Random forest

R Markdown

For this random forest model, downsampling are computed to improve the balance of data. The OOB error rate is 20.16%. Class error for 0 is 0.3340 and that for 1 is 0.0692. Train mse is 0.165. Test mse is 0.203. Though OOB error rate, class error for 0, train mse, and test mse are lower than model with neither downsampling nor smote, we could see the improvement of class error for 1 and ROC curve. This tradeoff the worthwhile since ROC curve is higher, which represents a better model.

Libraries

```
library(data.table)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1
```

```
library(caret)
```

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

```
## Loading required package: ggplot2
```

```
library(groupdata2)
library(InformationValue)
```

```
##
## Attaching package: 'InformationValue'
```

```
## The following objects are masked from 'package:caret':
##
##     confusionMatrix, precision, sensitivity, specificity
```

```
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
library(ggplot2)
library(ggthemes)
library(scales)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

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

```
library(dplyr)
```

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

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

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

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

```
library(tidyverse)
```

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

```
## v tibble  3.0.5     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() -
-
## x dplyr::between()    masks data.table::between()
## x readr::col_factor() masks scales::col_factor()
## x purrr::discard()    masks scales::discard()
## x tidyr::expand()     masks Matrix::expand()
## x dplyr::filter()     masks stats::filter()
## x dplyr::first()      masks data.table::first()
## x dplyr::lag()        masks stats::lag()
## x dplyr::last()       masks data.table::last()
## x purrr::lift()       masks caret::lift()
## x tidyr::pack()       masks Matrix::pack()
## x purrr::transpose()  masks data.table::transpose()
## x tidyr::unpack()     masks Matrix::unpack()
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(RColorBrewer)
library(leaps)
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.0.4
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(readr)
library(stringr)
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

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

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
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:dplyr':
##
##     combine
```

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

```
library(rpart)
library(DMwR)
```

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

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(pROC)
```

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

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

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

```
library(ROSE)
```

```
## Loaded ROSE 0.0-3
```

```
theme_set(theme_bw())
```

Loading data

```
db <- fread("D:/cs_dum_std.csv")
db$Response <- as.factor(db$Response)
colnames(db) <- make.names(colnames(db))
str(db)
```

```
## Classes 'data.table' and 'data.frame':   381109 obs. of  67 variables:
##  $ Age                       : num  0.334 2.397 0.527 -1.149 -0.633 ...
##  $ Annual_Premium            : num  0.575 0.173 0.449 -0.113 -0.178 ...
##  $ Vintage                   : num  0.749 0.342 -1.522 0.581 -1.379 ...
##  $ Response                  : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 1 2 1 1
...
##  $ Gender_Male               : int  1 1 1 1 0 0 1 0 0 0 ...
##  $ Driving_License_1         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Region_Code_2             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_3             : int  0 1 0 0 0 0 0 0 1 0 ...
##  $ Region_Code_6             : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Region_Code_7             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_8             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_9             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_10            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_11            : int  0 0 0 1 0 0 1 0 0 0 ...
##  $ Region_Code_12            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_13            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_14            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_15            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_16            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_17            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_18            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_21            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_24            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_25            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_26            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_27            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_28            : int  1 0 1 0 0 0 0 1 0 0 ...
##  $ Region_Code_29            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_30            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_31            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_32            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_33            : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Region_Code_35            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_36            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_37            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_38            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_39            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_41            : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Region_Code_43            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_45            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_46            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_47            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_48            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_50            : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ Previously_Insured_1      : int  0 0 0 1 1 0 0 0 1 1 ...
##  $ Vehicle_Age_..1.Year      : int  0 0 0 1 1 1 1 0 1 1 ...
##  $ Vehicle_Age_..2.Years     : int  1 0 1 0 0 0 0 0 0 0 ...
##  $ Vehicle_Damage_Yes        : int  1 0 1 0 0 1 1 1 0 0 ...
##  $ Policy_Sales_Channel_7    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_8    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_13   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_25   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_26   : int  1 1 1 0 0 0 0 1 0 0 ...
##  $ Policy_Sales_Channel_30   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_55   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_122  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_124  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_151  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_152  : int  0 0 0 1 1 0 1 0 1 1 ...
##  $ Policy_Sales_Channel_154  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_155  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_156  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_157  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_160  : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Policy_Sales_Channel_163  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Policy_Sales_Channel_Other: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region_Code_Other         : int  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Train Test split

```
set.seed(810)
test_p <- 0.3 # assign 30% random rows to the test set
rnorm(1)
```

```
## [1] 1.04923
```

```
test_index <- sample(nrow(db), round(test_p*nrow(db), digits=0)) # assign 30% rand
om rows to the test set
# now split
db.test <- db[test_index,]
db.train <- db[-test_index,]
# X
x.test <- db.test[, -"Response"]
x.train <- db.train[, -"Response"]
# Y
y.test <- db.test$Response
y.train <- db.train$Response
```

#Downsampling

```
downSample.train <- downsample(db.train, cat_col = "Response", id_col = NULL, id_m
ethod = "n_ids")
downSample.test <- downsample(db.test, cat_col = "Response", id_col = NULL, id_met
hod = "n_ids")
```

Defining relationship/formulas

```
f.all <- as.formula(Response ~ .)
f.num <- as.formula(Response ~ Age + Annual_Premium + Vintage)
```

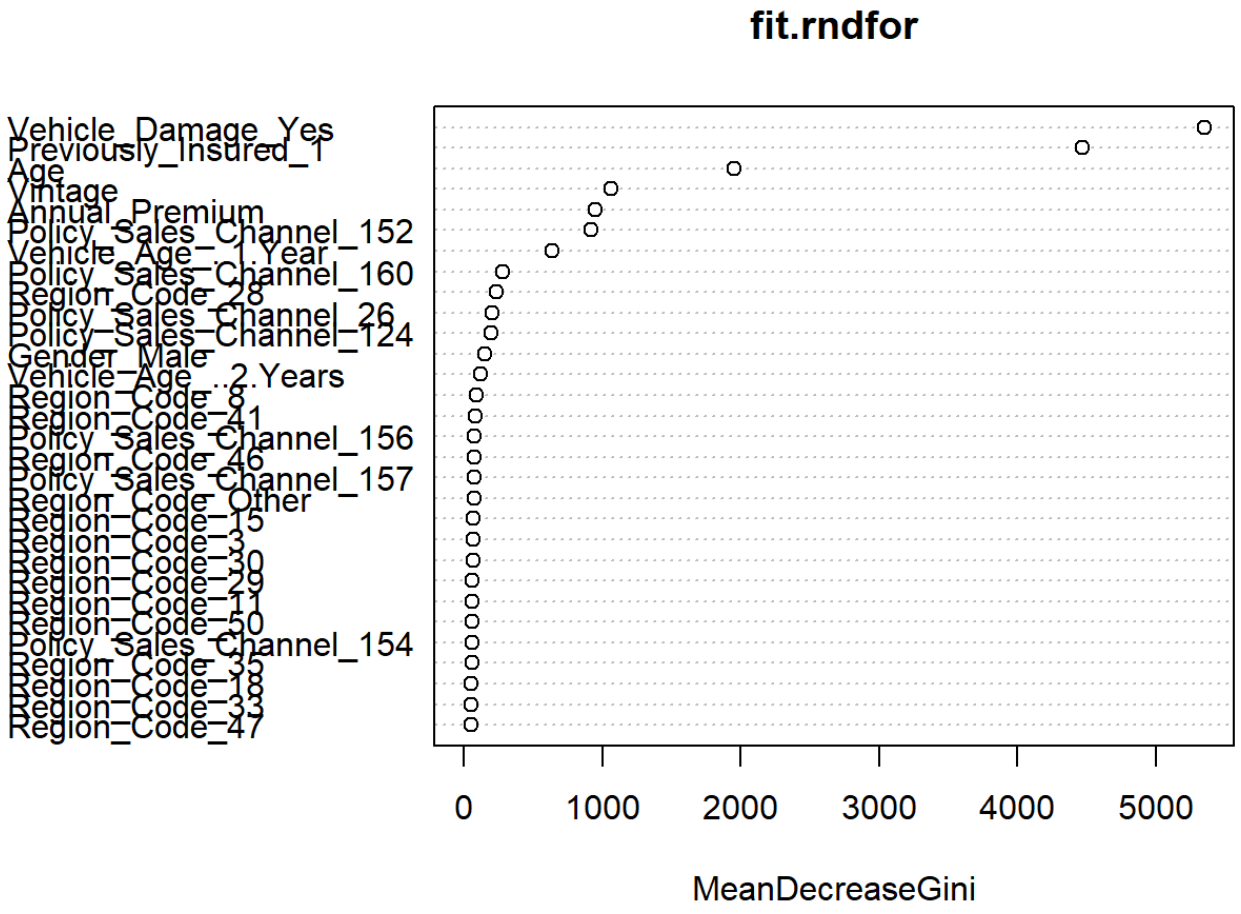
#Random forest

```
# which is an intercept added by default
x1.train.sample <- model.matrix(f.all, downSample.train)
# and this the response
y.train <- downSample.train$Response
y.train.sample <- downSample.train$Response

 # hack so that the following line works
x1.test <- model.matrix(f.all, downSample.test)
y.test <- downSample.test$Response

fit.rndfor <- randomForest(Response ~.,
                           downSample.train,
                           ntree=500,
                           do.trace=F)
```

```
print(fit.rndfor)
```

```
##
## Call:
##  randomForest(formula = Response ~ ., data = downSample.train,      ntree = 50
0, do.trace = F)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 8
##
##         OOB estimate of  error rate: 20.18%
## Confusion matrix:
##       0     1 class.error
## 0 21686 10893  0.33435649
## 1  2256 30323  0.06924706
```

```
varImpPlot(fit.rndfor)
```



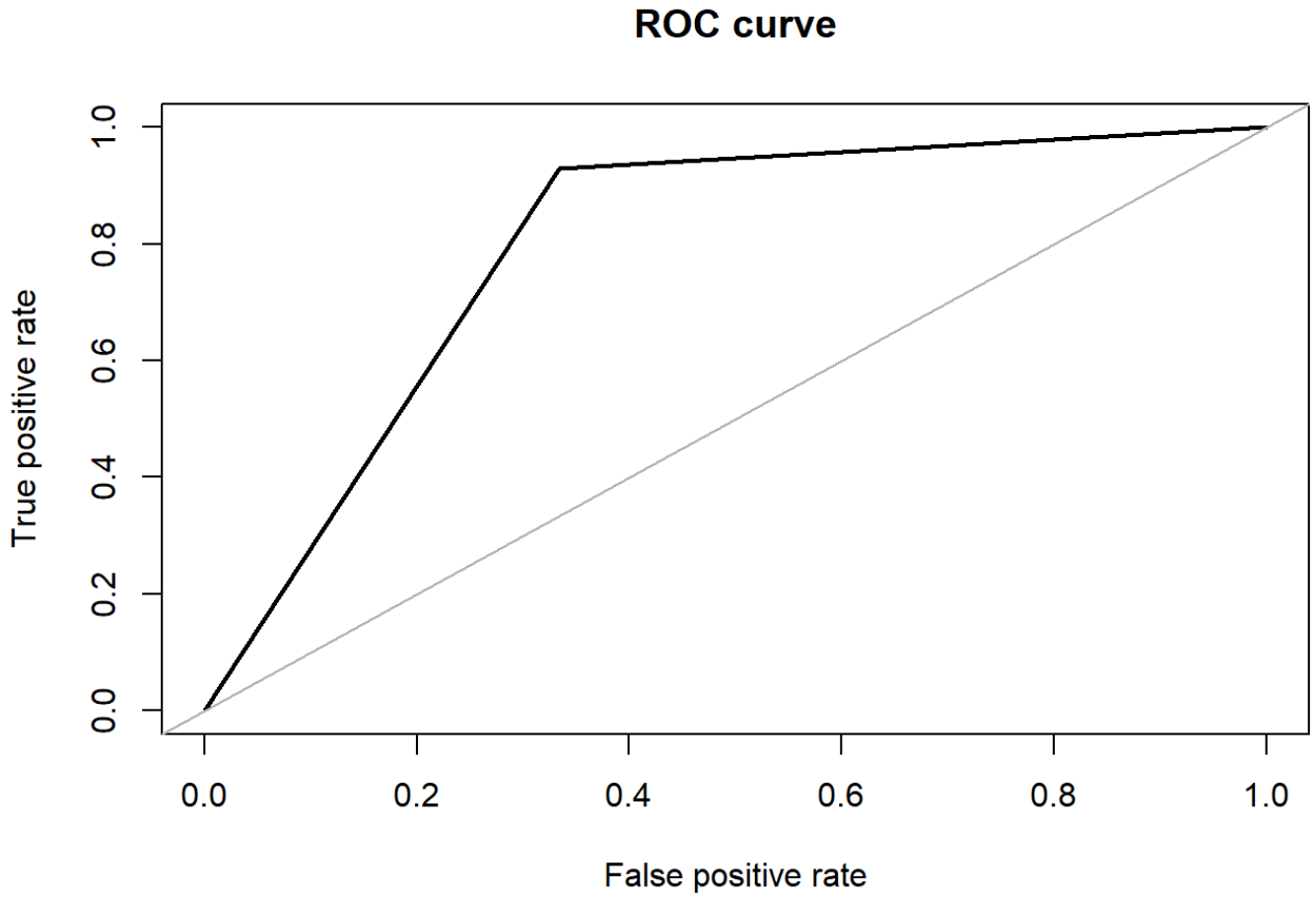
```
# Copmute train MSE
yhat.rndfor <- predict(fit.rndfor, downSample.train)
yhat.rndfor <- as.numeric(yhat.rndfor)
y.train.sample <- as.numeric(y.train.sample)
mse_data <- data.frame(pred = yhat.rndfor, actual = y.train.sample)
mse.tree <- mean((mse_data$actual - mse_data$pred)^2)

print(mse.tree)
```

```
## [1] 0.1664262
```

```
# Copmute test MSE
yhat.rndfor.test <- predict(fit.rndfor, downSample.test)
yhat.rndfor.test <- as.numeric(yhat.rndfor.test)
y.test <- as.numeric(y.test)
mse_test_data <- data.frame(pred = yhat.rndfor.test, actual = y.test)
mse.tree <- mean((mse_test_data$actual - mse_test_data$pred)^2)

print(mse.tree)
```

```
## [1] 0.2023565
```

```
roc.curve(y.test, yhat.rndfor.test,  plotit = TRUE, add.roc = FALSE)
```




```
## Area under the curve (AUC): 0.798
```