pl\_cart.R

sumedh

Sat Jun 16 00:03:58 2018

# Set the Working Directory  
setwd("D:/Great Lakes PGPDSE/Great Lakes/10 Supervised Learning - Classification/Supervised Learning Classification/Mini\_Project")  
# Import Dataset  
pl=read.csv("PL\_XSELL.csv")  
sum(is.na(pl)) # There is no NA value in the dataset

## [1] 0

# Converting Categorical value into the factor data type  
pl$GENDER=as.factor(pl$GENDER) # Converting into factor varaiable  
pl$OCCUPATION=as.factor(pl$OCCUPATION)  
pl$AGE\_BKT=as.factor(pl$AGE\_BKT)  
pl$ACC\_TYPE=as.factor(pl$ACC\_TYPE)  
pl=pl[ , -c(1,3,11,13,14,16,17,18,19,20,22,23,24,25,26,29,32,34,35,36,37,38,40)]  
dim(pl)

## [1] 20000 17

View(pl)  
  
summary(pl) # There is no missing value

## TARGET GENDER BALANCE OCCUPATION AGE\_BKT   
## Min. :0.0000 F: 5433 Min. : 0 PROF :5417 <25 :1753   
## 1st Qu.:0.0000 M:14376 1st Qu.: 64754 SAL :5855 >50 :3035   
## Median :0.0000 O: 191 Median : 231676 SELF-EMP:3568 26-30:3434   
## Mean :0.1256 Mean : 511362 SENP :5160 31-35:3404   
## 3rd Qu.:0.0000 3rd Qu.: 653877 36-40:2814   
## Max. :1.0000 Max. :8360431 41-45:3067   
## 46-50:2493   
## SCR HOLDING\_PERIOD ACC\_TYPE LEN\_OF\_RLTN\_IN\_MNTH  
## Min. :100.0 Min. : 1.00 CA: 4241 Min. : 29.0   
## 1st Qu.:227.0 1st Qu.: 7.00 SA:15759 1st Qu.: 79.0   
## Median :364.0 Median :15.00 Median :125.0   
## Mean :440.2 Mean :14.96 Mean :125.2   
## 3rd Qu.:644.0 3rd Qu.:22.00 3rd Qu.:172.0   
## Max. :999.0 Max. :31.00 Max. :221.0   
##   
## TOT\_NO\_OF\_L\_TXNS FLG\_HAS\_CC AMT\_L\_DR FLG\_HAS\_ANY\_CHGS  
## Min. : 0.00 Min. :0.0000 Min. : 0 Min. :0.0000   
## 1st Qu.: 9.00 1st Qu.:0.0000 1st Qu.: 237936 1st Qu.:0.0000   
## Median : 14.00 Median :0.0000 Median : 695115 Median :0.0000   
## Mean : 18.98 Mean :0.3054 Mean : 773717 Mean :0.1106   
## 3rd Qu.: 21.00 3rd Qu.:1.0000 3rd Qu.:1078927 3rd Qu.:0.0000   
## Max. :149.00 Max. :1.0000 Max. :6514921 Max. :1.0000   
##   
## AMT\_MIN\_BAL\_NMC\_CHGS NO\_OF\_IW\_CHQ\_BNC\_TXNS AVG\_AMT\_PER\_ATM\_TXN  
## Min. : 0.000 Min. :0.00000 Min. : 0   
## 1st Qu.: 0.000 1st Qu.:0.00000 1st Qu.: 0   
## Median : 0.000 Median :0.00000 Median : 6000   
## Mean : 1.292 Mean :0.04275 Mean : 7409   
## 3rd Qu.: 0.000 3rd Qu.:0.00000 3rd Qu.:13500   
## Max. :170.000 Max. :2.00000 Max. :25000   
##   
## FLG\_HAS\_OLD\_LOAN  
## Min. :0.0000   
## 1st Qu.:0.0000   
## Median :0.0000   
## Mean :0.4929   
## 3rd Qu.:1.0000   
## Max. :1.0000   
##

# Splitting the dataset into the Training set and Test set  
# install.packages('caTools')  
  
library(caTools)  
set.seed(123)  
split = sample.split(pl$TARGET, SplitRatio = 0.7)# Divided the data into 70:30 ratio  
dev\_sample = subset(pl, split == TRUE)  
hold\_sample = subset(pl, split == FALSE)  
  
nrow(dev\_sample) # 14000 rows

## [1] 14000

nrow(hold\_sample) # 6000 rows

## [1] 6000

# Feature Scaling  
#dev\_sample[,c(3,6,7,9)] = scale(dev\_sample[,c(3,6,7,9)])  
#hold\_sample[,c(3,6,7,9)] = scale(hold\_sample[,c(3,6,7,9)])  
  
  
## installing rpart package for CART  
## install.packages("rpart")  
## install.packages("rpart.plot")  
  
## loading the library  
library(rpart)  
library(rpart.plot)  
# For Development sample finding the target rate  
table(dev\_sample$TARGET) # 0's 12242 1's 1758

##   
## 0 1   
## 12242 1758

## Devlopment sample Target Rate   
sum(dev\_sample$TARGET)/14000 # 12.55% Target rate on development sample

## [1] 0.1255714

# For Hold sample finding the target rate  
table(hold\_sample$TARGET) # 0's 5246 1's 754

##   
## 0 1   
## 5246 754

## Hold sample Target Rate   
sum(hold\_sample$TARGET)/6000 # 12.56% Target rate on hold sample

## [1] 0.1256667

## setting the control paramter inputs for rpart  
r.ctrl = rpart.control(minsplit=100, minbucket = 10, cp = 0, xval = 10)  
  
# Fitting Decision Tree Classification to the Development Dataset  
# install.packages('rpart')  
  
m1 <- rpart(formula = dev\_sample$TARGET ~ .,   
 data = dev\_sample[,-1], method = "class",   
 control = r.ctrl)  
m1

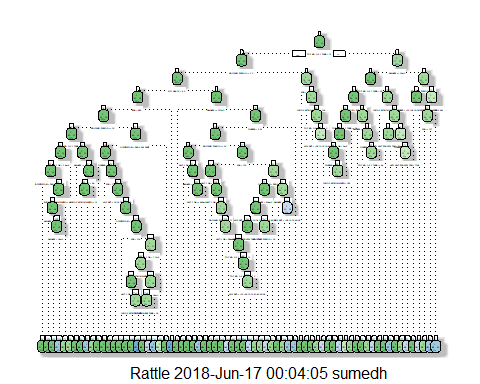
## n= 14000   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 14000 1758 0 (0.87442857 0.12557143)   
## 2) TOT\_NO\_OF\_L\_TXNS< 32.5 12135 1335 0 (0.88998764 0.11001236)   
## 4) HOLDING\_PERIOD>=2.5 11213 1116 0 (0.90047267 0.09952733)   
## 8) FLG\_HAS\_CC< 0.5 7768 615 0 (0.92082904 0.07917096)   
## 16) SCR< 995.5 7756 605 0 (0.92199587 0.07800413)   
## 32) HOLDING\_PERIOD>=15.5 4249 227 0 (0.94657566 0.05342434)   
## 64) TOT\_NO\_OF\_L\_TXNS< 18.5 3429 136 0 (0.96033829 0.03966171)   
## 128) AMT\_L\_DR< 1908034 3419 132 0 (0.96139222 0.03860778)   
## 256) OCCUPATION=PROF,SAL,SENP 2756 81 0 (0.97060958 0.02939042) \*  
## 257) OCCUPATION=SELF-EMP 663 51 0 (0.92307692 0.07692308)   
## 514) LEN\_OF\_RLTN\_IN\_MNTH>=43.5 626 38 0 (0.93929712 0.06070288)   
## 1028) BALANCE>=42525.39 497 15 0 (0.96981891 0.03018109) \*  
## 1029) BALANCE< 42525.39 129 23 0 (0.82170543 0.17829457)   
## 2058) BALANCE< 39065.02 110 8 0 (0.92727273 0.07272727) \*  
## 2059) BALANCE>=39065.02 19 4 1 (0.21052632 0.78947368) \*  
## 515) LEN\_OF\_RLTN\_IN\_MNTH< 43.5 37 13 0 (0.64864865 0.35135135) \*  
## 129) AMT\_L\_DR>=1908034 10 4 0 (0.60000000 0.40000000) \*  
## 65) TOT\_NO\_OF\_L\_TXNS>=18.5 820 91 0 (0.88902439 0.11097561)   
## 130) BALANCE>=81289.05 585 44 0 (0.92478632 0.07521368) \*  
## 131) BALANCE< 81289.05 235 47 0 (0.80000000 0.20000000)   
## 262) ACC\_TYPE=SA 154 16 0 (0.89610390 0.10389610)   
## 524) LEN\_OF\_RLTN\_IN\_MNTH>=32.5 143 9 0 (0.93706294 0.06293706) \*  
## 525) LEN\_OF\_RLTN\_IN\_MNTH< 32.5 11 4 1 (0.36363636 0.63636364) \*  
## 263) ACC\_TYPE=CA 81 31 0 (0.61728395 0.38271605) \*  
## 33) HOLDING\_PERIOD< 15.5 3507 378 0 (0.89221557 0.10778443)   
## 66) OCCUPATION=PROF,SAL,SENP 2902 272 0 (0.90627154 0.09372846)   
## 132) GENDER=F,O 758 38 0 (0.94986807 0.05013193) \*  
## 133) GENDER=M 2144 234 0 (0.89085821 0.10914179)   
## 266) SCR< 177.5 323 11 0 (0.96594427 0.03405573) \*  
## 267) SCR>=177.5 1821 223 0 (0.87753981 0.12246019)   
## 534) AGE\_BKT=26-30,31-35,36-40 888 79 0 (0.91103604 0.08896396) \*  
## 535) AGE\_BKT=<25,>50,41-45,46-50 933 144 0 (0.84565916 0.15434084)   
## 1070) OCCUPATION=SAL 370 33 0 (0.91081081 0.08918919) \*  
## 1071) OCCUPATION=PROF,SENP 563 111 0 (0.80284192 0.19715808)   
## 2142) SCR< 210.5 40 0 0 (1.00000000 0.00000000) \*  
## 2143) SCR>=210.5 523 111 0 (0.78776291 0.21223709)   
## 4286) AMT\_L\_DR< 1726047 494 99 0 (0.79959514 0.20040486)   
## 8572) LEN\_OF\_RLTN\_IN\_MNTH>=76.5 343 56 0 (0.83673469 0.16326531)   
## 17144) AMT\_L\_DR>=736606.5 171 17 0 (0.90058480 0.09941520) \*  
## 17145) AMT\_L\_DR< 736606.5 172 39 0 (0.77325581 0.22674419)   
## 34290) LEN\_OF\_RLTN\_IN\_MNTH< 207.5 162 31 0 (0.80864198 0.19135802) \*  
## 34291) LEN\_OF\_RLTN\_IN\_MNTH>=207.5 10 2 1 (0.20000000 0.80000000) \*  
## 8573) LEN\_OF\_RLTN\_IN\_MNTH< 76.5 151 43 0 (0.71523179 0.28476821)   
## 17146) BALANCE>=39424.26 127 28 0 (0.77952756 0.22047244)   
## 34292) TOT\_NO\_OF\_L\_TXNS< 19.5 96 12 0 (0.87500000 0.12500000) \*  
## 34293) TOT\_NO\_OF\_L\_TXNS>=19.5 31 15 1 (0.48387097 0.51612903) \*  
## 17147) BALANCE< 39424.26 24 9 1 (0.37500000 0.62500000) \*  
## 4287) AMT\_L\_DR>=1726047 29 12 0 (0.58620690 0.41379310) \*  
## 67) OCCUPATION=SELF-EMP 605 106 0 (0.82479339 0.17520661) \*  
## 17) SCR>=995.5 12 2 1 (0.16666667 0.83333333) \*  
## 9) FLG\_HAS\_CC>=0.5 3445 501 0 (0.85457184 0.14542816)   
## 18) BALANCE>=537034.8 1023 87 0 (0.91495601 0.08504399) \*  
## 19) BALANCE< 537034.8 2422 414 0 (0.82906689 0.17093311)   
## 38) GENDER=F,M 2409 405 0 (0.83188045 0.16811955)   
## 76) HOLDING\_PERIOD>=23.5 549 52 0 (0.90528233 0.09471767)   
## 152) AMT\_L\_DR< 432177.5 250 8 0 (0.96800000 0.03200000) \*  
## 153) AMT\_L\_DR>=432177.5 299 44 0 (0.85284281 0.14715719)   
## 306) BALANCE>=293239.8 70 0 0 (1.00000000 0.00000000) \*  
## 307) BALANCE< 293239.8 229 44 0 (0.80786026 0.19213974)   
## 614) AMT\_L\_DR>=466476.5 218 37 0 (0.83027523 0.16972477) \*  
## 615) AMT\_L\_DR< 466476.5 11 4 1 (0.36363636 0.63636364) \*  
## 77) HOLDING\_PERIOD< 23.5 1860 353 0 (0.81021505 0.18978495)   
## 154) SCR< 326.5 725 94 0 (0.87034483 0.12965517)   
## 308) AMT\_L\_DR< 373789.5 228 13 0 (0.94298246 0.05701754) \*  
## 309) AMT\_L\_DR>=373789.5 497 81 0 (0.83702213 0.16297787)   
## 618) AMT\_L\_DR>=925700 240 21 0 (0.91250000 0.08750000) \*  
## 619) AMT\_L\_DR< 925700 257 60 0 (0.76653696 0.23346304)   
## 1238) HOLDING\_PERIOD>=11.5 122 15 0 (0.87704918 0.12295082) \*  
## 1239) HOLDING\_PERIOD< 11.5 135 45 0 (0.66666667 0.33333333)   
## 2478) AMT\_L\_DR< 690316.5 65 9 0 (0.86153846 0.13846154) \*  
## 2479) AMT\_L\_DR>=690316.5 70 34 1 (0.48571429 0.51428571) \*  
## 155) SCR>=326.5 1135 259 0 (0.77180617 0.22819383)   
## 310) OCCUPATION=PROF,SAL,SENP 916 180 0 (0.80349345 0.19650655)   
## 620) BALANCE< 433516.1 823 143 0 (0.82624544 0.17375456)   
## 1240) AGE\_BKT=<25,>50,26-30,31-35,41-45,46-50 694 104 0 (0.85014409 0.14985591)   
## 2480) AVG\_AMT\_PER\_ATM\_TXN< 16650 609 81 0 (0.86699507 0.13300493)   
## 4960) TOT\_NO\_OF\_L\_TXNS>=12.5 298 26 0 (0.91275168 0.08724832) \*  
## 4961) TOT\_NO\_OF\_L\_TXNS< 12.5 311 55 0 (0.82315113 0.17684887)   
## 9922) TOT\_NO\_OF\_L\_TXNS< 7.5 136 11 0 (0.91911765 0.08088235) \*  
## 9923) TOT\_NO\_OF\_L\_TXNS>=7.5 175 44 0 (0.74857143 0.25142857)   
## 19846) AGE\_BKT=<25,>50,31-35,41-45,46-50 140 26 0 (0.81428571 0.18571429) \*  
## 19847) AGE\_BKT=26-30 35 17 1 (0.48571429 0.51428571) \*  
## 2481) AVG\_AMT\_PER\_ATM\_TXN>=16650 85 23 0 (0.72941176 0.27058824) \*  
## 1241) AGE\_BKT=36-40 129 39 0 (0.69767442 0.30232558)   
## 2482) HOLDING\_PERIOD< 19.5 93 20 0 (0.78494624 0.21505376) \*  
## 2483) HOLDING\_PERIOD>=19.5 36 17 1 (0.47222222 0.52777778) \*  
## 621) BALANCE>=433516.1 93 37 0 (0.60215054 0.39784946) \*  
## 311) OCCUPATION=SELF-EMP 219 79 0 (0.63926941 0.36073059)   
## 622) BALANCE>=78359.37 116 21 0 (0.81896552 0.18103448) \*  
## 623) BALANCE< 78359.37 103 45 1 (0.43689320 0.56310680)   
## 1246) AGE\_BKT=>50,31-35,46-50 40 12 0 (0.70000000 0.30000000) \*  
## 1247) AGE\_BKT=<25,26-30,36-40,41-45 63 17 1 (0.26984127 0.73015873) \*  
## 39) GENDER=O 13 4 1 (0.30769231 0.69230769) \*  
## 5) HOLDING\_PERIOD< 2.5 922 219 0 (0.76247289 0.23752711)   
## 10) SCR< 456.5 521 91 0 (0.82533589 0.17466411) \*  
## 11) SCR>=456.5 401 128 0 (0.68079800 0.31920200)   
## 22) LEN\_OF\_RLTN\_IN\_MNTH>=153.5 141 27 0 (0.80851064 0.19148936) \*  
## 23) LEN\_OF\_RLTN\_IN\_MNTH< 153.5 260 101 0 (0.61153846 0.38846154)   
## 46) TOT\_NO\_OF\_L\_TXNS< 6 35 2 0 (0.94285714 0.05714286) \*  
## 47) TOT\_NO\_OF\_L\_TXNS>=6 225 99 0 (0.56000000 0.44000000)   
## 94) AGE\_BKT=>50,31-35,36-40,46-50 131 43 0 (0.67175573 0.32824427) \*  
## 95) AGE\_BKT=<25,26-30,41-45 94 38 1 (0.40425532 0.59574468) \*  
## 3) TOT\_NO\_OF\_L\_TXNS>=32.5 1865 423 0 (0.77319035 0.22680965)   
## 6) BALANCE>=35866.76 1482 282 0 (0.80971660 0.19028340)   
## 12) TOT\_NO\_OF\_L\_TXNS< 66.5 1203 194 0 (0.83873649 0.16126351)   
## 24) OCCUPATION=PROF,SAL 776 93 0 (0.88015464 0.11984536)   
## 48) TOT\_NO\_OF\_L\_TXNS>=33.5 739 80 0 (0.89174560 0.10825440)   
## 96) AMT\_L\_DR< 5039950 729 75 0 (0.89711934 0.10288066)   
## 192) GENDER=M,O 624 53 0 (0.91506410 0.08493590) \*  
## 193) GENDER=F 105 22 0 (0.79047619 0.20952381)   
## 386) LEN\_OF\_RLTN\_IN\_MNTH< 196 88 13 0 (0.85227273 0.14772727) \*  
## 387) LEN\_OF\_RLTN\_IN\_MNTH>=196 17 8 1 (0.47058824 0.52941176) \*  
## 97) AMT\_L\_DR>=5039950 10 5 0 (0.50000000 0.50000000) \*  
## 49) TOT\_NO\_OF\_L\_TXNS< 33.5 37 13 0 (0.64864865 0.35135135) \*  
## 25) OCCUPATION=SELF-EMP,SENP 427 101 0 (0.76346604 0.23653396)   
## 50) BALANCE>=372034.7 187 25 0 (0.86631016 0.13368984) \*  
## 51) BALANCE< 372034.7 240 76 0 (0.68333333 0.31666667)   
## 102) GENDER=F,O 71 9 0 (0.87323944 0.12676056) \*  
## 103) GENDER=M 169 67 0 (0.60355030 0.39644970)   
## 206) TOT\_NO\_OF\_L\_TXNS>=44.5 95 26 0 (0.72631579 0.27368421) \*  
## 207) TOT\_NO\_OF\_L\_TXNS< 44.5 74 33 1 (0.44594595 0.55405405) \*  
## 13) TOT\_NO\_OF\_L\_TXNS>=66.5 279 88 0 (0.68458781 0.31541219)   
## 26) LEN\_OF\_RLTN\_IN\_MNTH>=200.5 25 0 0 (1.00000000 0.00000000) \*  
## 27) LEN\_OF\_RLTN\_IN\_MNTH< 200.5 254 88 0 (0.65354331 0.34645669)   
## 54) SCR< 307.5 100 22 0 (0.78000000 0.22000000)   
## 108) AVG\_AMT\_PER\_ATM\_TXN< 21966.67 88 13 0 (0.85227273 0.14772727) \*  
## 109) AVG\_AMT\_PER\_ATM\_TXN>=21966.67 12 3 1 (0.25000000 0.75000000) \*  
## 55) SCR>=307.5 154 66 0 (0.57142857 0.42857143)   
## 110) GENDER=O 20 1 0 (0.95000000 0.05000000) \*  
## 111) GENDER=F,M 134 65 0 (0.51492537 0.48507463)   
## 222) AVG\_AMT\_PER\_ATM\_TXN< 7350 48 14 0 (0.70833333 0.29166667) \*  
## 223) AVG\_AMT\_PER\_ATM\_TXN>=7350 86 35 1 (0.40697674 0.59302326) \*  
## 7) BALANCE< 35866.76 383 141 0 (0.63185379 0.36814621)   
## 14) HOLDING\_PERIOD>=14.5 135 24 0 (0.82222222 0.17777778)   
## 28) AGE\_BKT=<25,>50,26-30,31-35,36-40,46-50 106 8 0 (0.92452830 0.07547170) \*  
## 29) AGE\_BKT=41-45 29 13 1 (0.44827586 0.55172414) \*  
## 15) HOLDING\_PERIOD< 14.5 248 117 0 (0.52822581 0.47177419)   
## 30) OCCUPATION=PROF,SAL 163 59 0 (0.63803681 0.36196319)   
## 60) SCR>=237 115 30 0 (0.73913043 0.26086957) \*  
## 61) SCR< 237 48 19 1 (0.39583333 0.60416667) \*  
## 31) OCCUPATION=SELF-EMP,SENP 85 27 1 (0.31764706 0.68235294) \*

## install.packages("rattle")  
## install.packages("RcolorBrewer")  
library(rattle)

## Rattle: A free graphical interface for data science with R.  
## Version 5.1.0 Copyright (c) 2006-2017 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

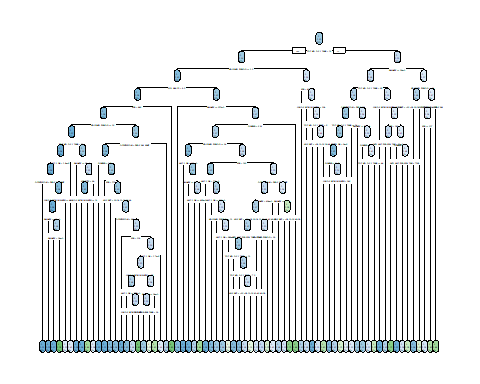
library(RColorBrewer)  
fancyRpartPlot(m1)

## Warning: labs do not fit even at cex 0.15, there may be some overplotting



rpart.plot(m1)

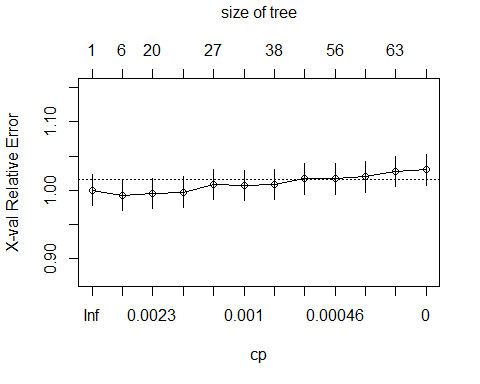
## Warning: labs do not fit even at cex 0.15, there may be some overplotting



printcp(m1)

##   
## Classification tree:  
## rpart(formula = dev\_sample$TARGET ~ ., data = dev\_sample[, -1],   
## method = "class", control = r.ctrl)  
##   
## Variables actually used in tree construction:  
## [1] ACC\_TYPE AGE\_BKT AMT\_L\_DR   
## [4] AVG\_AMT\_PER\_ATM\_TXN BALANCE FLG\_HAS\_CC   
## [7] GENDER HOLDING\_PERIOD LEN\_OF\_RLTN\_IN\_MNTH  
## [10] OCCUPATION SCR TOT\_NO\_OF\_L\_TXNS   
##   
## Root node error: 1758/14000 = 0.12557  
##   
## n= 14000   
##   
## CP nsplit rel error xerror xstd  
## 1 0.00440842 0 1.00000 1.00000 0.022302  
## 2 0.00243784 5 0.97668 0.99261 0.022232  
## 3 0.00208570 19 0.94255 0.99545 0.022259  
## 4 0.00170648 25 0.93003 0.99716 0.022275  
## 5 0.00113766 26 0.92833 1.00796 0.022378  
## 6 0.00089387 30 0.92378 1.00683 0.022367  
## 7 0.00062054 37 0.91752 1.00796 0.022378  
## 8 0.00056883 48 0.91069 1.01650 0.022459  
## 9 0.00037922 55 0.90671 1.01650 0.022459  
## 10 0.00028441 58 0.90557 1.01934 0.022486  
## 11 0.00014221 62 0.90444 1.02730 0.022561  
## 12 0.00000000 70 0.90330 1.02958 0.022582

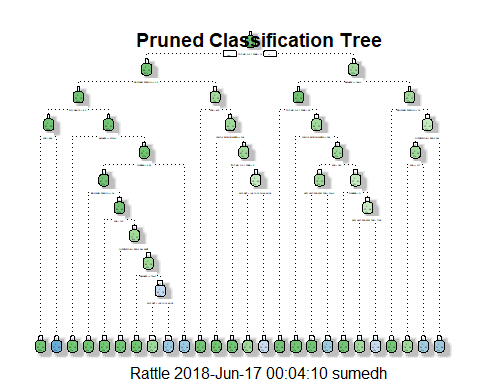
plotcp(m1)



# Pruning criteria based on cp table   
  
ptree<-prune(m1,cp=0.002,"CP")  
printcp(ptree)

##   
## Classification tree:  
## rpart(formula = dev\_sample$TARGET ~ ., data = dev\_sample[, -1],   
## method = "class", control = r.ctrl)  
##   
## Variables actually used in tree construction:  
## [1] AGE\_BKT AVG\_AMT\_PER\_ATM\_TXN BALANCE   
## [4] FLG\_HAS\_CC GENDER HOLDING\_PERIOD   
## [7] LEN\_OF\_RLTN\_IN\_MNTH OCCUPATION SCR   
## [10] TOT\_NO\_OF\_L\_TXNS   
##   
## Root node error: 1758/14000 = 0.12557  
##   
## n= 14000   
##   
## CP nsplit rel error xerror xstd  
## 1 0.0044084 0 1.00000 1.00000 0.022302  
## 2 0.0024378 5 0.97668 0.99261 0.022232  
## 3 0.0020857 19 0.94255 0.99545 0.022259  
## 4 0.0020000 25 0.93003 0.99716 0.022275

fancyRpartPlot(ptree,uniform= TRUE, main="Pruned Classification Tree")



## Let's use rattle to see various model evaluation measures  
##rattle()  
  
#View(dev\_sample)  
head(dev\_sample,2)

## TARGET GENDER BALANCE OCCUPATION AGE\_BKT SCR HOLDING\_PERIOD ACC\_TYPE  
## 1 0 M 3383.75 SELF-EMP 26-30 776 30 SA  
## 3 0 M 18216.88 SELF-EMP 36-40 603 2 SA  
## LEN\_OF\_RLTN\_IN\_MNTH TOT\_NO\_OF\_L\_TXNS FLG\_HAS\_CC AMT\_L\_DR  
## 1 146 10 0 986657  
## 3 61 15 0 1619210  
## FLG\_HAS\_ANY\_CHGS AMT\_MIN\_BAL\_NMC\_CHGS NO\_OF\_IW\_CHQ\_BNC\_TXNS  
## 1 0 0 0  
## 3 1 0 0  
## AVG\_AMT\_PER\_ATM\_TXN FLG\_HAS\_OLD\_LOAN  
## 1 13100 1  
## 3 11200 1

## deciling code  
decile <- function(x){  
 deciles <- vector(length=10)  
 for (i in seq(0.1,1,.1)){  
 deciles[i\*10] <- quantile(x, i, na.rm=T)  
 }  
 return (  
 ifelse(x<deciles[1], 1,  
 ifelse(x<deciles[2], 2,  
 ifelse(x<deciles[3], 3,  
 ifelse(x<deciles[4], 4,  
 ifelse(x<deciles[5], 5,  
 ifelse(x<deciles[6], 6,  
 ifelse(x<deciles[7], 7,  
 ifelse(x<deciles[8], 8,  
 ifelse(x<deciles[9], 9, 10  
 ))))))))))  
}  
  
dev\_sample$predict.class <- predict(ptree, dev\_sample, type="class")  
dev\_sample$predict.score <- predict(ptree, dev\_sample)  
class(dev\_sample$predict.score)

## [1] "matrix"

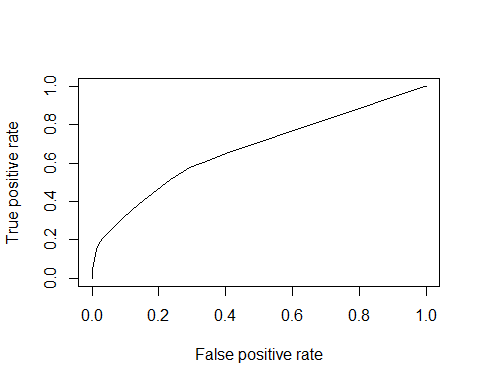
## deciling  
dev\_sample$deciles <- decile(dev\_sample$predict.score[,2])  
#View(dev\_sample)  
  
## Ranking code  
##install.packages("data.table")  
library(data.table)  
tmp\_DT = data.table(dev\_sample)  
  
rank <- tmp\_DT[, list(  
 cnt = length(TARGET),   
 cnt\_resp = sum(TARGET),   
 cnt\_non\_resp = sum(TARGET == 0)) ,   
 by=deciles][order(-deciles)]  
  
rank$rrate <- round(rank$cnt\_resp \* 100 / rank$cnt,2);  
rank$cum\_resp <- cumsum(rank$cnt\_resp)  
rank$cum\_non\_resp <- cumsum(rank$cnt\_non\_resp)  
rank$cum\_perct\_resp <- round(rank$cum\_resp \* 100 / sum(rank$cnt\_resp),2);  
rank$cum\_perct\_non\_resp <- round(rank$cum\_non\_resp \* 100 / sum(rank$cnt\_non\_resp),2);  
rank$ks <- abs(rank$cum\_perct\_resp - rank$cum\_perct\_non\_resp);  
  
View(rank)  
  
##install.packages("ROCR")  
## AUC for Development dataset  
library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

pred <- prediction(dev\_sample$predict.score[,2], dev\_sample$TARGET)  
perf <- performance(pred, "tpr", "fpr")  
plot(perf)



KS <- max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])  
auc <- performance(pred,"auc");   
auc <- as.numeric(auc@y.values)  
  
##install.packages("ineq")  
library(ineq)  
gini = ineq(dev\_sample$predict.score[,2], type="Gini")  
  
gini2=2\*auc-1  
with(dev\_sample, table(TARGET, predict.class))

## predict.class  
## TARGET 0 1  
## 0 12097 145  
## 1 1490 268

auc

## [1] 0.6723848

KS

## [1] 0.2825678

gini

## [1] 0.3014763

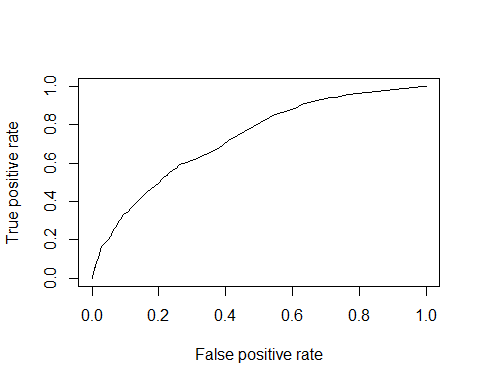
View(rank)  
## Syntax to get the node path  
tree.path <- path.rpart(ptree, node = c(16, 18))

##   
## node number: 16   
## root  
## TOT\_NO\_OF\_L\_TXNS< 32.5  
## HOLDING\_PERIOD>=2.5  
## FLG\_HAS\_CC< 0.5  
## SCR< 995.5  
##   
## node number: 18   
## root  
## TOT\_NO\_OF\_L\_TXNS< 32.5  
## HOLDING\_PERIOD>=2.5  
## FLG\_HAS\_CC>=0.5  
## BALANCE>=5.37e+05

nrow(hold\_sample)

## [1] 6000

## Scoring Holdout sample  
hold\_sample$predict.class <- predict(m1, hold\_sample, type="class")  
hold\_sample$predict.score <- predict(m1, hold\_sample)  
  
  
hold\_sample$deciles <- decile(hold\_sample$predict.score[,2])  
#View(hold\_sample)  
  
## Ranking code  
##install.packages("data.table")  
## Based upon the holding dataset.  
library(data.table)  
tmp\_DT = data.table(hold\_sample)  
h\_rank <- tmp\_DT[, list(  
 cnt = length(TARGET),   
 cnt\_resp = sum(TARGET),   
 cnt\_non\_resp = sum(TARGET == 0)) ,   
 by=deciles][order(-deciles)]  
#  
h\_rank$rrate <- round(h\_rank$cnt\_resp \* 100 / h\_rank$cnt,2);  
h\_rank$cum\_resp <- cumsum(h\_rank$cnt\_resp)  
h\_rank$cum\_non\_resp <- cumsum(h\_rank$cnt\_non\_resp)  
h\_rank$cum\_perct\_resp <- round(h\_rank$cum\_resp \* 100 / sum(h\_rank$cnt\_resp),2);  
h\_rank$cum\_perct\_non\_resp <- round(h\_rank$cum\_non\_resp \* 100 / sum(h\_rank$cnt\_non\_resp),2);  
h\_rank$ks <- abs(h\_rank$cum\_perct\_resp - h\_rank$cum\_perct\_non\_resp);  
  
View(h\_rank)  
  
## AUC for Development dataset  
library(ROCR)  
pred <- prediction(hold\_sample$predict.score[,2], hold\_sample$TARGET)  
perf <- performance(pred, "tpr", "fpr")  
plot(perf)



KS <- max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])  
auc <- performance(pred,"auc");   
auc <- as.numeric(auc@y.values)  
gini3=2\*auc-1  
  
auc

## [1] 0.7286962

KS

## [1] 0.331115

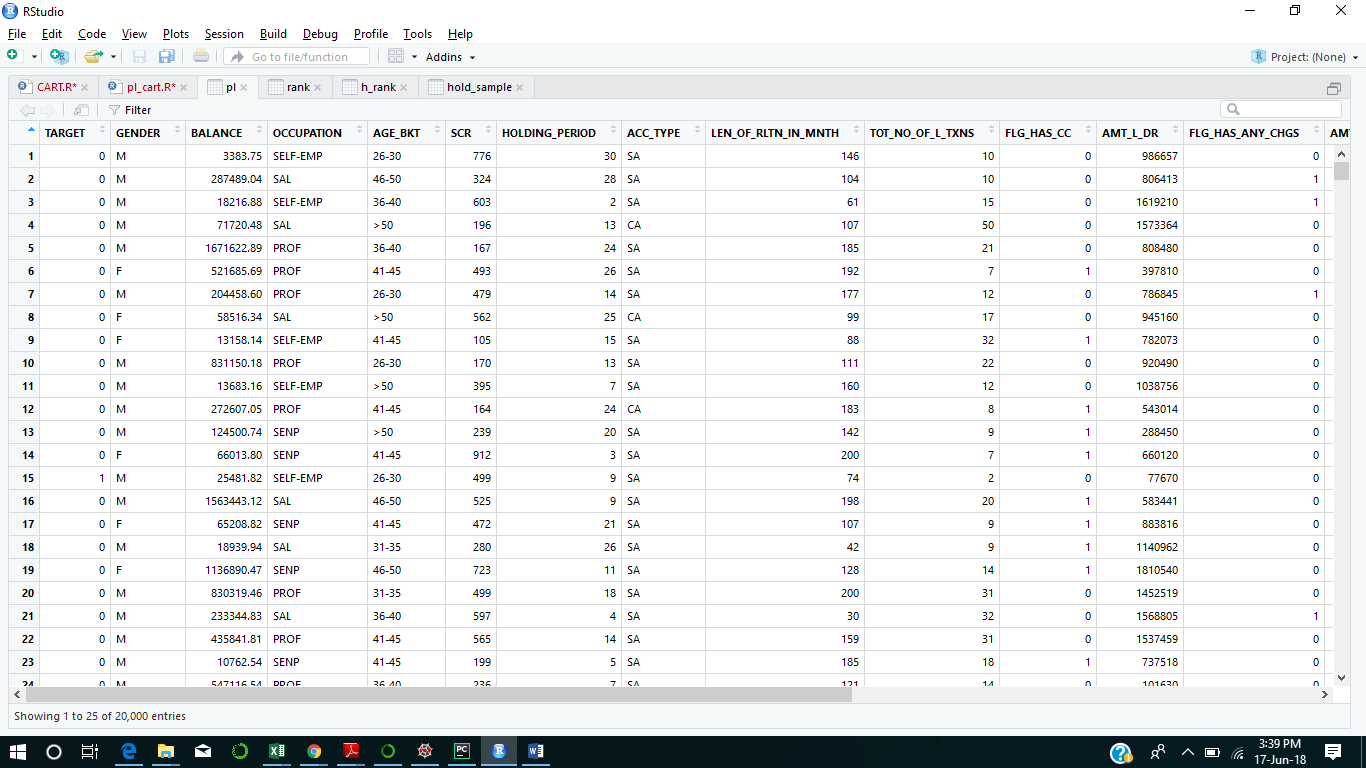
gini

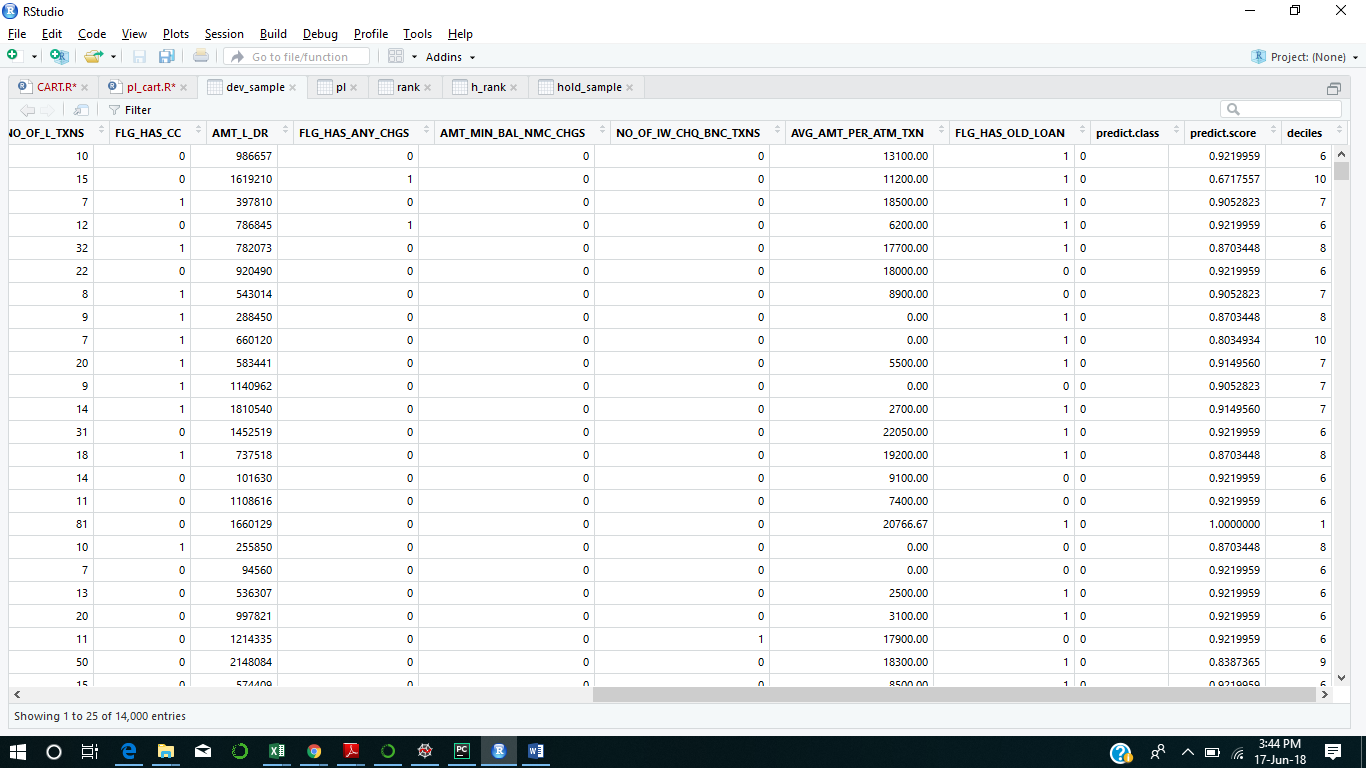
## [1] 0.3014763

with(hold\_sample, table(TARGET, predict.class))

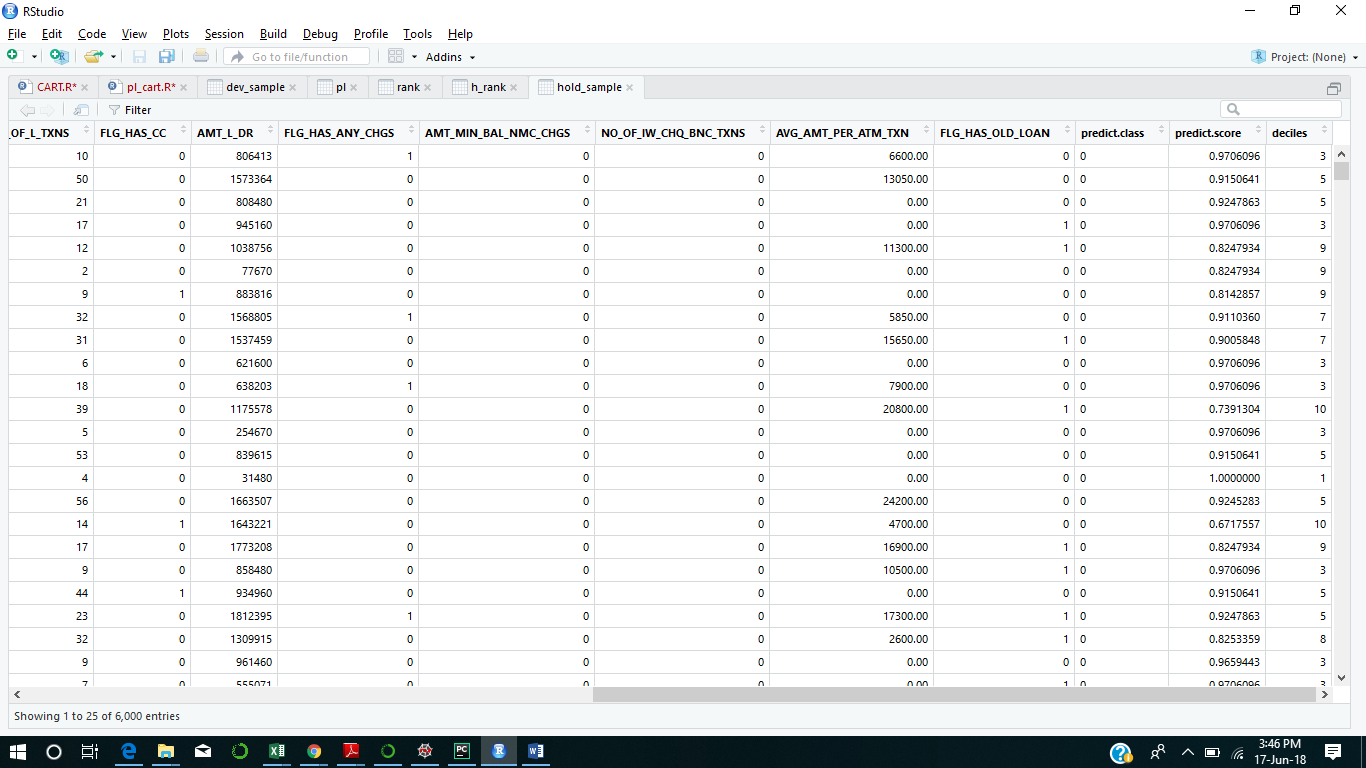
## predict.class  
## TARGET 0 1  
## 0 5083 163  
## 1 624 130

This is my original dataframe named.

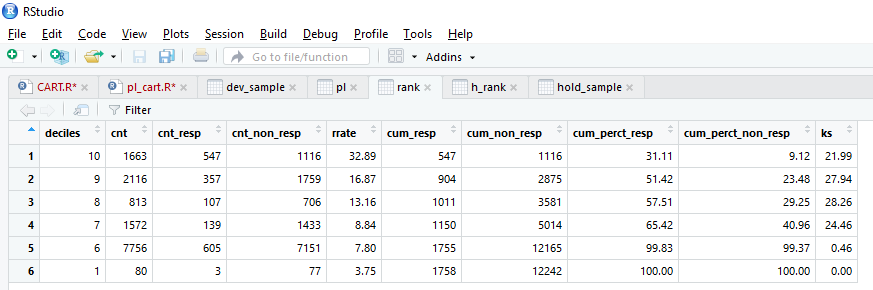


My development sample data frame. 

My hold sample data frame.



Data frame of the Rank (Development sample Rank )



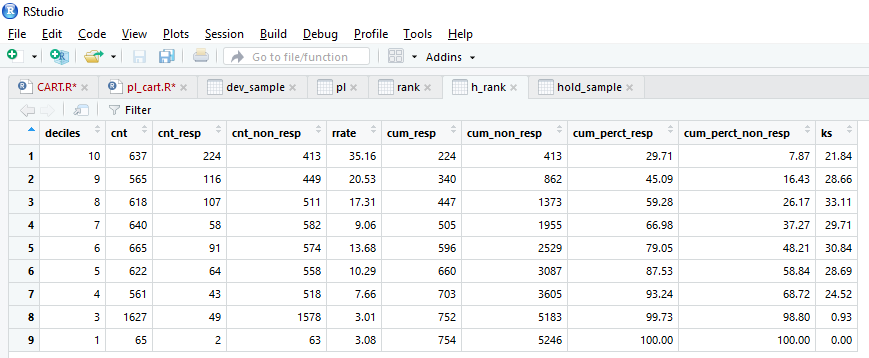
If We target top 3 decile i.e. deciles 10,9 and 8 and my maximum KS statistics is coming up at 8th decile i.e. 28.26.

Chance of getting the maximum from customer for selling Personal Loan is

((21.99\*547)+(27.94\*357)+(28.26\*813))/(547+357+107) = 44.48

It means that chance of buying the Personal Loan 44.48 times if we focus on top 3 decile.

Data frame of the h\_Rank ( Rank based on hold out data or test data)



If We target top 3 decile i.e. deciles 10,9 and 8 and my maximum KS statistics is coming up at 8th decile i.e. 33.11.

Chance of getting the maximum from customer for selling Personal Loan is

=((21.84\*224)+(28.66\*116)+(33.11\*618))/(224+116+107) =64.15

In test data my value has been increased to 64.15 ( test data) from 44.48 ( train data ).

It means that chance of buying the Personal Loan is 64.15 times if we focus on top 3 decile.

AUC ( Area under the Curve ) comparison

For my train dataset AUC is coming up 67.23% .

For my test dataset AUC is coming up 72.86%.

There is a increase of 7.7% in AUC it means there is not much variation between train and test dataset.