269	nan	0.1000	0.0006
## 10	0.0265	nan	0.1000	0.0005
## 20	0.0242	nan	0.1000	0.0001
## 40	0.0233	nan	0.1000	0.0000
## 60	0.0231	nan	0.1000	-0.0000
## 80	0.0229	nan	0.1000	-0.0000
## 100	0.0229	nan	0.1000	-0.0000
## 120	0.0228	nan	0.1000	-0.0000
## 140	0.0228	nan	0.1000	-0.0000
## 150	0.0228	nan	0.1000	-0.0000

```
##
## - Fold01.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

```
## + Fold01.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0371	nan	0.1000	0.0024

```
##      2      0.0347      nan      0.1000      0.0024
##      3      0.0326      nan      0.1000      0.0022
##      4      0.0310      nan      0.1000      0.0014
##      5      0.0297      nan      0.1000      0.0012
##      6      0.0285      nan      0.1000      0.0010
##      7      0.0276      nan      0.1000      0.0008
##      8      0.0269      nan      0.1000      0.0007
##      9      0.0263      nan      0.1000      0.0006
##     10      0.0258      nan      0.1000      0.0004
##     20      0.0236      nan      0.1000      0.0001
##     40      0.0229      nan      0.1000     -0.0000
##     60      0.0227      nan      0.1000      0.0000
##     80      0.0226      nan      0.1000     -0.0000
##    100      0.0225      nan      0.1000     -0.0000
##    120      0.0225      nan      0.1000     -0.0000
##    140      0.0224      nan      0.1000     -0.0000
##    150      0.0224      nan      0.1000     -0.0000
##
## - Fold01.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold02.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0376      nan      0.1000      0.0026
##      2      0.0358      nan      0.1000      0.0020
##      3      0.0341      nan      0.1000      0.0017
##      4      0.0327      nan      0.1000      0.0014
##      5      0.0314      nan      0.1000      0.0010
##      6      0.0306      nan      0.1000      0.0008
##      7      0.0298      nan      0.1000      0.0007
##      8      0.0293      nan      0.1000      0.0006
##      9      0.0289      nan      0.1000      0.0005
##     10      0.0285      nan      0.1000      0.0004
##     20      0.0269      nan      0.1000      0.0001
##     40      0.0261      nan      0.1000      0.0000
##     60      0.0259      nan      0.1000     -0.0000
##     80      0.0259      nan      0.1000     -0.0000
##    100      0.0259      nan      0.1000     -0.0000
##    120      0.0259      nan      0.1000     -0.0000
##    140      0.0259      nan      0.1000     -0.0000
##    150      0.0259      nan      0.1000     -0.0000
##
## - Fold02.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold02.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0370      nan      0.1000      0.0029
##      2      0.0345      nan      0.1000      0.0022
##      3      0.0324      nan      0.1000      0.0018
##      4      0.0308      nan      0.1000      0.0015
##      5      0.0295      nan      0.1000      0.0015
```

```
##      6      0.0284      nan      0.1000      0.0012
##      7      0.0276      nan      0.1000      0.0009
##      8      0.0268      nan      0.1000      0.0008
##      9      0.0263      nan      0.1000      0.0006
##     10      0.0258      nan      0.1000      0.0004
##     20      0.0235      nan      0.1000      0.0001
##     40      0.0227      nan      0.1000     -0.0000
##     60      0.0224      nan      0.1000     -0.0000
##     80      0.0223      nan      0.1000     -0.0000
##    100      0.0223      nan      0.1000     -0.0000
##    120      0.0222      nan      0.1000     -0.0000
##    140      0.0222      nan      0.1000     -0.0000
##    150      0.0222      nan      0.1000     -0.0000
##
## - Fold02.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold02.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0369           nan       0.1000    0.0028
##      2           0.0341           nan       0.1000    0.0027
##      3           0.0320           nan       0.1000    0.0021
##      4           0.0303           nan       0.1000    0.0018
##      5           0.0290           nan       0.1000    0.0012
##      6           0.0280           nan       0.1000    0.0010
##      7           0.0271           nan       0.1000    0.0009
##      8           0.0263           nan       0.1000    0.0007
##      9           0.0257           nan       0.1000    0.0005
##     10           0.0253           nan       0.1000    0.0004
##     20           0.0229           nan       0.1000    0.0001
##     40           0.0222           nan       0.1000   -0.0000
##     60           0.0220           nan       0.1000   -0.0000
##     80           0.0219           nan       0.1000   -0.0000
##    100           0.0218           nan       0.1000   -0.0000
##    120           0.0218           nan       0.1000   -0.0000
##    140           0.0217           nan       0.1000   -0.0000
##    150           0.0217           nan       0.1000   -0.0000
##
## - Fold02.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold03.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0387           nan       0.1000    0.0021
##      2           0.0368           nan       0.1000    0.0018
##      3           0.0352           nan       0.1000    0.0014
##      4           0.0339           nan       0.1000    0.0010
##      5           0.0328           nan       0.1000    0.0010
##      6           0.0321           nan       0.1000    0.0007
##      7           0.0314           nan       0.1000    0.0006
##      8           0.0309           nan       0.1000    0.0005
##      9           0.0305           nan       0.1000    0.0004
```

```
##      10      0.0302      nan      0.1000      0.0003
##      20      0.0289      nan      0.1000      0.0000
##      40      0.0281      nan      0.1000      0.0000
##      60      0.0279      nan      0.1000      0.0000
##      80      0.0278      nan      0.1000     -0.0000
##     100      0.0278      nan      0.1000     -0.0000
##     120      0.0278      nan      0.1000     -0.0000
##     140      0.0278      nan      0.1000     -0.0000
##     150      0.0278      nan      0.1000     -0.0000
##
## - Fold03.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold03.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0382      nan      0.1000      0.0032
##      2      0.0360      nan      0.1000      0.0023
##      3      0.0341      nan      0.1000      0.0020
##      4      0.0327      nan      0.1000      0.0016
##      5      0.0314      nan      0.1000      0.0012
##      6      0.0303      nan      0.1000      0.0010
##      7      0.0295      nan      0.1000      0.0007
##      8      0.0289      nan      0.1000      0.0007
##      9      0.0283      nan      0.1000      0.0006
##     10      0.0278      nan      0.1000      0.0005
##     20      0.0255      nan      0.1000      0.0001
##     40      0.0245      nan      0.1000     -0.0000
##     60      0.0243      nan      0.1000     -0.0000
##     80      0.0242      nan      0.1000     -0.0000
##    100      0.0241      nan      0.1000     -0.0000
##    120      0.0241      nan      0.1000     -0.0000
##    140      0.0241      nan      0.1000     -0.0000
##    150      0.0241      nan      0.1000     -0.0000
##
## - Fold03.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold03.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0379      nan      0.1000      0.0024
##      2      0.0355      nan      0.1000      0.0023
##      3      0.0338      nan      0.1000      0.0019
##      4      0.0321      nan      0.1000      0.0017
##      5      0.0308      nan      0.1000      0.0011
##      6      0.0296      nan      0.1000      0.0011
##      7      0.0288      nan      0.1000      0.0007
##      8      0.0281      nan      0.1000      0.0007
##      9      0.0274      nan      0.1000      0.0005
##     10      0.0270      nan      0.1000      0.0005
##     20      0.0249      nan      0.1000      0.0000
##     40      0.0242      nan      0.1000     -0.0000
##     60      0.0240      nan      0.1000     -0.0000
```

```
##      80      0.0239      nan      0.1000     -0.0000
##     100      0.0238      nan      0.1000     -0.0000
##     120      0.0238      nan      0.1000     -0.0000
##     140      0.0237      nan      0.1000     -0.0000
##     150      0.0237      nan      0.1000     -0.0000
##
## - Fold03.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold04.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1          0.0373           nan      0.1000    0.0022
##      2          0.0354           nan      0.1000    0.0018
##      3          0.0339           nan      0.1000    0.0015
##      4          0.0324           nan      0.1000    0.0013
##      5          0.0314           nan      0.1000    0.0009
##      6          0.0308           nan      0.1000    0.0007
##      7          0.0302           nan      0.1000    0.0007
##      8          0.0296           nan      0.1000    0.0006
##      9          0.0292           nan      0.1000    0.0004
##     10          0.0288           nan      0.1000    0.0004
##     20          0.0273           nan      0.1000    0.0001
##     40          0.0265           nan      0.1000    0.0000
##     60          0.0263           nan      0.1000   -0.0000
##     80          0.0262           nan      0.1000   -0.0000
##    100          0.0262           nan      0.1000   -0.0000
##    120          0.0262           nan      0.1000   -0.0000
##    140          0.0262           nan      0.1000   -0.0000
##    150          0.0262           nan      0.1000   -0.0000
##
## - Fold04.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold04.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1          0.0368           nan      0.1000    0.0027
##      2          0.0343           nan      0.1000    0.0023
##      3          0.0322           nan      0.1000    0.0020
##      4          0.0306           nan      0.1000    0.0013
##      5          0.0294           nan      0.1000    0.0013
##      6          0.0284           nan      0.1000    0.0011
##      7          0.0275           nan      0.1000    0.0009
##      8          0.0268           nan      0.1000    0.0007
##      9          0.0261           nan      0.1000    0.0005
##     10          0.0256           nan      0.1000    0.0004
##     20          0.0235           nan      0.1000    0.0001
##     40          0.0227           nan      0.1000    0.0000
##     60          0.0225           nan      0.1000   -0.0000
##     80          0.0224           nan      0.1000   -0.0000
##    100          0.0223           nan      0.1000   -0.0000
##    120          0.0223           nan      0.1000   -0.0000
##    140          0.0223           nan      0.1000   -0.0000
```

```
##      150      0.0223      nan      0.1000     -0.0000
##
## - Fold04.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold04.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0367      nan      0.1000      0.0032
##      2      0.0341      nan      0.1000      0.0026
##      3      0.0321      nan      0.1000      0.0020
##      4      0.0304      nan      0.1000      0.0014
##      5      0.0291      nan      0.1000      0.0011
##      6      0.0280      nan      0.1000      0.0011
##      7      0.0271      nan      0.1000      0.0008
##      8      0.0264      nan      0.1000      0.0006
##      9      0.0258      nan      0.1000      0.0006
##     10      0.0253      nan      0.1000      0.0005
##     20      0.0230      nan      0.1000      0.0001
##     40      0.0222      nan      0.1000      0.0000
##     60      0.0221      nan      0.1000     -0.0000
##     80      0.0221      nan      0.1000     -0.0000
##    100      0.0220      nan      0.1000     -0.0000
##    120      0.0220      nan      0.1000     -0.0000
##    140      0.0219      nan      0.1000     -0.0000
##    150      0.0219      nan      0.1000     -0.0000
##
## - Fold04.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold05.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0374      nan      0.1000      0.0022
##      2      0.0356      nan      0.1000      0.0016
##      3      0.0343      nan      0.1000      0.0014
##      4      0.0332      nan      0.1000      0.0011
##      5      0.0321      nan      0.1000      0.0010
##      6      0.0312      nan      0.1000      0.0008
##      7      0.0307      nan      0.1000      0.0006
##      8      0.0301      nan      0.1000      0.0004
##      9      0.0297      nan      0.1000      0.0004
##     10      0.0295      nan      0.1000      0.0003
##     20      0.0279      nan      0.1000      0.0001
##     40      0.0272      nan      0.1000     -0.0000
##     60      0.0270      nan      0.1000      0.0000
##     80      0.0269      nan      0.1000     -0.0000
##    100      0.0269      nan      0.1000     -0.0000
##    120      0.0269      nan      0.1000     -0.0000
##    140      0.0269      nan      0.1000     -0.0000
##    150      0.0269      nan      0.1000     -0.0000
##
## - Fold05.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
```

```
## + Fold05.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0369	nan	0.1000	0.0027
## 2	0.0346	nan	0.1000	0.0023
## 3	0.0328	nan	0.1000	0.0019
## 4	0.0312	nan	0.1000	0.0016
## 5	0.0301	nan	0.1000	0.0012
## 6	0.0291	nan	0.1000	0.0011
## 7	0.0284	nan	0.1000	0.0008
## 8	0.0277	nan	0.1000	0.0007
## 9	0.0271	nan	0.1000	0.0006
## 10	0.0265	nan	0.1000	0.0006
## 20	0.0243	nan	0.1000	0.0001
## 40	0.0235	nan	0.1000	-0.0000
## 60	0.0232	nan	0.1000	0.0000
## 80	0.0231	nan	0.1000	-0.0000
## 100	0.0231	nan	0.1000	-0.0000
## 120	0.0230	nan	0.1000	-0.0000
## 140	0.0230	nan	0.1000	-0.0000
## 150	0.0230	nan	0.1000	-0.0000

```
##
## - Fold05.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

```
## + Fold05.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0367	nan	0.1000	0.0031
## 2	0.0347	nan	0.1000	0.0025
## 3	0.0326	nan	0.1000	0.0016
## 4	0.0311	nan	0.1000	0.0014
## 5	0.0299	nan	0.1000	0.0011
## 6	0.0288	nan	0.1000	0.0011
## 7	0.0279	nan	0.1000	0.0010
## 8	0.0272	nan	0.1000	0.0007
## 9	0.0265	nan	0.1000	0.0006
## 10	0.0261	nan	0.1000	0.0004
## 20	0.0238	nan	0.1000	0.0001
## 40	0.0231	nan	0.1000	-0.0000
## 60	0.0229	nan	0.1000	-0.0000
## 80	0.0228	nan	0.1000	-0.0000
## 100	0.0227	nan	0.1000	-0.0000
## 120	0.0227	nan	0.1000	-0.0000
## 140	0.0226	nan	0.1000	-0.0000
## 150	0.0226	nan	0.1000	-0.0000

```
##
## - Fold05.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
```

```
## + Fold06.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0375	nan	0.1000	0.0023

```
##      2      0.0356      nan      0.1000      0.0019
##      3      0.0341      nan      0.1000      0.0017
##      4      0.0328      nan      0.1000      0.0013
##      5      0.0318      nan      0.1000      0.0011
##      6      0.0310      nan      0.1000      0.0008
##      7      0.0304      nan      0.1000      0.0007
##      8      0.0299      nan      0.1000      0.0006
##      9      0.0295      nan      0.1000      0.0005
##     10      0.0290      nan      0.1000      0.0004
##     20      0.0275      nan      0.1000      0.0001
##     40      0.0268      nan      0.1000      0.0000
##     60      0.0266      nan      0.1000     -0.0000
##     80      0.0265      nan      0.1000     -0.0000
##    100      0.0265      nan      0.1000     -0.0000
##    120      0.0265      nan      0.1000     -0.0000
##    140      0.0265      nan      0.1000     -0.0000
##    150      0.0265      nan      0.1000     -0.0000
##
## - Fold06.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold06.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0371      nan      0.1000      0.0031
##      2      0.0347      nan      0.1000      0.0025
##      3      0.0328      nan      0.1000      0.0020
##      4      0.0312      nan      0.1000      0.0015
##      5      0.0299      nan      0.1000      0.0011
##      6      0.0289      nan      0.1000      0.0009
##      7      0.0281      nan      0.1000      0.0008
##      8      0.0273      nan      0.1000      0.0007
##      9      0.0267      nan      0.1000      0.0005
##     10      0.0263      nan      0.1000      0.0003
##     20      0.0242      nan      0.1000      0.0001
##     40      0.0233      nan      0.1000     -0.0000
##     60      0.0230      nan      0.1000     -0.0000
##     80      0.0230      nan      0.1000     -0.0000
##    100      0.0229      nan      0.1000     -0.0000
##    120      0.0229      nan      0.1000     -0.0000
##    140      0.0228      nan      0.1000     -0.0000
##    150      0.0228      nan      0.1000     -0.0000
##
## - Fold06.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold06.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0371      nan      0.1000      0.0033
##      2      0.0347      nan      0.1000      0.0027
##      3      0.0327      nan      0.1000      0.0021
##      4      0.0309      nan      0.1000      0.0019
##      5      0.0296      nan      0.1000      0.0012
```



```
##      6      0.0285      nan      0.1000      0.0009
##      7      0.0275      nan      0.1000      0.0009
##      8      0.0269      nan      0.1000      0.0006
##      9      0.0263      nan      0.1000      0.0005
##     10      0.0258      nan      0.1000      0.0005
##     20      0.0238      nan      0.1000      0.0001
##     40      0.0229      nan      0.1000     -0.0000
##     60      0.0228      nan      0.1000     -0.0000
##     80      0.0227      nan      0.1000     -0.0000
##    100      0.0226      nan      0.1000     -0.0000
##    120      0.0225      nan      0.1000     -0.0000
##    140      0.0225      nan      0.1000     -0.0000
##    150      0.0224      nan      0.1000     -0.0000
##
## - Fold06.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold07.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0373           nan       0.1000     0.0024
##      2           0.0356           nan       0.1000     0.0020
##      3           0.0339           nan       0.1000     0.0017
##      4           0.0327           nan       0.1000     0.0012
##      5           0.0316           nan       0.1000     0.0010
##      6           0.0306           nan       0.1000     0.0009
##      7           0.0299           nan       0.1000     0.0007
##      8           0.0295           nan       0.1000     0.0005
##      9           0.0290           nan       0.1000     0.0005
##     10           0.0286           nan       0.1000     0.0003
##     20           0.0270           nan       0.1000     0.0000
##     40           0.0263           nan       0.1000     0.0000
##     60           0.0260           nan       0.1000    -0.0000
##     80           0.0260           nan       0.1000    -0.0000
##    100           0.0260           nan       0.1000    -0.0000
##    120           0.0259           nan       0.1000    -0.0000
##    140           0.0260           nan       0.1000     0.0000
##    150           0.0259           nan       0.1000    -0.0000
##
## - Fold07.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold07.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0370           nan       0.1000     0.0031
##      2           0.0346           nan       0.1000     0.0024
##      3           0.0327           nan       0.1000     0.0023
##      4           0.0310           nan       0.1000     0.0017
##      5           0.0297           nan       0.1000     0.0013
##      6           0.0287           nan       0.1000     0.0010
##      7           0.0277           nan       0.1000     0.0009
##      8           0.0269           nan       0.1000     0.0006
##      9           0.0263           nan       0.1000     0.0005
```

```
##      10      0.0258      nan      0.1000      0.0004
##      20      0.0237      nan      0.1000      0.0001
##      40      0.0229      nan      0.1000     -0.0000
##      60      0.0228      nan      0.1000     -0.0000
##      80      0.0227      nan      0.1000     -0.0000
##     100      0.0226      nan      0.1000     -0.0000
##     120      0.0225      nan      0.1000     -0.0000
##     140      0.0225      nan      0.1000     -0.0000
##     150      0.0225      nan      0.1000     -0.0000
##
## - Fold07.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold07.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0370      nan      0.1000      0.0030
##      2      0.0344      nan      0.1000      0.0021
##      3      0.0324      nan      0.1000      0.0019
##      4      0.0308      nan      0.1000      0.0018
##      5      0.0296      nan      0.1000      0.0012
##      6      0.0284      nan      0.1000      0.0011
##      7      0.0274      nan      0.1000      0.0010
##      8      0.0267      nan      0.1000      0.0008
##      9      0.0260      nan      0.1000      0.0005
##     10      0.0254      nan      0.1000      0.0005
##     20      0.0233      nan      0.1000      0.0001
##     40      0.0226      nan      0.1000     -0.0000
##     60      0.0224      nan      0.1000     -0.0000
##     80      0.0223      nan      0.1000     -0.0000
##    100      0.0223      nan      0.1000     -0.0000
##    120      0.0222      nan      0.1000      0.0000
##    140      0.0222      nan      0.1000     -0.0000
##    150      0.0222      nan      0.1000     -0.0000
##
## - Fold07.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold08.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0378      nan      0.1000      0.0026
##      2      0.0357      nan      0.1000      0.0018
##      3      0.0342      nan      0.1000      0.0016
##      4      0.0330      nan      0.1000      0.0013
##      5      0.0319      nan      0.1000      0.0011
##      6      0.0310      nan      0.1000      0.0010
##      7      0.0303      nan      0.1000      0.0007
##      8      0.0297      nan      0.1000      0.0005
##      9      0.0292      nan      0.1000      0.0004
##     10      0.0289      nan      0.1000      0.0004
##     20      0.0273      nan      0.1000      0.0001
##     40      0.0266      nan      0.1000      0.0000
##     60      0.0265      nan      0.1000     -0.0000
```

```
##      80      0.0264      nan      0.1000     -0.0000
##     100      0.0264      nan      0.1000     -0.0000
##     120      0.0264      nan      0.1000     -0.0000
##     140      0.0264      nan      0.1000     -0.0000
##     150      0.0264      nan      0.1000     -0.0000
##
## - Fold08.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold08.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.0371         nan         0.1000     0.0035
##      2         0.0347         nan         0.1000     0.0025
##      3         0.0328         nan         0.1000     0.0018
##      4         0.0313         nan         0.1000     0.0014
##      5         0.0300         nan         0.1000     0.0015
##      6         0.0290         nan         0.1000     0.0010
##      7         0.0280         nan         0.1000     0.0007
##      8         0.0273         nan         0.1000     0.0008
##      9         0.0267         nan         0.1000     0.0006
##     10         0.0262         nan         0.1000     0.0004
##     20         0.0240         nan         0.1000     0.0001
##     40         0.0231         nan         0.1000     0.0000
##     60         0.0229         nan         0.1000    -0.0000
##     80         0.0228         nan         0.1000    -0.0000
##    100         0.0228         nan         0.1000    -0.0000
##    120         0.0227         nan         0.1000    -0.0000
##    140         0.0227         nan         0.1000    -0.0000
##    150         0.0227         nan         0.1000    -0.0000
##
## - Fold08.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold08.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.0373         nan         0.1000     0.0027
##      2         0.0350         nan         0.1000     0.0022
##      3         0.0329         nan         0.1000     0.0023
##      4         0.0311         nan         0.1000     0.0018
##      5         0.0298         nan         0.1000     0.0013
##      6         0.0286         nan         0.1000     0.0012
##      7         0.0276         nan         0.1000     0.0008
##      8         0.0268         nan         0.1000     0.0007
##      9         0.0262         nan         0.1000     0.0006
##     10         0.0257         nan         0.1000     0.0005
##     20         0.0235         nan         0.1000     0.0001
##     40         0.0228         nan         0.1000    -0.0000
##     60         0.0227         nan         0.1000    -0.0000
##     80         0.0226         nan         0.1000    -0.0000
##    100         0.0225         nan         0.1000    -0.0000
##    120         0.0225         nan         0.1000    -0.0000
##    140         0.0224         nan         0.1000    -0.0000
```

```
##      150      0.0224      nan      0.1000     -0.0000
##
## - Fold08.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold09.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0371      nan      0.1000      0.0020
##      2      0.0352      nan      0.1000      0.0017
##      3      0.0339      nan      0.1000      0.0015
##      4      0.0326      nan      0.1000      0.0012
##      5      0.0317      nan      0.1000      0.0009
##      6      0.0310      nan      0.1000      0.0008
##      7      0.0303      nan      0.1000      0.0006
##      8      0.0299      nan      0.1000      0.0005
##      9      0.0296      nan      0.1000      0.0004
##     10      0.0293      nan      0.1000      0.0004
##     20      0.0278      nan      0.1000      0.0001
##     40      0.0272      nan      0.1000      0.0000
##     60      0.0269      nan      0.1000     -0.0000
##     80      0.0269      nan      0.1000     -0.0000
##    100      0.0269      nan      0.1000     -0.0000
##    120      0.0268      nan      0.1000     -0.0000
##    140      0.0268      nan      0.1000     -0.0000
##    150      0.0268      nan      0.1000     -0.0000
##
## - Fold09.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold09.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0367      nan      0.1000      0.0027
##      2      0.0344      nan      0.1000      0.0022
##      3      0.0326      nan      0.1000      0.0019
##      4      0.0311      nan      0.1000      0.0014
##      5      0.0299      nan      0.1000      0.0012
##      6      0.0290      nan      0.1000      0.0009
##      7      0.0282      nan      0.1000      0.0007
##      8      0.0275      nan      0.1000      0.0006
##      9      0.0271      nan      0.1000      0.0005
##     10      0.0265      nan      0.1000      0.0005
##     20      0.0246      nan      0.1000      0.0001
##     40      0.0237      nan      0.1000     -0.0000
##     60      0.0235      nan      0.1000     -0.0000
##     80      0.0233      nan      0.1000     -0.0000
##    100      0.0232      nan      0.1000     -0.0000
##    120      0.0232      nan      0.1000     -0.0000
##    140      0.0231      nan      0.1000     -0.0000
##    150      0.0231      nan      0.1000     -0.0000
##
## - Fold09.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

```
## + Fold09.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0361	nan	0.1000	0.0033
## 2	0.0339	nan	0.1000	0.0023
## 3	0.0320	nan	0.1000	0.0016
## 4	0.0305	nan	0.1000	0.0014
## 5	0.0293	nan	0.1000	0.0012
## 6	0.0284	nan	0.1000	0.0010
## 7	0.0277	nan	0.1000	0.0008
## 8	0.0270	nan	0.1000	0.0007
## 9	0.0264	nan	0.1000	0.0006
## 10	0.0260	nan	0.1000	0.0004
## 20	0.0239	nan	0.1000	0.0000
## 40	0.0232	nan	0.1000	-0.0000
## 60	0.0229	nan	0.1000	-0.0000
## 80	0.0228	nan	0.1000	-0.0000
## 100	0.0228	nan	0.1000	-0.0000
## 120	0.0227	nan	0.1000	-0.0000
## 140	0.0227	nan	0.1000	-0.0000
## 150	0.0227	nan	0.1000	-0.0000

```
##
## - Fold09.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold10.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0362	nan	0.1000	0.0023
## 2	0.0345	nan	0.1000	0.0021
## 3	0.0330	nan	0.1000	0.0015
## 4	0.0319	nan	0.1000	0.0012
## 5	0.0309	nan	0.1000	0.0010
## 6	0.0302	nan	0.1000	0.0008
## 7	0.0294	nan	0.1000	0.0004
## 8	0.0290	nan	0.1000	0.0005
## 9	0.0285	nan	0.1000	0.0004
## 10	0.0281	nan	0.1000	0.0002
## 20	0.0269	nan	0.1000	0.0001
## 40	0.0261	nan	0.1000	0.0000
## 60	0.0259	nan	0.1000	-0.0000
## 80	0.0258	nan	0.1000	-0.0000
## 100	0.0258	nan	0.1000	-0.0000
## 120	0.0258	nan	0.1000	-0.0000
## 140	0.0258	nan	0.1000	-0.0000
## 150	0.0258	nan	0.1000	-0.0000

```
##
## - Fold10.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold10.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0357	nan	0.1000	0.0021

```
##      2      0.0335      nan      0.1000      0.0019
##      3      0.0316      nan      0.1000      0.0018
##      4      0.0301      nan      0.1000      0.0013
##      5      0.0289      nan      0.1000      0.0012
##      6      0.0279      nan      0.1000      0.0010
##      7      0.0271      nan      0.1000      0.0007
##      8      0.0265      nan      0.1000      0.0006
##      9      0.0260      nan      0.1000      0.0005
##     10      0.0256      nan      0.1000      0.0005
##     20      0.0234      nan      0.1000      0.0001
##     40      0.0226      nan      0.1000      0.0000
##     60      0.0223      nan      0.1000     -0.0000
##     80      0.0222      nan      0.1000     -0.0000
##    100      0.0222      nan      0.1000     -0.0000
##    120      0.0221      nan      0.1000     -0.0000
##    140      0.0221      nan      0.1000     -0.0000
##    150      0.0221      nan      0.1000     -0.0000
##
## - Fold10.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold10.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0354      nan      0.1000      0.0027
##      2      0.0331      nan      0.1000      0.0019
##      3      0.0313      nan      0.1000      0.0017
##      4      0.0297      nan      0.1000      0.0016
##      5      0.0284      nan      0.1000      0.0015
##      6      0.0275      nan      0.1000      0.0010
##      7      0.0265      nan      0.1000      0.0008
##      8      0.0259      nan      0.1000      0.0007
##      9      0.0252      nan      0.1000      0.0006
##     10      0.0247      nan      0.1000      0.0003
##     20      0.0228      nan      0.1000      0.0001
##     40      0.0221      nan      0.1000     -0.0000
##     60      0.0219      nan      0.1000     -0.0000
##     80      0.0218      nan      0.1000     -0.0000
##    100      0.0218      nan      0.1000     -0.0000
##    120      0.0217      nan      0.1000     -0.0000
##    140      0.0217      nan      0.1000     -0.0000
##    150      0.0217      nan      0.1000     -0.0000
##
## - Fold10.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold01.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0371      nan      0.1000      0.0022
##      2      0.0354      nan      0.1000      0.0020
##      3      0.0339      nan      0.1000      0.0016
##      4      0.0326      nan      0.1000      0.0014
##      5      0.0316      nan      0.1000      0.0009
```

```
##      6      0.0308      nan      0.1000      0.0008
##      7      0.0302      nan      0.1000      0.0007
##      8      0.0296      nan      0.1000      0.0006
##      9      0.0290      nan      0.1000      0.0005
##     10      0.0286      nan      0.1000      0.0003
##     20      0.0270      nan      0.1000      0.0000
##     40      0.0263      nan      0.1000     -0.0000
##     60      0.0260      nan      0.1000      0.0000
##     80      0.0260      nan      0.1000     -0.0000
##    100      0.0260      nan      0.1000     -0.0000
##    120      0.0260      nan      0.1000     -0.0000
##    140      0.0260      nan      0.1000     -0.0000
##    150      0.0260      nan      0.1000     -0.0000
##
## - Fold01.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold01.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0367             nan      0.1000     0.0026
##      2           0.0344             nan      0.1000     0.0027
##      3           0.0326             nan      0.1000     0.0019
##      4           0.0310             nan      0.1000     0.0018
##      5           0.0296             nan      0.1000     0.0012
##      6           0.0285             nan      0.1000     0.0010
##      7           0.0275             nan      0.1000     0.0007
##      8           0.0268             nan      0.1000     0.0005
##      9           0.0263             nan      0.1000     0.0005
##     10           0.0258             nan      0.1000     0.0004
##     20           0.0237             nan      0.1000     0.0001
##     40           0.0227             nan      0.1000     -0.0000
##     60           0.0225             nan      0.1000      0.0000
##     80           0.0224             nan      0.1000     -0.0000
##    100           0.0223             nan      0.1000     -0.0000
##    120           0.0223             nan      0.1000     -0.0000
##    140           0.0223             nan      0.1000     -0.0000
##    150           0.0223             nan      0.1000     -0.0000
##
## - Fold01.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold01.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0365             nan      0.1000     0.0029
##      2           0.0340             nan      0.1000     0.0022
##      3           0.0321             nan      0.1000     0.0023
##      4           0.0305             nan      0.1000     0.0016
##      5           0.0292             nan      0.1000     0.0013
##      6           0.0281             nan      0.1000     0.0010
##      7           0.0273             nan      0.1000     0.0008
##      8           0.0266             nan      0.1000     0.0007
##      9           0.0260             nan      0.1000     0.0005
```

```
##      10      0.0255      nan      0.1000      0.0005
##      20      0.0231      nan      0.1000      0.0001
##      40      0.0225      nan      0.1000     -0.0000
##      60      0.0222      nan      0.1000     -0.0000
##      80      0.0222      nan      0.1000     -0.0000
##     100      0.0221      nan      0.1000     -0.0000
##     120      0.0220      nan      0.1000     -0.0000
##     140      0.0220      nan      0.1000     -0.0000
##     150      0.0220      nan      0.1000     -0.0000
##
## - Fold01.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold02.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0384      nan      0.1000      0.0026
##      2      0.0365      nan      0.1000      0.0017
##      3      0.0348      nan      0.1000      0.0016
##      4      0.0336      nan      0.1000      0.0012
##      5      0.0325      nan      0.1000      0.0011
##      6      0.0316      nan      0.1000      0.0008
##      7      0.0310      nan      0.1000      0.0007
##      8      0.0304      nan      0.1000      0.0005
##      9      0.0299      nan      0.1000      0.0005
##     10      0.0295      nan      0.1000      0.0003
##     20      0.0281      nan      0.1000      0.0001
##     40      0.0272      nan      0.1000      0.0000
##     60      0.0270      nan      0.1000      0.0000
##     80      0.0269      nan      0.1000     -0.0000
##    100      0.0269      nan      0.1000     -0.0000
##    120      0.0269      nan      0.1000     -0.0000
##    140      0.0269      nan      0.1000     -0.0000
##    150      0.0269      nan      0.1000     -0.0000
##
## - Fold02.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold02.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0377      nan      0.1000      0.0028
##      2      0.0356      nan      0.1000      0.0025
##      3      0.0335      nan      0.1000      0.0021
##      4      0.0320      nan      0.1000      0.0015
##      5      0.0306      nan      0.1000      0.0013
##      6      0.0295      nan      0.1000      0.0012
##      7      0.0287      nan      0.1000      0.0008
##      8      0.0279      nan      0.1000      0.0008
##      9      0.0273      nan      0.1000      0.0006
##     10      0.0268      nan      0.1000      0.0005
##     20      0.0244      nan      0.1000      0.0001
##     40      0.0235      nan      0.1000     -0.0000
##     60      0.0233      nan      0.1000     -0.0000
```



```

##      80      0.0232      nan      0.1000     -0.0000
##     100      0.0231      nan      0.1000     -0.0000
##     120      0.0230      nan      0.1000     -0.0000
##     140      0.0230      nan      0.1000     -0.0000
##     150      0.0230      nan      0.1000     -0.0000
##
## - Fold02.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold02.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0377           nan      0.1000    0.0033
##      2           0.0352           nan      0.1000    0.0025
##      3           0.0331           nan      0.1000    0.0022
##      4           0.0314           nan      0.1000    0.0017
##      5           0.0299           nan      0.1000    0.0014
##      6           0.0287           nan      0.1000    0.0012
##      7           0.0279           nan      0.1000    0.0008
##      8           0.0272           nan      0.1000    0.0007
##      9           0.0266           nan      0.1000    0.0005
##     10           0.0260           nan      0.1000    0.0004
##     20           0.0238           nan      0.1000    0.0001
##     40           0.0231           nan      0.1000   -0.0000
##     60           0.0229           nan      0.1000   -0.0000
##     80           0.0228           nan      0.1000   -0.0000
##    100           0.0227           nan      0.1000   -0.0000
##    120           0.0226           nan      0.1000   -0.0000
##    140           0.0225           nan      0.1000   -0.0000
##    150           0.0225           nan      0.1000   -0.0000
##
## - Fold02.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold03.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0373           nan      0.1000    0.0022
##      2           0.0354           nan      0.1000    0.0021
##      3           0.0338           nan      0.1000    0.0014
##      4           0.0327           nan      0.1000    0.0012
##      5           0.0315           nan      0.1000    0.0011
##      6           0.0308           nan      0.1000    0.0007
##      7           0.0302           nan      0.1000    0.0006
##      8           0.0297           nan      0.1000    0.0005
##      9           0.0292           nan      0.1000    0.0004
##     10           0.0289           nan      0.1000    0.0003
##     20           0.0275           nan      0.1000    0.0001
##     40           0.0268           nan      0.1000    0.0000
##     60           0.0265           nan      0.1000    0.0000
##     80           0.0264           nan      0.1000   -0.0000
##    100           0.0264           nan      0.1000   -0.0000
##    120           0.0264           nan      0.1000   -0.0000
##    140           0.0264           nan      0.1000   -0.0000

```

```
##      150      0.0264      nan      0.1000     -0.0000
##
## - Fold03.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold03.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0369      nan      0.1000      0.0023
##      2      0.0346      nan      0.1000      0.0021
##      3      0.0327      nan      0.1000      0.0020
##      4      0.0311      nan      0.1000      0.0014
##      5      0.0297      nan      0.1000      0.0015
##      6      0.0287      nan      0.1000      0.0010
##      7      0.0278      nan      0.1000      0.0009
##      8      0.0271      nan      0.1000      0.0006
##      9      0.0265      nan      0.1000      0.0005
##     10      0.0261      nan      0.1000      0.0005
##     20      0.0240      nan      0.1000      0.0001
##     40      0.0231      nan      0.1000      0.0000
##     60      0.0228      nan      0.1000     -0.0000
##     80      0.0228      nan      0.1000     -0.0000
##    100      0.0227      nan      0.1000     -0.0000
##    120      0.0227      nan      0.1000     -0.0000
##    140      0.0226      nan      0.1000     -0.0000
##    150      0.0226      nan      0.1000     -0.0000
##
## - Fold03.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold03.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0369      nan      0.1000      0.0026
##      2      0.0345      nan      0.1000      0.0021
##      3      0.0324      nan      0.1000      0.0023
##      4      0.0308      nan      0.1000      0.0016
##      5      0.0295      nan      0.1000      0.0013
##      6      0.0284      nan      0.1000      0.0011
##      7      0.0275      nan      0.1000      0.0008
##      8      0.0268      nan      0.1000      0.0006
##      9      0.0263      nan      0.1000      0.0006
##     10      0.0259      nan      0.1000      0.0005
##     20      0.0235      nan      0.1000      0.0001
##     40      0.0227      nan      0.1000     -0.0000
##     60      0.0225      nan      0.1000     -0.0000
##     80      0.0224      nan      0.1000     -0.0000
##    100      0.0223      nan      0.1000     -0.0001
##    120      0.0222      nan      0.1000     -0.0000
##    140      0.0222      nan      0.1000     -0.0000
##    150      0.0222      nan      0.1000     -0.0000
##
## - Fold03.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
```

```
## + Fold04.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0378	nan	0.1000	0.0021
## 2	0.0359	nan	0.1000	0.0019
## 3	0.0344	nan	0.1000	0.0016
## 4	0.0332	nan	0.1000	0.0013
## 5	0.0322	nan	0.1000	0.0010
## 6	0.0314	nan	0.1000	0.0007
## 7	0.0307	nan	0.1000	0.0006
## 8	0.0302	nan	0.1000	0.0005
## 9	0.0298	nan	0.1000	0.0004
## 10	0.0294	nan	0.1000	0.0004
## 20	0.0280	nan	0.1000	0.0000
## 40	0.0272	nan	0.1000	0.0000
## 60	0.0270	nan	0.1000	-0.0000
## 80	0.0269	nan	0.1000	-0.0000
## 100	0.0269	nan	0.1000	-0.0000
## 120	0.0269	nan	0.1000	-0.0000
## 140	0.0269	nan	0.1000	-0.0000
## 150	0.0269	nan	0.1000	-0.0000

```
##
## - Fold04.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
```

```
## + Fold04.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0373	nan	0.1000	0.0031
## 2	0.0354	nan	0.1000	0.0019
## 3	0.0335	nan	0.1000	0.0021
## 4	0.0318	nan	0.1000	0.0018
## 5	0.0305	nan	0.1000	0.0013
## 6	0.0293	nan	0.1000	0.0010
## 7	0.0285	nan	0.1000	0.0008
## 8	0.0277	nan	0.1000	0.0007
## 9	0.0271	nan	0.1000	0.0006
## 10	0.0267	nan	0.1000	0.0003
## 20	0.0244	nan	0.1000	0.0001
## 40	0.0235	nan	0.1000	0.0000
## 60	0.0232	nan	0.1000	-0.0000
## 80	0.0231	nan	0.1000	-0.0000
## 100	0.0231	nan	0.1000	-0.0000
## 120	0.0230	nan	0.1000	-0.0000
## 140	0.0230	nan	0.1000	-0.0000
## 150	0.0230	nan	0.1000	-0.0000

```
##
## - Fold04.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

```
## + Fold04.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0374	nan	0.1000	0.0031

```
##      2      0.0350      nan      0.1000      0.0025
##      3      0.0331      nan      0.1000      0.0020
##      4      0.0312      nan      0.1000      0.0019
##      5      0.0299      nan      0.1000      0.0013
##      6      0.0288      nan      0.1000      0.0010
##      7      0.0279      nan      0.1000      0.0008
##      8      0.0272      nan      0.1000      0.0007
##      9      0.0266      nan      0.1000      0.0007
##     10      0.0261      nan      0.1000      0.0005
##     20      0.0238      nan      0.1000      0.0001
##     40      0.0231      nan      0.1000      0.0000
##     60      0.0228      nan      0.1000     -0.0000
##     80      0.0227      nan      0.1000     -0.0000
##    100      0.0226      nan      0.1000     -0.0000
##    120      0.0226      nan      0.1000     -0.0000
##    140      0.0225      nan      0.1000     -0.0000
##    150      0.0225      nan      0.1000     -0.0000
##
## - Fold04.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold05.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0372      nan      0.1000      0.0022
##      2      0.0354      nan      0.1000      0.0021
##      3      0.0341      nan      0.1000      0.0015
##      4      0.0329      nan      0.1000      0.0014
##      5      0.0318      nan      0.1000      0.0012
##      6      0.0310      nan      0.1000      0.0008
##      7      0.0303      nan      0.1000      0.0008
##      8      0.0296      nan      0.1000      0.0005
##      9      0.0292      nan      0.1000      0.0004
##     10      0.0288      nan      0.1000      0.0004
##     20      0.0273      nan      0.1000      0.0000
##     40      0.0265      nan      0.1000     -0.0000
##     60      0.0263      nan      0.1000     -0.0000
##     80      0.0263      nan      0.1000     -0.0000
##    100      0.0263      nan      0.1000     -0.0000
##    120      0.0263      nan      0.1000     -0.0000
##    140      0.0263      nan      0.1000     -0.0000
##    150      0.0263      nan      0.1000     -0.0000
##
## - Fold05.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold05.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0364      nan      0.1000      0.0028
##      2      0.0344      nan      0.1000      0.0022
##      3      0.0324      nan      0.1000      0.0018
##      4      0.0309      nan      0.1000      0.0015
##      5      0.0296      nan      0.1000      0.0013
```

```
##      6      0.0286      nan      0.1000      0.0009
##      7      0.0277      nan      0.1000      0.0008
##      8      0.0270      nan      0.1000      0.0007
##      9      0.0263      nan      0.1000      0.0006
##     10      0.0259      nan      0.1000      0.0004
##     20      0.0237      nan      0.1000      0.0001
##     40      0.0229      nan      0.1000      0.0000
##     60      0.0227      nan      0.1000     -0.0001
##     80      0.0226      nan      0.1000     -0.0000
##    100      0.0226      nan      0.1000     -0.0000
##    120      0.0225      nan      0.1000     -0.0000
##    140      0.0225      nan      0.1000     -0.0000
##    150      0.0225      nan      0.1000     -0.0000
##
## - Fold05.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold05.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0365             nan      0.1000     0.0031
##      2           0.0340             nan      0.1000     0.0026
##      3           0.0322             nan      0.1000     0.0020
##      4           0.0306             nan      0.1000     0.0017
##      5           0.0293             nan      0.1000     0.0012
##      6           0.0282             nan      0.1000     0.0010
##      7           0.0273             nan      0.1000     0.0009
##      8           0.0266             nan      0.1000     0.0007
##      9           0.0260             nan      0.1000     0.0007
##     10           0.0255             nan      0.1000     0.0005
##     20           0.0233             nan      0.1000     0.0001
##     40           0.0226             nan      0.1000     -0.0000
##     60           0.0224             nan      0.1000     -0.0000
##     80           0.0224             nan      0.1000     -0.0000
##    100           0.0223             nan      0.1000     -0.0000
##    120           0.0223             nan      0.1000     -0.0000
##    140           0.0222             nan      0.1000     -0.0000
##    150           0.0222             nan      0.1000     -0.0000
##
## - Fold05.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold06.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0373             nan      0.1000     0.0026
##      2           0.0354             nan      0.1000     0.0018
##      3           0.0338             nan      0.1000     0.0017
##      4           0.0324             nan      0.1000     0.0012
##      5           0.0315             nan      0.1000     0.0009
##      6           0.0306             nan      0.1000     0.0009
##      7           0.0300             nan      0.1000     0.0006
##      8           0.0295             nan      0.1000     0.0005
##      9           0.0291             nan      0.1000     0.0004
```

```
##      10      0.0287      nan      0.1000      0.0004
##      20      0.0273      nan      0.1000      0.0001
##      40      0.0267      nan      0.1000      0.0000
##      60      0.0266      nan      0.1000      0.0000
##      80      0.0265      nan      0.1000     -0.0000
##     100      0.0265      nan      0.1000     -0.0000
##     120      0.0265      nan      0.1000     -0.0000
##     140      0.0265      nan      0.1000     -0.0000
##     150      0.0265      nan      0.1000     -0.0000
##
## - Fold06.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold06.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0369      nan      0.1000      0.0030
##      2      0.0345      nan      0.1000      0.0024
##      3      0.0328      nan      0.1000      0.0021
##      4      0.0312      nan      0.1000      0.0015
##      5      0.0300      nan      0.1000      0.0013
##      6      0.0289      nan      0.1000      0.0010
##      7      0.0281      nan      0.1000      0.0008
##      8      0.0274      nan      0.1000      0.0007
##      9      0.0268      nan      0.1000      0.0006
##     10      0.0263      nan      0.1000      0.0004
##     20      0.0241      nan      0.1000      0.0001
##     40      0.0233      nan      0.1000      0.0000
##     60      0.0231      nan      0.1000     -0.0000
##     80      0.0230      nan      0.1000     -0.0000
##    100      0.0229      nan      0.1000     -0.0000
##    120      0.0229      nan      0.1000     -0.0000
##    140      0.0229      nan      0.1000     -0.0000
##    150      0.0228      nan      0.1000     -0.0000
##
## - Fold06.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold06.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0367      nan      0.1000      0.0032
##      2      0.0342      nan      0.1000      0.0023
##      3      0.0323      nan      0.1000      0.0020
##      4      0.0308      nan      0.1000      0.0015
##      5      0.0296      nan      0.1000      0.0012
##      6      0.0285      nan      0.1000      0.0008
##      7      0.0276      nan      0.1000      0.0010
##      8      0.0268      nan      0.1000      0.0007
##      9      0.0262      nan      0.1000      0.0005
##     10      0.0258      nan      0.1000      0.0004
##     20      0.0237      nan      0.1000      0.0001
##     40      0.0229      nan      0.1000     -0.0000
##     60      0.0228      nan      0.1000     -0.0000
```

```
##      80      0.0227      nan      0.1000     -0.0000
##     100      0.0226      nan      0.1000     -0.0001
##     120      0.0226      nan      0.1000     -0.0000
##     140      0.0225      nan      0.1000     -0.0000
##     150      0.0225      nan      0.1000     -0.0001
##
## - Fold06.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold07.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.0378         nan         0.1000     0.0026
##      2         0.0359         nan         0.1000     0.0020
##      3         0.0346         nan         0.1000     0.0017
##      4         0.0331         nan         0.1000     0.0012
##      5         0.0320         nan         0.1000     0.0010
##      6         0.0311         nan         0.1000     0.0009
##      7         0.0303         nan         0.1000     0.0007
##      8         0.0296         nan         0.1000     0.0006
##      9         0.0293         nan         0.1000     0.0004
##     10         0.0289         nan         0.1000     0.0003
##     20         0.0274         nan         0.1000     0.0001
##     40         0.0268         nan         0.1000     0.0000
##     60         0.0265         nan         0.1000     0.0000
##     80         0.0265         nan         0.1000     -0.0000
##    100         0.0264         nan         0.1000     -0.0000
##    120         0.0264         nan         0.1000     -0.0000
##    140         0.0264         nan         0.1000     -0.0000
##    150         0.0264         nan         0.1000     -0.0000
##
## - Fold07.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold07.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.0375         nan         0.1000     0.0027
##      2         0.0352         nan         0.1000     0.0023
##      3         0.0332         nan         0.1000     0.0021
##      4         0.0316         nan         0.1000     0.0017
##      5         0.0304         nan         0.1000     0.0014
##      6         0.0292         nan         0.1000     0.0011
##      7         0.0284         nan         0.1000     0.0009
##      8         0.0277         nan         0.1000     0.0007
##      9         0.0271         nan         0.1000     0.0006
##     10         0.0266         nan         0.1000     0.0005
##     20         0.0243         nan         0.1000     0.0001
##     40         0.0233         nan         0.1000     0.0000
##     60         0.0231         nan         0.1000     -0.0000
##     80         0.0230         nan         0.1000     -0.0000
##    100         0.0229         nan         0.1000     -0.0000
##    120         0.0228         nan         0.1000     -0.0000
##    140         0.0228         nan         0.1000     -0.0000
```

```
##      150      0.0228      nan      0.1000     -0.0000
##
## - Fold07.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold07.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0373      nan      0.1000      0.0030
##      2      0.0349      nan      0.1000      0.0025
##      3      0.0329      nan      0.1000      0.0020
##      4      0.0312      nan      0.1000      0.0018
##      5      0.0298      nan      0.1000      0.0013
##      6      0.0286      nan      0.1000      0.0012
##      7      0.0276      nan      0.1000      0.0010
##      8      0.0269      nan      0.1000      0.0008
##      9      0.0263      nan      0.1000      0.0006
##     10      0.0258      nan      0.1000      0.0006
##     20      0.0236      nan      0.1000      0.0000
##     40      0.0229      nan      0.1000     -0.0000
##     60      0.0227      nan      0.1000     -0.0000
##     80      0.0226      nan      0.1000     -0.0000
##    100      0.0225      nan      0.1000     -0.0000
##    120      0.0225      nan      0.1000     -0.0000
##    140      0.0224      nan      0.1000     -0.0000
##    150      0.0224      nan      0.1000     -0.0000
##
## - Fold07.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold08.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0388      nan      0.1000      0.0025
##      2      0.0367      nan      0.1000      0.0020
##      3      0.0352      nan      0.1000      0.0017
##      4      0.0339      nan      0.1000      0.0014
##      5      0.0327      nan      0.1000      0.0012
##      6      0.0318      nan      0.1000      0.0009
##      7      0.0311      nan      0.1000      0.0007
##      8      0.0305      nan      0.1000      0.0005
##      9      0.0301      nan      0.1000      0.0005
##     10      0.0297      nan      0.1000      0.0003
##     20      0.0282      nan      0.1000      0.0001
##     40      0.0274      nan      0.1000      0.0000
##     60      0.0271      nan      0.1000     -0.0000
##     80      0.0271      nan      0.1000     -0.0000
##    100      0.0270      nan      0.1000     -0.0000
##    120      0.0270      nan      0.1000     -0.0000
##    140      0.0270      nan      0.1000     -0.0000
##    150      0.0270      nan      0.1000     -0.0000
##
## - Fold08.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
```



```
## + Fold08.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0381	nan	0.1000	0.0030
## 2	0.0355	nan	0.1000	0.0023
## 3	0.0337	nan	0.1000	0.0020
## 4	0.0320	nan	0.1000	0.0016
## 5	0.0308	nan	0.1000	0.0013
## 6	0.0296	nan	0.1000	0.0012
## 7	0.0287	nan	0.1000	0.0008
## 8	0.0279	nan	0.1000	0.0007
## 9	0.0273	nan	0.1000	0.0005
## 10	0.0267	nan	0.1000	0.0004
## 20	0.0245	nan	0.1000	0.0001
## 40	0.0237	nan	0.1000	-0.0000
## 60	0.0234	nan	0.1000	0.0000
## 80	0.0233	nan	0.1000	-0.0000
## 100	0.0233	nan	0.1000	-0.0000
## 120	0.0232	nan	0.1000	-0.0000
## 140	0.0232	nan	0.1000	-0.0000
## 150	0.0232	nan	0.1000	-0.0000

```
##
## - Fold08.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

```
## + Fold08.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0379	nan	0.1000	0.0032
## 2	0.0355	nan	0.1000	0.0020
## 3	0.0335	nan	0.1000	0.0019
## 4	0.0316	nan	0.1000	0.0013
## 5	0.0302	nan	0.1000	0.0014
## 6	0.0292	nan	0.1000	0.0011
## 7	0.0283	nan	0.1000	0.0008
## 8	0.0274	nan	0.1000	0.0008
## 9	0.0267	nan	0.1000	0.0006
## 10	0.0262	nan	0.1000	0.0005
## 20	0.0239	nan	0.1000	0.0000
## 40	0.0232	nan	0.1000	-0.0000
## 60	0.0231	nan	0.1000	-0.0000
## 80	0.0229	nan	0.1000	-0.0000
## 100	0.0229	nan	0.1000	-0.0000
## 120	0.0228	nan	0.1000	-0.0000
## 140	0.0228	nan	0.1000	-0.0000
## 150	0.0228	nan	0.1000	-0.0000

```
##
## - Fold08.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
```

```
## + Fold09.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0359	nan	0.1000	0.0023

```
##      2      0.0343      nan      0.1000      0.0017
##      3      0.0329      nan      0.1000      0.0015
##      4      0.0319      nan      0.1000      0.0013
##      5      0.0310      nan      0.1000      0.0010
##      6      0.0302      nan      0.1000      0.0007
##      7      0.0298      nan      0.1000      0.0006
##      8      0.0293      nan      0.1000      0.0005
##      9      0.0288      nan      0.1000      0.0004
##     10      0.0285      nan      0.1000      0.0004
##     20      0.0270      nan      0.1000      0.0000
##     40      0.0263      nan      0.1000      0.0000
##     60      0.0261      nan      0.1000     -0.0000
##     80      0.0261      nan      0.1000     -0.0000
##    100      0.0260      nan      0.1000     -0.0000
##    120      0.0260      nan      0.1000     -0.0000
##    140      0.0260      nan      0.1000     -0.0000
##    150      0.0260      nan      0.1000     -0.0000
##
## - Fold09.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold09.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0356      nan      0.1000      0.0026
##      2      0.0333      nan      0.1000      0.0021
##      3      0.0315      nan      0.1000      0.0015
##      4      0.0300      nan      0.1000      0.0013
##      5      0.0288      nan      0.1000      0.0013
##      6      0.0279      nan      0.1000      0.0008
##      7      0.0271      nan      0.1000      0.0007
##      8      0.0264      nan      0.1000      0.0005
##      9      0.0260      nan      0.1000      0.0004
##     10      0.0256      nan      0.1000      0.0005
##     20      0.0235      nan      0.1000      0.0001
##     40      0.0228      nan      0.1000      0.0000
##     60      0.0226      nan      0.1000     -0.0000
##     80      0.0225      nan      0.1000     -0.0000
##    100      0.0225      nan      0.1000     -0.0000
##    120      0.0224      nan      0.1000     -0.0000
##    140      0.0224      nan      0.1000     -0.0000
##    150      0.0224      nan      0.1000     -0.0000
##
## - Fold09.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold09.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0353      nan      0.1000      0.0029
##      2      0.0331      nan      0.1000      0.0022
##      3      0.0312      nan      0.1000      0.0017
##      4      0.0299      nan      0.1000      0.0014
##      5      0.0286      nan      0.1000      0.0011
```

```
##      6      0.0275      nan      0.1000      0.0011
##      7      0.0268      nan      0.1000      0.0007
##      8      0.0261      nan      0.1000      0.0007
##      9      0.0255      nan      0.1000      0.0005
##     10      0.0251      nan      0.1000      0.0004
##     20      0.0231      nan      0.1000      0.0001
##     40      0.0225      nan      0.1000      0.0000
##     60      0.0222      nan      0.1000      0.0000
##     80      0.0221      nan      0.1000     -0.0000
##    100      0.0221      nan      0.1000     -0.0000
##    120      0.0220      nan      0.1000     -0.0000
##    140      0.0220      nan      0.1000     -0.0001
##    150      0.0220      nan      0.1000     -0.0000
##
## - Fold09.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold10.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0365           nan      0.1000      0.0020
##      2           0.0349           nan      0.1000      0.0016
##      3           0.0334           nan      0.1000      0.0014
##      4           0.0322           nan      0.1000      0.0011
##      5           0.0313           nan      0.1000      0.0009
##      6           0.0306           nan      0.1000      0.0007
##      7           0.0300           nan      0.1000      0.0006
##      8           0.0295           nan      0.1000      0.0005
##      9           0.0291           nan      0.1000      0.0004
##     10           0.0288           nan      0.1000      0.0003
##     20           0.0274           nan      0.1000      0.0001
##     40           0.0267           nan      0.1000      0.0000
##     60           0.0265           nan      0.1000     -0.0001
##     80           0.0264           nan      0.1000     -0.0000
##    100           0.0264           nan      0.1000     -0.0000
##    120           0.0264           nan      0.1000     -0.0000
##    140           0.0264           nan      0.1000     -0.0000
##    150           0.0264           nan      0.1000     -0.0000
##
## - Fold10.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold10.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0361           nan      0.1000      0.0031
##      2           0.0341           nan      0.1000      0.0022
##      3           0.0325           nan      0.1000      0.0016
##      4           0.0309           nan      0.1000      0.0014
##      5           0.0297           nan      0.1000      0.0013
##      6           0.0289           nan      0.1000      0.0008
##      7           0.0280           nan      0.1000      0.0009
##      8           0.0273           nan      0.1000      0.0006
##      9           0.0268           nan      0.1000      0.0006
```

```
##      10      0.0263      nan      0.1000      0.0004
##      20      0.0242      nan      0.1000      0.0001
##      40      0.0235      nan      0.1000     -0.0000
##      60      0.0232      nan      0.1000     -0.0000
##      80      0.0231      nan      0.1000     -0.0000
##     100      0.0231      nan      0.1000     -0.0000
##     120      0.0230      nan      0.1000     -0.0000
##     140      0.0230      nan      0.1000     -0.0000
##     150      0.0230      nan      0.1000     -0.0000
##
## - Fold10.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold10.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0362           nan      0.1000    0.0027
##      2           0.0339           nan      0.1000    0.0022
##      3           0.0322           nan      0.1000    0.0019
##      4           0.0307           nan      0.1000    0.0019
##      5           0.0293           nan      0.1000    0.0013
##      6           0.0283           nan      0.1000    0.0010
##      7           0.0275           nan      0.1000    0.0006
##      8           0.0268           nan      0.1000    0.0006
##      9           0.0263           nan      0.1000    0.0006
##     10           0.0257           nan      0.1000    0.0003
##     20           0.0238           nan      0.1000    0.0001
##     40           0.0231           nan      0.1000   -0.0000
##     60           0.0230           nan      0.1000   -0.0000
##     80           0.0228           nan      0.1000   -0.0000
##    100           0.0228           nan      0.1000   -0.0000
##    120           0.0227           nan      0.1000   -0.0000
##    140           0.0226           nan      0.1000   -0.0000
##    150           0.0226           nan      0.1000   -0.0000
##
## - Fold10.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold01.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           0.0368           nan      0.1000    0.0025
##      2           0.0349           nan      0.1000    0.0019
##      3           0.0333           nan      0.1000    0.0015
##      4           0.0321           nan      0.1000    0.0011
##      5           0.0311           nan      0.1000    0.0010
##      6           0.0303           nan      0.1000    0.0008
##      7           0.0298           nan      0.1000    0.0007
##      8           0.0293           nan      0.1000    0.0006
##      9           0.0288           nan      0.1000    0.0005
##     10           0.0284           nan      0.1000    0.0004
##     20           0.0268           nan      0.1000    0.0000
##     40           0.0261           nan      0.1000    0.0000
##     60           0.0259           nan      0.1000    0.0000
```

```
##      80      0.0258      nan      0.1000     -0.0000
##     100      0.0258      nan      0.1000     -0.0000
##     120      0.0258      nan      0.1000     -0.0000
##     140      0.0258      nan      0.1000     -0.0000
##     150      0.0258      nan      0.1000     -0.0000
##
## - Fold01.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold01.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.0363         nan         0.1000     0.0028
##      2         0.0339         nan         0.1000     0.0024
##      3         0.0319         nan         0.1000     0.0020
##      4         0.0304         nan         0.1000     0.0014
##      5         0.0292         nan         0.1000     0.0011
##      6         0.0282         nan         0.1000     0.0010
##      7         0.0273         nan         0.1000     0.0006
##      8         0.0267         nan         0.1000     0.0006
##      9         0.0261         nan         0.1000     0.0005
##     10         0.0257         nan         0.1000     0.0004
##     20         0.0235         nan         0.1000     0.0001
##     40         0.0227         nan         0.1000     0.0000
##     60         0.0225         nan         0.1000    -0.0000
##     80         0.0225         nan         0.1000    -0.0000
##    100         0.0224         nan         0.1000    -0.0000
##    120         0.0224         nan         0.1000    -0.0000
##    140         0.0224         nan         0.1000    -0.0000
##    150         0.0223         nan         0.1000    -0.0000
##
## - Fold01.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold01.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.0362         nan         0.1000     0.0029
##      2         0.0339         nan         0.1000     0.0022
##      3         0.0318         nan         0.1000     0.0019
##      4         0.0304         nan         0.1000     0.0015
##      5         0.0290         nan         0.1000     0.0012
##      6         0.0279         nan         0.1000     0.0012
##      7         0.0270         nan         0.1000     0.0008
##      8         0.0264         nan         0.1000     0.0006
##      9         0.0258         nan         0.1000     0.0005
##     10         0.0253         nan         0.1000     0.0004
##     20         0.0232         nan         0.1000     0.0000
##     40         0.0225         nan         0.1000    -0.0000
##     60         0.0223         nan         0.1000    -0.0000
##     80         0.0222         nan         0.1000    -0.0000
##    100         0.0222         nan         0.1000    -0.0000
##    120         0.0222         nan         0.1000    -0.0000
##    140         0.0221         nan         0.1000    -0.0000
```

```
##      150      0.0221      nan      0.1000     -0.0000
##
## - Fold01.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold02.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0376      nan      0.1000      0.0020
##      2      0.0358      nan      0.1000      0.0020
##      3      0.0342      nan      0.1000      0.0016
##      4      0.0329      nan      0.1000      0.0014
##      5      0.0320      nan      0.1000      0.0011
##      6      0.0311      nan      0.1000      0.0008
##      7      0.0304      nan      0.1000      0.0007
##      8      0.0298      nan      0.1000      0.0006
##      9      0.0294      nan      0.1000      0.0004
##     10      0.0291      nan      0.1000      0.0003
##     20      0.0276      nan      0.1000      0.0000
##     40      0.0268      nan      0.1000      0.0000
##     60      0.0266      nan      0.1000     -0.0000
##     80      0.0265      nan      0.1000     -0.0000
##    100      0.0265      nan      0.1000     -0.0000
##    120      0.0265      nan      0.1000     -0.0000
##    140      0.0265      nan      0.1000     -0.0000
##    150      0.0265      nan      0.1000     -0.0000
##
## - Fold02.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold02.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0369      nan      0.1000      0.0026
##      2      0.0346      nan      0.1000      0.0020
##      3      0.0328      nan      0.1000      0.0017
##      4      0.0313      nan      0.1000      0.0017
##      5      0.0300      nan      0.1000      0.0013
##      6      0.0289      nan      0.1000      0.0010
##      7      0.0281      nan      0.1000      0.0009
##      8      0.0273      nan      0.1000      0.0007
##      9      0.0267      nan      0.1000      0.0006
##     10      0.0263      nan      0.1000      0.0004
##     20      0.0241      nan      0.1000      0.0001
##     40      0.0232      nan      0.1000     -0.0000
##     60      0.0229      nan      0.1000      0.0000
##     80      0.0228      nan      0.1000     -0.0000
##    100      0.0228      nan      0.1000      0.0000
##    120      0.0227      nan      0.1000     -0.0000
##    140      0.0227      nan      0.1000     -0.0000
##    150      0.0227      nan      0.1000     -0.0000
##
## - Fold02.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

```
## + Fold02.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0368	nan	0.1000	0.0031
## 2	0.0344	nan	0.1000	0.0028
## 3	0.0325	nan	0.1000	0.0019
## 4	0.0307	nan	0.1000	0.0016
## 5	0.0293	nan	0.1000	0.0014
## 6	0.0283	nan	0.1000	0.0011
## 7	0.0274	nan	0.1000	0.0007
## 8	0.0267	nan	0.1000	0.0006
## 9	0.0262	nan	0.1000	0.0005
## 10	0.0256	nan	0.1000	0.0004
## 20	0.0235	nan	0.1000	0.0001
## 40	0.0228	nan	0.1000	-0.0000
## 60	0.0226	nan	0.1000	-0.0000
## 80	0.0224	nan	0.1000	-0.0000
## 100	0.0224	nan	0.1000	-0.0000
## 120	0.0223	nan	0.1000	-0.0000
## 140	0.0223	nan	0.1000	-0.0000
## 150	0.0222	nan	0.1000	-0.0000

```
##
## - Fold02.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold03.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0382	nan	0.1000	0.0022
## 2	0.0362	nan	0.1000	0.0019
## 3	0.0347	nan	0.1000	0.0015
## 4	0.0333	nan	0.1000	0.0012
## 5	0.0324	nan	0.1000	0.0010
## 6	0.0316	nan	0.1000	0.0009
## 7	0.0309	nan	0.1000	0.0007
## 8	0.0304	nan	0.1000	0.0005
## 9	0.0300	nan	0.1000	0.0005
## 10	0.0297	nan	0.1000	0.0004
## 20	0.0281	nan	0.1000	0.0000
## 40	0.0273	nan	0.1000	-0.0000
## 60	0.0270	nan	0.1000	-0.0000
## 80	0.0269	nan	0.1000	-0.0000
## 100	0.0269	nan	0.1000	-0.0000
## 120	0.0269	nan	0.1000	-0.0000
## 140	0.0269	nan	0.1000	-0.0000
## 150	0.0269	nan	0.1000	-0.0000

```
##
## - Fold03.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold03.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0377	nan	0.1000	0.0028

```
##      2      0.0351      nan      0.1000      0.0021
##      3      0.0332      nan      0.1000      0.0020
##      4      0.0317      nan      0.1000      0.0015
##      5      0.0303      nan      0.1000      0.0011
##      6      0.0293      nan      0.1000      0.0011
##      7      0.0284      nan      0.1000      0.0008
##      8      0.0277      nan      0.1000      0.0007
##      9      0.0271      nan      0.1000      0.0006
##     10      0.0266      nan      0.1000      0.0005
##     20      0.0244      nan      0.1000      0.0001
##     40      0.0235      nan      0.1000      0.0000
##     60      0.0233      nan      0.1000     -0.0000
##     80      0.0231      nan      0.1000     -0.0000
##    100      0.0231      nan      0.1000     -0.0000
##    120      0.0230      nan      0.1000     -0.0000
##    140      0.0230      nan      0.1000     -0.0000
##    150      0.0230      nan      0.1000     -0.0000
##
## - Fold03.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold03.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0375      nan      0.1000      0.0031
##      2      0.0349      nan      0.1000      0.0028
##      3      0.0329      nan      0.1000      0.0020
##      4      0.0315      nan      0.1000      0.0016
##      5      0.0301      nan      0.1000      0.0013
##      6      0.0289      nan      0.1000      0.0009
##      7      0.0279      nan      0.1000      0.0009
##      8      0.0273      nan      0.1000      0.0006
##      9      0.0267      nan      0.1000      0.0005
##     10      0.0262      nan      0.1000      0.0006
##     20      0.0239      nan      0.1000      0.0001
##     40      0.0231      nan      0.1000      0.0000
##     60      0.0229      nan      0.1000     -0.0000
##     80      0.0228      nan      0.1000     -0.0000
##    100      0.0227      nan      0.1000     -0.0000
##    120      0.0227      nan      0.1000     -0.0000
##    140      0.0226      nan      0.1000     -0.0000
##    150      0.0226      nan      0.1000     -0.0000
##
## - Fold03.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold04.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0381      nan      0.1000      0.0024
##      2      0.0366      nan      0.1000      0.0018
##      3      0.0349      nan      0.1000      0.0013
##      4      0.0336      nan      0.1000      0.0013
##      5      0.0326      nan      0.1000      0.0010
```



```
##      6      0.0319      nan      0.1000      0.0007
##      7      0.0314      nan      0.1000      0.0006
##      8      0.0309      nan      0.1000      0.0005
##      9      0.0305      nan      0.1000      0.0004
##     10      0.0303      nan      0.1000      0.0003
##     20      0.0288      nan      0.1000      0.0000
##     40      0.0281      nan      0.1000      0.0000
##     60      0.0278      nan      0.1000      0.0000
##     80      0.0278      nan      0.1000     -0.0000
##    100      0.0278      nan      0.1000     -0.0000
##    120      0.0278      nan      0.1000     -0.0000
##    140      0.0278      nan      0.1000     -0.0000
##    150      0.0278      nan      0.1000     -0.0000
##
## - Fold04.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold04.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.0378           nan       0.1000     0.0029
##      2         0.0355           nan       0.1000     0.0026
##      3         0.0337           nan       0.1000     0.0017
##      4         0.0322           nan       0.1000     0.0015
##      5         0.0309           nan       0.1000     0.0012
##      6         0.0299           nan       0.1000     0.0010
##      7         0.0291           nan       0.1000     0.0008
##      8         0.0284           nan       0.1000     0.0006
##      9         0.0278           nan       0.1000     0.0006
##     10         0.0274           nan       0.1000     0.0004
##     20         0.0252           nan       0.1000     0.0001
##     40         0.0243           nan       0.1000     0.0000
##     60         0.0241           nan       0.1000     0.0000
##     80         0.0240           nan       0.1000     -0.0000
##    100         0.0239           nan       0.1000     -0.0000
##    120         0.0239           nan       0.1000     -0.0000
##    140         0.0239           nan       0.1000     -0.0000
##    150         0.0238           nan       0.1000     -0.0000
##
## - Fold04.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold04.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.0376           nan       0.1000     0.0025
##      2         0.0353           nan       0.1000     0.0023
##      3         0.0332           nan       0.1000     0.0017
##      4         0.0317           nan       0.1000     0.0014
##      5         0.0304           nan       0.1000     0.0011
##      6         0.0293           nan       0.1000     0.0012
##      7         0.0286           nan       0.1000     0.0008
##      8         0.0279           nan       0.1000     0.0008
##      9         0.0274           nan       0.1000     0.0005
```

```
##      10      0.0268      nan      0.1000      0.0005
##      20      0.0247      nan      0.1000      0.0001
##      40      0.0239      nan      0.1000     -0.0001
##      60      0.0237      nan      0.1000     -0.0000
##      80      0.0236      nan      0.1000     -0.0000
##     100      0.0236      nan      0.1000     -0.0000
##     120      0.0235      nan      0.1000     -0.0000
##     140      0.0234      nan      0.1000     -0.0000
##     150      0.0234      nan      0.1000     -0.0000
##
## - Fold04.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold05.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0366      nan      0.1000      0.0024
##      2      0.0348      nan      0.1000      0.0017
##      3      0.0335      nan      0.1000      0.0016
##      4      0.0325      nan      0.1000      0.0014
##      5      0.0315      nan      0.1000      0.0011
##      6      0.0305      nan      0.1000      0.0009
##      7      0.0298      nan      0.1000      0.0006
##      8      0.0293      nan      0.1000      0.0005
##      9      0.0290      nan      0.1000      0.0004
##     10      0.0285      nan      0.1000      0.0003
##     20      0.0270      nan      0.1000      0.0001
##     40      0.0263      nan      0.1000     -0.0000
##     60      0.0261      nan      0.1000     -0.0000
##     80      0.0260      nan      0.1000     -0.0000
##    100      0.0260      nan      0.1000     -0.0000
##    120      0.0260      nan      0.1000     -0.0000
##    140      0.0260      nan      0.1000     -0.0000
##    150      0.0260      nan      0.1000     -0.0000
##
## - Fold05.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold05.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0363      nan      0.1000      0.0031
##      2      0.0339      nan      0.1000      0.0025
##      3      0.0320      nan      0.1000      0.0018
##      4      0.0304      nan      0.1000      0.0015
##      5      0.0292      nan      0.1000      0.0011
##      6      0.0282      nan      0.1000      0.0008
##      7      0.0273      nan      0.1000      0.0008
##      8      0.0267      nan      0.1000      0.0006
##      9      0.0261      nan      0.1000      0.0006
##     10      0.0256      nan      0.1000      0.0004
##     20      0.0236      nan      0.1000      0.0001
##     40      0.0228      nan      0.1000     -0.0000
##     60      0.0226      nan      0.1000     -0.0000
```

```

##      80      0.0225      nan      0.1000      0.0000
##     100      0.0224      nan      0.1000     -0.0000
##     120      0.0224      nan      0.1000     -0.0000
##     140      0.0223      nan      0.1000     -0.0000
##     150      0.0223      nan      0.1000      0.0000
##
## - Fold05.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold05.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1          0.0359             nan      0.1000    0.0028
##      2          0.0334             nan      0.1000    0.0024
##      3          0.0316             nan      0.1000    0.0019
##      4          0.0300             nan      0.1000    0.0017
##      5          0.0287             nan      0.1000    0.0013
##      6          0.0276             nan      0.1000    0.0011
##      7          0.0267             nan      0.1000    0.0008
##      8          0.0260             nan      0.1000    0.0006
##      9          0.0254             nan      0.1000    0.0005
##     10          0.0249             nan      0.1000    0.0005
##     20          0.0229             nan      0.1000    0.0001
##     40          0.0223             nan      0.1000    0.0000
##     60          0.0221             nan      0.1000   -0.0000
##     80          0.0220             nan      0.1000   -0.0000
##    100          0.0220             nan      0.1000   -0.0000
##    120          0.0219             nan      0.1000   -0.0000
##    140          0.0219             nan      0.1000   -0.0000
##    150          0.0219             nan      0.1000   -0.0000
##
## - Fold05.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold06.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1          0.0372             nan      0.1000    0.0024
##      2          0.0355             nan      0.1000    0.0020
##      3          0.0340             nan      0.1000    0.0017
##      4          0.0325             nan      0.1000    0.0011
##      5          0.0315             nan      0.1000    0.0011
##      6          0.0307             nan      0.1000    0.0009
##      7          0.0300             nan      0.1000    0.0007
##      8          0.0294             nan      0.1000    0.0006
##      9          0.0289             nan      0.1000    0.0005
##     10          0.0285             nan      0.1000    0.0003
##     20          0.0270             nan      0.1000    0.0000
##     40          0.0262             nan      0.1000    0.0000
##     60          0.0260             nan      0.1000   -0.0000
##     80          0.0260             nan      0.1000   -0.0000
##    100          0.0259             nan      0.1000   -0.0000
##    120          0.0259             nan      0.1000   -0.0000
##    140          0.0259             nan      0.1000   -0.0000

```

```
##      150      0.0259      nan      0.1000     -0.0000
##
## - Fold06.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold06.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0367      nan      0.1000      0.0026
##      2      0.0342      nan      0.1000      0.0023
##      3      0.0324      nan      0.1000      0.0018
##      4      0.0307      nan      0.1000      0.0016
##      5      0.0295      nan      0.1000      0.0013
##      6      0.0283      nan      0.1000      0.0010
##      7      0.0275      nan      0.1000      0.0007
##      8      0.0267      nan      0.1000      0.0008
##      9      0.0261      nan      0.1000      0.0005
##     10      0.0256      nan      0.1000      0.0005
##     20      0.0235      nan      0.1000      0.0001
##     40      0.0225      nan      0.1000      0.0000
##     60      0.0223      nan      0.1000     -0.0000
##     80      0.0222      nan      0.1000      0.0000
##    100      0.0221      nan      0.1000     -0.0000
##    120      0.0221      nan      0.1000     -0.0000
##    140      0.0220      nan      0.1000     -0.0000
##    150      0.0220      nan      0.1000     -0.0000
##
## - Fold06.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold06.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0366      nan      0.1000      0.0031
##      2      0.0342      nan      0.1000      0.0019
##      3      0.0324      nan      0.1000      0.0018
##      4      0.0308      nan      0.1000      0.0016
##      5      0.0292      nan      0.1000      0.0015
##      6      0.0280      nan      0.1000      0.0009
##      7      0.0270      nan      0.1000      0.0011
##      8      0.0262      nan      0.1000      0.0007
##      9      0.0256      nan      0.1000      0.0005
##     10      0.0250      nan      0.1000      0.0007
##     20      0.0228      nan      0.1000      0.0001
##     40      0.0221      nan      0.1000     -0.0001
##     60      0.0219      nan      0.1000     -0.0000
##     80      0.0218      nan      0.1000     -0.0000
##    100      0.0217      nan      0.1000     -0.0000
##    120      0.0217      nan      0.1000     -0.0000
##    140      0.0216      nan      0.1000     -0.0000
##    150      0.0216      nan      0.1000     -0.0000
##
## - Fold06.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
```

```
## + Fold07.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0378	nan	0.1000	0.0026
## 2	0.0359	nan	0.1000	0.0021
## 3	0.0343	nan	0.1000	0.0015
## 4	0.0331	nan	0.1000	0.0013
## 5	0.0320	nan	0.1000	0.0011
## 6	0.0313	nan	0.1000	0.0009
## 7	0.0306	nan	0.1000	0.0007
## 8	0.0300	nan	0.1000	0.0005
## 9	0.0295	nan	0.1000	0.0005
## 10	0.0292	nan	0.1000	0.0003
## 20	0.0277	nan	0.1000	0.0001
## 40	0.0270	nan	0.1000	-0.0000
## 60	0.0268	nan	0.1000	0.0000
## 80	0.0267	nan	0.1000	-0.0000
## 100	0.0267	nan	0.1000	-0.0000
## 120	0.0267	nan	0.1000	-0.0000
## 140	0.0267	nan	0.1000	-0.0000
## 150	0.0267	nan	0.1000	-0.0000

```
##
## - Fold07.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
```

```
## + Fold07.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0375	nan	0.1000	0.0029
## 2	0.0352	nan	0.1000	0.0026
## 3	0.0334	nan	0.1000	0.0021
## 4	0.0317	nan	0.1000	0.0016
## 5	0.0307	nan	0.1000	0.0013
## 6	0.0296	nan	0.1000	0.0010
## 7	0.0288	nan	0.1000	0.0010
## 8	0.0280	nan	0.1000	0.0008
## 9	0.0273	nan	0.1000	0.0007
## 10	0.0268	nan	0.1000	0.0005
## 20	0.0245	nan	0.1000	0.0000
## 40	0.0237	nan	0.1000	-0.0000
## 60	0.0235	nan	0.1000	-0.0000
## 80	0.0234	nan	0.1000	-0.0000
## 100	0.0233	nan	0.1000	-0.0000
## 120	0.0233	nan	0.1000	-0.0000
## 140	0.0232	nan	0.1000	-0.0000
## 150	0.0232	nan	0.1000	-0.0000

```
##
## - Fold07.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
```

```
## + Fold07.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	0.0377	nan	0.1000	0.0030

```
##      2      0.0351      nan      0.1000      0.0024
##      3      0.0330      nan      0.1000      0.0017
##      4      0.0313      nan      0.1000      0.0016
##      5      0.0300      nan      0.1000      0.0014
##      6      0.0289      nan      0.1000      0.0010
##      7      0.0281      nan      0.1000      0.0007
##      8      0.0274      nan      0.1000      0.0007
##      9      0.0267      nan      0.1000      0.0006
##     10      0.0262      nan      0.1000      0.0004
##     20      0.0240      nan      0.1000      0.0001
##     40      0.0234      nan      0.1000     -0.0000
##     60      0.0232      nan      0.1000     -0.0000
##     80      0.0231      nan      0.1000     -0.0000
##    100      0.0230      nan      0.1000     -0.0000
##    120      0.0229      nan      0.1000     -0.0000
##    140      0.0229      nan      0.1000     -0.0000
##    150      0.0229      nan      0.1000     -0.0000
##
## - Fold07.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold08.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0379      nan      0.1000      0.0025
##      2      0.0360      nan      0.1000      0.0017
##      3      0.0344      nan      0.1000      0.0017
##      4      0.0332      nan      0.1000      0.0013
##      5      0.0322      nan      0.1000      0.0010
##      6      0.0314      nan      0.1000      0.0008
##      7      0.0309      nan      0.1000      0.0006
##      8      0.0304      nan      0.1000      0.0005
##      9      0.0299      nan      0.1000      0.0004
##     10      0.0296      nan      0.1000      0.0004
##     20      0.0280      nan      0.1000      0.0000
##     40      0.0272      nan      0.1000     -0.0000
##     60      0.0270      nan      0.1000     -0.0000
##     80      0.0269      nan      0.1000     -0.0000
##    100      0.0269      nan      0.1000     -0.0000
##    120      0.0269      nan      0.1000     -0.0000
##    140      0.0269      nan      0.1000     -0.0000
##    150      0.0269      nan      0.1000     -0.0000
##
## - Fold08.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold08.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0373      nan      0.1000      0.0030
##      2      0.0351      nan      0.1000      0.0024
##      3      0.0332      nan      0.1000      0.0016
##      4      0.0317      nan      0.1000      0.0015
##      5      0.0305      nan      0.1000      0.0009
```

```

##      6      0.0295      nan      0.1000      0.0010
##      7      0.0286      nan      0.1000      0.0010
##      8      0.0279      nan      0.1000      0.0006
##      9      0.0273      nan      0.1000      0.0006
##     10      0.0268      nan      0.1000      0.0005
##     20      0.0245      nan      0.1000      0.0000
##     40      0.0236      nan      0.1000      0.0000
##     60      0.0233      nan      0.1000     -0.0000
##     80      0.0233      nan      0.1000     -0.0000
##    100      0.0232      nan      0.1000     -0.0000
##    120      0.0232      nan      0.1000     -0.0000
##    140      0.0231      nan      0.1000     -0.0000
##    150      0.0231      nan      0.1000     -0.0000
##
## - Fold08.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold08.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.0370           nan      0.1000     0.0026
##      2         0.0346           nan      0.1000     0.0021
##      3         0.0328           nan      0.1000     0.0018
##      4         0.0313           nan      0.1000     0.0017
##      5         0.0299           nan      0.1000     0.0015
##      6         0.0288           nan      0.1000     0.0012
##      7         0.0279           nan      0.1000     0.0006
##      8         0.0272           nan      0.1000     0.0006
##      9         0.0265           nan      0.1000     0.0006
##     10         0.0260           nan      0.1000     0.0003
##     20         0.0239           nan      0.1000     0.0001
##     40         0.0233           nan      0.1000     0.0000
##     60         0.0230           nan      0.1000     -0.0000
##     80         0.0229           nan      0.1000     -0.0000
##    100         0.0229           nan      0.1000     -0.0000
##    120         0.0227           nan      0.1000     -0.0000
##    140         0.0227           nan      0.1000     -0.0000
##    150         0.0227           nan      0.1000     -0.0000
##
## - Fold08.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold09.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.0370           nan      0.1000     0.0020
##      2         0.0353           nan      0.1000     0.0019
##      3         0.0337           nan      0.1000     0.0018
##      4         0.0324           nan      0.1000     0.0012
##      5         0.0314           nan      0.1000     0.0010
##      6         0.0306           nan      0.1000     0.0008
##      7         0.0299           nan      0.1000     0.0007
##      8         0.0295           nan      0.1000     0.0005
##      9         0.0291           nan      0.1000     0.0005

```

```
##      10      0.0286      nan      0.1000      0.0004
##      20      0.0270      nan      0.1000      0.0001
##      40      0.0263      nan      0.1000      0.0000
##      60      0.0262      nan      0.1000      0.0000
##      80      0.0261      nan      0.1000     -0.0000
##     100      0.0261      nan      0.1000     -0.0000
##     120      0.0261      nan      0.1000     -0.0000
##     140      0.0261      nan      0.1000     -0.0000
##     150      0.0260      nan      0.1000     -0.0000
##
## - Fold09.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold09.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0368      nan      0.1000      0.0031
##      2      0.0346      nan      0.1000      0.0022
##      3      0.0326      nan      0.1000      0.0018
##      4      0.0311      nan      0.1000      0.0017
##      5      0.0298      nan      0.1000      0.0015
##      6      0.0286      nan      0.1000      0.0013
##      7      0.0278      nan      0.1000      0.0009
##      8      0.0271      nan      0.1000      0.0007
##      9      0.0264      nan      0.1000      0.0006
##     10      0.0259      nan      0.1000      0.0005
##     20      0.0236      nan      0.1000      0.0001
##     40      0.0227      nan      0.1000     -0.0000
##     60      0.0226      nan      0.1000     -0.0000
##     80      0.0224      nan      0.1000     -0.0000
##    100      0.0223      nan      0.1000     -0.0000
##    120      0.0223      nan      0.1000     -0.0000
##    140      0.0223      nan      0.1000     -0.0000
##    150      0.0223      nan      0.1000     -0.0000
##
## - Fold09.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold09.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      0.0365      nan      0.1000      0.0029
##      2      0.0340      nan      0.1000      0.0024
##      3      0.0320      nan      0.1000      0.0018
##      4      0.0304      nan      0.1000      0.0014
##      5      0.0291      nan      0.1000      0.0013
##      6      0.0280      nan      0.1000      0.0010
##      7      0.0271      nan      0.1000      0.0008
##      8      0.0264      nan      0.1000      0.0006
##      9      0.0257      nan      0.1000      0.0006
##     10      0.0252      nan      0.1000      0.0005
##     20      0.0231      nan      0.1000      0.0000
##     40      0.0224      nan      0.1000      0.0000
##     60      0.0222      nan      0.1000     -0.0000
```



```

##      80      0.0221      nan      0.1000     -0.0000
##     100      0.0220      nan      0.1000     -0.0000
##     120      0.0220      nan      0.1000     -0.0000
##     140      0.0219      nan      0.1000     -0.0000
##     150      0.0219      nan      0.1000     -0.0000
##
## - Fold09.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## + Fold10.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.0371           nan      0.1000    0.0024
##      2         0.0355           nan      0.1000    0.0019
##      3         0.0338           nan      0.1000    0.0017
##      4         0.0325           nan      0.1000    0.0012
##      5         0.0314           nan      0.1000    0.0009
##      6         0.0306           nan      0.1000    0.0008
##      7         0.0299           nan      0.1000    0.0006
##      8         0.0295           nan      0.1000    0.0005
##      9         0.0291           nan      0.1000    0.0004
##     10         0.0287           nan      0.1000    0.0004
##     20         0.0274           nan      0.1000    0.0000
##     40         0.0266           nan      0.1000    0.0000
##     60         0.0264           nan      0.1000    0.0000
##     80         0.0263           nan      0.1000   -0.0000
##    100         0.0263           nan      0.1000   -0.0000
##    120         0.0263           nan      0.1000   -0.0000
##    140         0.0263           nan      0.1000   -0.0000
##    150         0.0263           nan      0.1000   -0.0000
##
## - Fold10.Rep3: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150
## + Fold10.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.0364           nan      0.1000    0.0028
##      2         0.0343           nan      0.1000    0.0022
##      3         0.0324           nan      0.1000    0.0019
##      4         0.0308           nan      0.1000    0.0014
##      5         0.0296           nan      0.1000    0.0012
##      6         0.0286           nan      0.1000    0.0009
##      7         0.0278           nan      0.1000    0.0008
##      8         0.0270           nan      0.1000    0.0007
##      9         0.0263           nan      0.1000    0.0005
##     10         0.0258           nan      0.1000    0.0005
##     20         0.0238           nan      0.1000    0.0001
##     40         0.0230           nan      0.1000   -0.0000
##     60         0.0229           nan      0.1000   -0.0000
##     80         0.0228           nan      0.1000   -0.0000
##    100         0.0227           nan      0.1000   -0.0000
##    120         0.0227           nan      0.1000   -0.0000
##    140         0.0227           nan      0.1000   -0.0000

```

```
##      150      0.0226      nan      0.1000     -0.0000
##
## - Fold10.Rep3: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150
## + Fold10.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0366      nan      0.1000      0.0031
##      2      0.0343      nan      0.1000      0.0025
##      3      0.0324      nan      0.1000      0.0019
##      4      0.0308      nan      0.1000      0.0016
##      5      0.0293      nan      0.1000      0.0014
##      6      0.0282      nan      0.1000      0.0010
##      7      0.0273      nan      0.1000      0.0010
##      8      0.0265      nan      0.1000      0.0007
##      9      0.0260      nan      0.1000      0.0006
##     10      0.0255      nan      0.1000      0.0005
##     20      0.0234      nan      0.1000      0.0000
##     40      0.0228      nan      0.1000     -0.0000
##     60      0.0226      nan      0.1000     -0.0000
##     80      0.0225      nan      0.1000     -0.0000
##    100      0.0224      nan      0.1000     -0.0000
##    120      0.0224      nan      0.1000     -0.0000
##    140      0.0223      nan      0.1000     -0.0000
##    150      0.0223      nan      0.1000     -0.0000
##
## - Fold10.Rep3: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150
## Aggregating results
## Selecting tuning parameters
## Fitting n.trees = 100, interaction.depth = 2, shrinkage = 0.1, n.minobsinnode = 10 on full training set
## Iter      TrainDeviance      ValidDeviance      StepSize      Improve
##      1      0.0368      nan      0.1000      0.0032
##      2      0.0345      nan      0.1000      0.0022
##      3      0.0326      nan      0.1000      0.0018
##      4      0.0311      nan      0.1000      0.0015
##      5      0.0300      nan      0.1000      0.0013
##      6      0.0289      nan      0.1000      0.0011
##      7      0.0280      nan      0.1000      0.0009
##      8      0.0275      nan      0.1000      0.0006
##      9      0.0269      nan      0.1000      0.0006
##     10      0.0264      nan      0.1000      0.0006
##     20      0.0242      nan      0.1000      0.0001
##     40      0.0233      nan      0.1000      0.0000
##     60      0.0230      nan      0.1000     -0.0000
##     80      0.0230      nan      0.1000     -0.0000
##    100      0.0229      nan      0.1000     -0.0000
```

```
# Predicting in the test dataset
x.boo.pred <- predict(mod_GBM, newdata=test_set)
f<-as.matrix(x.boo.pred)
#pred_prob2 <- predict(x.cf, test_set, type = "response")

##Evaluating Logistic Regression Model
# Converting from probability to actual output
pred_class6 <- ifelse(x.boo.pred >= 0.017, "1", "0")

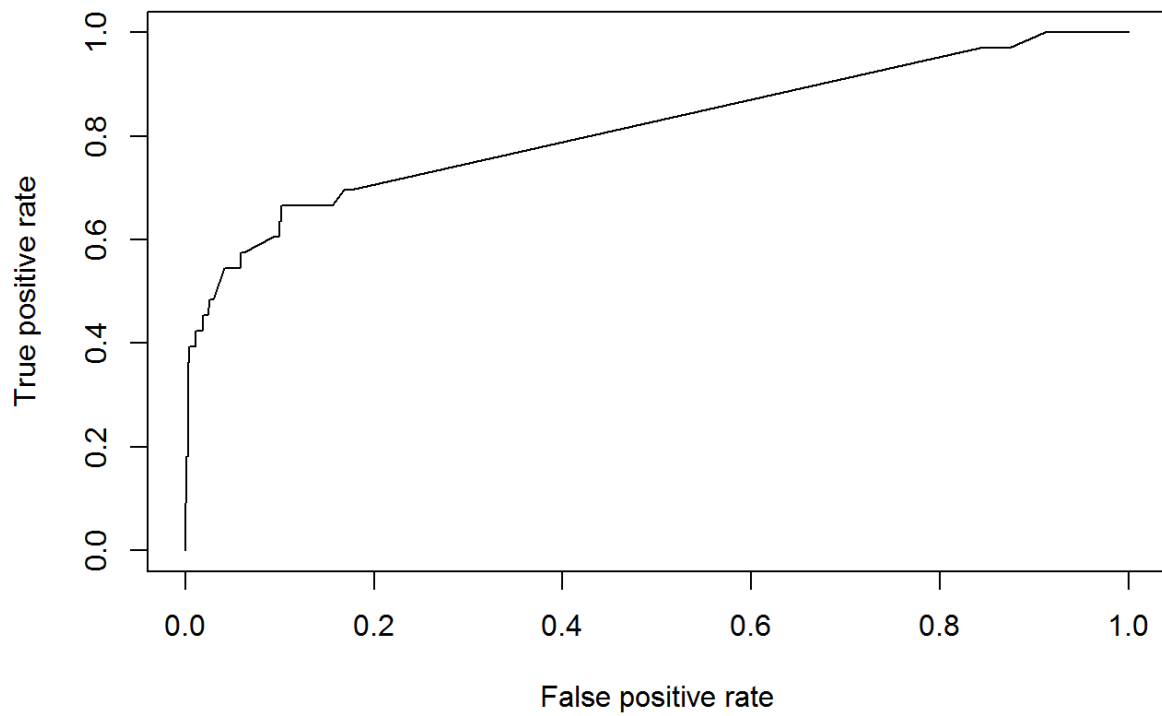
# Generating the classification table
ctab_test6 <- table(test_set$IC50, pred_class6)
ctab_test6
```

```
##      pred_class6
##      0    1
## YES 643 124
## NO   6   10
```

```
#Accuracy = (TP + TN)/(TN + FP + FN + TP)
# Accuracy in Test dataset
accuracy_test6 <- sum(diag(ctab_test6))/sum(ctab_test6)*100
accuracy_test6
```

```
## [1] 83.39719
```

```
library(ROCR)
p6 <- predict(mod_GBM, newdata=valid_set)
pr6 <- prediction(p6, valid_set$IC50)
prf6 <- performance(pr6, measure = "tpr", x.measure = "fpr")
plot(prf6)
```



```
auc <- performance(pr6, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.8149134
```

```
auc_result[8,1]<- auc
```