Appendix 3 - Customer’s decision to open a term deposit account

## Part 3 - Decision Tree

### Step 1 : Collecting Data

Same as of Part 1

### Step 2 : Exploring, preprocessing and cleaning the data

Primary setup

knitr::opts\_knit$set(root.dir = '/Users/sobil/Documents/MSC/Sem 1/Data Mining & Machine Learning/Project/Bank/')  
remove(list = ls())  
set.seed(1)

loading all the libraries required

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(C50)  
library(RWeka)

rest same as part 1

source("bank\_import\_primaryExplore.R")

## 'data.frame': 41188 obs. of 21 variables:  
## $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ duration : int 261 149 226 151 307 198 139 217 380 50 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
## $ cons.price.idx: num 94 94 94 94 94 ...  
## $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
## $ nr.employed : num 5191 5191 5191 5191 5191 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 41188   
##   
##   
## | no | yes |   
## |-----------|-----------|  
## | 36548 | 4640 |   
## | 0.887 | 0.113 |   
## |-----------|-----------|  
##   
##   
##   
##

### Step 3 - Data transformation & preparation

creating training and testing dataset from exisitng sample

set.seed(1)  
indx <- createDataPartition(bank$y, p = 0.8, list = FALSE)  
  
bank\_train <- bank[indx,]  
bank\_test <- bank[- indx,]  
  
prop.table(table(bank\_train$y))

##   
## Yes No   
## 0.1126521 0.8873479

prop.table(table(bank\_test$y))

##   
## Yes No   
## 0.1126624 0.8873376

### Step 4 : Train the model

bank\_model <- C50::C5.0(bank\_train[-21],bank\_train$y)

### Step 5 : Evaluate the model

bank\_model

##   
## Call:  
## C5.0.default(x = bank\_train[-21], y = bank\_train$y)  
##   
## Classification Tree  
## Number of samples: 32951   
## Number of predictors: 20   
##   
## Tree size: 172   
##   
## Non-standard options: attempt to group attributes

bank\_predict <- predict(object = bank\_model, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict, bank\_test$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 519 262  
## No 409 7047  
##   
## Accuracy : 0.9185   
## 95% CI : (0.9124, 0.9244)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5623   
##   
## Mcnemar's Test P-Value : 1.738e-08   
##   
## Sensitivity : 0.55927   
## Specificity : 0.96415   
## Pos Pred Value : 0.66453   
## Neg Pred Value : 0.94514   
## Prevalence : 0.11266   
## Detection Rate : 0.06301   
## Detection Prevalence : 0.09482   
## Balanced Accuracy : 0.76171   
##   
## 'Positive' Class : Yes   
##

### Step 6 : Improving model performance

#### 1) Method 1 - Boosting trials = 10

bank\_boost10 <- C50::C5.0(bank\_train[-21], bank\_train$y, trials = 10)  
bank\_boost10

##   
## Call:  
## C5.0.default(x = bank\_train[-21], y = bank\_train$y, trials = 10)  
##   
## Classification Tree  
## Number of samples: 32951   
## Number of predictors: 20   
##   
## Number of boosting iterations: 10   
## Average tree size: 167.1   
##   
## Non-standard options: attempt to group attributes

bank\_boost\_predict10 <- predict(object = bank\_boost10, newdata = bank\_test)  
caret::confusionMatrix(bank\_boost\_predict10, bank\_test$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 510 287  
## No 418 7022  
##   
## Accuracy : 0.9144   
## 95% CI : (0.9082, 0.9204)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.065e-16   
##   
## Kappa : 0.5438   
##   
## Mcnemar's Test P-Value : 9.777e-07   
##   
## Sensitivity : 0.54957   
## Specificity : 0.96073   
## Pos Pred Value : 0.63990   
## Neg Pred Value : 0.94382   
## Prevalence : 0.11266   
## Detection Rate : 0.06192   
## Detection Prevalence : 0.09676   
## Balanced Accuracy : 0.75515   
##   
## 'Positive' Class : Yes   
##

#### 2) Method 2 - Boosting trials = 5

bank\_boost5 <- C50::C5.0(bank\_train[-21], bank\_train$y, trials = 5)  
bank\_boost5

##   
## Call:  
## C5.0.default(x = bank\_train[-21], y = bank\_train$y, trials = 5)  
##   
## Classification Tree  
## Number of samples: 32951   
## Number of predictors: 20   
##   
## Number of boosting iterations: 5   
## Average tree size: 129.4   
##   
## Non-standard options: attempt to group attributes

bank\_boost\_predict5 <- predict(object = bank\_boost5, newdata = bank\_test)  
caret::confusionMatrix(bank\_boost\_predict5, bank\_test$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 521 305  
## No 407 7004  
##   
## Accuracy : 0.9136   
## 95% CI : (0.9073, 0.9195)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 3.393e-15   
##   
## Kappa : 0.5459   
##   
## Mcnemar's Test P-Value : 0.0001536   
##   
## Sensitivity : 0.56142   
## Specificity : 0.95827   
## Pos Pred Value : 0.63075   
## Neg Pred Value : 0.94508   
## Prevalence : 0.11266   
## Detection Rate : 0.06325   
## Detection Prevalence : 0.10028   
## Balanced Accuracy : 0.75985   
##   
## 'Positive' Class : Yes   
##

#### 3) Method 3 - Adding Cost matrix

creating cost matrix - “yes” more costlier than others

mtr\_dim <- list(c("no", "yes"), c("no","yes"))  
names(mtr\_dim) <- c("predict","actual")  
mtr\_dim

## $predict  
## [1] "no" "yes"  
##   
## $actual  
## [1] "no" "yes"

err\_cst <- matrix(c(0,1,0,4), nrow = 2, dimnames = mtr\_dim)  
err\_cst

## actual  
## predict no yes  
## no 0 0  
## yes 1 4

Model Costs

bank\_cost <- C50::C5.0(bank\_train[-21], bank\_train$y, costs = err\_cst)  
bank\_cost\_predict <- predict(object = bank\_cost, newdata = bank\_test)  
caret::confusionMatrix(bank\_cost\_predict, bank\_test$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 519 262  
## No 409 7047  
##   
## Accuracy : 0.9185   
## 95% CI : (0.9124, 0.9244)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5623   
##   
## Mcnemar's Test P-Value : 1.738e-08   
##   
## Sensitivity : 0.55927   
## Specificity : 0.96415   
## Pos Pred Value : 0.66453   
## Neg Pred Value : 0.94514   
## Prevalence : 0.11266   
## Detection Rate : 0.06301   
## Detection Prevalence : 0.09482   
## Balanced Accuracy : 0.76171   
##   
## 'Positive' Class : Yes   
##

#### 4) Method 4 - 1R algo model

bank\_1R <- OneR(y ~ ., data = bank\_train)  
bank\_1R

## duration:  
## < 457.5 -> No  
## < 458.5 -> Yes  
## < 525.5 -> No  
## < 526.5 -> Yes  
## < 554.5 -> No  
## < 555.5 -> Yes  
## < 562.5 -> No  
## < 563.5 -> Yes  
## < 582.5 -> No  
## < 583.5 -> Yes  
## < 606.5 -> No  
## < 608.5 -> Yes  
## < 617.5 -> No  
## < 618.5 -> Yes  
## < 622.5 -> No  
## < 623.5 -> Yes  
## < 644.5 -> No  
## < 646.5 -> Yes  
## < 648.5 -> No  
## < 651.5 -> Yes  
## < 653.5 -> No  
## < 655.5 -> Yes  
## < 657.5 -> No  
## < 658.5 -> Yes  
## < 661.5 -> No  
## < 664.5 -> Yes  
## < 666.5 -> No  
## < 667.5 -> Yes  
## < 668.5 -> No  
## < 671.5 -> Yes  
## < 676.5 -> No  
## < 679.5 -> Yes  
## < 699.5 -> No  
## < 701.5 -> Yes  
## < 702.5 -> No  
## < 704.5 -> Yes  
## < 706.5 -> No  
## < 712.5 -> Yes  
## < 714.5 -> No  
## < 716.5 -> Yes  
## < 718.5 -> No  
## < 721.5 -> Yes  
## < 730.5 -> No  
## < 734.5 -> Yes  
## < 738.5 -> No  
## < 741.5 -> Yes  
## < 744.5 -> No  
## < 746.5 -> Yes  
## < 761.5 -> No  
## < 763.5 -> Yes  
## < 766.5 -> No  
## < 768.5 -> Yes  
## < 780.5 -> No  
## < 788.5 -> Yes  
## < 800.5 -> No  
## < 803.5 -> Yes  
## < 807.5 -> No  
## < 809.5 -> Yes  
## < 811.5 -> No  
## < 813.5 -> Yes  
## < 816.5 -> No  
## < 820.5 -> Yes  
## < 827.5 -> No  
## < 832.5 -> Yes  
## < 835.5 -> No  
## < 842.0 -> Yes  
## < 847.5 -> No  
## < 850.5 -> Yes  
## < 853.5 -> No  
## < 883.5 -> Yes  
## < 887.0 -> No  
## < 897.5 -> Yes  
## < 899.5 -> No  
## < 904.5 -> Yes  
## < 907.5 -> No  
## < 914.0 -> Yes  
## < 919.5 -> No  
## < 924.5 -> Yes  
## < 930.5 -> No  
## < 971.5 -> Yes  
## < 977.5 -> No  
## < 985.5 -> Yes  
## < 990.5 -> No  
## < 1009.5 -> Yes  
## < 1014.5 -> No  
## < 1082.5 -> Yes  
## < 1089.5 -> No  
## < 1093.5 -> Yes  
## < 1100.5 -> No  
## < 1155.0 -> Yes  
## < 1168.0 -> No  
## < 1179.0 -> Yes  
## < 1198.0 -> No  
## < 1219.0 -> Yes  
## < 1225.5 -> No  
## < 1270.0 -> Yes  
## < 1292.5 -> No  
## < 1375.0 -> Yes  
## < 1390.5 -> No  
## < 1424.5 -> Yes  
## < 1438.5 -> No  
## < 1546.5 -> Yes  
## < 1592.0 -> No  
## < 1737.0 -> Yes  
## < 1805.5 -> No  
## < 1869.5 -> Yes  
## < 1960.0 -> No  
## < 2781.0 -> Yes  
## < 3570.0 -> No  
## >= 3570.0 -> Yes  
## (29647/32951 instances correct)

bank\_predict\_1R <- predict(object = bank\_1R, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_1R, bank\_test$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 196 164  
## No 732 7145  
##   
## Accuracy : 0.8912   
## 95% CI : (0.8843, 0.8979)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.136   
##   
## Kappa : 0.2576   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.21121   
## Specificity : 0.97756   
## Pos Pred Value : 0.54444   
## Neg Pred Value : 0.90707   
## Prevalence : 0.11266   
## Detection Rate : 0.02380   
## Detection Prevalence : 0.04371   
## Balanced Accuracy : 0.59438   
##   
## 'Positive' Class : Yes   
##

#### 5) Method 5 - RIPPER algo model

bank\_JRip <- JRip(y ~ . , data = bank\_train)  
bank\_JRip

## JRIP rules:  
## ===========  
##   
## (nr.employed <= 5076.2) and (duration >= 190) and (pdays <= 15) => y=Yes (639.0/112.0)  
## (duration >= 854) => y=Yes (1155.0/485.0)  
## (duration >= 381) and (euribor3m <= 1.4) and (cons.price.idx >= 92.963) => y=Yes (363.0/105.0)  
## (nr.employed <= 5076.2) and (duration >= 251) => y=Yes (801.0/354.0)  
## (duration >= 608) and (contact = cellular) and (month = may) => y=Yes (213.0/93.0)  
## (euribor3m >= 4.866) and (duration >= 758) and (age <= 37) => y=Yes (76.0/33.0)  
## (euribor3m <= 3.053) and (duration >= 148) and (nr.employed <= 5076.2) and (pdays <= 15) => y=Yes (140.0/50.0)  
## => y=No (29564.0/1557.0)  
##   
## Number of Rules : 8

bank\_predict\_JRip <- predict(object = bank\_JRip, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_JRip, bank\_test$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 543 300  
## No 385 7009  
##   
## Accuracy : 0.9168   
## 95% CI : (0.9107, 0.9227)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2e-16   
##   
## Kappa : 0.5667   
##   
## Mcnemar's Test P-Value : 0.00133   
##   
## Sensitivity : 0.58513   
## Specificity : 0.95895   
## Pos Pred Value : 0.64413   
## Neg Pred Value : 0.94793   
## Prevalence : 0.11266   
## Detection Rate : 0.06592   
## Detection Prevalence : 0.10234   
## Balanced Accuracy : 0.77204   
##   
## 'Positive' Class : Yes   
##

#kappa = 0.5667, Sensitivity = 0.58513, Specificity = 0.95895 ====> BEST

caret::confusionMatrix(bank\_predict\_JRip, bank\_test$y, mode="prec\_recall")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 543 300  
## No 385 7009  
##   
## Accuracy : 0.9168   
## 95% CI : (0.9107, 0.9227)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2e-16   
##   
## Kappa : 0.5667   
##   
## Mcnemar's Test P-Value : 0.00133   
##   
## Precision : 0.64413   
## Recall : 0.58513   
## F1 : 0.61321   
## Prevalence : 0.11266   
## Detection Rate : 0.06592   
## Detection Prevalence : 0.10234   
## Balanced Accuracy : 0.77204   
##   
## 'Positive' Class : Yes   
##