All 7 models - nb, C5.0, JRip, glm, kknn, nnet, svmlinear

jean wills

13/07/2020

# knitr::opts\_chunk$set(echo = TRUE, message=FALSE, warning=FALSE)

# step 1a - need to get to “BM\_mini\_sc” (added 95% CI + numeric scaled)

library(plyr)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

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

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

BM\_mini <- read.csv("/Users/jeanwills/Desktop/CKME136/1\_data/bank.csv", header=T, sep = ";", stringsAsFactors = T, na.strings = "NA")  
# Step 1 - NUMERIC DATA Cleaning - change numeric data outside the 2.5% and the 97.5% percentiles to this maximum/minimum value  
BM<-BM\_mini  
  
Lq\_bal<- quantile(BM$balance, probs=c(0.025))  
Hq\_bal<- quantile(BM$balance, probs=c(0.975))  
#Lq\_bal # -393  
#Hq\_bal # 8969  
Lq\_dur<- quantile(BM$duration, probs=c(0.025))  
Hq\_dur<- quantile(BM$duration, probs=c(0.975))  
#Lq\_dur # 19  
#Hq\_dur # 986  
Lq\_cam<- quantile(BM$campaign, probs=c(0.025))  
Hq\_cam<- quantile(BM$campaign, probs=c(0.975))  
#Lq\_cam # 1  
#Hq\_cam # 11  
Lq\_days<- quantile(BM$pdays, probs=c(0.025))  
Hq\_days<- quantile(BM$pdays, probs=c(0.975))  
#Lq\_days # -1  
#Hq\_days # 356  
Lq\_prv<- quantile(BM$previous, probs=c(0.025))  
Hq\_prv<- quantile(BM$previous, probs=c(0.975))  
#Lq\_prv # 0  
#Hq\_prv # 5  
  
BM$balance[BM$balance < Lq\_bal] <- Lq\_bal  
BM$balance[BM$balance > Hq\_bal] <- Hq\_bal  
  
BM$duration[BM$duration < Lq\_dur] <- Lq\_dur  
BM$duration[BM$duration > Hq\_dur] <- Hq\_dur  
  
BM$campaign[BM$campaign < Lq\_cam] <- Lq\_cam  
BM$campaign[BM$campaign > Hq\_cam] <- Hq\_cam  
  
BM$pdays[BM$pdays < Lq\_days] <- Lq\_days  
BM$pdays[BM$pdays > Hq\_days] <- Hq\_days  
  
BM$previous[BM$previous < Lq\_prv] <- Lq\_prv  
BM$previous[BM$previous > Hq\_prv] <- Hq\_prv  
  
# now make minor adjsutments  
# switch -1 -> 0 in 'pdays'  
BM$pdays<- ifelse(BM$pdays == -1, 0, BM$pdays)  
# switch duration in seconds to minutes for easier use  
BM$duration<- BM$duration/60  
  
# now have file BM ..

## step 1b - BM -> BM\_scale or BM\_mini\_sc"

# KEEP: age-1, balance-6, day-10, duration-12, campaign-13, pdays-14, previous-15  
BMS<- BM  
# BMS<-BM\_mini  
normalize<- function(x) {  
 return ((x - min(x)) / (max(x) - min(x)))  
}  
BM\_s<- as.data.frame(lapply(BMS[,c(1,6,10,12:15)], normalize))  
# now recombine dataframes with the nominal components  
BM\_s$job<-BMS$job  
BM\_s$marital<-BMS$marital  
BM\_s$education<-BMS$education  
BM\_s$default<-BMS$default  
BM\_s$housing<-BMS$housing  
BM\_s$loan<-BMS$loan  
BM\_s$contact<-BMS$contact  
BM\_s$month<-BMS$month  
BM\_s$poutcome<-BMS$poutcome  
BM\_s$y<-BMS$y  
# convert  
BM\_mini\_sc<-BM\_s  
# result used BM\_num file but now BM\_num\_scale with normalized numeric data  
# and y is factor  
# to convert y to numeric use next line  
# BM\_scale$y<- ifelse(BM\_scale$y==c("yes"), 1, 0)  
rm(BMS)  
rm(BM\_s)  
rm(BM)

# step 2 - run training and test datasets FOR ALL CONFIGURATIONS

# str(BM\_mini\_sc)  
# rename dataset here:  
BM<- BM\_mini\_sc  
set.seed(30)  
# get train and test datasets  
# note that if we use cross validation, we can use the complete dataset for training or keep a portion for validation after  
#  
BM\_train\_index <- sample(nrow(BM), 0.7 \* nrow(BM))  
BM\_train<- BM[BM\_train\_index, ]  
BM\_test <- BM[-BM\_train\_index, ]  
BM\_train\_labels <- BM[BM\_train\_index, 17]  
BM\_test\_labels <- BM[-BM\_train\_index, 17]  
# note that we only balance the training sets  
# and leave test set as is.

# step 3a - run SPECIFIC Balancing step to get balanced data version:

## OVER\_SAMPLE MAJOR/MINOR - each for training

set.seed(40)  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

up\_train <- upSample(x = BM\_train[, -ncol(BM\_train)], y = BM\_train$y)  
table(up\_train$Class)

##   
## no yes   
## 2804 2804

#use these now in each model if needed:  
x=up\_train[,-17]  
trainsv=up\_train  
y=up\_train$Class  
test\_noy=BM\_test[,-17]  
test\_labels=BM\_test$y

# step 4 - now run models

## MODEL 1 NAIVE BAYES - uses caret

# FINAL PARAMETER VALUES USED WERE fL = 0, usekernel = TRUE and adjust = 1.  
library(klaR)

## Loading required package: MASS

##   
## Attaching package: 'MASS'

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

library(caret)  
library(e1071)  
library(MASS)  
set.seed(520)  
nb\_grid <- expand.grid(fL= 0, usekernel= c("TRUE"), adjust=1)  
# nb\_grid <- expand.grid(fL= c(0,1), usekernel= c("TRUE", "FALSE"), adjust=c(0,1,2,3))  
nab\_mod<- train(x=x, y=y, method="nb", metric="ROC", tuneGrid=nb\_grid, trControl = trainControl(method="repeatedcv", number=10, repeats=10, classProbs=TRUE, summaryFunction = twoClassSummary, verboseIter=FALSE))

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 397

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 452

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 470

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 515

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 397

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 452

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 470

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 515

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 348

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 451

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 348

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 451

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 382  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 382

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 282

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 509

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 282

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 509

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 449

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 508

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 449

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 508

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 169

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 251

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 288

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 462

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 169

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 251

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 288

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 462

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 453

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 552

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 453

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 552

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 554  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 554

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 264

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 500

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 264

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 500

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 343

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 394

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 476

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 522

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 343

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 394

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 476

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 522

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 244

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 454

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 244

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 454

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 433

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 497

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 433

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 497

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 168

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 289

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 398

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 453

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 469

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 168

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 289

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 398

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 453

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 469

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 535  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 535

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 292  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 292

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 284  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 284

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 252

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 392

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 252

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 392

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 464

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 465

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 502

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 464

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 465

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 502

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 183

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 551

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 183

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 551

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 377

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 440

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 377

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 440

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 437  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 437

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 441

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 508

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 441

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 508

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 345

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 479

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 509

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 532

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 345

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 479

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 509

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 532

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 286

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 437

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 286

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 437

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 282  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 282

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 253

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 439

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 253

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 439

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 400  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 400

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 377

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 520

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 555

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 377

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 520

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 555

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 438  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 438

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 352

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 473

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 506

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 352

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 473

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 506

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 182  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 182

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 256  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 256

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 383

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 557

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 383

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 557

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 462  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 462

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 292  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 292

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 282

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 434

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 282

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 434

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 344  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 344

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 174  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 174

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 442

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 447

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 442

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 447

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 376

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 449

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 453

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 496

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 514

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 376

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 449

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 453

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 496

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 514

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 170

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 435

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 460

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 500

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 518

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 170

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 435

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 460

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 500

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 518

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 495  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 495

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 246

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 437

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 551

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 246

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 437

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 551

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 382

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 445

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 382

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 445

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 536  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 536

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 283

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 378

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 452

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 283

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 378

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 452

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 471  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 471

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 285  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 285

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 353  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 353

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 446

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 514

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 446

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 514

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 282

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 397

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 459

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 505

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 282

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 397

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 459

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 505

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 287

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 341

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 373

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 287

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 341

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 373

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 251  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 251

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 448

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 463

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 542

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 448

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 463

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 542

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 169

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 491

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 169

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 491

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 549  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 549

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 266

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 502

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 266

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 502

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 534  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 534

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 281

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 435

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 281

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 435

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 385

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 386

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 506

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 385

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 386

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 506

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 172

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 250

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 551

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 172

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 250

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 551

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 290

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 436

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 290

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 436

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 334

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 446

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 334

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 446

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 530  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 530

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 380  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 380

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 549  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 549

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 385

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 448

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 385

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 448

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 252

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 343

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 458

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 252

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 343

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 458

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 189

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 514

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 189

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 514

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 450

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 509

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 450

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 509

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 290  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 290

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 283  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 283

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 456

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 456

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 333  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 333

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 170

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 459

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 170

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 459

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 383

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 434

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 383

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 434

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 495

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 507

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 554

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 495

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 507

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 554

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 462  
  
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 462

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 247

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 295

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 432

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 247

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 295

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 432

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 381

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 448

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 381

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 448

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 460

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 505

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 460

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 505

# to see model results:  
# nab\_mod  
# predict output using test set - even though we did 10x10 cv above, we are also using 'validation' set of 30% of the data:  
nab\_pred<- predict(nab\_mod, test\_noy)

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 962

s<-table(nab\_pred, test\_labels)  
# Confusion matrix  
print(confusionMatrix(s))

## Confusion Matrix and Statistics  
##   
## test\_labels  
## nab\_pred no yes  
## no 982 55  
## yes 214 106  
##   
## Accuracy : 0.8018   
## 95% CI : (0.7795, 0.8227)  
## No Information Rate : 0.8814   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.3359   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8211   
## Specificity : 0.6584   
## Pos Pred Value : 0.9470   
## Neg Pred Value : 0.3313   
## Prevalence : 0.8814   
## Detection Rate : 0.7237   
## Detection Prevalence : 0.7642   
## Balanced Accuracy : 0.7397   
##   
## 'Positive' Class : no   
##

### we can now run 10-fold on test dataset:

# copy in files you need and use test dataset only   
banking<-BM\_test  
  
# the other way is to run 10-fold on the test dataset and take the average of the (10 times) F1 measure   
folds<- createFolds(banking$y, k=10)  
 # create a function to do 10 folds of the data and run the statistics...  
 results <- lapply(folds, function(x) {  
 test<- banking[x,]  
 pred<- predict(nab\_mod, test[-17])  
 actual<- test$y  
 # PPV = TP/(TP+FP)  
 # pos<-posPredValue(table(pred, actual))  
 # I actually want: NPV= TN/(TN+FN) for precision of minority class  
 pr<-negPredValue(table(pred, actual))  
 # pr<-precision(table(pred, actual ))  
 # rec<- recall(table(pred, actual))  
 # i actually want specificity for recall of minority class  
 rec<- specificity(table(pred, actual))  
 F1<- 2 \* pr \* rec /(pr + rec)  
 return(F1)  
 })

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 95

#  
 # print(results)  
 value<-mean(unlist(results))  
 print(value)

## [1] 0.4379057

# print the average of the 10 F1 results for test set

## model 2 - C5.0 Decision Tree algorithm

strengths: numeric/nominal, easy interpretation weaknesses: biased splits, overfitting

# The final tuning parameters used for the original model were trials = 1, model = tree and winnow = FALSE.  
# Trials = an integer specifying the number of boosting iterations. A value of one indicates that a single model is used.  
# winnow: A logical: should predictor winnowing (i.e feature selection) be used.  
library(caret)  
set.seed(40)  
# c5\_grid <- expand.grid(trials=c(1,3,5), model = c("tree", "rules"), winnow = c(TRUE, FALSE))  
c5\_grid <- expand.grid(trials = 1, model = "tree", winnow = FALSE)  
c5\_ctrl<- trainControl(method="repeatedcv", number = 10, repeats=10, classProbs=TRUE, summaryFunction = twoClassSummary)  
c5\_mod<- train(x,y, method="C5.0", metric="ROC", tuneGrid=c5\_grid, trControl = c5\_ctrl, verbose=FALSE)  
c\_pred<- predict(c5\_mod, test\_noy)  
# Testing the result output  
s<-table(c\_pred, test\_labels)  
# Confusion matrix  
print(confusionMatrix(s))

## Confusion Matrix and Statistics  
##   
## test\_labels  
## c\_pred no yes  
## no 1063 80  
## yes 133 81  
##   
## Accuracy : 0.843   
## 95% CI : (0.8226, 0.862)  
## No Information Rate : 0.8814   
## P-Value [Acc > NIR] : 0.9999886   
##   
## Kappa : 0.343   
##   
## Mcnemar's Test P-Value : 0.0003667   
##   
## Sensitivity : 0.8888   
## Specificity : 0.5031   
## Pos Pred Value : 0.9300   
## Neg Pred Value : 0.3785   
## Prevalence : 0.8814   
## Detection Rate : 0.7833   
## Detection Prevalence : 0.8423   
## Balanced Accuracy : 0.6960   
##   
## 'Positive' Class : no   
##

### we can now run 10-fold on test dataset:

# copy in files you need and use test dataset only   
banking<-BM\_test  
  
# the other way is to run 10-fold on the test dataset and take the average of the (10 times) F1 measure   
folds<- createFolds(banking$y, k=10)  
 # create a function to do 10 folds of the data and run the statistics...  
 results <- lapply(folds, function(x) {  
 test<- banking[x,]  
 pred<- predict(c5\_mod, test[-17])  
 actual<- test$y  
 # PPV = TP/(TP+FP)  
 # pos<-posPredValue(table(pred, actual))  
 # I actually want: NPV= TN/(TN+FN) for precision of minority class  
 pr<-negPredValue(table(pred, actual))  
 # pr<-precision(table(pred, actual ))  
 # rec<- recall(table(pred, actual))  
 # i actually want specificity for recall of minority class  
 rec<- specificity(table(pred, actual))  
 F1<- 2 \* pr \* rec /(pr + rec)  
 return(F1)  
 })  
 #  
 # print(results)  
 value<-mean(unlist(results))  
 print(value)

## [1] 0.4315609

# print the average of the 10 F1 results for test set

### to see model results:

#c5\_mod  
#c5\_mod$finalModel  
# another way to see the tree's decisions and best attributes:  
summary(c5\_mod)

##   
## Call:  
## (function (x, y, trials = 1, rules = FALSE, weights = NULL, control  
## = FALSE, CF = 0.25, minCases = 2, fuzzyThreshold = FALSE, sample =  
## 0, earlyStopping = TRUE, label = "outcome", seed = 2372L), verbose = FALSE)  
##   
##   
## C5.0 [Release 2.07 GPL Edition] Mon Aug 3 17:34:53 2020  
## -------------------------------  
##   
## Class specified by attribute `outcome'  
##   
## Read 5608 cases (17 attributes) from undefined.data  
##   
## Decision tree:  
##   
## duration <= 0.1985522:  
## :...poutcome = success:  
## : :...education in {primary,unknown}: no (6)  
## : : education in {secondary,tertiary}:  
## : : :...day > 0.7333333: no (4)  
## : : day <= 0.7333333:  
## : : :...housing = no: yes (62/1)  
## : : housing = yes:  
## : : :...duration <= 0.1706308: no (5)  
## : : duration > 0.1706308: yes (16)  
## : poutcome in {failure,other,unknown}:  
## : :...month in {aug,dec,jan,jul,jun,may,nov}:  
## : :...job = student:  
## : : :...month in {dec,jan,jun,may}: no (13)  
## : : : month in {aug,jul,nov}:  
## : : : :...marital = divorced: yes (0)  
## : : : marital = married: no (3)  
## : : : marital = single:  
## : : : :...previous > 0.6: no (2)  
## : : : previous <= 0.6:  
## : : : :...duration <= 0.08066184: no (2)  
## : : : duration > 0.08066184: yes (27/1)  
## : : job in {admin.,blue-collar,entrepreneur,housemaid,management,  
## : : : retired,self-employed,services,technician,unemployed,  
## : : : unknown}:  
## : : :...poutcome = other:  
## : : :...balance <= 0.1531724: no (43)  
## : : : balance > 0.1531724:  
## : : : :...previous > 0.4: no (5)  
## : : : previous <= 0.4:  
## : : : :...duration <= 0.08893485: yes (21/1)  
## : : : duration > 0.08893485: no (2)  
## : : poutcome in {failure,unknown}:  
## : : :...age <= 0.2205882:  
## : : :...housing = no:  
## : : : :...month = dec: no (0)  
## : : : : month = jun:  
## : : : : :...job in {admin.,entrepreneur,services,  
## : : : : : : technician}: no (6)  
## : : : : : job in {blue-collar,housemaid,management,  
## : : : : : retired,self-employed,unemployed,  
## : : : : : unknown}: yes (18)  
## : : : : month in {aug,jan,jul,may,nov}:  
## : : : : :...balance <= 0.06248665: no (67)  
## : : : : balance > 0.06248665:  
## : : : : :...balance <= 0.06323435: yes (27)  
## : : : : balance > 0.06323435: no (67)  
## : : : housing = yes:  
## : : : :...duration <= 0.1427094: no (216)  
## : : : duration > 0.1427094:  
## : : : :...duration > 0.14788: no (54)  
## : : : duration <= 0.14788:  
## : : : :...campaign <= 0.1: yes (12/1)  
## : : : campaign > 0.1: no (4)  
## : : age > 0.2205882:  
## : : :...pdays > 0.9859551:  
## : : :...duration <= 0.188211: no (21)  
## : : : duration > 0.188211: yes (4)  
## : : pdays <= 0.9859551:  
## : : :...marital in {divorced,married}: no (873)  
## : : marital = single:  
## : : :...age <= 0.4264706: no (117)  
## : : age > 0.4264706:  
## : : :...day > 0.2666667: no (18)  
## : : day <= 0.2666667:  
## : : :...month in {aug,dec,jan,jul,  
## : : : nov}: yes (8)  
## : : month in {jun,may}: no (4)  
## : month in {apr,feb,mar,oct,sep}:  
## : :...pdays > 0.5280899: no (48)  
## : pdays <= 0.5280899:  
## : :...marital = divorced:  
## : :...loan = yes: no (4)  
## : : loan = no:  
## : : :...housing = yes: yes (37/1)  
## : : housing = no:  
## : : :...balance <= 0.1351207: no (3)  
## : : balance > 0.1351207: yes (16/1)  
## : marital in {married,single}:  
## : :...job in {blue-collar,entrepreneur,self-employed,services,  
## : : student,unemployed,unknown}: no (58)  
## : job in {admin.,housemaid,management,retired,technician}:  
## : :...duration <= 0.07238883: no (30)  
## : duration > 0.07238883:  
## : :...previous > 0.4: no (8)  
## : previous <= 0.4:  
## : :...contact = unknown: yes (10)  
## : contact = telephone: no (3)  
## : contact = cellular:  
## : :...duration <= 0.07962772: yes (34/1)  
## : duration > 0.07962772:  
## : :...loan = yes: no (5)  
## : loan = no:  
## : :...age <= 0.1323529: no (6)  
## : age > 0.1323529: [S1]  
## duration > 0.1985522:  
## :...contact = unknown:  
## :...duration <= 0.434333:  
## : :...month in {apr,aug,dec,feb,jan,mar,oct,sep}: no (0)  
## : : month = nov: yes (8)  
## : : month in {jul,jun,may}:  
## : : :...age > 0.07352941: no (243)  
## : : age <= 0.07352941:  
## : : :...education = primary: yes (8)  
## : : education in {secondary,tertiary,unknown}: no (6)  
## : duration > 0.434333:  
## : :...education = unknown: no (5)  
## : education in {primary,secondary,tertiary}:  
## : :...duration > 0.7631851:  
## : :...loan = yes:  
## : : :...marital = divorced: yes (7)  
## : : : marital in {married,single}: no (6)  
## : : loan = no:  
## : : :...job in {entrepreneur,retired,self-employed,student,  
## : : : unknown}: yes (23)  
## : : job in {services,unemployed}: no (2)  
## : : job = admin.:  
## : : :...marital = married: no (1)  
## : : : marital in {divorced,single}: yes (10)  
## : : job = housemaid:  
## : : :...day <= 0.3: no (2)  
## : : : day > 0.3: yes (5)  
## : : job = management:  
## : : :...marital in {divorced,single}: yes (10)  
## : : : marital = married: no (3)  
## : : job = technician:  
## : : :...month = jun: no (1)  
## : : : month in {apr,aug,dec,feb,jan,jul,mar,may,nov,oct,  
## : : : sep}: yes (33/1)  
## : : job = blue-collar:  
## : : :...marital in {divorced,single}: yes (31)  
## : : marital = married:  
## : : :...housing = no: yes (9)  
## : : housing = yes: no (6)  
## : duration <= 0.7631851:  
## : :...job = unknown: yes (0)  
## : job in {entrepreneur,housemaid,self-employed,student,  
## : : technician,unemployed}: no (26)  
## : job in {admin.,blue-collar,management,retired,services}:  
## : :...day > 0.6333333:  
## : :...age <= 0.1323529: yes (4)  
## : : age > 0.1323529: no (17)  
## : day <= 0.6333333:  
## : :...day <= 0.03333334: yes (18)  
## : day > 0.03333334:  
## : :...campaign > 0.5: no (5)  
## : campaign <= 0.5:  
## : :...day <= 0.4333333:  
## : :...education in {primary,  
## : : : secondary}: no (18)  
## : : education = tertiary: yes (17/1)  
## : day > 0.4333333:  
## : :...marital in {divorced,single}: no (4)  
## : marital = married:  
## : :...day <= 0.6: yes (32)  
## : day > 0.6:  
## : :...age <= 0.2205882: no (2)  
## : age > 0.2205882: yes (6)  
## contact in {cellular,telephone}:  
## :...duration > 0.6483971:  
## :...contact = telephone:  
## : :...balance <= 0.08096561: no (6)  
## : : balance > 0.08096561:  
## : : :...month in {apr,may}: no (3)  
## : : month in {aug,dec,feb,jan,jul,jun,mar,oct,  
## : : : sep}: yes (54/1)  
## : : month = nov:  
## : : :...loan = no: no (3)  
## : : loan = yes: yes (8)  
## : contact = cellular:  
## : :...education = unknown:  
## : :...housing = no: no (3)  
## : : housing = yes: yes (6)  
## : education = primary:  
## : :...campaign <= 0: no (6)  
## : : campaign > 0:  
## : : :...balance > 0.06024354: yes (52)  
## : : balance <= 0.06024354:  
## : : :...balance <= 0.01570177: yes (6)  
## : : balance > 0.01570177: no (4)  
## : education in {secondary,tertiary}:  
## : :...month in {feb,mar,may,sep}: yes (154/8)  
## : month = apr:  
## : :...campaign <= 0.1: yes (54/3)  
## : : campaign > 0.1: no (3)  
## : month = dec:  
## : :...job = admin.: no (1)  
## : : job in {blue-collar,entrepreneur,housemaid,management,  
## : : retired,self-employed,services,student,  
## : : technician,unemployed,unknown}: yes (8)  
## : month = jan:  
## : :...poutcome = other: no (1)  
## : : poutcome in {failure,success,unknown}: yes (28/1)  
## : month = jun:  
## : :...job in {admin.,blue-collar,entrepreneur,housemaid,  
## : : : retired,self-employed,services,student,  
## : : : technician,unemployed,unknown}: yes (11)  
## : : job = management: no (1)  
## : month = oct:  
## : :...age <= 0.3382353: yes (5)  
## : : age > 0.3382353: no (2)  
## : month = aug:  
## : :...poutcome in {other,success}: yes (0)  
## : : poutcome = failure: no (1)  
## : : poutcome = unknown:  
## : : :...duration > 0.7797312: yes (151/2)  
## : : duration <= 0.7797312:  
## : : :...day > 0.6: no (5)  
## : : day <= 0.6:  
## : : :...duration <= 0.7300931: yes (34)  
## : : duration > 0.7300931: no (2)  
## : month = nov:  
## : :...age <= 0.2205882: no (5)  
## : : age > 0.2205882:  
## : : :...poutcome = failure: no (1)  
## : : poutcome in {other,success}: yes (15)  
## : : poutcome = unknown:  
## : : :...age <= 0.2794118: yes (37)  
## : : age > 0.2794118:  
## : : :...education = secondary: yes (13)  
## : : education = tertiary: no (4)  
## : month = jul:  
## : :...poutcome in {failure,other}: yes (0)  
## : poutcome = success: no (1)  
## : poutcome = unknown:  
## : :...campaign > 0.2: yes (78/1)  
## : campaign <= 0.2:  
## : :...marital in {divorced,single}: yes (38/1)  
## : marital = married:  
## : :...day <= 0.5333334: no (6)  
## : day > 0.5333334:  
## : :...housing = no: yes (13)  
## : housing = yes:  
## : :...day <= 0.6: yes (6)  
## : day > 0.6: no (2)  
## duration <= 0.6483971:  
## :...poutcome in {other,success}:  
## :...poutcome = success: yes (349/11)  
## : poutcome = other:  
## : :...education in {primary,unknown}: no (10)  
## : education in {secondary,tertiary}:  
## : :...month = dec: yes (0)  
## : month in {jan,jul,mar}: no (6)  
## : month in {apr,aug,feb,jun,may,nov,oct,sep}:  
## : :...job in {admin.,housemaid,management,retired,  
## : : self-employed,services,student,technician,  
## : : unknown}: yes (190/11)  
## : job in {blue-collar,entrepreneur,  
## : unemployed}: no (5)  
## poutcome in {failure,unknown}:  
## :...month in {dec,feb,jun,mar,oct,sep}:  
## :...job = blue-collar: no (7)  
## : job in {admin.,entrepreneur,housemaid,management,retired,  
## : : self-employed,services,student,technician,  
## : : unemployed,unknown}:  
## : :...day > 0.5:  
## : :...day <= 0.7333333: yes (145)  
## : : day > 0.7333333:  
## : : :...contact = telephone: no (2)  
## : : contact = cellular:  
## : : :...poutcome = failure: no (1)  
## : : poutcome = unknown: yes (41/3)  
## : day <= 0.5:  
## : :...month = dec:  
## : :...job = technician: no (1)  
## : : job in {admin.,entrepreneur,housemaid,  
## : : management,retired,self-employed,  
## : : services,student,unemployed,  
## : : unknown}: yes (6)  
## : month = oct:  
## : :...day <= 0.3333333: yes (7/1)  
## : : day > 0.3333333: no (2)  
## : month = jun:  
## : :...poutcome = failure: no (3)  
## : : poutcome = unknown:  
## : : :...contact = cellular: yes (80/5)  
## : : contact = telephone: no (1)  
## : month = mar:  
## : :...age > 0.6764706: no (2)  
## : : age <= 0.6764706:  
## : : :...marital in {divorced,married}: yes (20)  
## : : marital = single: no (1)  
## : month = sep:  
## : :...day <= 0.1666667: no (3)  
## : : day > 0.1666667:  
## : : :...day <= 0.3333333: yes (22/1)  
## : : day > 0.3333333: no (2)  
## : month = feb:  
## : :...job in {housemaid,unknown}: yes (0)  
## : job in {admin.,entrepreneur,self-employed,  
## : : services,student,  
## : : unemployed}: no (14)  
## : job in {management,retired,technician}:  
## : :...education in {primary,tertiary,  
## : : unknown}: yes (48/2)  
## : education = secondary:  
## : :...balance <= 0.2063662: no (7)  
## : balance > 0.2063662: yes (7)  
## month in {apr,aug,jan,jul,may,nov}:  
## :...age > 0.6029412:  
## :...poutcome = failure: no (3)  
## : poutcome = unknown:  
## : :...marital = single: no (2)  
## : marital in {divorced,married}:  
## : :...education = tertiary: yes (25/1)  
## : education = unknown: no (1)  
## : education = primary:  
## : :...day <= 0.6: no (2)  
## : : day > 0.6: yes (11)  
## : education = secondary:  
## : :...age <= 0.75: yes (32)  
## : age > 0.75: no (2)  
## age <= 0.6029412:  
## :...duration <= 0.3422958:  
## :...balance <= 0.0520188: no (81)  
## : balance > 0.0520188:  
## : :...loan = yes: no (46)  
## : loan = no:  
## : :...job in {blue-collar,housemaid,retired,  
## : : self-employed,services,technician,  
## : : unknown}:  
## : :...campaign > 0:  
## : : :...campaign <= 0.2: no (66)  
## : : : campaign > 0.2:  
## : : : :...month in {apr,aug,jan,jul,  
## : : : : nov}: no (17)  
## : : : month = may: yes (3)  
## : : campaign <= 0:  
## : : :...housing = yes:  
## : : :...day <= 0.9666666: no (37)  
## : : : day > 0.9666666: yes (9)  
## : : housing = no: [S2]  
## : job in {admin.,entrepreneur,management,  
## : : student,unemployed}:  
## : :...month in {jan,jul}:  
## : :...contact = cellular: no (17)  
## : : contact = telephone: [S3]  
## : month in {apr,aug,may,nov}:  
## : :...day > 0.6333333:  
## : :...month in {apr,  
## : : : may}: yes (59/2)  
## : : month = nov: no (1)  
## : : month = aug:  
## : : :...pdays <= 0.5: no (5)  
## : : pdays > 0.5: yes (3)  
## : day <= 0.6333333:  
## : :...day > 0.5: no (20)  
## : day <= 0.5:  
## : :...month = may: no (6)  
## : month in {apr,aug,nov}: [S4]  
## duration > 0.3422958:  
## :...contact = telephone:  
## :...housing = no: yes (43)  
## : housing = yes: no (4)  
## contact = cellular:  
## :...month in {jan,jul,nov}:  
## :...day <= 0.1: yes (13)  
## : day > 0.1:  
## : :...job in {admin.,entrepreneur,management,  
## : : retired,services,student,  
## : : unemployed}: no (56)  
## : job in {blue-collar,housemaid,  
## : : self-employed,technician,  
## : : unknown}:  
## : :...age <= 0.1617647: [S5]  
## : age > 0.1617647:  
## : :...duration <= 0.4529473: no (30)  
## : duration > 0.4529473: [S6]  
## month in {apr,aug,may}:  
## :...loan = yes: no (18)  
## loan = no:  
## :...age <= 0.1617647: no (18)  
## age > 0.1617647:  
## :...day > 0.9: yes (23)  
## day <= 0.9:  
## :...month = apr:  
## :...age <= 0.4705882: no (15)  
## : age > 0.4705882:  
## : :...pdays <= 0.1797753: yes (23)  
## : pdays > 0.1797753: no (3)  
## month in {aug,may}: [S7]  
##   
## SubTree [S1]  
##   
## education = unknown: yes (0)  
## education = primary: no (3)  
## education = secondary:  
## :...month in {apr,mar,sep}: no (10)  
## : month in {feb,oct}: yes (13)  
## education = tertiary:  
## :...job in {housemaid,retired}: no (3)  
## job in {admin.,management,technician}:  
## :...campaign <= 0.1: yes (53/4)  
## campaign > 0.1:  
## :...month in {apr,feb,oct,sep}: no (5)  
## month = mar: yes (5)  
##   
## SubTree [S2]  
##   
## job in {housemaid,unknown}: yes (0)  
## job in {blue-collar,retired,self-employed}: no (8)  
## job in {services,technician}:  
## :...education in {primary,unknown}: yes (6)  
## education = secondary:  
## :...age <= 0.3235294: yes (19)  
## : age > 0.3235294: no (2)  
## education = tertiary:  
## :...day <= 0.8: no (3)  
## day > 0.8: yes (5)  
##   
## SubTree [S3]  
##   
## education = secondary: no (3)  
## education in {primary,tertiary,unknown}: yes (6)  
##   
## SubTree [S4]  
##   
## campaign <= 0: yes (42/1)  
## campaign > 0:  
## :...month in {apr,nov}: yes (20)  
## month = aug:  
## :...contact = cellular: no (8)  
## contact = telephone: yes (3)  
##   
## SubTree [S5]  
##   
## education = primary: no (2)  
## education in {secondary,tertiary,unknown}: yes (35/3)  
##   
## SubTree [S6]  
##   
## job in {self-employed,technician}: no (8)  
## job in {blue-collar,housemaid,unknown}:  
## :...housing = no: yes (20/1)  
## housing = yes:  
## :...day <= 0.9666666: no (4)  
## day > 0.9666666: yes (5)  
##   
## SubTree [S7]  
##   
## job in {entrepreneur,housemaid}: yes (38/1)  
## job in {self-employed,services,student,unemployed,unknown}: no (8)  
## job = admin.:  
## :...balance <= 0.1381115: yes (24/1)  
## : balance > 0.1381115: no (2)  
## job = retired:  
## :...age <= 0.5441176: yes (12)  
## : age > 0.5441176: no (2)  
## job = blue-collar:  
## :...balance <= 0.126896: no (8)  
## : balance > 0.126896:  
## : :...duration <= 0.4250259: no (2)  
## : duration > 0.4250259:  
## : :...campaign <= 0: yes (17)  
## : campaign > 0:  
## : :...month = aug: yes (10)  
## : month = may: no (3)  
## job = management:  
## :...day > 0.5: no (6)  
## : day <= 0.5:  
## : :...age > 0.3529412:  
## : :...marital in {divorced,married}: no (5)  
## : : marital = single: yes (3)  
## : age <= 0.3529412:  
## : :...pdays > 0.8061798: no (4)  
## : pdays <= 0.8061798:  
## : :...education = secondary: no (1)  
## : education in {primary,tertiary,unknown}: yes (48/1)  
## job = technician:  
## :...poutcome = failure: no (2)  
## poutcome = unknown:  
## :...education in {primary,secondary,unknown}: yes (32/1)  
## education = tertiary:  
## :...balance <= 0.08224738: no (3)  
## balance > 0.08224738:  
## :...balance <= 0.1936552: yes (21)  
## balance > 0.1936552: no (2)  
##   
##   
## Evaluation on training data (5608 cases):  
##   
## Decision Tree   
## ----------------   
## Size Errors   
##   
## 221 76( 1.4%) <<  
##   
##   
## (a) (b) <-classified as  
## ---- ----  
## 2728 76 (a): class no  
## 2804 (b): class yes  
##   
##   
## Attribute usage:  
##   
## 100.00% duration  
## 85.66% month  
## 82.83% poutcome  
## 65.57% contact  
## 63.91% job  
## 56.19% age  
## 31.97% education  
## 29.53% marital  
## 26.50% pdays  
## 25.93% day  
## 19.44% loan  
## 17.62% balance  
## 14.62% housing  
## 12.64% campaign  
## 3.87% previous  
##   
##   
## Time: 0.1 secs

# we see the decision trees and....  
# ... and best attributes: see below at bottom....

## model 3 - JRip (rule learner)

# The final values used for the model were NumOpt = 10, NumFolds = 10 and MinWeights = 10.  
library(caret)  
library(RWeka)  
set.seed(50)  
# JR\_grid <- expand.grid(NumOpt=c(1,3,5,10), NumFolds=c(1,3,5,10),MinWeights=c(1,3,5,10))  
JR\_grid <- expand.grid(NumOpt=10, NumFolds=10, MinWeights=10)  
JR\_ctrl<- trainControl(method="repeatedcv", number = 10, repeats = 10, classProbs=TRUE, summaryFunction = twoClassSummary)  
JR\_mod<- train(x, y, method="JRip", metric="ROC", tuneGrid=JR\_grid, trControl = JR\_ctrl)  
JR\_pred<- predict(JR\_mod, test\_noy)  
s<-table(JR\_pred, test\_labels)  
# Confusion matrix  
print(confusionMatrix(s))

## Confusion Matrix and Statistics  
##   
## test\_labels  
## JR\_pred no yes  
## no 1013 48  
## yes 183 113  
##   
## Accuracy : 0.8298   
## 95% CI : (0.8087, 0.8494)  
## No Information Rate : 0.8814   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.4027   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8470   
## Specificity : 0.7019   
## Pos Pred Value : 0.9548   
## Neg Pred Value : 0.3818   
## Prevalence : 0.8814   
## Detection Rate : 0.7465   
## Detection Prevalence : 0.7819   
## Balanced Accuracy : 0.7744   
##   
## 'Positive' Class : no   
##

### we can now run 10-fold on test dataset:

# copy in files you need and use test dataset only   
banking<-BM\_test  
  
# the other way is to run 10-fold on the test dataset and take the average of the (10 times) F1 measure   
folds<- createFolds(banking$y, k=10)  
 # create a function to do 10 folds of the data and run the statistics...  
 results <- lapply(folds, function(x) {  
 test<- banking[x,]  
 pred<- predict(JR\_mod, test[-17])  
 actual<- test$y  
 # PPV = TP/(TP+FP)  
 # pos<-posPredValue(table(pred, actual))  
 # I actually want: NPV= TN/(TN+FN) for precision of minority class  
 pr<-negPredValue(table(pred, actual))  
 # pr<-precision(table(pred, actual ))  
 # rec<- recall(table(pred, actual))  
 # i actually want specificity for recall of minority class  
 rec<- specificity(table(pred, actual))  
 F1<- 2 \* pr \* rec /(pr + rec)  
 return(F1)  
 })  
 #  
 # print(results)  
 value<-mean(unlist(results))  
 print(value)

## [1] 0.492946

# print the average of the 10 F1 results for test set

### JRip rules compiled…

JR\_mod$finalModel

## JRIP rules:  
## ===========  
##   
## (duration >= 0.653568) and (contact = cellular) => .outcome=yes (757.0/64.0)  
## (duration >= 0.199586) and (pdays >= 0.106742) and (poutcome = success) => .outcome=yes (351.0/13.0)  
## (duration >= 0.377456) and (housing = no) and (balance >= 0.087374) and (day <= 0.233333) and (age >= 0.352941) => .outcome=yes (95.0/3.0)  
## (duration >= 0.409514) and (housing = no) and (balance >= 0.185217) => .outcome=yes (171.0/21.0)  
## (duration >= 0.199586) and (contact = cellular) and (previous >= 0.4) and (day >= 0.566667) and (balance <= 0.08919) => .outcome=yes (99.0/4.0)  
## (duration >= 0.768356) => .outcome=yes (143.0/28.0)  
## (duration >= 0.199586) and (contact = cellular) and (balance >= 0.083209) and (job = management) and (age <= 0.264706) => .outcome=yes (220.0/33.0)  
## (duration >= 0.344364) and (campaign <= 0) and (housing = no) and (age >= 0.602941) => .outcome=yes (40.0/3.0)  
## (duration >= 0.344364) and (day <= 0.2) and (education = tertiary) => .outcome=yes (91.0/10.0)  
## (campaign <= 0) and (duration >= 0.430196) and (duration <= 0.497415) and (balance <= 0.009293) => .outcome=yes (25.0/1.0)  
## (duration >= 0.199586) and (balance >= 0.093036) and (contact = cellular) and (age >= 0.426471) and (day <= 0.2) => .outcome=yes (84.0/9.0)  
## (duration >= 0.199586) and (housing = no) and (month = apr) and (day >= 0.666667) => .outcome=yes (65.0/1.0)  
## (duration >= 0.199586) and (month = oct) => .outcome=yes (75.0/6.0)  
## (duration >= 0.562565) and (day >= 0.366667) and (age <= 0.25) and (job = blue-collar) => .outcome=yes (40.0/3.0)  
## (loan = no) and (month = aug) and (duration >= 0.423992) => .outcome=yes (42.0/7.0)  
## (duration >= 0.163392) and (balance >= 0.150182) and (housing = no) and (age >= 0.602941) => .outcome=yes (76.0/12.0)  
## (duration >= 0.210962) and (job = admin.) and (day >= 0.766667) => .outcome=yes (50.0/11.0)  
## (duration >= 0.210962) and (day <= 0.066667) and (age >= 0.279412) and (pdays <= 0) and (balance <= 0.063768) => .outcome=yes (27.0/0.0)  
## (duration >= 0.205791) and (balance >= 0.083209) and (housing = no) and (day <= 0.4) and (previous >= 0.8) => .outcome=yes (25.0/1.0)  
## (duration >= 0.211996) and (balance >= 0.093142) and (housing = no) and (day <= 0.166667) and (age <= 0.191176) => .outcome=yes (24.0/3.0)  
## (age <= 0.147059) and (housing = no) and (balance >= 0.060137) and (balance <= 0.077334) => .outcome=yes (68.0/8.0)  
## (duration >= 0.211996) and (balance >= 0.093142) and (age >= 0.397059) and (month = aug) and (balance <= 0.288293) => .outcome=yes (21.0/3.0)  
## (duration >= 0.364012) and (age <= 0.088235) => .outcome=yes (21.0/3.0)  
## (balance >= 0.148473) and (pdays >= 0.002809) and (pdays <= 0.280899) => .outcome=yes (60.0/12.0)  
## (duration >= 0.423992) and (month = may) and (job = admin.) and (balance >= 0.093142) => .outcome=yes (22.0/1.0)  
## (education = tertiary) and (month = mar) and (age >= 0.161765) => .outcome=yes (28.0/0.0)  
## (age >= 0.279412) and (job = retired) and (duration >= 0.291624) and (age <= 0.544118) and (campaign <= 0.1) => .outcome=yes (22.0/1.0)  
## (duration >= 0.058945) and (balance >= 0.290323) and (month = feb) => .outcome=yes (38.0/5.0)  
## (education = tertiary) and (month = apr) and (balance <= 0.043367) => .outcome=yes (24.0/2.0)  
## (duration >= 0.079628) and (month = oct) and (day >= 0.666667) => .outcome=yes (34.0/2.0)  
## (poutcome = success) and (pdays <= 0.258427) => .outcome=yes (27.0/1.0)  
## (balance >= 0.17069) and (job = management) and (duration <= 0.268873) and (duration >= 0.233713) => .outcome=yes (27.0/1.0)  
## (month = sep) and (contact = telephone) => .outcome=yes (13.0/0.0)  
## (age <= 0.279412) and (campaign <= 0) and (balance >= 0.15627) and (balance <= 0.156911) => .outcome=yes (17.0/0.0)  
## (duration >= 0.291624) and (day <= 0.033333) => .outcome=yes (20.0/6.0)  
## (duration <= 0.441572) and (duration >= 0.437435) and (month = may) => .outcome=yes (15.0/0.0)  
## (age <= 0.264706) and (balance <= 0.063021) and (balance >= 0.063021) => .outcome=yes (15.0/0.0)  
## (duration >= 0.080662) and (duration <= 0.093071) and (day <= 0.333333) and (day >= 0.233333) and (marital = single) and (balance <= 0.058107) and (balance >= 0.042192) => .outcome=yes (21.0/0.0)  
## (duration >= 0.080662) and (month = apr) and (day >= 0.633333) and (duration <= 0.131334) => .outcome=yes (29.0/9.0)  
## (duration >= 0.143744) and (day <= 0.466667) and (month = nov) => .outcome=yes (20.0/2.0)  
## (age <= 0.220588) and (age >= 0.191176) and (campaign <= 0.1) and (balance <= 0.06035) and (balance >= 0.056185) => .outcome=yes (28.0/7.0)  
## (balance >= 0.150182) and (balance <= 0.155843) and (age >= 0.323529) and (day >= 0.333333) => .outcome=yes (11.0/0.0)  
## => .outcome=no (2527.0/19.0)  
##   
## Number of Rules : 43

# all for outcome yes

## model 4 - Logistic Regression

# no parameters  
library(caret)  
set.seed(520)  
log\_mod<- train(x, y, method="glm", metric="ROC", family=binomial(link="logit"), trControl = trainControl(method="repeatedcv", number=10, repeats=10, verboseIter=FALSE, classProbs = TRUE, summaryFunction = twoClassSummary))  
log\_pred<- predict(log\_mod, test\_noy)  
s<-table(log\_pred, test\_labels)  
# Confusion matrix  
print(confusionMatrix(s))

## Confusion Matrix and Statistics  
##   
## test\_labels  
## log\_pred no yes  
## no 994 34  
## yes 202 127  
##   
## Accuracy : 0.8261   
## 95% CI : (0.8049, 0.8459)  
## No Information Rate : 0.8814   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.4271   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8311   
## Specificity : 0.7888   
## Pos Pred Value : 0.9669   
## Neg Pred Value : 0.3860   
## Prevalence : 0.8814   
## Detection Rate : 0.7325   
## Detection Prevalence : 0.7576   
## Balanced Accuracy : 0.8100   
##   
## 'Positive' Class : no   
##

### we can now run 10-fold on test dataset:

# copy in files you need and use test dataset only   
banking<-BM\_test  
  
# the other way is to run 10-fold on the test dataset and take the average of the (10 times) F1 measure   
folds<- createFolds(banking$y, k=10)  
 # create a function to do 10 folds of the data and run the statistics...  
 results <- lapply(folds, function(x) {  
 test<- banking[x,]  
 pred<- predict(log\_mod, test[-17])  
 actual<- test$y  
 # PPV = TP/(TP+FP)  
 # pos<-posPredValue(table(pred, actual))  
 # I actually want: NPV= TN/(TN+FN) for precision of minority class  
 pr<-negPredValue(table(pred, actual))  
 # pr<-precision(table(pred, actual ))  
 # rec<- recall(table(pred, actual))  
 # i actually want specificity for recall of minority class  
 rec<- specificity(table(pred, actual))  
 F1<- 2 \* pr \* rec /(pr + rec)  
 return(F1)  
 })  
 #  
 # print(results)  
 value<-mean(unlist(results))  
 print(value)

## [1] 0.5181706

# print the average of the 10 F1 results for test set

### model results

# to see model results:  
summary(log\_mod)

##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.0751 -0.5775 -0.0160 0.5627 2.3579   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.7828882 0.4043346 -1.936 0.052838 .   
## age -0.3908221 0.3335590 -1.172 0.241328   
## balance 0.2990409 0.1802698 1.659 0.097146 .   
## day 0.0420382 0.1596081 0.263 0.792255   
## duration 6.1603835 0.1846725 33.358 < 2e-16 \*\*\*  
## campaign -1.1486741 0.2141031 -5.365 8.09e-08 \*\*\*  
## pdays -0.3310399 0.2886749 -1.147 0.251482   
## previous 0.6394368 0.2640733 2.421 0.015459 \*   
## jobblue-collar -1.0603682 0.1624103 -6.529 6.62e-11 \*\*\*  
## jobentrepreneur -0.2010562 0.2246436 -0.895 0.370787   
## jobhousemaid -0.9685234 0.2750557 -3.521 0.000430 \*\*\*  
## jobmanagement -0.2294053 0.1586167 -1.446 0.148097   
## jobretired 0.1796008 0.2120545 0.847 0.397020   
## jobself-employed -0.4094192 0.2421491 -1.691 0.090880 .   
## jobservices -0.5541746 0.1792588 -3.091 0.001992 \*\*   
## jobstudent 0.1982894 0.2719377 0.729 0.465896   
## jobtechnician -0.4565498 0.1485758 -3.073 0.002120 \*\*   
## jobunemployed -0.9751550 0.2873443 -3.394 0.000690 \*\*\*  
## jobunknown -0.1312502 0.4496665 -0.292 0.770376   
## maritalmarried -0.2458346 0.1247108 -1.971 0.048697 \*   
## maritalsingle 0.0005812 0.1450005 0.004 0.996802   
## educationsecondary -0.0620101 0.1349464 -0.460 0.645863   
## educationtertiary 0.0315484 0.1545057 0.204 0.838206   
## educationunknown -0.5965558 0.2386940 -2.499 0.012446 \*   
## defaultyes 0.3172674 0.3332387 0.952 0.341060   
## housingyes -0.2993707 0.0876310 -3.416 0.000635 \*\*\*  
## loanyes -1.0712360 0.1292442 -8.288 < 2e-16 \*\*\*  
## contacttelephone 0.0971061 0.1540105 0.631 0.528357   
## contactunknown -1.6509368 0.1428892 -11.554 < 2e-16 \*\*\*  
## monthaug -0.4057871 0.1611281 -2.518 0.011789 \*   
## monthdec 0.6376095 0.4794198 1.330 0.183531   
## monthfeb 0.4568610 0.2004408 2.279 0.022650 \*   
## monthjan -1.9766328 0.2830019 -6.985 2.86e-12 \*\*\*  
## monthjul -0.7970542 0.1658584 -4.806 1.54e-06 \*\*\*  
## monthjun 0.1880138 0.2010374 0.935 0.349676   
## monthmar 1.7946929 0.3165783 5.669 1.44e-08 \*\*\*  
## monthmay -0.7023544 0.1603531 -4.380 1.19e-05 \*\*\*  
## monthnov -0.9311400 0.1811305 -5.141 2.74e-07 \*\*\*  
## monthoct 1.9626950 0.2826783 6.943 3.83e-12 \*\*\*  
## monthsep 0.8632296 0.3199385 2.698 0.006973 \*\*   
## poutcomeother 0.8997820 0.1880149 4.786 1.70e-06 \*\*\*  
## poutcomesuccess 2.5787477 0.2290729 11.257 < 2e-16 \*\*\*  
## poutcomeunknown 0.1214048 0.2616391 0.464 0.642636   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 7774.3 on 5607 degrees of freedom  
## Residual deviance: 4404.9 on 5565 degrees of freedom  
## AIC: 4490.9  
##   
## Number of Fisher Scoring iterations: 5

### plot to see the most important attributes (those that “stand out” at far left or right)

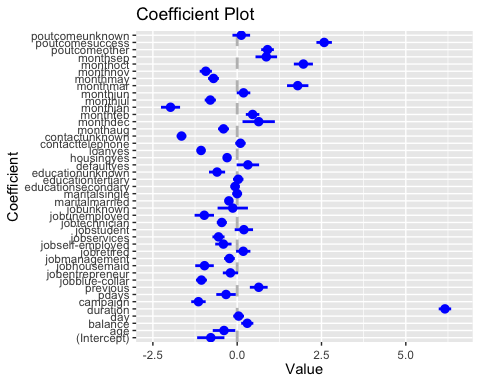
require(coefplot)

## Loading required package: coefplot

##   
## Attaching package: 'coefplot'

## The following object is masked from 'package:e1071':  
##   
## extractPath

coefplot(log\_mod)



# to reinterpret data properly  
# invlogit<- function (x) {1/(1+exp(-x))}  
# invlogit(log\_mod$coefficients)  
# results of plot: duration, campaign, balance, poutcomesuccess, some months, , some jobs

## Model #5 - K-Nearest Neighbours

# The final values used for the model were kmax = 9, distance = 2 and kernel = optimal  
library(caret)  
library(lattice)  
library(ggplot2)  
knn\_ctrl<- trainControl(method="repeatedcv", number = 10, repeats = 10, classProbs=TRUE, summaryFunction = twoClassSummary)  
# knn\_grid<-expand.grid(kmax=c(5,7,9,13), distance=c(1,2,4,6), kernel=c("rectangular","rank","optimal"))  
knn\_grid<-expand.grid(kmax=9, distance=2, kernel="optimal")  
knn\_mod<- train(x=x,y=y, method="kknn", metric="ROC", tuneGrid=knn\_grid, trControl=knn\_ctrl, verbose=FALSE)  
knn\_pred<- predict(knn\_mod, test\_noy)  
# Testing the result output  
s<-table( knn\_pred, test\_labels)  
# Confusion matrix  
print(confusionMatrix(s))

## Confusion Matrix and Statistics  
##   
## test\_labels  
## knn\_pred no yes  
## no 1130 100  
## yes 66 61  
##   
## Accuracy : 0.8777   
## 95% CI : (0.859, 0.8946)  
## No Information Rate : 0.8814   
## P-Value [Acc > NIR] : 0.68085   
##   
## Kappa : 0.3563   
##   
## Mcnemar's Test P-Value : 0.01043   
##   
## Sensitivity : 0.9448   
## Specificity : 0.3789   
## Pos Pred Value : 0.9187   
## Neg Pred Value : 0.4803   
## Prevalence : 0.8814   
## Detection Rate : 0.8327   
## Detection Prevalence : 0.9064   
## Balanced Accuracy : 0.6618   
##   
## 'Positive' Class : no   
##

### we can now run 10-fold on test dataset:

# copy in files you need and use test dataset only   
banking<-BM\_test  
  
# the other way is to run 10-fold on the test dataset and take the average of the (10 times) F1 measure   
folds<- createFolds(banking$y, k=10)  
 # create a function to do 10 folds of the data and run the statistics...  
 results <- lapply(folds, function(x) {  
 test<- banking[x,]  
 pred<- predict(knn\_mod, test[-17])  
 actual<- test$y  
 # PPV = TP/(TP+FP)  
 # pos<-posPredValue(table(pred, actual))  
 # I actually want: NPV= TN/(TN+FN) for precision of minority class  
 pr<-negPredValue(table(pred, actual))  
 # pr<-precision(table(pred, actual ))  
 # rec<- recall(table(pred, actual))  
 # i actually want specificity for recall of minority class  
 rec<- specificity(table(pred, actual))  
 F1<- 2 \* pr \* rec /(pr + rec)  
 return(F1)  
 })  
 #  
 # print(results)  
 value<-mean(unlist(results))  
 print(value)

## [1] 0.4165229

# print the average of the 10 F1 results for test set

## NEURAL NET - model 6

Positive: can be used for classification or numeric prediction, makes few assumptions about the data (doesnt have to be normalized). Negative: SLOW..can overfit, ‘black box’.

# #1: The final parameter values used for the model were size = 16 and decay = 0.1.  
# Size is the number of units in hidden layer (nnet fit a single hidden layer neural network) and   
# decay is the regularization parameter to avoid over-fitting.  
require(mlbench)

## Loading required package: mlbench

require(caret)  
require (nnet)

## Loading required package: nnet

nnctrl = trainControl(method="repeatedcv", number=10, repeats = 10, classProbs=TRUE, summaryFunction = twoClassSummary)  
# initially, create a grid list to find best parameters:  
# nn\_grid = expand.grid(size=c(1,4,8,16),decay=c(0,0.1,0.2,0.3,0.4))  
nn\_grid = expand.grid(size=16, decay=0.1)  
nn\_mod <- train(x=x, y=y, method="nnet", metric="ROC", trControl=nnctrl, tuneGrid=nn\_grid, trace=FALSE)  
nn\_pred<- predict(nn\_mod, test\_noy)  
# Testing the result output  
s<-table( nn\_pred, test\_labels)  
# Confusion matrix  
print(confusionMatrix(s))

## Confusion Matrix and Statistics  
##   
## test\_labels  
## nn\_pred no yes  
## no 1066 74  
## yes 130 87  
##   
## Accuracy : 0.8497   
## 95% CI : (0.8295, 0.8683)  
## No Information Rate : 0.8814   
## P-Value [Acc > NIR] : 0.9997955   
##   
## Kappa : 0.3752   
##   
## Mcnemar's Test P-Value : 0.0001177   
##   
## Sensitivity : 0.8913   
## Specificity : 0.5404   
## Pos Pred Value : 0.9351   
## Neg Pred Value : 0.4009   
## Prevalence : 0.8814   
## Detection Rate : 0.7856   
## Detection Prevalence : 0.8401   
## Balanced Accuracy : 0.7158   
##   
## 'Positive' Class : no   
##

### we can now run 10-fold on test dataset:

# copy in files you need and use test dataset only   
banking<-BM\_test  
  
# the other way is to run 10-fold on the test dataset and take the average of the (10 times) F1 measure   
folds<- createFolds(banking$y, k=10)  
 # create a function to do 10 folds of the data and run the statistics...  
 results <- lapply(folds, function(x) {  
 test<- banking[x,]  
 pred<- predict(nn\_mod, test[-17])  
 actual<- test$y  
 # PPV = TP/(TP+FP)  
 # pos<-posPredValue(table(pred, actual))  
 # I actually want: NPV= TN/(TN+FN) for precision of minority class  
 pr<-negPredValue(table(pred, actual))  
 # pr<-precision(table(pred, actual ))  
 # rec<- recall(table(pred, actual))  
 # i actually want specificity for recall of minority class  
 rec<- specificity(table(pred, actual))  
 F1<- 2 \* pr \* rec /(pr + rec)  
 return(F1)  
 })  
 #  
 # print(results)  
 value<-mean(unlist(results))  
 print(value)

## [1] 0.458881

# print the average of the 10 F1 results for test set

## SVM model 7 - using caret package and linear kernel.

### Another supervised learning model. A SVM can be imagined as a surface thatc reates a boundary between points of data plotted in an n-space representing examples and their feature values. SVM creates a flat boundary called a hyperplane, which divides the space into homogeneous partitions on either side. This way, SVM combines nearest neighbour instance-based learning with linear regression modeling.

negatives: need to test various parameters and kernels to get best solution, can be slow, ’black box". positives: good for binary classification, classification/numeric prediction

# Linear (vanilla) kernel function.   
# The final values used for the model were Cost = 1 for classification.  
library(caret)  
library(kernlab)

##   
## Attaching package: 'kernlab'

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

fitctrl<- trainControl(method="repeatedcv", number = 10, repeats = 10, classProbs=TRUE, summaryFunction = twoClassSummary)  
grid<-expand.grid(C=1)  
sv\_m<- train(Class~., data=trainsv, method="svmLinear", metric="ROC", trControl=fitctrl, tunegrid=grid, verbose=FALSE)

## line search fails -1.262654 0.02477884 1.84966e-05 1.985099e-06 -2.103301e-08 -3.754597e-09 -3.964924e-13

## Warning in method$predict(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class prediction calculations failed; returning NAs

## Warning in method$prob(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class probability calculations failed; returning NAs

## Warning in data.frame(..., check.names = FALSE): row names were found from a  
## short variable and have been discarded

## line search fails -1.30235 6.48553e-05 2.08313e-05 2.386534e-06 -2.518121e-08 -5.07854e-09 -5.366774e-13

## Warning in method$predict(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class prediction calculations failed; returning NAs

## Warning in method$prob(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class probability calculations failed; returning NAs

## Warning in data.frame(..., check.names = FALSE): row names were found from a  
## short variable and have been discarded

## line search fails -1.328557 0.007437473 3.026781e-05 3.354336e-06 -3.817605e-08 -7.028079e-09 -1.17908e-12

## Warning in method$predict(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class prediction calculations failed; returning NAs

## Warning in method$prob(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class probability calculations failed; returning NAs

## Warning in data.frame(..., check.names = FALSE): row names were found from a  
## short variable and have been discarded

## line search fails -1.316453 -0.0002504906 2.527373e-05 2.647994e-06 -3.154742e-08 -5.732852e-09 -8.125016e-13

## Warning in method$predict(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class prediction calculations failed; returning NAs

## Warning in method$prob(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class probability calculations failed; returning NAs

## Warning in data.frame(..., check.names = FALSE): row names were found from a  
## short variable and have been discarded

## line search fails -1.294566 -0.006840061 2.360271e-05 2.799796e-06 -2.832214e-08 -6.178883e-09 -6.857789e-13

## Warning in method$predict(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class prediction calculations failed; returning NAs

## Warning in method$prob(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class probability calculations failed; returning NAs

## Warning in data.frame(..., check.names = FALSE): row names were found from a  
## short variable and have been discarded

## line search fails -1.30676 0.01094271 1.845987e-05 2.066398e-06 -2.249503e-08 -4.188298e-09 -4.2391e-13

## Warning in method$predict(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class prediction calculations failed; returning NAs

## Warning in method$prob(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class probability calculations failed; returning NAs

## Warning in data.frame(..., check.names = FALSE): row names were found from a  
## short variable and have been discarded

## line search fails -1.278014 0.01730588 1.962094e-05 1.931042e-06 -2.296435e-08 -3.765724e-09 -4.578539e-13

## Warning in method$predict(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class prediction calculations failed; returning NAs

## Warning in method$prob(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class probability calculations failed; returning NAs

## Warning in data.frame(..., check.names = FALSE): row names were found from a  
## short variable and have been discarded

## line search fails -1.316534 -0.003707503 2.491635e-05 2.861108e-06 -3.07669e-08 -6.238751e-09 -7.844486e-13

## Warning in method$predict(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class prediction calculations failed; returning NAs

## Warning in method$prob(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class probability calculations failed; returning NAs

## Warning in data.frame(..., check.names = FALSE): row names were found from a  
## short variable and have been discarded

## line search fails -1.315039 -0.005282885 1.079656e-05 9.069267e-07 -1.328684e-08 -2.023724e-09 -1.452875e-13

## Warning in method$predict(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class prediction calculations failed; returning NAs

## Warning in method$prob(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class probability calculations failed; returning NAs

## Warning in data.frame(..., check.names = FALSE): row names were found from a  
## short variable and have been discarded

## line search fails -1.291192 0.02889126 2.534992e-05 2.598665e-06 -3.021612e-08 -4.993925e-09 -7.789539e-13

## Warning in method$predict(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class prediction calculations failed; returning NAs

## Warning in method$prob(modelFit = modelFit, newdata = newdata, submodels =  
## param): kernlab class probability calculations failed; returning NAs

## Warning in data.frame(..., check.names = FALSE): row names were found from a  
## short variable and have been discarded

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

sp<- predict(sv\_m, test\_noy)  
s<-table( sp, test\_labels)  
# Confusion matrix  
print(confusionMatrix(s))

## Confusion Matrix and Statistics  
##   
## test\_labels  
## sp no yes  
## no 1002 35  
## yes 194 126  
##   
## Accuracy : 0.8312   
## 95% CI : (0.8102, 0.8508)  
## No Information Rate : 0.8814   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.4347   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8378   
## Specificity : 0.7826   
## Pos Pred Value : 0.9662   
## Neg Pred Value : 0.3937   
## Prevalence : 0.8814   
## Detection Rate : 0.7384   
## Detection Prevalence : 0.7642   
## Balanced Accuracy : 0.8102   
##   
## 'Positive' Class : no   
##

### we can now run 10-fold on test dataset:

# copy in files you need and use test dataset only   
banking<-BM\_test  
  
# the other way is to run 10-fold on the test dataset and take the average of the (10 times) F1 measure   
folds<- createFolds(banking$y, k=10)  
 # create a function to do 10 folds of the data and run the statistics...  
 results <- lapply(folds, function(x) {  
 test<- banking[x,]  
 pred<- predict(sv\_m, test[-17])  
 actual<- test$y  
 # PPV = TP/(TP+FP)  
 # pos<-posPredValue(table(pred, actual))  
 # I actually want: NPV= TN/(TN+FN) for precision of minority class  
 pr<-negPredValue(table(pred, actual))  
 # pr<-precision(table(pred, actual ))  
 # rec<- recall(table(pred, actual))  
 # i actually want specificity for recall of minority class  
 rec<- specificity(table(pred, actual))  
 F1<- 2 \* pr \* rec /(pr + rec)  
 return(F1)  
 })  
 #  
 # print(results)  
 value<-mean(unlist(results))  
 print(value)

## [1] 0.524552

# print the average of the 10 F1 results for test set