Data Modelling

Matheus C. Andrade

6/15/2021

Stage 0: Collecting the data

```
## Setting the working directory
setwd("~/Development/DataScienceAcademy/FCD/BigDataRAzure/ProjetoFinal/TalkingData-AdTracking-Fraud-Dete
getwd()
```

```
## [1] "/home/matheus/Development/DataScienceAcademy/FCD/BigDataRAzure/ProjetoFinal/TalkingData-AdTracki
```

```
source("../FeatureEngineering/FeatureEngeneering.R")

## Libraries
library(data.table)
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(ROSE)
```

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

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(ROCR)
library(pROC)
```

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

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

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

```
## Loading dataset
df <- fread("../Datasets/train_sample.csv", header=T)
# df <- fread(file.choose(), header=T)
```

Stage 1: Data Munging

```
df <- featureEngineering(df, 0)
```

Stage 2: Splitting into train and test datasets

```
set.seed(123)
rows = sample(1:nrow(df), 0.8*nrow(df))
train = df[rows,]
test = df[-rows,]
```

Stage 3: Applying Random Undersampling to balance the dataset's class variables:

```
table(train$is_attributed) # 191*2 = N
```

```
##
##     0     1
## 79809   191
```

```
under_train <- ovun.sample(is_attributed ~.,
                           data = train,
                           method = "under",
                           N = 382, seed=123)$data  #320 #138104

table(test$is_attributed) #36*2 = N
```

```
##
##     0     1
## 19964    36
```

```
under_test <- ovun.sample(is_attributed ~.,
                          data = test,
                          method = "under",
                          N = 72, seed=123)$data #134 #134
table(under_train$is_attributed)
```

```
##
##   0   1
## 191 191
```

```
table(under_test$is_attributed)
```

```
##
##  0  1
```

```
## 36 36
```

Stage 4: Creating the undersampled models

```r
### Model 1: Modelo Naive Bayes
# Creating the Naive Bayes model
set.seed(12345)
model_underNB = train(under_train[,-7], under_train[,7], method='naive_bayes')
# Making predictions on test data:
pred_model_underNB = predict(model_underNB, under_test[,-7])
# creating Confusion Matrix from predictions
confusionMatrix(pred_model_underNB, under_test$is_attributed)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 29  6
##          1  7 30
##
##                Accuracy : 0.8194
##                  95% CI : (0.7111, 0.9002)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 1.904e-08
##
##                   Kappa : 0.6389
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8056
##             Specificity : 0.8333
##          Pos Pred Value : 0.8286
##          Neg Pred Value : 0.8108
##              Prevalence : 0.5000
##          Detection Rate : 0.4028
##    Detection Prevalence : 0.4861
##       Balanced Accuracy : 0.8194
##
##        'Positive' Class : 0
##
```

```r
# calculating AUC for this model
underNB_AUC <- auc(roc(as.integer(under_test$is_attributed), as.integer(pred_model_underNB)))
```

```
## Setting levels: control = 1, case = 2
```

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

```r
### Model 2: Linear Discriminant Analysis (LDA)
# Creating
set.seed(12345)
model_underLDA = train(under_train[,-7], under_train[,7], method='lda')
# Making predictions
pred_model_underLDA = predict(model_underLDA, under_test[,-7])
# Confusion Matrix
```

```r
confusionMatrix(pred_model_underLDA, under_test$is_attributed)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 25 18
##          1 11 18
##
##                Accuracy : 0.5972
##                  95% CI : (0.475, 0.7112)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 0.06246
##
##                   Kappa : 0.1944
##
##  Mcnemar's Test P-Value : 0.26521
##
##             Sensitivity : 0.6944
##             Specificity : 0.5000
##          Pos Pred Value : 0.5814
##          Neg Pred Value : 0.6207
##              Prevalence : 0.5000
##          Detection Rate : 0.3472
##    Detection Prevalence : 0.5972
##       Balanced Accuracy : 0.5972
##
##        'Positive' Class : 0
##
```

```r
# calculating AUC and ROC curve for this model
underLDA_AUC <- auc(roc(as.integer(under_test$is_attributed), as.integer(pred_model_underLDA)))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```r
### Model 3: Decision Tree (rpart)
set.seed(12345)
model_underDT = train(under_train[,-7], under_train[,7], method='rpart')
pred_model_underDT = predict(model_underDT, under_test[,-7])
confusionMatrix(pred_model_underDT, under_test$is_attributed)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 31  4
##          1  5 32
##
##                Accuracy : 0.875
##                  95% CI : (0.7759, 0.9412)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 2.091e-11
##
##                   Kappa : 0.75
```

```
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8611
##             Specificity : 0.8889
##          Pos Pred Value : 0.8857
##          Neg Pred Value : 0.8649
##              Prevalence : 0.5000
##          Detection Rate : 0.4306
##    Detection Prevalence : 0.4861
##       Balanced Accuracy : 0.8750
##
##        'Positive' Class : 0
##
```

```
underDT_AUC <- auc(roc(as.integer(under_test$is_attributed), as.integer(pred_model_underDT)))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
### Model 4: Random Forest
set.seed(12345)
model_underRF = train(under_train[,-7], under_train[,7], method='rf')
pred_model_underRF = predict(model_underRF, under_test[,-7])
confusionMatrix(pred_model_underRF, under_test$is_attributed)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 34  4
##          1  2 32
##
##                Accuracy : 0.9167
##                  95% CI : (0.8274, 0.9688)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 3.628e-14
##
##                   Kappa : 0.8333
##
##  Mcnemar's Test P-Value : 0.6831
##
##             Sensitivity : 0.9444
##             Specificity : 0.8889
##          Pos Pred Value : 0.8947
##          Neg Pred Value : 0.9412
##              Prevalence : 0.5000
##          Detection Rate : 0.4722
##    Detection Prevalence : 0.5278
##       Balanced Accuracy : 0.9167
##
##        'Positive' Class : 0
##
```

```
underRF_AUC <- auc(roc(as.integer(under_test$is_attributed), as.integer(pred_model_underRF)))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
# ### Model 5: AdaBoost
# set.seed(12345)
# model_underAda = train(under_train[,-7], under_train[,7], method='adaboost')
# pred_model_underAda = predict(model_underAda, under_test[,-7])
# confusionMatrix(pred_model_underAda, under_test$is_attributed)
# underAda_AUC <- auc(roc(as.integer(under_test$is_attributed), as.integer(pred_model_underAda)))
#
# ### Model 6: Support Vector Machines
# set.seed(12345)
# model_underAda = train(under_train[,-7], under_train[,7], method='svmLinear')
# pred_model_underSVM = predict(model_underSVM, under_test[,-7])
# confusionMatrix(pred_model_underSVM, under_test$is_attributed)
# underSVM_AUC <- auc(roc(as.integer(under_test$is_attributed), as.integer(pred_model_underSVM)))
#
# ### Model 7: Support Vector Machines
# set.seed(12345)
# model_underGBM = train(under_train[,-7], under_train[,7], method='gbm')
# pred_model_underGBM = predict(model_underGBM, under_test[,-7])
# confusionMatrix(pred_model_underGBM, under_test$is_attributed)
# underGBM_AUC <- auc(roc(as.integer(under_test$is_attributed), as.integer(pred_model_underGBM)))
#
# ### Model 8: Neural Network
# set.seed(12345)
# model_underNN = train(under_train[,-7], under_train[,7], method='nnet')
# pred_model_underNN = predict(model_underNN, under_test[,-7])
# confusionMatrix(pred_model_underNN, under_test$is_attributed)
# underNN_AUC <- auc(roc(as.integer(under_test$is_attributed), as.integer(pred_model_underNN)))
```

ROC curves and AUC of undersampled models

```
roc.curve(under_test$is_attributed, pred_model_underNB, plotit = T, col = 1)
```

```
## Area under the curve (AUC): 0.819
```

```
roc.curve(under_test$is_attributed, pred_model_underLDA, plotit = T, col = 2, add=T)
```

```
## Area under the curve (AUC): 0.597
```

```
roc.curve(under_test$is_attributed, pred_model_underDT, plotit = T, col = 3, add=T)
```

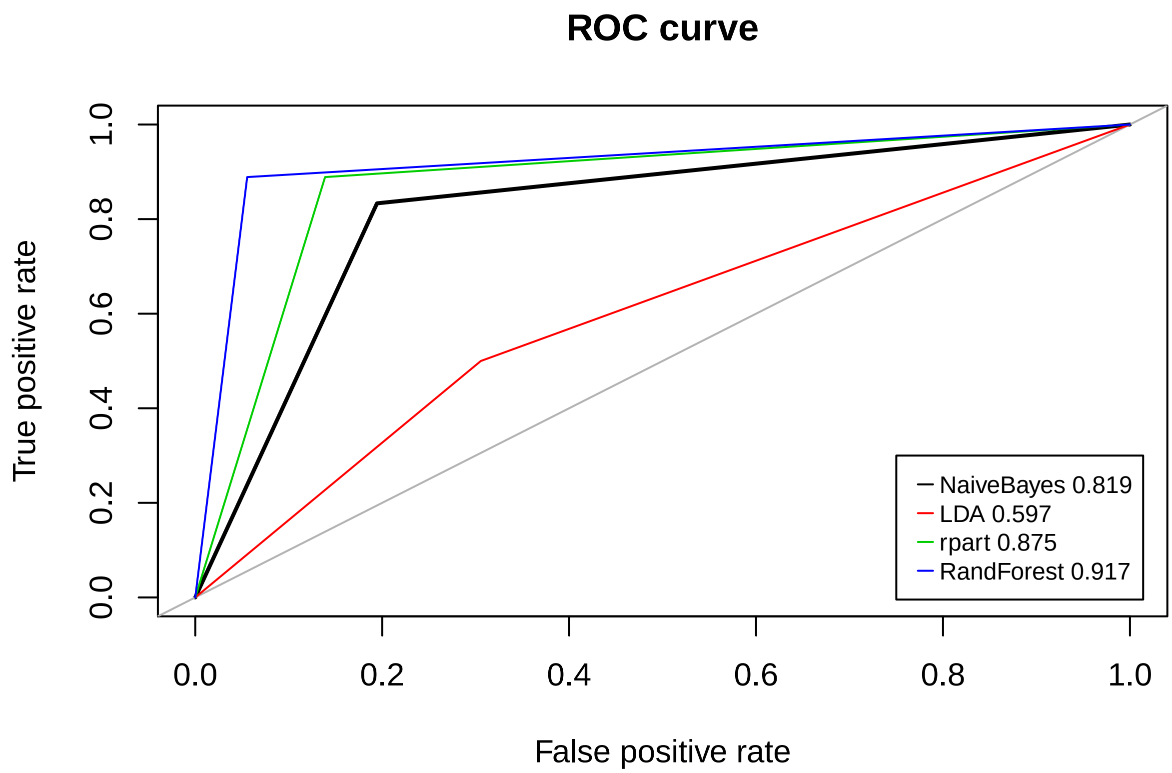
```
## Area under the curve (AUC): 0.875
```

```
roc.curve(under_test$is_attributed, pred_model_underRF, plotit = T, col = 4, add=T)
```

```
## Area under the curve (AUC): 0.917
```

```
# roc.curve(under_test$is_attributed, pred_model_underAda, plotit = T, col = 5, add=T)
# roc.curve(under_test$is_attributed, pred_model_underSVM, plotit = T, col = 6, add=T)#
# roc.curve(under_test$is_attributed, pred_model_underGBM, plotit = T, col = 7, add=T)#
# roc.curve(under_test$is_attributed, pred_model_underNN, plotit = T, col = 8, add=T)#
legend(y=0.3,x=0.75, bty="o", #"bottomright"
       c(paste("NaiveBayes", round(underNB_AUC, 3)),
         paste("LDA", round(underLDA_AUC, 3)),
         paste("rpart", round(underDT_AUC, 3)),
```

```
          paste("RandForest", round(underRF_AUC, 3))
          # paste("AdaBoost", round(underAda_AUC, 3)),
          # paste("SVM", round(underAda_AUC, 3)),
          # paste("GBM", round(underGBM_AUC, 3)),
          # paste("NN", round(underGBM_AUC, 3))
        ),
      col=1:4, lty=1:1, lwd=1, seg.len=0.7, cex=0.75,
       x.intersp=0.3, xjust=0)
```




```
## creating a AUC values vector for each undersampled model
under_aucModels <- c(
  underNB_AUC   = underNB_AUC,
  underLDA_AUC = underLDA_AUC,
  underDT_AUC   = underDT_AUC,
  underRF_AUC   = underRF_AUC
  # underAda_AUC = underAda_AUC,
  # underSVM_AUC = underSVM_AUC,
  # underGBM_AUC = underGBM_AUC,
  # underNN_AUC  = underNN_AUC
)
### best undersampled model
head(sort(under_aucModels, decreasing = T), 1)
```

```
## underRF_AUC
##   0.9166667
```

```r
# the best is randomForest
```

Stage 5: Applying Random Oversampling to balance the dataset's class variables:

```r
table(train$is_attributed) # 79809*2 = N
```

```
##
##     0     1
## 79809   191
```

```r
over_train <- ovun.sample(is_attributed ~.,
                          data = train,
                          method = "over",
                          N = 159618, seed = 123)$data  #320 #138104

table(test$is_attributed) #19964*2 = N
```

```
##
##     0     1
## 19964    36
```

```r
over_test <- ovun.sample(is_attributed ~.,
                         data = test,
                         method = "over",
                         N = 39926, seed = 123)$data #134 #134

table(over_train$is_attributed)
```

```
##
##     0     1
## 79809 79809
```

```r
table(over_test$is_attributed)
```

```
##
##     0     1
## 19964 19962
```

Stage 6: Creating the oversampled models

```r
### Model 1: Modelo Naive Bayes
set.seed(12345)
model_overNB = train(over_train[,-7], over_train[,7], method='naive_bayes')
pred_model_overNB = predict(model_overNB, over_test[,-7])
confusionMatrix(pred_model_overNB, over_test$is_attributed)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 16225  3842
##          1  3739 16120
##
```

```
##                Accuracy : 0.8101
##                  95% CI : (0.8062, 0.814)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6202
##
##  Mcnemar's Test P-Value : 0.2414
##
##             Sensitivity : 0.8127
##             Specificity : 0.8075
##          Pos Pred Value : 0.8085
##          Neg Pred Value : 0.8117
##              Prevalence : 0.5000
##          Detection Rate : 0.4064
##    Detection Prevalence : 0.5026
##       Balanced Accuracy : 0.8101
##
##        'Positive' Class : 0
##
```

```
overNB_AUC <- auc(roc(as.integer(over_test$is_attributed), as.integer(pred_model_overNB)))
```

```
## Setting levels: control = 1, case = 2
```

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

```
### Model 2: Linear Discriminant Analysis (LDA)
set.seed(12345)
model_overLDA = train(over_train[,-7], over_train[,7], method='lda')
pred_model_overLDA = predict(model_overLDA, over_train[,-7])
confusionMatrix(pred_model_overLDA, over_train$is_attributed)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 59744 22917
##          1 20065 56892
##
##                Accuracy : 0.7307
##                  95% CI : (0.7285, 0.7329)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4614
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7486
##             Specificity : 0.7129
##          Pos Pred Value : 0.7228
##          Neg Pred Value : 0.7393
##              Prevalence : 0.5000
##          Detection Rate : 0.3743
##    Detection Prevalence : 0.5179
```

```
##         Balanced Accuracy : 0.7307
##
##          'Positive' Class : 0
##
```

```r
overLDA_AUC <- auc(roc(as.integer(over_train$is_attributed), as.integer(pred_model_overLDA)))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```r
### Model 3: Decision Tree (rpart)
set.seed(12345)
model_overDT = train(over_train[,-7], over_train[,7], method='rpart')
pred_model_overDT = predict(model_overDT, over_test[,-7])
confusionMatrix(pred_model_overDT, over_test$is_attributed)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction     0     1
##          0 17412  2172
##          1  2552 17790
##
##                Accuracy : 0.8817
##                  95% CI : (0.8785, 0.8848)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7634
##
##  Mcnemar's Test P-Value : 3.503e-08
##
##             Sensitivity : 0.8722
##             Specificity : 0.8912
##          Pos Pred Value : 0.8891
##          Neg Pred Value : 0.8745
##              Prevalence : 0.5000
##          Detection Rate : 0.4361
##    Detection Prevalence : 0.4905
##       Balanced Accuracy : 0.8817
##
##        'Positive' Class : 0
##
```

```r
overDT_AUC <- auc(roc(as.integer(over_test$is_attributed), as.integer(pred_model_overDT)))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```r
# ### Model 4: Random Forest
# set.seed(12345)
# model_overRF = train(over_train[,-7], over_train[,7], method='rf')
# pred_model_overRF <- predict(model_overRF, over_test)
# confusionMatrix(pred_model_overRF,
#                 over_test$is_attributed,
#                 positive='1')
# overRF_AUC <- auc(roc(as.integer(over_test$is_attributed), as.integer(pred_model_overRF)))
```

```
# ### Model 5: AdaBoost (the longest)
# set.seed(12345)
# model_overAda = train(over_train[,-7], over_train[,7], method='adaboost', nIter=1)
# ?adaboost
# pred_model_overAda = predict(model_overAda, over_test[,-7])
# confusionMatrix(pred_model_overAda, over_test$is_attributed)
# overAda_AUC <- auc(roc(as.integer(over_test$is_attributed), as.integer(pred_model_overAda)))
#
# ### Model 6: Support Vector Machines
# library(e1071)
# set.seed(12345)
# model_overSVM = svm(is_attributed ~ .,
#                     data = over_train,
#                     type = 'C-classification',
#                     kernel = 'linear')
# pred_model_overSVM = predict(model_overSVM, over_test[,-7])
# confusionMatrix(pred_model_overSVM, over_test$is_attributed)
# overSVM_AUC <- auc(roc(as.integer(over_test$is_attributed), as.integer(pred_model_overSVM)))
#
# ### Model 7: GBM
# set.seed(12345)
# model_overGBM = train(over_train[,-7], over_train[,7], method='gbm', n.trees=3)
#
# pred_model_overGBM = predict(model_overGBM, over_test[,-7])
# confusionMatrix(pred_model_overGBM, over_test$is_attributed)
# overGBM_AUC <- auc(roc(as.integer(over_test$is_attributed), as.integer(pred_model_overGBM)))
#
# # ### Model 8: Neural Network
# set.seed(12345)
# model_overNN = train(over_train[,-7], over_train[,7], method='nnet', nIter=2)
# pred_model_overNN = predict(model_overNN, over_test[,-7])
# confusionMatrix(pred_model_overNN, over_test$is_attributed)
# overNN_AUC <- auc(roc(as.integer(over_test$is_attributed), as.integer(pred_model_overNN)))
```

ROC curves and AUC of oversampled models

```
roc.curve(over_test$is_attributed, pred_model_overNB, plotit = T, col = 1)
```

```
## Area under the curve (AUC): 0.810
```

```
roc.curve(over_train$is_attributed, pred_model_overLDA, plotit = T, col = 2, add=T)
```

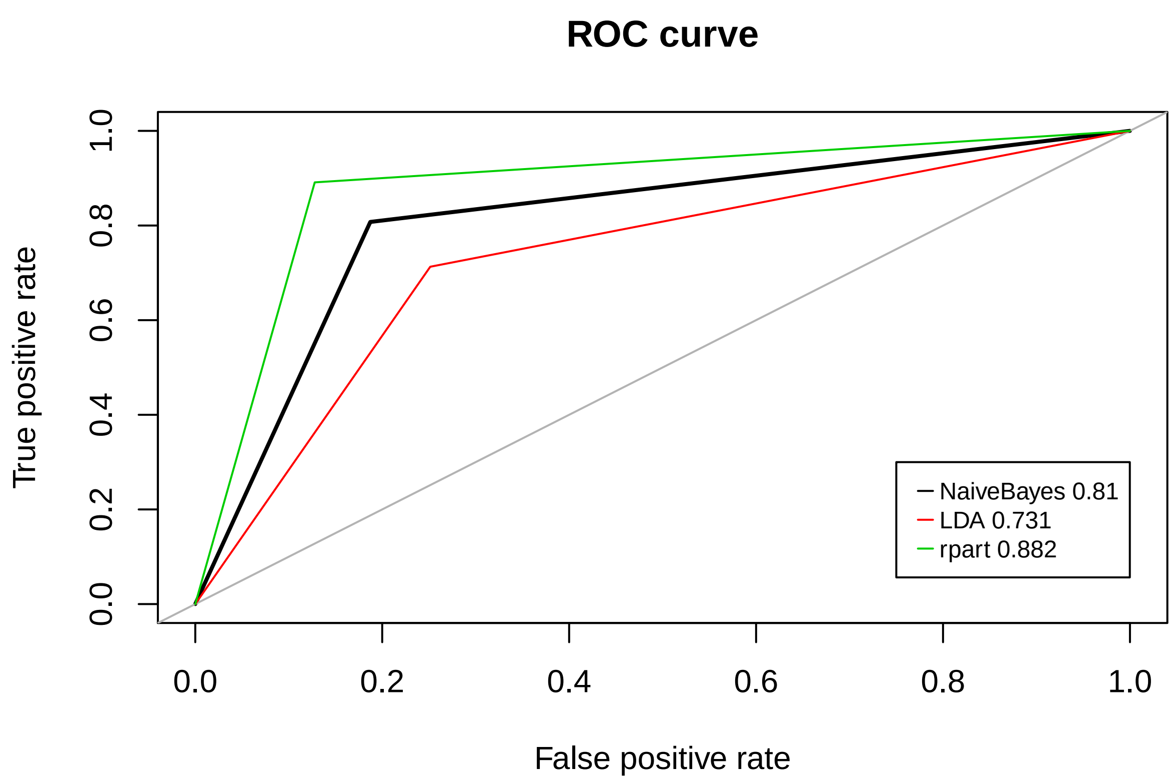
```
## Area under the curve (AUC): 0.731
```

```
roc.curve(over_test$is_attributed, pred_model_overDT, plotit = T, col = 3, add=T)
```

```
## Area under the curve (AUC): 0.882
```

```
#roc.curve(over_test$is_attributed, pred_model_overRF, plotit = T, col = 4, add=T)
# roc.curve(over_test$is_attributed, pred_model_overAda, plotit = T, col = 5, add=T)
# roc.curve(over_test$is_attributed, pred_model_overSVM, plotit = T, col = 6, add=T)
# roc.curve(over_test$is_attributed, pred_model_overGBM, plotit = T, col = 7, add=T)
# roc.curve(over_test$is_attributed, pred_model_overNN, plotit = T, col = 8, add=T)
legend(y=0.3,x=0.75, bty="o", #"bottomright"
       c(paste("NaiveBayes", round(overNB_AUC, 3)),
```

```r
         paste("LDA", round(overLDA_AUC, 3)),
         paste("rpart", round(overDT_AUC, 3))
         #paste("RandForest", round(overRF_AUC, 3))
         # paste("AdaBoost", round(overAda_AUC, 3)),
         # paste("SVM", round(overSVM_AUC, 3)),
         # paste("GBM", round(overGBM_AUC, 3)),
         # paste("NN", round(overNN_AUC, 3))
       ),
       col=1:3, lty=1:1, lwd=1, seg.len=0.7, cex=0.75,
       x.intersp=0.3, xjust=0)
```




```r
# creating a AUC values vector for each oversampled model
over_aucModels <- c(
  overNB_AUC  = overNB_AUC,
  overLDA_AUC = overLDA_AUC,
  overDT_AUC  = overDT_AUC
  #overRF_AUC  = overRF_AUC
  # overAda_AUC = overAda_AUC,
  # overSVM_AUC = overSVM_AUC,
  # overGBM_AUC = overGBM_AUC
  # overNN_AUC = overNN_AUC
)
# best oversampled model
head(sort(over_aucModels, decreasing = T), 1)
```

```
## overDT_AUC
```

```
## 0.8816816
```

```r
# So, the best of all models is really the undersampled randomForest
```

Stage 7: Optimizing the best model: Undersampled Random Forest

```r
set.seed(12345)
model_underRFop = train(under_train[,-7], under_train[,7],
                        method='rf',
                        metric="Accuracy",
                        ntree = 100,
                        nodesize = 10)
pred_model_underRFop = predict(model_underRFop, under_test[,-7])
confusionMatrix(pred_model_underRFop, under_test$is_attributed)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 34  3
##          1  2 33
##
##                Accuracy : 0.9306
##                  95% CI : (0.8453, 0.9771)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 3.194e-15
##
##                   Kappa : 0.8611
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9444
##             Specificity : 0.9167
##          Pos Pred Value : 0.9189
##          Neg Pred Value : 0.9429
##              Prevalence : 0.5000
##          Detection Rate : 0.4722
##    Detection Prevalence : 0.5139
##       Balanced Accuracy : 0.9306
##
##        'Positive' Class : 0
##
```

```r
underRF_AUCop <- auc(roc(as.integer(under_test$is_attributed), as.integer(pred_model_underRFop)))
```

```
## Setting levels: control = 1, case = 2
```

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

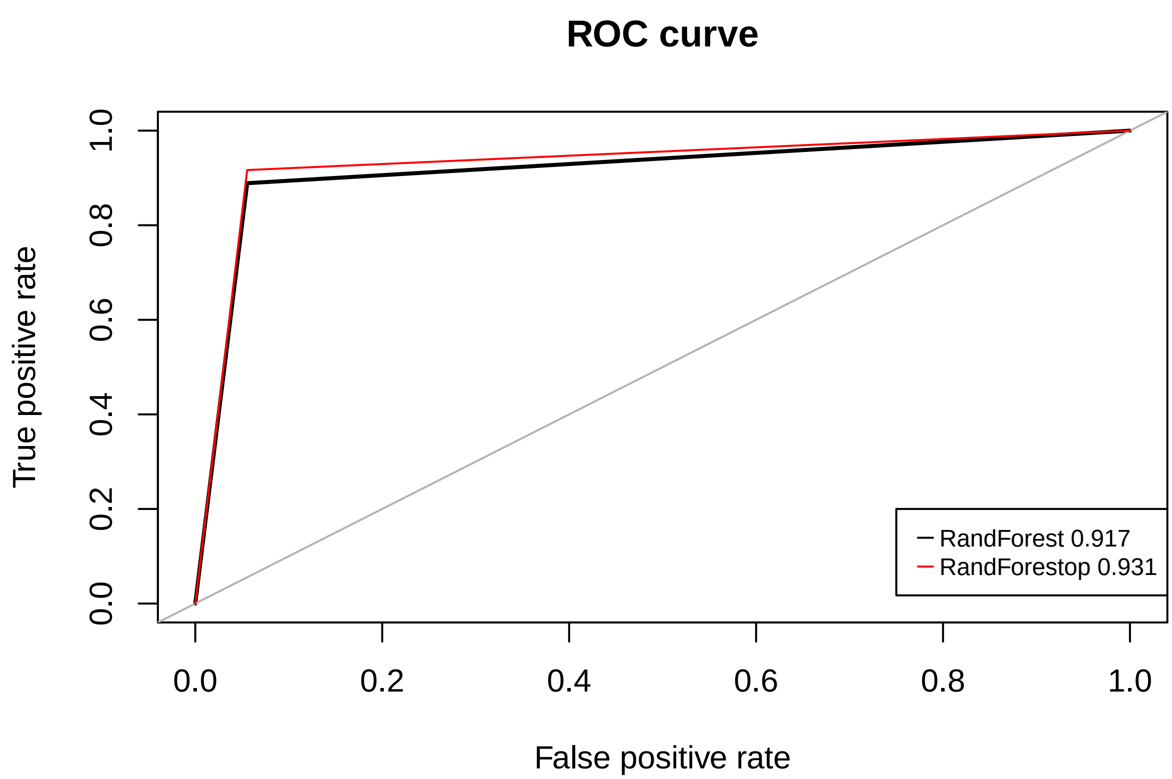
```r
# the model was improved
roc.curve(under_test$is_attributed, pred_model_underRF, plotit = T, col = 1)
```

```
## Area under the curve (AUC): 0.917
```

```r
roc.curve(under_test$is_attributed, pred_model_underRFop, plotit = T, col = 2, add=T)
```

```
## Area under the curve (AUC): 0.931
```

```
legend(y=0.2,x=0.75, bty="o", #"bottomright"
       c(paste("RandForest", round(underRF_AUC, 3)),
         paste("RandForestop", round(underRF_AUCop, 3))
       ),
       col=1:2, lty=1:1, lwd=1, seg.len=0.7, cex=0.75,
       x.intersp=0.3, xjust=0)
```




```
# so the best model was the "improved undersampled random forest"
chosenModel <- model_underRFop
```

Stage 8: File Submission on Kaggle

```
## loading "test" kaggle dataset
#submit <- fread(file.choose(), header=T)
submit <- fread("../Datasets/test.csv", header=T)

# to keep in safe 'click_id' column
submit1 <- submit[,-1]
submit1$attributed_time <- 0

## preparing data for modelling
submit1 <- featureEngineering(submit1, 1)
#gc() # to free memory
```

```r
## predicting the test
p <- predict(chosenModel, submit1)

## generating the dataframe that will be submited
d <- data.frame(click_id = submit$click_id, is_attributed = p)
fwrite(d, "submit_op_under_rf.csv")
# Finally, the scores AUC metric in Kaggle by this model was:
# Private Score: 0.87182~0.87193
# Public Score: 0.87675~0.87668
```