MODEL 1 BAGGING

REY P. PENDANG

2022-12-16

LOAD PACKAGES

```
library(dplyr)       # for data wrangling
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)     # for awesome plotting
library(doParallel)  # for parallel backend to foreach
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
library(foreach)     # for parallel processing with for loops
library(rsample)     # for creating our train-test splits
library(tidyverse)   # for filtering
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v tibble  3.1.8     v purrr   0.3.5
## v tidyr   1.2.1     v stringr 1.4.1
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x purrr::accumulate() masks foreach::accumulate()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## x purrr::when()       masks foreach::when()
```

```
library(readr)       #load dataset
library(caret)       # for general model fitting
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
```

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

```
library(rpart)       # for fitting decision trees
library(ipred)       # for fitting bagged decision trees
library(ROCR)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

LOAD THE DATA SET

```
set.seed(123) # for reproducibility
dt<- read_csv("normalRad.csv")
```

```
## Rows: 197 Columns: 431
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (1): Institution
## dbl (430): Failure.binary, Failure, Entropy_cooc.W.ADC, GLNU_align.H.PET, Mi...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

SPLIT TRAINING AND TESTING

```
for_splitted  <-  sample(1:nrow(dt), round(nrow(dt) * 0.8))
traindt <- dt[for_splitted,]
radiomicsdata_test  <- dt[-for_splitted,]
```

BAGGING

```
set.seed(123)
bagging_model1 <- bagging(
  formula = Failure.binary ~ .,
  data = traindt,
  nbagg = 200,
  coob = TRUE,
  control = rpart.control(minsplit = 2, cp = 0)
)

bagging_model1
```

```
##
## Bagging regression trees with 200 bootstrap replications
```

```
##
## Call: bagging.data.frame(formula = Failure.binary ~ ., data = traindt,
##     nbagg = 200, coob = TRUE, control = rpart.control(minsplit = 2,
##         cp = 0))
##
## Out-of-bag estimate of root mean squared error:  0.2945
```

Out-of-bag estimate of root mean squared error is 0.2945.

Bagging Model 2

```
set.seed(123)
bagging_model2 <- train(
  Failure.binary ~ .,
  data = traindt,
  method = "treebag",
  trControl = trainControl(method = "cv", number = 10),
  nbagg = 200,
  control = rpart.control(minsplit = 2, cp = 0)
)
bagging_model2
```

```
## Bagged CART
##
## 158 samples
## 430 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 142, 142, 142, 142, 142, 143, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.2875722  0.6208728  0.1688396
```

Used 10 cross fold validation. And found out that the RMSE value is 0.2875. Using RMSE, the best model will have lowest RMSE. Thus, Bagging Model 2 is better than Bagging Model 1 since model 2 RMSE is 0.2875 which is less than 0.2945 in model 1.

This part will create a parallel socket cluster using 8.

```
cl <- makeCluster(8)
```

Register the parallel backend.

```
registerDoParallel(cl)
```

```
predictions <- foreach(
  icount(160),
  .packages = "rpart",
  .combine = cbind
) %dopar% {
  index <- sample(nrow(traindt), replace = TRUE)
  trainDF_boot <- traindt[index, ]
  bagged_tree <- rpart(
    Failure.binary ~ .,
    control = rpart.control(minsplit = 2, cp = 0),
```

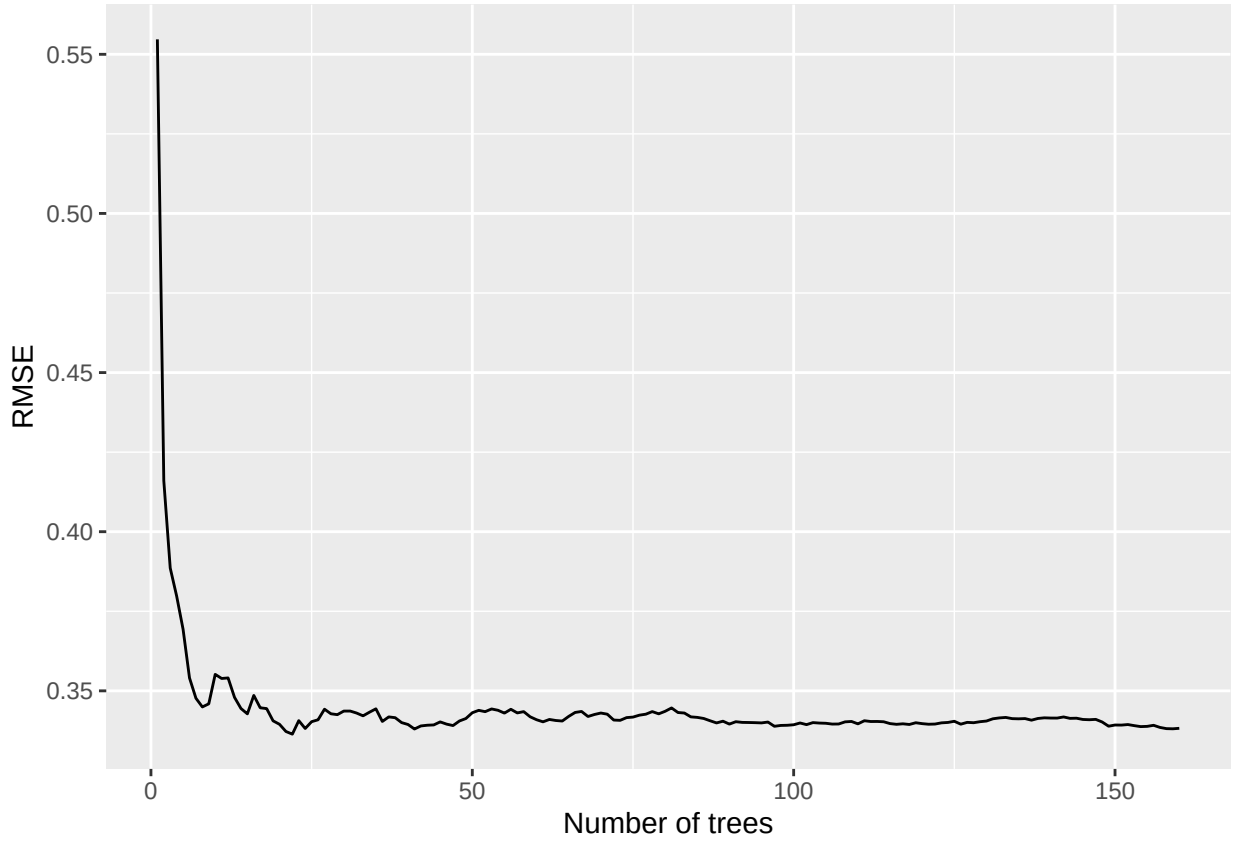
```
    data = trainDF_boot
  )

  predict(bagged_tree, newdata = radiomicsdata_test)
}

predictions[1:5, 1:7]
```

```
##   result.1 result.2 result.3 result.4 result.5 result.6 result.7
## 1        0        0        0        0        0        0        0
## 2        1        1        1        1        0        0        1
## 3        1        1        1        1        1        1        1
## 4        1        1        1        1        1        1        1
## 5        0        0        0        0        0        0        0
```

```
predictions %>%
  as.data.frame() %>%
  mutate(
    observation = 1:n(),
    actual = radiomicsdata_test$Failure.binary) %>%
  tidyr::gather(tree,
                predicted,
                -c(observation,
                   actual)) %>%
  group_by(observation) %>%
  mutate(tree = stringr::str_extract(tree, '\\d+') %>% as.numeric()) %>%
  ungroup() %>%
  arrange(observation, tree) %>%
  group_by(observation) %>%
  mutate(avg_prediction = cummean(predicted)) %>%
  group_by(tree) %>%
  summarize(RMSE = RMSE(avg_prediction, actual)) %>%
  ggplot(aes(tree, RMSE)) +
  geom_line() +
  xlab('Number of trees')
```



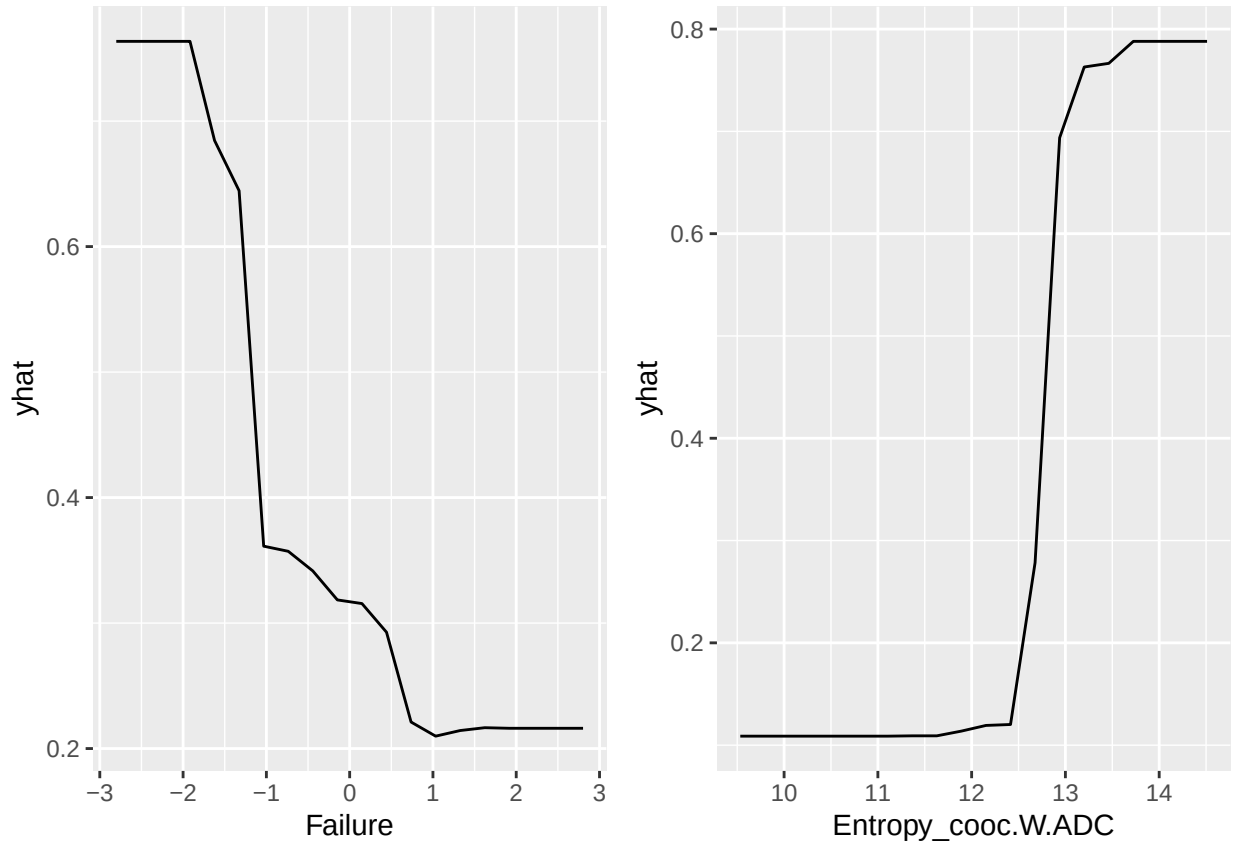
Error stabilizing graphat 50 to 75 number of trees which implies there is no gain for additional trees in making the model.

Construct partial dependence plots

```
plot_1 <- pdp::partial(
  bagging_model2,
  pred.var = names(dt)[3],
  grid.resolution = 20
) %>%
  autoplot()

plot_2 <- pdp::partial(
  bagging_model2,
  pred.var = names(dt)[4],
  grid.resolution = 20
) %>%
  autoplot()

gridExtra::grid.arrange(plot_1, plot_2, nrow = 1)
```




```
traindt$Failure.binary=as.factor(traindt$Failure.binary)
bagging_model2 <- train(
  Failure.binary ~ .,
  data = traindt,
  method = "treebag",
  trControl = trainControl(method = "cv", number = 10),
  nbagg = 100,
  control = rpart.control(minsplit = 2, cp = 0)
)
# Shutdown parallel cluster
stopCluster(cl)
```

Compute predicted probabilities on training data

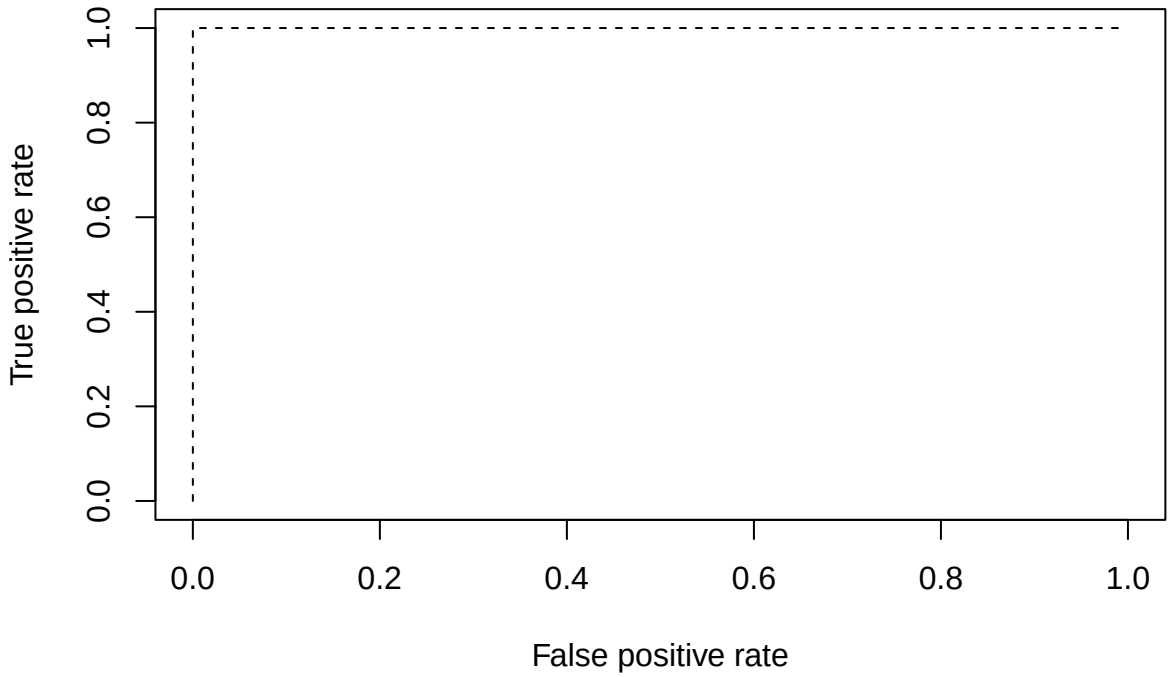
```
pred_prob1 <- predict(bagging_model2, traindt, type = "prob")[,2]
```

Compute AUC metrics for cv_model1,2 and 3

```
perf1 <- prediction(pred_prob1,traindt$Failure.binary) %>%
  performance(measure = "tpr", x.measure = "fpr")
```

Plot ROC curves for cv_model1,2 and 3

```
plot(perf1, col = "black", lty = 2)
```

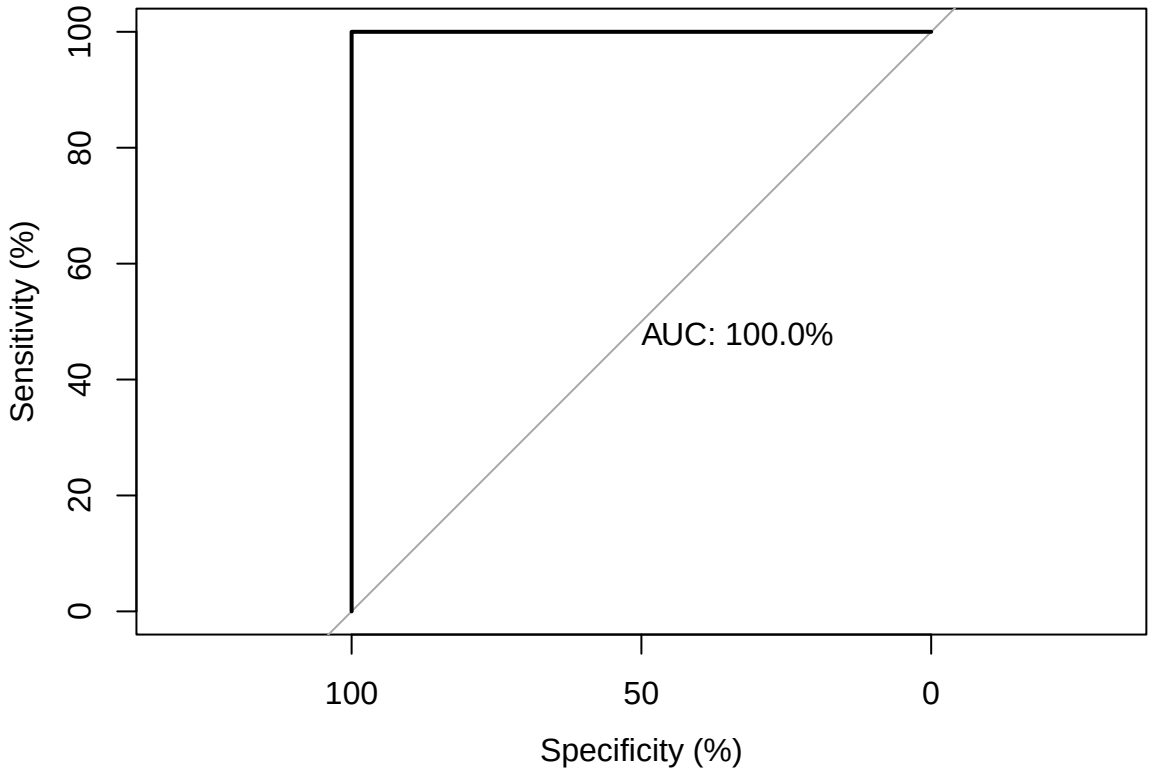


ROC plot for training data

```
roc( traindt$Failure.binary ~ pred_prob1, plot=TRUE, legacy.axes=FALSE,
    percent=TRUE, col="black", lwd=2, print.auc=TRUE)
```

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

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




```
##
## Call:
## roc.formula(formula = traindt$Failure.binary ~ pred_prob1, plot = TRUE,     legacy.axes = FALSE, per
##
## Data: pred_prob1 in 106 controls (traindt$Failure.binary 0) < 52 cases (traindt$Failure.binary 1).
## Area under the curve: 100%
```

Compute predicted probabilities on training data

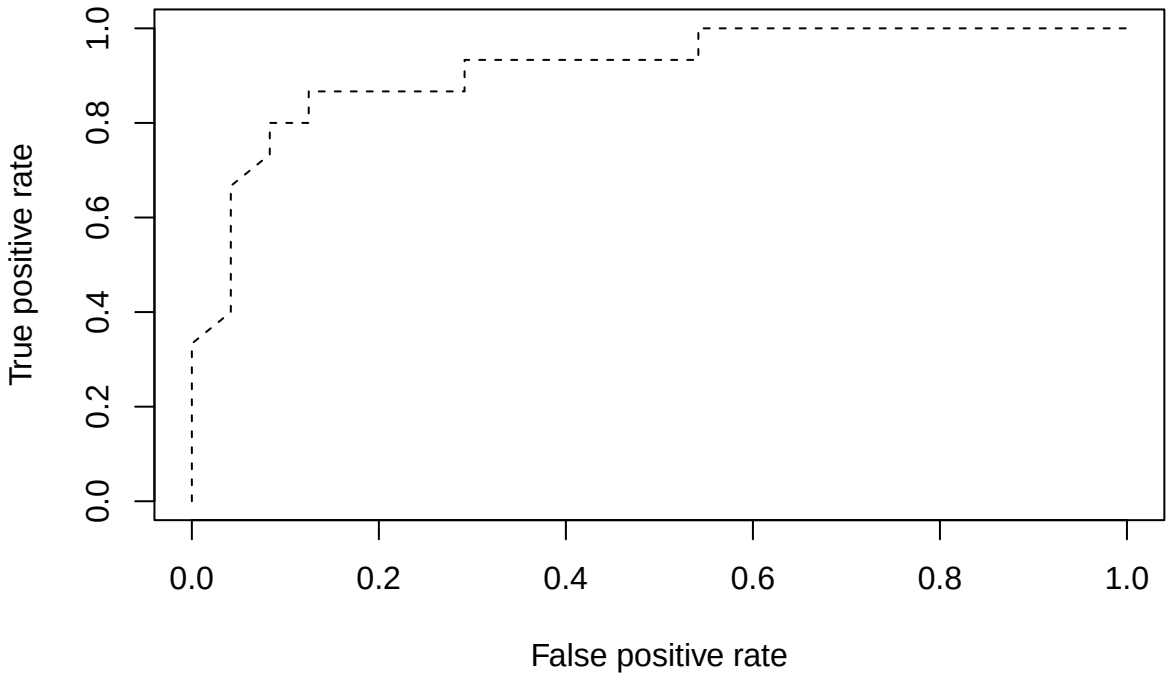
```
pred_prob2 <- predict(bagging_model2, radiomicsdata_test, type = "prob")[,2]
```

Compute AUC metrics for cv_model1,2 and 3

```
perf2 <- prediction(pred_prob2,radiomicsdata_test$Failure.binary) %>%
  performance(measure = "tpr", x.measure = "fpr")
```

Plot ROC curves for cv_model1,2 and 3

```
plot(perf2, col = "black", lty = 2)
```

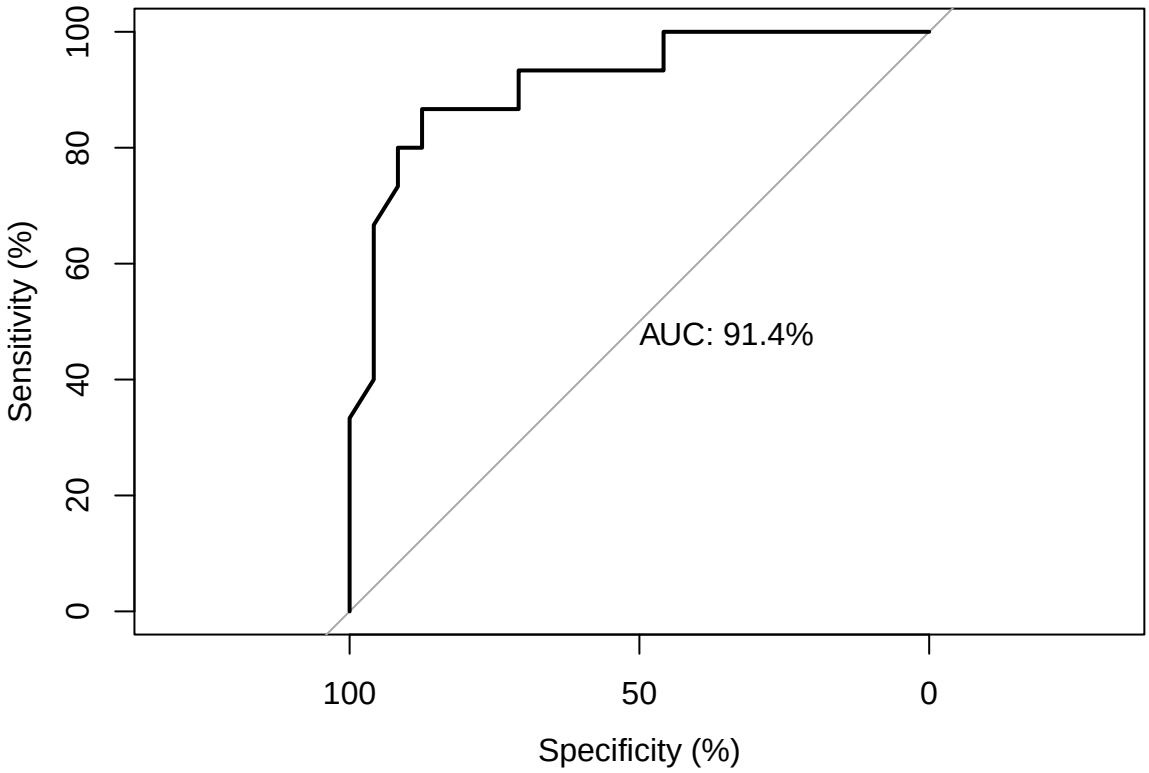


ROC plot for training data

```
roc( radiomicsdata_test$Failure.binary ~ pred_prob2, plot=TRUE, legacy.axes=FALSE,
    percent=TRUE, col="black", lwd=2, print.auc=TRUE)
```

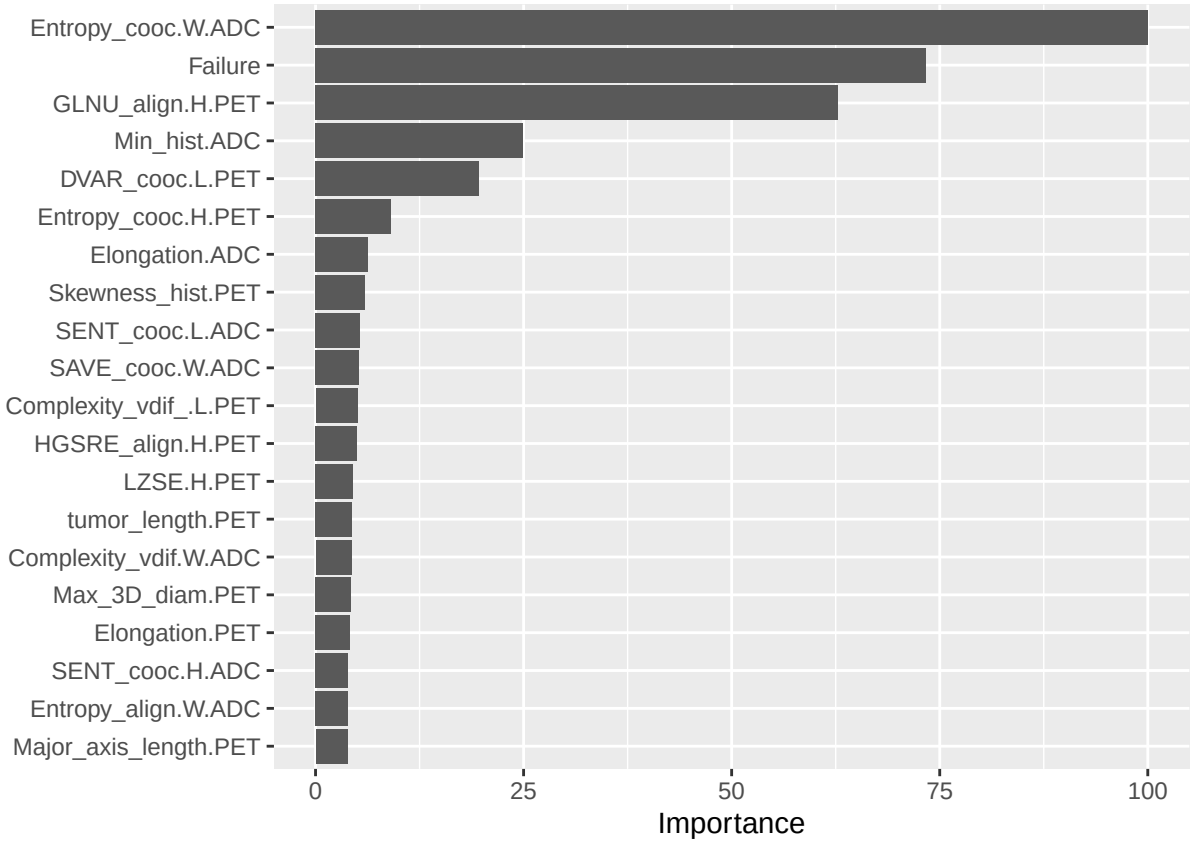
```
## Setting levels: control = 0, case = 1
```

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




```
##
## Call:
## roc.formula(formula = radiomicsdata_test$Failure.binary ~ pred_prob2,     plot = TRUE, legacy.axes =
##
## Data: pred_prob2 in 24 controls (radiomicsdata_test$Failure.binary 0) < 15 cases (radiomicsdata_test$
## Area under the curve: 91.39%
```

```
vip::vip(bagging_model2, num_features = 20)
```


Entropy_cooc.W.ADC
Failure
GLNU_align.H.PET
Min_hist.ADC
DVAR_cooc.L.PET
Entropy_cooc.H.PET
Elongation.ADC
Skewness_hist.PET
SENT_cooc.L.ADC
SAVE_cooc.W.ADC
Complexity_vdif_.L.PET
HGSRE_align.H.PET
LZSE.H.PET
tumor_length.PET
Complexity_vdif.W.ADC
Max_3D_diam.PET
Elongation.PET
SENT_cooc.H.ADC
Entropy_align.W.ADC
Major_axis_length.PET
0
25
50
75
100
Importance