Create a train-test split of the data.

```
library(rpart)
library(rpart.plot)
library(RColorBrewer)
library(rattle)
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.3.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
#library(RGtk2)
require(tree)
```

```
## Loading required package: tree
```

```
library(tidyr)
library(dplyr)
```

```
##
## 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(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 3.0-2
```

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```r
library(lattice)
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

```r
library(tidyverse)
```

```
## Registered S3 method overwritten by 'cli':
##   method     from
##   print.tree tree
```

```
## -- Attaching packages --------------------------------------------------------------- tidyverse 1.3.0
```

```
## v tibble  3.0.1     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## v purrr   0.3.4
```

```
## Warning: package 'tibble' was built under R version 3.6.2
```

```
## Warning: package 'purrr' was built under R version 3.6.2
```

```
## -- Conflicts ------------------------------------------------------------------ tidyverse_conflicts()
## x Matrix::expand() masks tidyr::expand()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x purrr::lift()    masks caret::lift()
## x Matrix::pack()   masks tidyr::pack()
## x Matrix::unpack() masks tidyr::unpack()
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```r
library(class)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:rattle':
##
##     importance
```

```r
require(caTools)
```

```
## Loading required package: caTools
```

```r
library(leaps)
fifa = read.csv("fifa_cleaned_dj.csv")
fifa$Improved <- as.factor(fifa$Improved)
fifa = subset(fifa, select = -c(Nationality, Club, Potential,Jersey.Number))
train_index = sample(nrow(fifa), 0.8*nrow(fifa))
train = fifa[train_index,]
test = fifa[-train_index,]
test.Improved = fifa[-train_index,"Improved"]
dim(test)
```

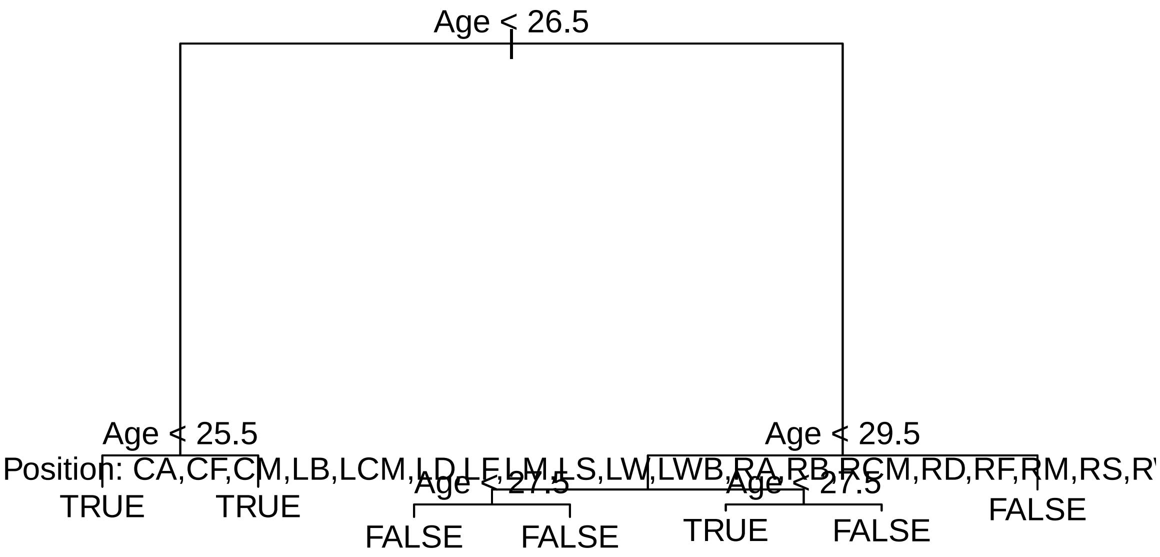
```
## [1] 3582    9
```

tree

```r
set.seed(3)
fifa.tree = tree(Improved~.,data=train)
summary(fifa.tree)
```

```
##
## Classification tree:
## tree(formula = Improved ~ ., data = train)
## Variables actually used in tree construction:
## [1] "Age"      "Position"
## Number of terminal nodes:  7
## Residual mean deviance:  0.217 = 3106 / 14320
## Misclassification error rate: 0.04768 = 683 / 14325
```

```r
plot(fifa.tree)
text(fifa.tree,pretty = 1)
```



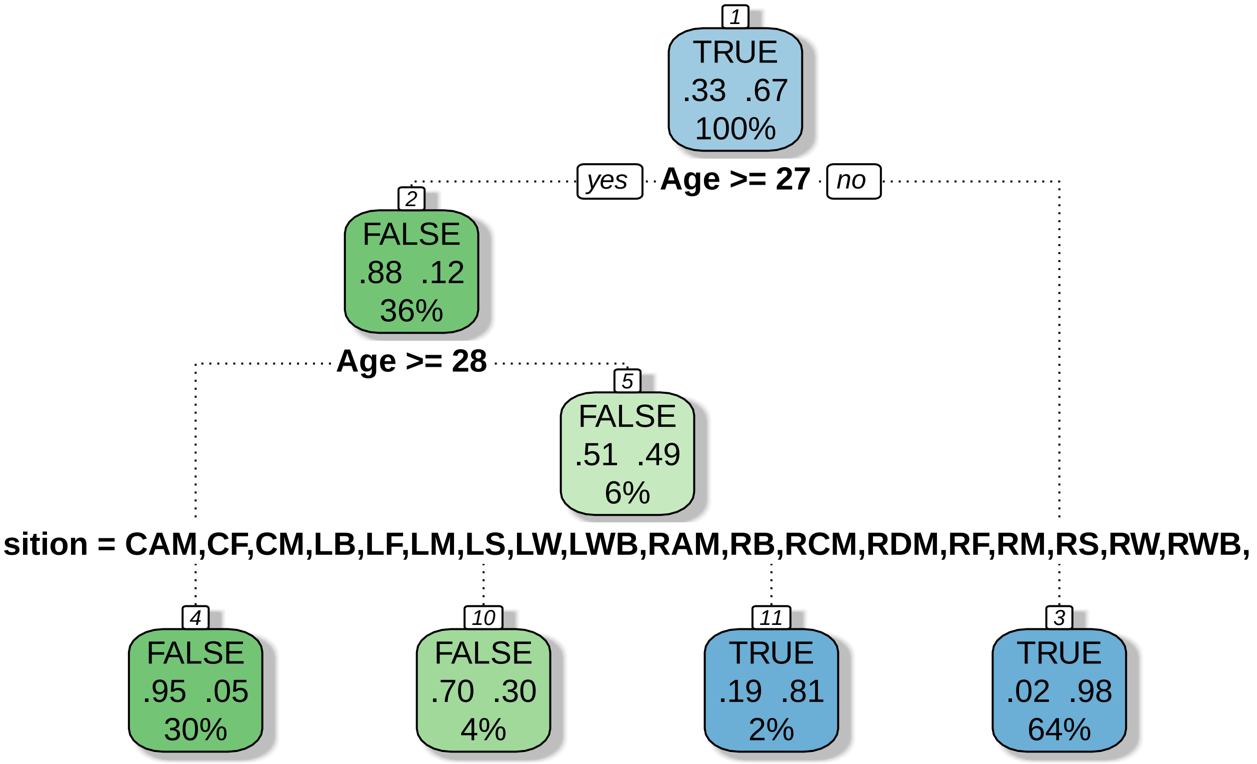
```
tree.pred = predict(fifa.tree,test,type="class")
table(tree.pred,test.Improved)
```

```
##          test.Improved
## tree.pred FALSE TRUE
##     FALSE  1195  117
##     TRUE     65 2205
```

```
(1067+2326)/3582
```

```
## [1] 0.9472362
```

```
fifa.tree2 = rpart(Improved~.,data = train, method = "class")
fancyRpartPlot(fifa.tree2)
```



Rattle 2020−Apr−30 17:01:59 ning

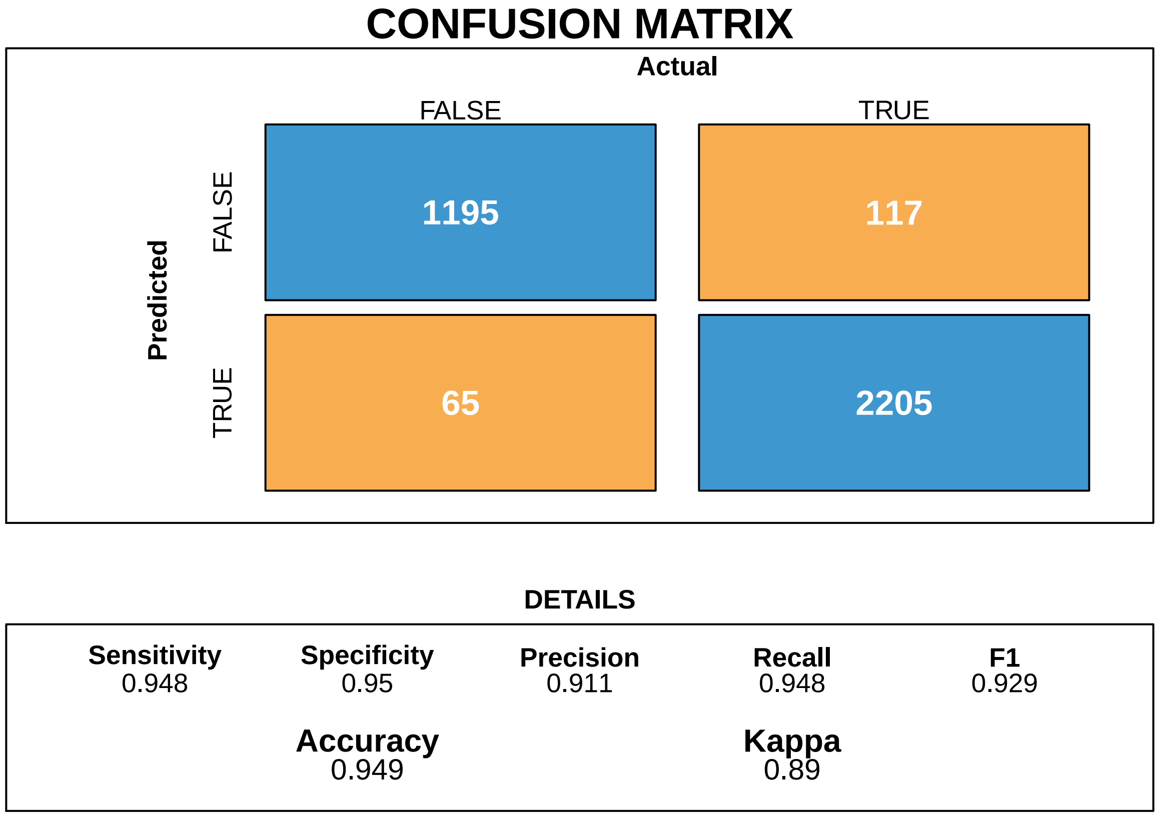
```
summary(fifa.tree2)
```

```
## Call:
## rpart(formula = Improved ~ ., data = train, method = "class")
##   n= 14325
##
##           CP nsplit rel error    xerror        xstd
## 1 0.81665969      0 1.0000000 1.0000000 0.011810358
## 2 0.02113855      1 0.1833403 0.1833403 0.006002105
## 3 0.01000000      3 0.1410632 0.1439933 0.005356242
##
## Variable importance
##               Age          Position Contract.Duration              Wage
##                90                 3                 3                 1
##            Weight             Value            Height
##                 1                 1                 1
##
## Node number 1: 14325 observations,    complexity param=0.8166597
##   predicted class=TRUE   expected loss=0.3335428  P(node) =1
##     class counts:  4778  9547
##    probabilities: 0.334 0.666
##   left son=2 (5200 obs) right son=3 (9125 obs)
##   Primary splits:
##       Age               < 26.5       to the right, improve=4789.96000, (0 missing)
##       Wage              < -0.3856722 to the right, improve= 318.45720, (0 missing)
##       Value             < 0.3497221  to the right, improve= 114.41950, (0 missing)
##       Contract.Duration < 11.5       to the right, improve=  82.05653, (0 missing)
##       Position          splits as  RRRRRRLRLLLLRLRRLRLLLLRLRRR, improve=  70.06266, (0 missing)
##   Surrogate splits:
##       Contract.Duration < 9.5        to the right, agree=0.649, adj=0.032, (0 split)
##       Position          splits as  RRRRRRLRLRLRRRRRRRLRRRRRRRR, agree=0.643, adj=0.018, (0 split)
##       Wage              < 1.377786   to the right, agree=0.643, adj=0.016, (0 split)
##       Weight            < 195        to the right, agree=0.641, adj=0.010, (0 split)
##       Value             < -2.252842  to the left,  agree=0.639, adj=0.006, (0 split)
##
## Node number 2: 5200 observations,    complexity param=0.02113855
##   predicted class=FALSE  expected loss=0.1248077  P(node) =0.3630017
##     class counts:  4551   649
##    probabilities: 0.875 0.125
##   left son=4 (4292 obs) right son=5 (908 obs)
##   Primary splits:
##       Age      < 27.5       to the right, improve=286.532200, (0 missing)
##       Position splits as  LRLLLRLLRLLLLLLLLRLLLLLLLLL, improve= 61.216990, (0 missing)
##       Height   < 1.845      to the left,  improve= 29.006850, (0 missing)
##       Weight   < 171        to the left,  improve= 21.160630, (0 missing)
##       Value    < 1.273533   to the left,  improve=  4.873298, (0 missing)
##
## Node number 3: 9125 observations
##   predicted class=TRUE   expected loss=0.02487671  P(node) =0.6369983
##     class counts:   227  8898
##    probabilities: 0.025 0.975
##
## Node number 4: 4292 observations
##   predicted class=FALSE  expected loss=0.04846226  P(node) =0.2996161
##     class counts:  4084   208
##    probabilities: 0.952 0.048
##
## Node number 5: 908 observations,    complexity param=0.02113855
##   predicted class=FALSE  expected loss=0.4856828  P(node) =0.06338569
##     class counts:   467   441
##    probabilities: 0.514 0.486
##   left son=10 (578 obs) right son=11 (330 obs)
##   Primary splits:
##       Position          splits as  LRRLLRRLRRRLLLLLLRLLLLLLLLL, improve=106.420900, (0 missing)
##       Height            < 1.815      to the left,  improve= 57.136450, (0 missing)
##       Weight            < 171        to the left,  improve= 47.496490, (0 missing)
##       Value             < -1.188356  to the right, improve=  4.304014, (0 missing)
##       Contract.Duration < 3.5        to the left,  improve=  3.791995, (0 missing)
##   Surrogate splits:
##       Height            < 1.845      to the left,  agree=0.739, adj=0.282, (0 split)
##       Weight            < 177.5      to the left,  agree=0.721, adj=0.233, (0 split)
##       Value             < -1.341428  to the right, agree=0.645, adj=0.024, (0 split)
##       Contract.Duration < 0.5        to the right, agree=0.638, adj=0.003, (0 split)
##
## Node number 10: 578 observations
##   predicted class=FALSE  expected loss=0.3027682  P(node) =0.04034904
##     class counts:   403   175
##    probabilities: 0.697 0.303
##
## Node number 11: 330 observations
##   predicted class=TRUE   expected loss=0.1939394  P(node) =0.02303665
##     class counts:    64   266
##    probabilities: 0.194 0.806
```

```
tree.pred2 = predict(fifa.tree2,test,type = "class")
```

```
cm1 = confusionMatrix(data = tree.pred,reference = test.Improved)



draw_confusion_matrix <- function(cm) {

  layout(matrix(c(1,1,2)))
  par(mar=c(2,2,2,2))
  plot(c(100, 345), c(300, 450), type = "n", xlab="", ylab="", xaxt='n', yaxt='n')
  title('CONFUSION MATRIX', cex.main=2)

  # create the matrix
  rect(150, 430, 240, 370, col='#3F97D0')
  text(195, 435, 'FALSE', cex=1.2)
  rect(250, 430, 340, 370, col='#F7AD50')
  text(295, 435, 'TRUE', cex=1.2)
  text(125, 370, 'Predicted', cex=1.3, srt=90, font=2)
  text(245, 450, 'Actual', cex=1.3, font=2)
  rect(150, 305, 240, 365, col='#F7AD50')
  rect(250, 305, 340, 365, col='#3F97D0')
  text(140, 400, 'FALSE', cex=1.2, srt=90)
  text(140, 335, 'TRUE', cex=1.2, srt=90)

  # add in the cm results
  res <- as.numeric(cm$table)
  text(195, 400, res[1], cex=1.6, font=2, col='white')
  text(195, 335, res[2], cex=1.6, font=2, col='white')
  text(295, 400, res[3], cex=1.6, font=2, col='white')
  text(295, 335, res[4], cex=1.6, font=2, col='white')

  # add in the specifics
  plot(c(100, 0), c(100, 0), type = "n", xlab="", ylab="", main = "DETAILS", xaxt='n', yaxt='n')
  text(10, 85, names(cm$byClass[1]), cex=1.2, font=2)
  text(10, 70, round(as.numeric(cm$byClass[1]), 3), cex=1.2)
  text(30, 85, names(cm$byClass[2]), cex=1.2, font=2)
  text(30, 70, round(as.numeric(cm$byClass[2]), 3), cex=1.2)
  text(50, 85, names(cm$byClass[5]), cex=1.2, font=2)
  text(50, 70, round(as.numeric(cm$byClass[5]), 3), cex=1.2)
  text(70, 85, names(cm$byClass[6]), cex=1.2, font=2)
  text(70, 70, round(as.numeric(cm$byClass[6]), 3), cex=1.2)
  text(90, 85, names(cm$byClass[7]), cex=1.2, font=2)
  text(90, 70, round(as.numeric(cm$byClass[7]), 3), cex=1.2)

  # add in the accuracy information
  text(30, 35, names(cm$overall[1]), cex=1.5, font=2)
  text(30, 20, round(as.numeric(cm$overall[1]), 3), cex=1.4)
  text(70, 35, names(cm$overall[2]), cex=1.5, font=2)
  text(70, 20, round(as.numeric(cm$overall[2]), 3), cex=1.4)
}
```

```r
draw_confusion_matrix(cm1)
```

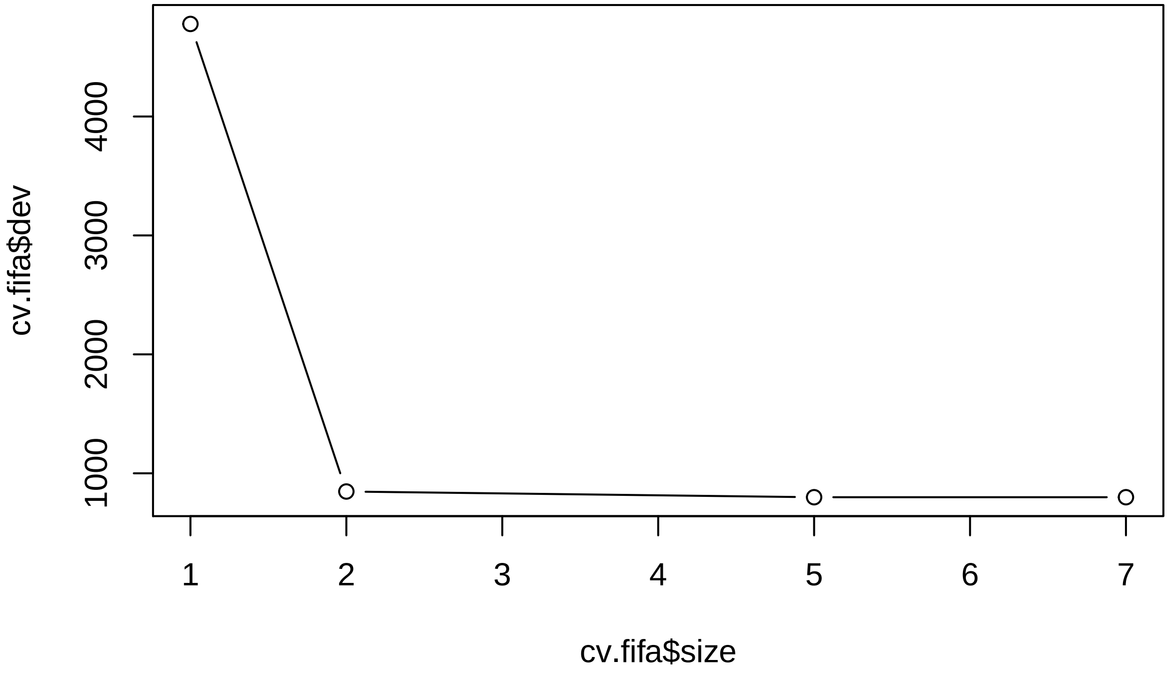


```r
## cv
cv.fifa = cv.tree(fifa.tree,FUN = prune.misclass)
names(cv.fifa)
```

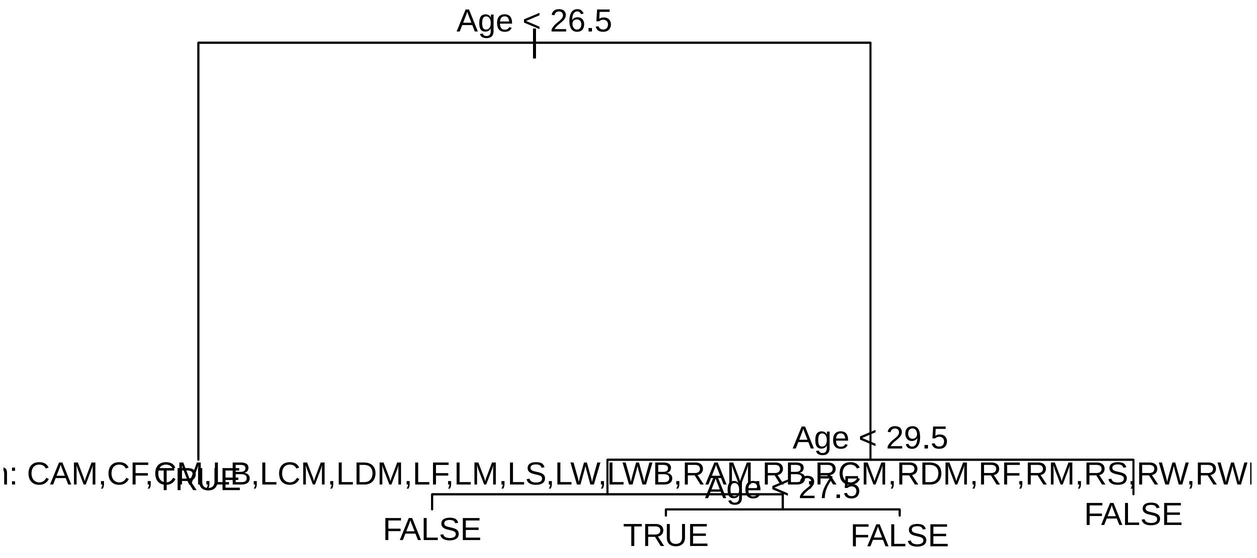
```
## [1] "size"   "dev"    "k"      "method"
```

```r
cv.fifa
```

```
## $size
## [1] 7 5 2 1
## 
## $dev
## [1]  799  799  847 4778
## 
## $k
## [1]       -Inf    0.00000   64.33333 3902.00000
## 
## $method
## [1] "misclass"
## 
## attr(,"class")
## [1] "prune"         "tree.sequence"
## tree with 4 terminal nodes results in lowest cv error rate, with 811 cv error



plot(cv.fifa$size,cv.fifa$dev,type = "b")
```



```
#prune
prune.fifa = prune.misclass(fifa.tree,best = 3)
plot(prune.fifa)
text(prune.fifa,pretty=11)
```


Age < 26.5
ı: CAM,CF,CM,LB,LCM,LDM,LF,LM,LS,LW,LWB,RAM,RB,RCM,RDM,RF,RM,RS,RW,RWI
TRUE
Age < 29.5
Age < 27.5
FALSE
TRUE
FALSE
FALSE

```
tree.pred = predict(prune.fifa,test,type = "class")
table(tree.pred,test.Improved)
```

```
##           test.Improved
## tree.pred FALSE TRUE
##     FALSE  1195  117
##     TRUE     65 2205
```

```
(1067+2326)/3582
```

```
## [1] 0.9472362
```

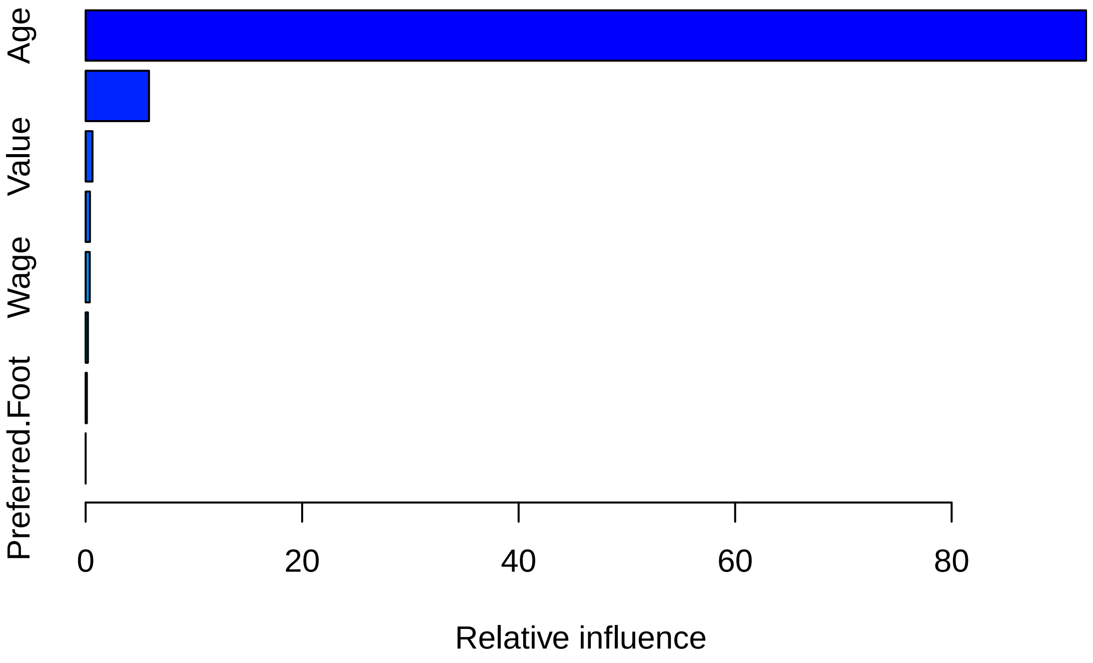
```
summary(prune.fifa)
```

```
## 
## Classification tree:
## snip.tree(tree = fifa.tree, nodes = c(2L, 12L))
## Variables actually used in tree construction:
## [1] "Age"      "Position"
## Number of terminal nodes:  5 
## Residual mean deviance:  0.3076 = 4405 / 14320 
## Misclassification error rate: 0.04768 = 683 / 14325
```

```
#Boosting
library(gbm)
```

```
## Loaded gbm 2.1.5
```

```
boost.fifa = gbm(Improved~.,train,distribution = "multinomial",n.trees = 100,interaction.depth = 4)
summary(boost.fifa)
```



```
##                                 var    rel.inf
## Age                             Age 92.4180266
## Position                   Position  5.8537700
## Value                         Value  0.6325627
## Contract.Duration Contract.Duration  0.3971496
## Wage                           Wage  0.3784392
## Height                       Height  0.2083869
## Weight                       Weight  0.1116650
## Preferred.Foot       Preferred.Foot  0.0000000
```

```
#par(mfrow=c(1,2))
#plot(boost.fifa,i="Age")
#plot(boost.fifa,i="Position")

#yhat.boost=predict(boost.fifa,newdata=test,n.tree = 100)
```

```
library(randomForest)
rf1 = randomForest(Improved~.,data=train,importance=TRUE)
summary(rf1)
```

```
##                 Length Class  Mode
## call                4  -none- call
## type                1  -none- character
## predicted       14325  factor numeric
## err.rate         1500  -none- numeric
## confusion           6  -none- numeric
## votes           28650  matrix numeric
## oob.times       14325  -none- numeric
## classes             2  -none- character
## importance         32  -none- numeric
## importanceSD       24  -none- numeric
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             14  -none- list
## y               14325  factor numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
## terms               3  terms  call
```

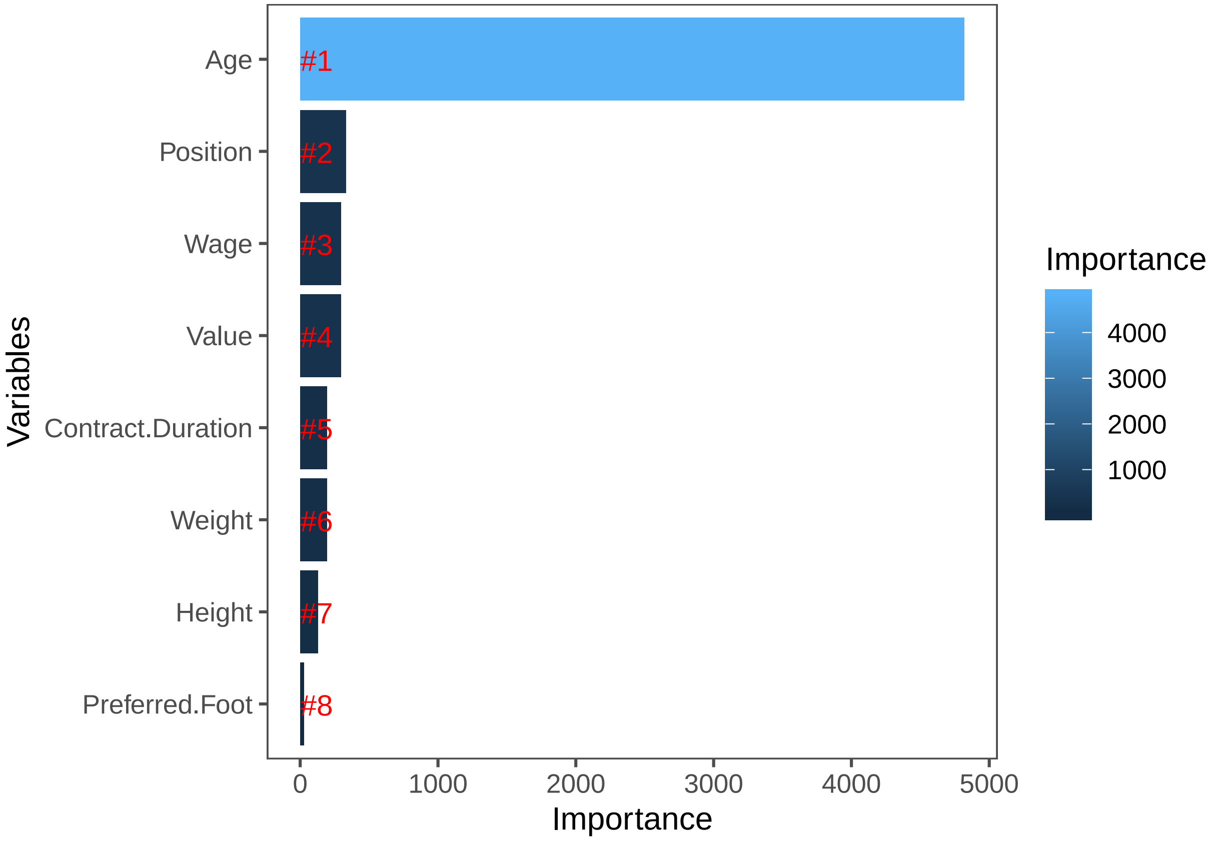
```
rf1.test<- predict(rf1, test, type = "class")
table(rf1.test,test.Improved)
```

```
##         test.Improved
## rf1.test FALSE TRUE
##    FALSE  1176   79
##    TRUE     84 2243
```

```
(1113+2308)/(1113+2308+72+89)
```

```
## [1] 0.955053
```

```
#visualize the importance
library("ggplot2")
library('ggthemes')
importance=randomForest::importance(rf1)
varImportance=data.frame(Variables=row.names(importance),
                        Importance=round(importance[ ,'MeanDecreaseGini'],2))

rankImportance <- varImportance %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance))))

#visualize the relative importance of variables
ggplot(rankImportance, aes(x = reorder(Variables, Importance),
                           y = Importance, fill = Importance)) +
  geom_bar(stat='identity') +
  geom_text(aes(x = Variables, y = 0.5, label = Rank),
            hjust=0, vjust=0.55, size = 4, colour = 'red') +
  labs(x = 'Variables') +
  coord_flip() +
  theme_few()
```


Variables
Age
Position
Wage
Value
Contract.Duration
Weight
Height
Preferred.Foot
#1
#2
#3
#4
#5
#6
#7
#8
Importance
Importance