HW\_5\_Tatum.R

Jennifer

Thu Apr 27 14:16:24 2017

library(caret)

## Warning: package 'caret' was built under R version 3.3.3

## Loading required package: lattice

## Loading required package: ggplot2

library (party)

## Warning: package 'party' was built under R version 3.3.3

## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 3.3.3

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: sandwich

library(mlbench)

## Warning: package 'mlbench' was built under R version 3.3.3

setwd("C:/Users/Jennifer/Documents/ADM/HW 5")  
bankfull<-read.csv("bankfull.csv")  
bankfull2<-bankfull[,c(17,1:11, 13:16)]  
   
summary(bankfull2)

## y..category age..number job..category marital..category  
## no :39922 Min. :18.00 blue-collar:9732 divorced: 5207   
## yes: 5289 1st Qu.:33.00 management :9458 married :27214   
## Median :39.00 technician :7597 single :12790   
## Mean :40.94 admin. :5171   
## 3rd Qu.:48.00 services :4154   
## Max. :95.00 retired :2264   
## (Other) :6835   
## education..category default..category balance..number housing..category  
## primary : 6851 no :44396 Min. : -8019 no :20081   
## secondary:23202 yes: 815 1st Qu.: 72 yes:25130   
## tertiary :13301 Median : 448   
## unknown : 1857 Mean : 1362   
## 3rd Qu.: 1428   
## Max. :102127   
##   
## loan..category contact..category day..number month..category  
## no :37967 cellular :29285 Min. : 1.00 may :13766   
## yes: 7244 telephone: 2906 1st Qu.: 8.00 jul : 6895   
## unknown :13020 Median :16.00 aug : 6247   
## Mean :15.81 jun : 5341   
## 3rd Qu.:21.00 nov : 3970   
## Max. :31.00 apr : 2932   
## (Other): 6060   
## campaign..number pdays..number previous..number poutcome..category  
## Min. : 1.000 Min. : -1.0 Min. : 0.0000 failure: 4901   
## 1st Qu.: 1.000 1st Qu.: -1.0 1st Qu.: 0.0000 other : 1840   
## Median : 2.000 Median : -1.0 Median : 0.0000 success: 1511   
## Mean : 2.764 Mean : 40.2 Mean : 0.5803 unknown:36959   
## 3rd Qu.: 3.000 3rd Qu.: -1.0 3rd Qu.: 0.0000   
## Max. :63.000 Max. :871.0 Max. :275.0000   
##

dim(bankfull2)

## [1] 45211 16

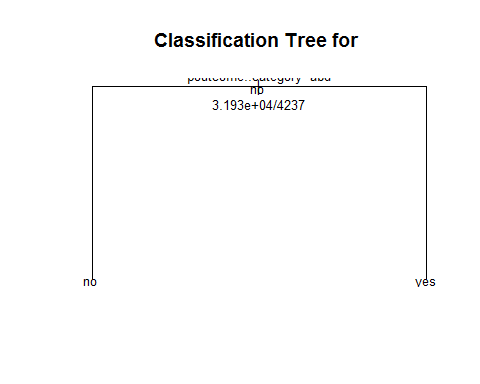
names(bankfull2)

## [1] "y..category" "age..number" "job..category"   
## [4] "marital..category" "education..category" "default..category"   
## [7] "balance..number" "housing..category" "loan..category"   
## [10] "contact..category" "day..number" "month..category"   
## [13] "campaign..number" "pdays..number" "previous..number"   
## [16] "poutcome..category"

set.seed(123)  
bankfull\_rand <- bankfull2[order(runif(45211)), ]   
bankfull\_train <- bankfull\_rand[1:36169, ]   
bankfull\_test <-bankfull\_rand[36170:45211, ]  
  
library(rpart)  
bankfull\_rpart <- rpart(bankfull\_train$y..category~., method="class", parms = list(split="informationgain"), data=bankfull\_train)  
summary(bankfull\_rpart)

## Call:  
## rpart(formula = bankfull\_train$y..category ~ ., data = bankfull\_train,   
## method = "class", parms = list(split = "informationgain"))  
## n= 36169   
##   
## CP nsplit rel error xerror xstd  
## 1 0.08803399 0 1.000000 1.000000 0.01443496  
## 2 0.01000000 1 0.911966 0.911966 0.01386522  
##   
## Variable importance  
## poutcome..category   
## 100   
##   
## Node number 1: 36169 observations, complexity param=0.08803399  
## predicted class=no expected loss=0.1171445 P(node) =1  
## class counts: 31932 4237  
## probabilities: 0.883 0.117   
## left son=2 (34950 obs) right son=3 (1219 obs)  
## Primary splits:  
## poutcome..category splits as LLRL, improve=724.4511, (0 missing)  
## month..category splits as LLRLLLLRLLRR, improve=417.9430, (0 missing)  
## pdays..number < 8.5 to the left, improve=220.4904, (0 missing)  
## previous..number < 0.5 to the left, improve=217.3589, (0 missing)  
## age..number < 60.5 to the left, improve=182.9319, (0 missing)  
##   
## Node number 2: 34950 observations  
## predicted class=no expected loss=0.09845494 P(node) =0.9662971  
## class counts: 31509 3441  
## probabilities: 0.902 0.098   
##   
## Node number 3: 1219 observations  
## predicted class=yes expected loss=0.3470057 P(node) =0.03370289  
## class counts: 423 796  
## probabilities: 0.347 0.653

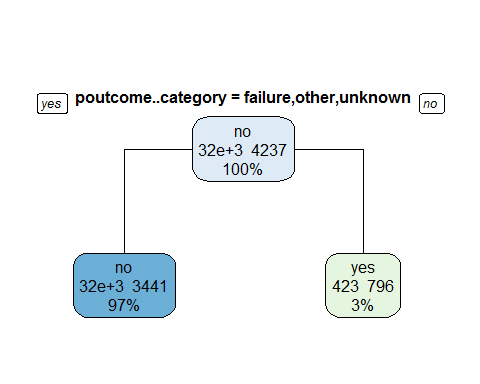
plot(bankfull\_rpart, uniform=TRUE, main="Classification Tree for ")  
text(bankfull\_rpart, use.n=TRUE, all=TRUE, cex=0.8)



library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 3.3.3

rpart.plot(bankfull\_rpart, type=1, extra=101)



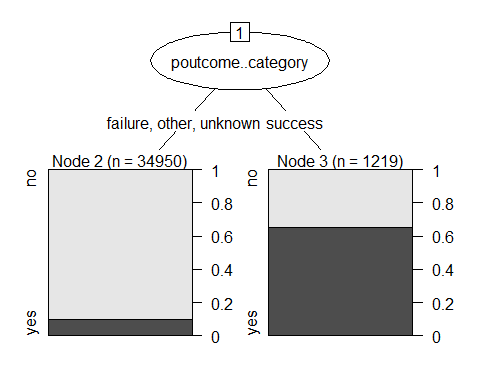
library(party)  
library(partykit)

## Warning: package 'partykit' was built under R version 3.3.3

##   
## Attaching package: 'partykit'

## The following objects are masked from 'package:party':  
##   
## cforest, ctree, ctree\_control, edge\_simple, mob, mob\_control,  
## node\_barplot, node\_bivplot, node\_boxplot, node\_inner,  
## node\_surv, node\_terminal

bankfull\_party<-as.party(bankfull\_rpart)  
plot(bankfull\_party)



library(caret)  
actual <- bankfull\_test$y..category  
predicted <- predict(bankfull\_rpart, bankfull\_test, type="class")  
results.matrix <- confusionMatrix(predicted, actual, positive="yes")  
print(results.matrix)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 7880 870  
## yes 110 182  
##   
## Accuracy : 0.8916   
## 95% CI : (0.885, 0.898)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.009025   
##   
## Kappa : 0.232   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.17300   
## Specificity : 0.98623   
## Pos Pred Value : 0.62329   
## Neg Pred Value : 0.90057   
## Prevalence : 0.11635   
## Detection Rate : 0.02013   
## Detection Prevalence : 0.03229   
## Balanced Accuracy : 0.57962   
##   
## 'Positive' Class : yes   
##

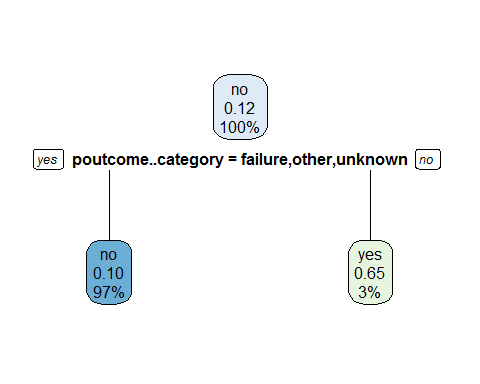
########################################################################################  
  
library(caret)  
  
set.seed(123)  
folds<-createFolds(y=bankfull\_train$y..category, k=10, list=TRUE, returnTrain=FALSE)  
sapply(folds, length)

## Fold01 Fold02 Fold03 Fold04 Fold05 Fold06 Fold07 Fold08 Fold09 Fold10   
## 3616 3617 3617 3617 3617 3616 3617 3618 3617 3617

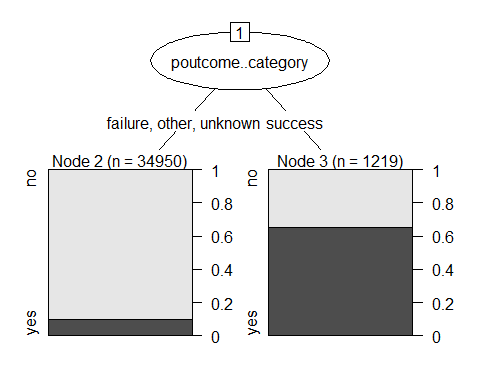
folds$Fold01

## [1] 28 39 51 53 54 65 77 84 88 89 95  
## [12] 97 98 104 108 111 115 119 157 162 164 166  
## [23] 167 173 180 190 206 207 208 210 211 216 235  
## [34] 238 240 244 268 271 275 288 291 298 302 337  
## [45] 343 354 355 364 370 384 389 402 410 414 423  
## [56] 465 489 520 521 527 529 539 541 589 604 607  
## [67] 615 616 617 650 653 656 685 691 715 728 732  
## [78] 745 748 763 781 788 801 806 828 829 844 861  
## [89] 885 887 918 925 933 949 955 962 963 973 988  
## [100] 1021 1034 1045 1056 1057 1066 1070 1071 1084 1099 1125  
## [111] 1135 1148 1151 1152 1178 1182 1228 1275 1276 1281 1283  
## [122] 1290 1291 1292 1293 1352 1358 1361 1375 1381 1382 1396  
## [133] 1403 1412 1417 1423 1446 1448 1462 1464 1483 1484 1503  
## [144] 1504 1515 1517 1518 1547 1548 1549 1550 1568 1583 1586  
## [155] 1587 1608 1610 1616 1619 1629 1631 1641 1646 1677 1679  
## [166] 1696 1699 1718 1721 1726 1728 1770 1805 1815 1817 1818  
## [177] 1826 1841 1849 1874 1905 1909 1931 1937 1938 1939 1948  
## [188] 1981 2002 2033 2039 2051 2052 2076 2089 2098 2101 2105  
## [199] 2116 2118 2119 2124 2145 2148 2149 2171 2177 2183 2187  
## [210] 2192 2202 2222 2256 2257 2262 2264 2265 2266 2268 2298  
## [221] 2300 2302 2311 2313 2331 2337 2345 2347 2349 2364 2379  
## [232] 2381 2387 2400 2402 2403 2412 2415 2449 2461 2464 2469  
## [243] 2479 2503 2525 2529 2539 2541 2562 2607 2616 2629 2632  
## [254] 2677 2678 2679 2689 2698 2699 2733 2738 2752 2776 2777  
## [265] 2780 2798 2805 2809 2829 2837 2838 2841 2846 2851 2867  
## [276] 2885 2888 2893 2901 2908 2920 2927 2928 2942 2972 2973  
## [287] 2976 3000 3011 3014 3015 3018 3027 3030 3037 3041 3065  
## [298] 3067 3071 3077 3080 3089 3100 3110 3129 3133 3144 3152  
## [309] 3175 3177 3192 3207 3211 3243 3249 3285 3287 3290 3315  
## [320] 3320 3325 3327 3344 3346 3357 3369 3371 3384 3390 3410  
## [331] 3411 3417 3422 3431 3438 3455 3456 3467 3497 3503 3508  
## [342] 3519 3520 3525 3541 3553 3564 3565 3582 3585 3610 3626  
## [353] 3628 3638 3653 3668 3671 3688 3711 3714 3728 3738 3740  
## [364] 3742 3746 3761 3781 3789 3798 3821 3826 3833 3836 3839  
## [375] 3842 3849 3856 3874 3878 3882 3887 3892 3895 3899 3907  
## [386] 3918 3924 3934 3936 3945 3947 3950 3978 3985 3986 3995  
## [397] 3997 4028 4029 4046 4068 4080 4082 4110 4117 4138 4145  
## [408] 4155 4162 4171 4186 4191 4194 4200 4207 4213 4221 4229  
## [419] 4237 4246 4251 4270 4280 4282 4287 4295 4304 4307 4324  
## [430] 4328 4331 4382 4396 4402 4414 4420 4431 4445 4452 4460  
## [441] 4463 4479 4492 4495 4510 4550 4570 4587 4612 4615 4616  
## [452] 4621 4623 4626 4663 4664 4667 4673 4727 4729 4745 4749  
## [463] 4750 4770 4772 4788 4794 4812 4834 4843 4871 4875 4895  
## [474] 4916 4921 4931 4946 4951 4964 4990 4998 5010 5014 5045  
## [485] 5051 5064 5065 5068 5073 5100 5104 5106 5113 5116 5128  
## [496] 5130 5152 5165 5176 5179 5180 5187 5190 5191 5196 5198  
## [507] 5203 5209 5214 5215 5216 5241 5292 5295 5297 5302 5356  
## [518] 5368 5389 5397 5403 5438 5440 5465 5466 5474 5494 5547  
## [529] 5556 5579 5590 5599 5605 5625 5626 5654 5698 5705 5707  
## [540] 5721 5724 5766 5771 5783 5787 5810 5821 5838 5840 5843  
## [551] 5849 5880 5894 5911 5918 5920 5925 5931 5943 5944 5946  
## [562] 5947 5949 5953 5955 5961 5965 5975 5989 5997 5999 6010  
## [573] 6011 6013 6015 6024 6063 6082 6083 6091 6106 6110 6118  
## [584] 6127 6128 6143 6145 6157 6159 6176 6177 6195 6203 6208  
## [595] 6222 6224 6227 6251 6259 6266 6270 6274 6277 6282 6285  
## [606] 6291 6292 6296 6297 6309 6334 6337 6340 6357 6362 6381  
## [617] 6386 6405 6410 6411 6416 6434 6476 6513 6514 6516 6526  
## [628] 6528 6530 6543 6545 6556 6591 6608 6610 6612 6617 6636  
## [639] 6639 6648 6653 6665 6675 6677 6685 6691 6698 6699 6726  
## [650] 6732 6742 6772 6773 6778 6781 6786 6810 6844 6846 6849  
## [661] 6852 6873 6882 6901 6902 6904 6914 6941 6942 6946 6953  
## [672] 6955 6962 6974 6990 6992 6996 7001 7006 7008 7013 7015  
## [683] 7026 7036 7037 7038 7039 7041 7045 7051 7065 7070 7098  
## [694] 7133 7142 7152 7161 7166 7177 7187 7197 7219 7222 7228  
## [705] 7229 7259 7294 7309 7317 7328 7335 7338 7339 7346 7352  
## [716] 7363 7369 7370 7373 7389 7409 7410 7415 7420 7431 7433  
## [727] 7445 7448 7462 7485 7487 7494 7502 7512 7513 7547 7562  
## [738] 7566 7568 7579 7587 7596 7597 7605 7607 7611 7614 7622  
## [749] 7629 7651 7662 7669 7670 7689 7690 7701 7703 7706 7721  
## [760] 7727 7733 7743 7748 7773 7776 7778 7806 7809 7825 7826  
## [771] 7827 7840 7848 7868 7871 7889 7890 7907 7913 7934 7943  
## [782] 7973 7979 7980 7991 7992 7998 8002 8032 8035 8038 8050  
## [793] 8060 8087 8100 8123 8156 8158 8165 8167 8171 8181 8182  
## [804] 8187 8206 8215 8224 8229 8235 8239 8240 8245 8259 8288  
## [815] 8292 8308 8326 8328 8341 8342 8345 8369 8399 8400 8426  
## [826] 8438 8458 8459 8480 8495 8498 8504 8513 8524 8529 8532  
## [837] 8548 8568 8570 8579 8580 8582 8587 8604 8605 8610 8613  
## [848] 8615 8616 8617 8635 8638 8648 8671 8677 8686 8687 8692  
## [859] 8697 8712 8729 8734 8745 8753 8756 8762 8765 8876 8889  
## [870] 8891 8896 8897 8904 8906 8915 8948 8967 8999 9000 9022  
## [881] 9050 9097 9107 9128 9141 9164 9175 9194 9217 9219 9222  
## [892] 9228 9242 9253 9257 9261 9263 9268 9275 9280 9287 9300  
## [903] 9303 9316 9336 9344 9369 9375 9378 9389 9393 9424 9427  
## [914] 9428 9432 9443 9449 9453 9455 9460 9464 9470 9478 9494  
## [925] 9498 9520 9527 9548 9553 9565 9566 9576 9580 9589 9596  
## [936] 9609 9610 9613 9653 9655 9660 9678 9683 9689 9694 9698  
## [947] 9705 9711 9712 9722 9730 9739 9740 9741 9750 9757 9847  
## [958] 9849 9884 9917 9959 9961 9963 9965 9969 9979 9995 10001  
## [969] 10023 10039 10048 10056 10063 10067 10072 10085 10088 10095 10116  
## [980] 10119 10132 10140 10146 10163 10180 10199 10201 10208 10236 10241  
## [991] 10245 10259 10276 10305 10318 10323 10324 10327 10329 10346 10364  
## [1002] 10371 10380 10396 10411 10424 10434 10435 10463 10467 10470 10476  
## [1013] 10480 10489 10492 10493 10516 10544 10545 10548 10560 10572 10583  
## [1024] 10586 10606 10657 10663 10700 10715 10726 10737 10750 10760 10767  
## [1035] 10769 10779 10782 10797 10829 10839 10848 10855 10876 10890 10894  
## [1046] 10896 10897 10904 10965 10966 10967 10970 10974 11000 11003 11011  
## [1057] 11012 11029 11031 11050 11053 11080 11081 11090 11122 11125 11139  
## [1068] 11141 11147 11156 11167 11201 11212 11219 11242 11247 11254 11256  
## [1079] 11274 11305 11313 11328 11330 11335 11336 11355 11356 11366 11370  
## [1090] 11391 11394 11397 11400 11420 11427 11429 11434 11435 11440 11442  
## [1101] 11454 11457 11459 11507 11508 11522 11525 11537 11548 11581 11609  
## [1112] 11615 11627 11651 11657 11675 11676 11691 11693 11703 11708 11744  
## [1123] 11755 11757 11760 11765 11768 11769 11774 11784 11790 11791 11807  
## [1134] 11827 11856 11899 11905 11921 11924 11925 11927 11943 11970 11973  
## [1145] 11980 11981 12001 12004 12019 12049 12053 12060 12072 12076 12077  
## [1156] 12091 12101 12103 12106 12107 12112 12124 12143 12144 12145 12147  
## [1167] 12152 12155 12161 12165 12191 12209 12213 12238 12246 12251 12267  
## [1178] 12286 12291 12307 12308 12313 12358 12360 12392 12403 12404 12434  
## [1189] 12440 12445 12456 12459 12466 12471 12472 12473 12478 12482 12485  
## [1200] 12509 12521 12541 12579 12582 12594 12604 12607 12624 12625 12641  
## [1211] 12646 12664 12668 12691 12697 12700 12707 12720 12726 12730 12732  
## [1222] 12739 12751 12768 12773 12782 12784 12785 12803 12818 12841 12842  
## [1233] 12843 12849 12850 12852 12868 12872 12875 12904 12905 12924 12942  
## [1244] 12946 12947 12963 12976 12979 12986 12996 13012 13051 13052 13058  
## [1255] 13065 13074 13111 13116 13127 13133 13143 13155 13187 13192 13208  
## [1266] 13214 13226 13296 13303 13318 13331 13341 13343 13352 13359 13379  
## [1277] 13384 13388 13389 13400 13414 13416 13418 13428 13439 13464 13482  
## [1288] 13484 13494 13503 13505 13523 13542 13543 13574 13615 13622 13624  
## [1299] 13627 13632 13640 13642 13650 13655 13668 13690 13707 13713 13718  
## [1310] 13739 13744 13765 13772 13792 13795 13797 13807 13813 13829 13835  
## [1321] 13836 13850 13856 13883 13884 13894 13901 13905 13907 13930 13933  
## [1332] 13934 13945 13963 13972 13977 14006 14017 14018 14019 14023 14027  
## [1343] 14030 14034 14036 14038 14039 14048 14052 14055 14057 14066 14069  
## [1354] 14077 14086 14102 14127 14141 14143 14146 14147 14148 14156 14158  
## [1365] 14159 14164 14216 14220 14225 14234 14235 14241 14250 14252 14258  
## [1376] 14278 14291 14299 14312 14322 14337 14339 14347 14348 14354 14355  
## [1387] 14373 14375 14411 14419 14424 14427 14470 14478 14489 14507 14511  
## [1398] 14529 14537 14546 14548 14591 14597 14603 14631 14633 14655 14672  
## [1409] 14678 14691 14698 14713 14716 14738 14751 14752 14753 14765 14775  
## [1420] 14790 14801 14805 14810 14820 14835 14852 14885 14886 14922 14924  
## [1431] 14929 14933 14937 14939 14942 14956 14966 14984 14985 14987 14990  
## [1442] 14993 14995 15001 15011 15019 15038 15043 15063 15071 15078 15097  
## [1453] 15101 15107 15121 15126 15127 15166 15175 15207 15224 15229 15232  
## [1464] 15233 15243 15263 15289 15304 15307 15322 15331 15339 15349 15358  
## [1475] 15392 15393 15404 15406 15415 15429 15431 15433 15437 15447 15455  
## [1486] 15467 15468 15483 15488 15491 15494 15501 15504 15520 15539 15554  
## [1497] 15569 15576 15615 15620 15627 15632 15639 15665 15682 15683 15701  
## [1508] 15705 15730 15733 15744 15769 15792 15800 15838 15854 15869 15880  
## [1519] 15898 15900 15908 15936 15985 16043 16062 16069 16070 16095 16100  
## [1530] 16116 16118 16128 16134 16163 16169 16170 16173 16180 16183 16200  
## [1541] 16217 16219 16230 16243 16256 16262 16263 16274 16280 16292 16296  
## [1552] 16306 16316 16320 16333 16334 16356 16359 16373 16380 16393 16396  
## [1563] 16403 16416 16421 16424 16442 16445 16459 16467 16490 16494 16500  
## [1574] 16533 16535 16536 16552 16570 16582 16586 16591 16600 16601 16621  
## [1585] 16630 16632 16636 16656 16662 16663 16668 16673 16687 16694 16695  
## [1596] 16696 16697 16700 16716 16724 16735 16740 16742 16760 16764 16778  
## [1607] 16784 16785 16800 16811 16819 16829 16832 16833 16836 16844 16860  
## [1618] 16869 16870 16872 16888 16894 16905 16913 16915 16930 16941 16949  
## [1629] 16950 16952 16959 16965 16971 16976 16990 16993 17000 17011 17016  
## [1640] 17030 17044 17048 17054 17067 17069 17086 17106 17107 17145 17154  
## [1651] 17160 17164 17170 17178 17198 17215 17235 17249 17266 17272 17306  
## [1662] 17315 17317 17329 17337 17368 17371 17373 17375 17383 17384 17399  
## [1673] 17421 17426 17427 17430 17431 17434 17460 17474 17475 17482 17493  
## [1684] 17494 17514 17515 17529 17551 17557 17559 17579 17601 17614 17617  
## [1695] 17623 17633 17647 17657 17668 17684 17694 17696 17709 17711 17714  
## [1706] 17716 17719 17720 17740 17742 17744 17745 17757 17763 17764 17781  
## [1717] 17782 17794 17802 17810 17827 17830 17831 17840 17841 17845 17850  
## [1728] 17872 17911 17914 17929 17937 17941 17957 17964 17965 17973 17982  
## [1739] 17990 17999 18022 18024 18034 18041 18051 18056 18073 18101 18111  
## [1750] 18115 18127 18155 18156 18159 18164 18179 18181 18188 18195 18208  
## [1761] 18232 18261 18269 18270 18292 18296 18302 18303 18371 18381 18403  
## [1772] 18423 18428 18430 18456 18464 18469 18510 18532 18544 18563 18568  
## [1783] 18575 18593 18611 18615 18621 18624 18653 18663 18667 18690 18695  
## [1794] 18710 18715 18720 18731 18755 18763 18772 18795 18808 18819 18829  
## [1805] 18848 18849 18867 18872 18875 18888 18889 18893 18896 18908 18911  
## [1816] 18928 18937 18961 18963 18967 18970 18993 19005 19011 19018 19031  
## [1827] 19046 19050 19052 19060 19064 19065 19087 19093 19111 19112 19124  
## [1838] 19145 19146 19160 19164 19169 19183 19190 19205 19209 19217 19222  
## [1849] 19236 19251 19266 19297 19332 19339 19354 19359 19361 19365 19372  
## [1860] 19384 19416 19418 19419 19439 19443 19445 19448 19490 19495 19503  
## [1871] 19505 19508 19520 19538 19547 19552 19595 19601 19606 19613 19618  
## [1882] 19645 19646 19672 19691 19695 19707 19708 19719 19721 19725 19733  
## [1893] 19740 19749 19760 19761 19785 19815 19829 19833 19835 19844 19863  
## [1904] 19866 19872 19873 19876 19893 19894 19907 19911 19925 19936 19937  
## [1915] 19948 19971 19972 19983 19988 20003 20013 20018 20040 20043 20046  
## [1926] 20054 20058 20064 20073 20079 20082 20087 20093 20111 20118 20120  
## [1937] 20131 20137 20138 20144 20147 20150 20173 20182 20191 20207 20208  
## [1948] 20236 20250 20260 20265 20268 20280 20303 20306 20319 20338 20342  
## [1959] 20350 20351 20353 20356 20364 20366 20371 20376 20388 20390 20396  
## [1970] 20407 20409 20412 20424 20432 20436 20438 20453 20460 20485 20500  
## [1981] 20502 20507 20512 20532 20533 20537 20550 20554 20558 20564 20567  
## [1992] 20590 20592 20610 20612 20631 20632 20633 20639 20657 20691 20696  
## [2003] 20705 20724 20743 20773 20776 20777 20792 20798 20802 20803 20837  
## [2014] 20851 20852 20858 20869 20876 20915 20917 20946 20951 20954 20958  
## [2025] 20967 20977 20983 20986 21031 21034 21043 21049 21050 21057 21060  
## [2036] 21062 21063 21075 21076 21090 21096 21102 21104 21109 21128 21134  
## [2047] 21146 21157 21172 21179 21182 21198 21207 21214 21215 21216 21236  
## [2058] 21240 21251 21252 21262 21273 21276 21282 21297 21308 21311 21320  
## [2069] 21321 21331 21351 21356 21375 21382 21386 21393 21400 21408 21411  
## [2080] 21414 21417 21437 21442 21447 21462 21463 21492 21505 21514 21577  
## [2091] 21597 21611 21615 21629 21631 21637 21640 21643 21647 21650 21656  
## [2102] 21660 21664 21679 21684 21685 21686 21690 21692 21697 21699 21702  
## [2113] 21713 21727 21732 21735 21736 21738 21753 21759 21767 21775 21793  
## [2124] 21796 21808 21828 21858 21865 21876 21884 21914 21928 21933 21936  
## [2135] 21942 21945 21946 21947 21952 21959 21968 21980 21987 21999 22000  
## [2146] 22006 22007 22008 22011 22015 22025 22029 22035 22046 22060 22062  
## [2157] 22098 22100 22113 22120 22126 22130 22136 22137 22147 22148 22158  
## [2168] 22166 22177 22183 22191 22241 22243 22244 22253 22256 22259 22261  
## [2179] 22282 22298 22306 22309 22310 22311 22319 22323 22326 22331 22340  
## [2190] 22351 22357 22363 22364 22382 22385 22386 22392 22395 22397 22415  
## [2201] 22432 22438 22467 22482 22485 22487 22491 22496 22504 22508 22514  
## [2212] 22529 22541 22554 22557 22565 22570 22580 22606 22607 22612 22623  
## [2223] 22627 22634 22661 22666 22672 22673 22677 22689 22703 22705 22707  
## [2234] 22711 22727 22729 22741 22755 22759 22767 22787 22801 22808 22809  
## [2245] 22811 22815 22833 22835 22846 22849 22850 22863 22865 22866 22872  
## [2256] 22878 22882 22884 22896 22924 22994 22995 23021 23032 23034 23046  
## [2267] 23049 23053 23057 23059 23081 23094 23105 23128 23139 23152 23161  
## [2278] 23165 23166 23170 23176 23187 23191 23203 23211 23212 23244 23294  
## [2289] 23315 23322 23330 23345 23369 23375 23377 23378 23381 23383 23389  
## [2300] 23403 23404 23433 23437 23450 23456 23459 23468 23469 23478 23502  
## [2311] 23509 23518 23523 23536 23574 23577 23601 23614 23625 23627 23645  
## [2322] 23649 23676 23683 23689 23718 23729 23748 23751 23752 23762 23766  
## [2333] 23794 23801 23802 23804 23815 23825 23827 23832 23837 23847 23848  
## [2344] 23867 23882 23918 23922 23929 23941 23944 23966 23984 23991 24001  
## [2355] 24011 24012 24024 24026 24054 24093 24107 24116 24119 24120 24148  
## [2366] 24152 24155 24157 24167 24195 24196 24223 24225 24238 24240 24262  
## [2377] 24264 24270 24272 24303 24321 24339 24363 24396 24400 24401 24409  
## [2388] 24414 24425 24427 24442 24459 24461 24476 24477 24484 24488 24502  
## [2399] 24503 24509 24520 24527 24528 24529 24531 24532 24537 24566 24567  
## [2410] 24571 24581 24583 24584 24587 24597 24604 24611 24614 24621 24627  
## [2421] 24652 24656 24681 24683 24688 24713 24727 24740 24743 24765 24772  
## [2432] 24783 24784 24802 24806 24811 24812 24825 24841 24847 24853 24865  
## [2443] 24872 24878 24888 24897 24935 24945 24946 24954 24967 24969 25041  
## [2454] 25043 25044 25048 25059 25063 25073 25080 25091 25093 25102 25109  
## [2465] 25119 25137 25145 25146 25147 25173 25177 25190 25196 25202 25203  
## [2476] 25205 25212 25214 25215 25216 25221 25236 25269 25290 25291 25311  
## [2487] 25322 25324 25338 25347 25353 25360 25362 25368 25371 25374 25380  
## [2498] 25385 25393 25409 25411 25417 25423 25425 25432 25435 25438 25440  
## [2509] 25459 25461 25464 25468 25476 25478 25480 25486 25491 25494 25515  
## [2520] 25529 25541 25578 25591 25597 25603 25610 25644 25676 25687 25691  
## [2531] 25694 25696 25705 25711 25716 25727 25733 25734 25740 25742 25743  
## [2542] 25766 25772 25776 25796 25809 25821 25825 25832 25837 25840 25846  
## [2553] 25852 25877 25884 25886 25891 25893 25894 25910 25915 25918 25942  
## [2564] 25987 26008 26009 26018 26020 26064 26065 26083 26089 26106 26126  
## [2575] 26128 26134 26136 26139 26149 26162 26173 26176 26183 26186 26200  
## [2586] 26209 26215 26230 26247 26265 26267 26285 26289 26313 26324 26351  
## [2597] 26354 26358 26365 26374 26389 26412 26459 26466 26467 26479 26489  
## [2608] 26515 26519 26532 26534 26544 26545 26552 26553 26572 26589 26590  
## [2619] 26596 26600 26608 26626 26636 26638 26661 26663 26672 26689 26698  
## [2630] 26702 26707 26727 26734 26737 26746 26750 26759 26765 26772 26775  
## [2641] 26779 26786 26817 26819 26829 26832 26835 26842 26844 26858 26862  
## [2652] 26871 26891 26894 26910 26924 26929 26930 26948 26957 26961 26965  
## [2663] 26971 26979 26985 27003 27013 27014 27038 27039 27041 27053 27062  
## [2674] 27069 27076 27080 27084 27094 27096 27097 27103 27122 27148 27153  
## [2685] 27159 27183 27191 27209 27234 27235 27239 27245 27276 27279 27286  
## [2696] 27290 27300 27304 27319 27321 27327 27345 27361 27375 27376 27380  
## [2707] 27381 27383 27392 27402 27413 27414 27428 27430 27440 27443 27453  
## [2718] 27458 27463 27478 27498 27503 27518 27521 27524 27526 27531 27543  
## [2729] 27557 27560 27563 27575 27586 27608 27619 27632 27639 27676 27677  
## [2740] 27678 27685 27712 27713 27715 27716 27731 27736 27739 27750 27775  
## [2751] 27776 27802 27811 27822 27828 27830 27844 27846 27855 27900 27906  
## [2762] 27909 27914 27915 27921 27924 27932 27950 27958 27961 27984 27986  
## [2773] 27989 27996 28002 28017 28024 28030 28036 28039 28041 28055 28067  
## [2784] 28078 28084 28090 28092 28101 28103 28133 28142 28144 28147 28161  
## [2795] 28162 28175 28182 28185 28193 28213 28222 28227 28228 28237 28242  
## [2806] 28250 28257 28267 28284 28335 28343 28348 28352 28354 28361 28371  
## [2817] 28390 28405 28412 28422 28430 28436 28441 28443 28460 28461 28468  
## [2828] 28495 28505 28512 28516 28531 28546 28549 28550 28553 28557 28561  
## [2839] 28564 28568 28584 28608 28616 28620 28621 28623 28631 28656 28664  
## [2850] 28676 28695 28703 28716 28727 28743 28779 28784 28790 28792 28795  
## [2861] 28803 28808 28815 28829 28832 28833 28837 28842 28852 28855 28856  
## [2872] 28861 28874 28888 28902 28917 28918 28920 28928 28931 28949 28957  
## [2883] 28961 28986 28998 29001 29006 29008 29016 29026 29027 29036 29037  
## [2894] 29042 29043 29045 29047 29062 29069 29080 29101 29138 29144 29145  
## [2905] 29150 29201 29206 29217 29221 29226 29231 29240 29241 29245 29249  
## [2916] 29261 29272 29277 29287 29294 29316 29321 29352 29353 29400 29408  
## [2927] 29419 29439 29448 29450 29464 29470 29475 29493 29500 29518 29537  
## [2938] 29541 29555 29557 29566 29582 29583 29626 29637 29646 29653 29655  
## [2949] 29656 29658 29675 29680 29701 29717 29733 29742 29782 29787 29801  
## [2960] 29804 29814 29836 29899 29913 29922 29931 29960 29961 29967 29968  
## [2971] 29970 29979 29988 29994 30000 30002 30007 30013 30025 30032 30036  
## [2982] 30038 30041 30048 30050 30052 30053 30055 30077 30086 30088 30096  
## [2993] 30106 30117 30120 30121 30128 30143 30159 30169 30177 30178 30181  
## [3004] 30192 30194 30209 30210 30236 30244 30253 30264 30270 30287 30310  
## [3015] 30318 30319 30325 30331 30334 30341 30351 30356 30357 30360 30401  
## [3026] 30414 30419 30421 30429 30437 30440 30450 30462 30478 30510 30528  
## [3037] 30543 30549 30551 30554 30575 30592 30598 30601 30607 30632 30646  
## [3048] 30654 30655 30669 30679 30715 30720 30739 30746 30757 30762 30763  
## [3059] 30797 30804 30810 30813 30817 30820 30825 30843 30855 30861 30873  
## [3070] 30875 30879 30899 30900 30901 30928 30935 30955 30962 30966 30978  
## [3081] 30991 31010 31023 31041 31052 31060 31062 31067 31091 31094 31129  
## [3092] 31144 31149 31151 31157 31172 31173 31179 31183 31196 31202 31214  
## [3103] 31220 31240 31245 31254 31266 31277 31282 31294 31301 31313 31314  
## [3114] 31354 31358 31360 31362 31391 31395 31400 31406 31407 31424 31427  
## [3125] 31444 31451 31458 31482 31497 31505 31531 31545 31555 31559 31562  
## [3136] 31588 31623 31624 31626 31627 31658 31693 31711 31730 31731 31735  
## [3147] 31750 31754 31778 31785 31796 31804 31808 31810 31825 31834 31840  
## [3158] 31846 31858 31859 31861 31863 31891 31900 31929 31930 31936 31962  
## [3169] 32003 32004 32006 32011 32033 32039 32055 32056 32061 32062 32064  
## [3180] 32077 32080 32081 32084 32095 32096 32098 32113 32117 32138 32167  
## [3191] 32177 32188 32190 32204 32205 32212 32213 32214 32217 32222 32245  
## [3202] 32255 32265 32273 32293 32303 32307 32311 32327 32351 32369 32376  
## [3213] 32401 32421 32433 32434 32443 32486 32488 32541 32550 32554 32556  
## [3224] 32583 32587 32592 32593 32597 32602 32604 32606 32651 32657 32663  
## [3235] 32665 32673 32678 32683 32684 32686 32690 32696 32713 32725 32730  
## [3246] 32733 32737 32740 32742 32745 32753 32759 32770 32772 32773 32797  
## [3257] 32798 32800 32802 32816 32824 32834 32837 32838 32845 32847 32871  
## [3268] 32872 32878 32880 32895 32908 32912 32916 32940 32943 32956 32969  
## [3279] 32987 33000 33018 33022 33030 33054 33058 33060 33068 33072 33087  
## [3290] 33098 33100 33112 33118 33132 33135 33146 33159 33167 33171 33190  
## [3301] 33202 33212 33219 33240 33246 33281 33283 33306 33314 33331 33340  
## [3312] 33341 33342 33346 33357 33358 33375 33381 33386 33394 33397 33398  
## [3323] 33406 33426 33435 33456 33463 33464 33467 33470 33471 33490 33513  
## [3334] 33515 33529 33531 33534 33538 33539 33544 33548 33559 33562 33588  
## [3345] 33592 33611 33623 33635 33637 33638 33653 33663 33673 33696 33702  
## [3356] 33706 33709 33710 33712 33714 33718 33733 33737 33755 33765 33774  
## [3367] 33781 33805 33816 33848 33853 33859 33860 33867 33868 33885 33906  
## [3378] 33929 33940 33947 33953 33961 33964 33984 33994 34002 34007 34015  
## [3389] 34025 34047 34054 34055 34067 34076 34094 34119 34129 34142 34148  
## [3400] 34150 34162 34164 34173 34177 34183 34185 34197 34215 34216 34221  
## [3411] 34225 34240 34257 34288 34291 34296 34315 34319 34321 34334 34340  
## [3422] 34349 34365 34369 34370 34376 34389 34420 34423 34424 34446 34449  
## [3433] 34454 34484 34498 34507 34543 34555 34560 34566 34567 34573 34574  
## [3444] 34581 34584 34590 34593 34619 34621 34627 34693 34696 34711 34712  
## [3455] 34722 34728 34760 34764 34776 34779 34792 34803 34806 34809 34810  
## [3466] 34834 34844 34846 34848 34856 34879 34890 34895 34909 34911 34922  
## [3477] 34925 34931 34938 34941 34942 34957 34975 34976 34990 34996 35013  
## [3488] 35014 35018 35059 35064 35075 35101 35102 35115 35121 35143 35152  
## [3499] 35172 35173 35174 35177 35187 35198 35200 35209 35222 35228 35233  
## [3510] 35265 35266 35283 35286 35319 35325 35333 35338 35343 35348 35356  
## [3521] 35362 35371 35378 35383 35386 35433 35437 35441 35448 35451 35452  
## [3532] 35481 35486 35490 35493 35499 35501 35528 35539 35543 35578 35596  
## [3543] 35600 35601 35603 35613 35626 35642 35657 35660 35662 35666 35672  
## [3554] 35690 35699 35707 35709 35719 35723 35727 35733 35734 35735 35739  
## [3565] 35747 35751 35759 35774 35778 35779 35823 35827 35837 35839 35847  
## [3576] 35853 35857 35859 35862 35865 35887 35890 35893 35898 35900 35911  
## [3587] 35913 35923 35938 35952 35958 35966 35969 35988 35995 35997 35999  
## [3598] 36001 36021 36038 36054 36060 36064 36079 36083 36085 36091 36099  
## [3609] 36104 36110 36114 36126 36130 36131 36144 36147

cvCtrl <- trainControl(method="cv", number=10)  
set.seed(123)  
bankfull.caret.10folds<-train(y..category~., data=bankfull\_train, method="rpart", metric="Accuracy", tuneLength=10, na.action = na.omit, trControl=cvCtrl)  
  
bankfull.rpart.pruned<-prune(bankfull\_rpart, cp=0)  
rpart.plot(bankfull.rpart.pruned)



bankfull\_party2<-as.party(bankfull.rpart.pruned)  
plot(bankfull\_party2)



actual <- bankfull\_test$y..category   
predicted2 <- predict(bankfull.rpart.pruned, bankfull\_test, type="class")   
bankfull\_results.matrix.2 <- confusionMatrix(predicted2, actual, positive="yes")   
print(bankfull\_results.matrix.2)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 7880 870  
## yes 110 182  
##   
## Accuracy : 0.8916   
## 95% CI : (0.885, 0.898)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.009025   
##   
## Kappa : 0.232   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.17300   
## Specificity : 0.98623   
## Pos Pred Value : 0.62329   
## Neg Pred Value : 0.90057   
## Prevalence : 0.11635   
## Detection Rate : 0.02013   
## Detection Prevalence : 0.03229   
## Balanced Accuracy : 0.57962   
##   
## 'Positive' Class : yes   
##

###########################################################################################  
  
library(randomForest)

## Warning: package 'randomForest' was built under R version 3.3.3

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

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

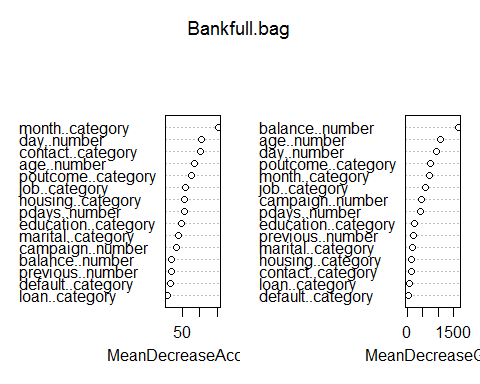
set.seed(123)   
  
Bankfull.bag <- randomForest(y..category~., data=bankfull\_train, mtry=15, na.action=na.omit, importance=TRUE)  
print(Bankfull.bag)

##   
## Call:  
## randomForest(formula = y..category ~ ., data = bankfull\_train, mtry = 15, importance = TRUE, na.action = na.omit)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 15  
##   
## OOB estimate of error rate: 10.94%  
## Confusion matrix:  
## no yes class.error  
## no 31049 883 0.02765251  
## yes 3073 1164 0.72527732

importance(Bankfull.bag) #

## no yes MeanDecreaseAccuracy  
## age..number 87.355129 -4.705515 85.218236  
## job..category 67.102769 -12.392482 59.172343  
## marital..category 40.894908 1.291384 39.244137  
## education..category 49.848605 -7.713502 48.096282  
## default..category 16.979833 -1.089667 16.954513  
## balance..number 19.279585 4.869645 20.060503  
## housing..category 50.693595 14.694509 57.600466  
## loan..category 5.284106 11.369190 9.755525  
## contact..category 95.441150 -15.514590 101.715420  
## day..number 103.461838 -25.020405 104.496200  
## month..category 149.816835 -18.653537 154.175444  
## campaign..number 32.068022 10.507085 34.904531  
## pdays..number 43.643777 27.393813 56.915679  
## previous..number 19.454827 -4.345604 19.486397  
## poutcome..category 47.726713 100.970020 78.127320  
## MeanDecreaseGini  
## age..number 1086.75414  
## job..category 572.39621  
## marital..category 144.01651  
## education..category 213.27724  
## default..category 15.56974  
## balance..number 1683.43242  
## housing..category 121.29738  
## loan..category 64.88782  
## contact..category 106.49376  
## day..number 951.78902  
## month..category 699.75977  
## campaign..number 456.87344  
## pdays..number 407.55746  
## previous..number 190.51393  
## poutcome..category 762.28941

varImpPlot(Bankfull.bag)



actual3 <- bankfull\_test$y..category   
predicted3 <- predict(Bankfull.bag, bankfull\_test, type="class")   
Bankfull\_results.matrix.bag <- confusionMatrix(predicted3, actual3, positive="yes")   
print("Bagged results")

## [1] "Bagged results"

print(Bankfull\_results.matrix.bag)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 7786 777  
## yes 204 275  
##   
## Accuracy : 0.8915   
## 95% CI : (0.8849, 0.8978)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.009873   
##   
## Kappa : 0.3089   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.26141   
## Specificity : 0.97447   
## Pos Pred Value : 0.57411   
## Neg Pred Value : 0.90926   
## Prevalence : 0.11635   
## Detection Rate : 0.03041   
## Detection Prevalence : 0.05298   
## Balanced Accuracy : 0.61794   
##   
## 'Positive' Class : yes   
##

print("Best pruned tree")

## [1] "Best pruned tree"

print(bankfull\_results.matrix.2)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 7880 870  
## yes 110 182  
##   
## Accuracy : 0.8916   
## 95% CI : (0.885, 0.898)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.009025   
##   
## Kappa : 0.232   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.17300   
## Specificity : 0.98623   
## Pos Pred Value : 0.62329   
## Neg Pred Value : 0.90057   
## Prevalence : 0.11635   
## Detection Rate : 0.02013   
## Detection Prevalence : 0.03229   
## Balanced Accuracy : 0.57962   
##   
## 'Positive' Class : yes   
##

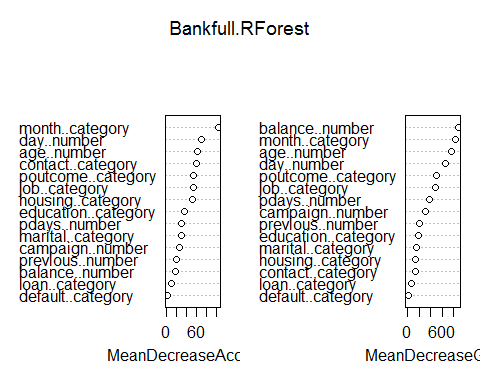
######################################################################################################  
  
ptm <- proc.time()  
Bankfull.RForest <- randomForest(y..category~., data=bankfull\_train, mtry=3, ntree=600,na.action = na.omit, importance=TRUE)   
print(Bankfull.RForest)

##   
## Call:  
## randomForest(formula = y..category ~ ., data = bankfull\_train, mtry = 3, ntree = 600, importance = TRUE, na.action = na.omit)   
## Type of random forest: classification  
## Number of trees: 600  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 10.51%  
## Confusion matrix:  
## no yes class.error  
## no 31380 552 0.01728673  
## yes 3248 989 0.76658013

importance(Bankfull.RForest)

## no yes MeanDecreaseAccuracy  
## age..number 58.253692 4.39325361 61.078003  
## job..category 57.096987 -7.85925042 54.671033  
## marital..category 26.632966 8.82076699 29.990415  
## education..category 37.222368 -0.99431253 36.864953  
## default..category 3.029608 0.04011965 3.069654  
## balance..number 14.029937 10.30986100 17.359370  
## housing..category 44.448580 15.25374548 52.058446  
## loan..category 1.077220 18.06481248 9.578854  
## contact..category 56.367551 -1.66124977 58.917721  
## day..number 70.374735 -14.91910598 69.921255  
## month..category 99.883972 -1.08883781 103.681908  
## campaign..number 21.865437 13.10419223 25.769689  
## pdays..number 27.908889 20.97684032 30.618202  
## previous..number 19.605149 14.50324123 20.023854  
## poutcome..category 39.453357 10.64458968 54.749022  
## MeanDecreaseGini  
## age..number 755.95610  
## job..category 490.23347  
## marital..category 149.32120  
## education..category 191.64929  
## default..category 13.71908  
## balance..number 890.33866  
## housing..category 138.32499  
## loan..category 67.56889  
## contact..category 137.39134  
## day..number 650.18344  
## month..category 834.74743  
## campaign..number 306.35856  
## pdays..number 379.66430  
## previous..number 201.60661  
## poutcome..category 504.58226

varImpPlot(Bankfull.RForest)



proc.time() - ptm

## user system elapsed   
## 74.98 0.64 75.73

actual4 <- bankfull\_test$y..category   
Bankfull.RForest\_predict<-predict(Bankfull.RForest, bankfull\_test,type="response")   
Bankfull.RForest\_results.matrix <- confusionMatrix(Bankfull.RForest\_predict, actual4,positive="yes")   
print("Random Forest results")

## [1] "Random Forest results"

print(Bankfull.RForest\_results.matrix)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 7854 819  
## yes 136 233  
##   
## Accuracy : 0.8944   
## 95% CI : (0.8879, 0.9006)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.0006764   
##   
## Kappa : 0.2847   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.22148   
## Specificity : 0.98298   
## Pos Pred Value : 0.63144   
## Neg Pred Value : 0.90557   
## Prevalence : 0.11635   
## Detection Rate : 0.02577   
## Detection Prevalence : 0.04081   
## Balanced Accuracy : 0.60223   
##   
## 'Positive' Class : yes   
##

print("Previous best bagging results")

## [1] "Previous best bagging results"

print(Bankfull\_results.matrix.bag)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 7786 777  
## yes 204 275  
##   
## Accuracy : 0.8915   
## 95% CI : (0.8849, 0.8978)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.009873   
##   
## Kappa : 0.3089   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.26141   
## Specificity : 0.97447   
## Pos Pred Value : 0.57411   
## Neg Pred Value : 0.90926   
## Prevalence : 0.11635   
## Detection Rate : 0.03041   
## Detection Prevalence : 0.05298   
## Balanced Accuracy : 0.61794   
##   
## 'Positive' Class : yes   
##

######################################################################################################  
  
setwd("C:/Users/Jennifer/Documents/ADM/HW 5")  
bankfull<-read.csv("bankfull.csv")  
  
bankfull3<-bankfull[,c(17,1,2,10,11,16)]  
  
set.seed(123)  
bankfull\_rand2 <- bankfull3[order(runif(45211)), ]   
bankfull\_train2 <- bankfull\_rand2[1:36169, ]   
bankfull\_test2 <-bankfull\_rand2[36170:45211, ]  
  
bankfull\_train\_logit<-bankfull\_train2  
bankfull.logit <- glm(bankfull\_train\_logit$y..category~., data=bankfull\_train\_logit, family=binomial())   
summary(bankfull.logit)

##   
## Call:  
## glm(formula = bankfull\_train\_logit$y..category ~ ., family = binomial(),   
## data = bankfull\_train\_logit)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4957 -0.4560 -0.3911 -0.3354 2.5558   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.280280 0.117063 -10.937 < 2e-16 \*\*\*  
## age..number -0.004568 0.001919 -2.380 0.017313 \*   
## job..categoryblue-collar -0.332551 0.069618 -4.777 1.78e-06 \*\*\*  
## job..categoryentrepreneur -0.158904 0.121936 -1.303 0.192515   
## job..categoryhousemaid -0.334995 0.132937 -2.520 0.011737 \*   
## job..categorymanagement 0.196961 0.063499 3.102 0.001923 \*\*   
## job..categoryretired 0.620029 0.093459 6.634 3.26e-11 \*\*\*  
## job..categoryself-employed 0.080559 0.105637 0.763 0.445702   
## job..categoryservices -0.143591 0.082826 -1.734 0.082979 .   
## job..categorystudent 0.844082 0.106655 7.914 2.49e-15 \*\*\*  
## job..categorytechnician -0.014977 0.068329 -0.219 0.826498   
## job..categoryunemployed 0.329965 0.106699 3.092 0.001985 \*\*   
## job..categoryunknown 0.128209 0.215815 0.594 0.552467   
## day..number -0.001704 0.002356 -0.723 0.469599   
## month..categoryaug -0.716005 0.075241 -9.516 < 2e-16 \*\*\*  
## month..categorydec 0.630082 0.180982 3.481 0.000499 \*\*\*  
## month..categoryfeb -0.338469 0.086739 -3.902 9.53e-05 \*\*\*  
## month..categoryjan -1.028196 0.125181 -8.214 < 2e-16 \*\*\*  
## month..categoryjul -0.685315 0.074185 -9.238 < 2e-16 \*\*\*  
## month..categoryjun -0.622385 0.078513 -7.927 2.24e-15 \*\*\*  
## month..categorymar 1.278905 0.120709 10.595 < 2e-16 \*\*\*  
## month..categorymay -1.090554 0.067808 -16.083 < 2e-16 \*\*\*  
## month..categorynov -0.806928 0.083079 -9.713 < 2e-16 \*\*\*  
## month..categoryoct 0.803352 0.107449 7.477 7.63e-14 \*\*\*  
## month..categorysep 0.724266 0.119095 6.081 1.19e-09 \*\*\*  
## poutcome..categoryother 0.224017 0.089048 2.516 0.011880 \*   
## poutcome..categorysuccess 2.413286 0.081684 29.544 < 2e-16 \*\*\*  
## poutcome..categoryunknown -0.189955 0.056527 -3.360 0.000778 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 26128 on 36168 degrees of freedom  
## Residual deviance: 22529 on 36141 degrees of freedom  
## AIC: 22585  
##   
## Number of Fisher Scoring iterations: 5

odds<-exp(cbind(Odds\_Ratio=coef(bankfull.logit)))  
odds

## Odds\_Ratio  
## (Intercept) 0.2779594  
## age..number 0.9954427  
## job..categoryblue-collar 0.7170924  
## job..categoryentrepreneur 0.8530784  
## job..categoryhousemaid 0.7153415  
## job..categorymanagement 1.2176960  
## job..categoryretired 1.8589812  
## job..categoryself-employed 1.0838926  
## job..categoryservices 0.8662419  
## job..categorystudent 2.3258427  
## job..categorytechnician 0.9851341  
## job..categoryunemployed 1.3909197  
## job..categoryunknown 1.1367904  
## day..number 0.9982976  
## month..categoryaug 0.4887006  
## month..categorydec 1.8777649  
## month..categoryfeb 0.7128612  
## month..categoryjan 0.3576516  
## month..categoryjul 0.5039315  
## month..categoryjun 0.5366629  
## month..categorymar 3.5927029  
## month..categorymay 0.3360303  
## month..categorynov 0.4462266  
## month..categoryoct 2.2330128  
## month..categorysep 2.0632155  
## poutcome..categoryother 1.2510926  
## poutcome..categorysuccess 11.1706046  
## poutcome..categoryunknown 0.8269963

prob<-odds/(1+odds)  
prob

## Odds\_Ratio  
## (Intercept) 0.2175025  
## age..number 0.4988581  
## job..categoryblue-collar 0.4176201  
## job..categoryentrepreneur 0.4603574  
## job..categoryhousemaid 0.4170257  
## job..categorymanagement 0.5490816  
## job..categoryretired 0.6502250  
## job..categoryself-employed 0.5201288  
## job..categoryservices 0.4641638  
## job..categorystudent 0.6993243  
## job..categorytechnician 0.4962557  
## job..categoryunemployed 0.5817509  
## job..categoryunknown 0.5320084  
## day..number 0.4995740  
## month..categoryaug 0.3282733  
## month..categorydec 0.6525081  
## month..categoryfeb 0.4161815  
## month..categoryjan 0.2634340  
## month..categoryjul 0.3350761  
## month..categoryjun 0.3492392  
## month..categorymar 0.7822633  
## month..categorymay 0.2515140  
## month..categorynov 0.3085454  
## month..categoryoct 0.6906910  
## month..categorysep 0.6735457  
## poutcome..categoryother 0.5557713  
## poutcome..categorysuccess 0.9178348  
## poutcome..categoryunknown 0.4526535

anova(bankfull.logit,test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: bankfull\_train\_logit$y..category  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 36168 26128   
## age..number 1 18.27 36167 26110 1.918e-05 \*\*\*  
## job..category 11 639.48 36156 25471 < 2.2e-16 \*\*\*  
## day..number 1 23.65 36155 25447 1.155e-06 \*\*\*  
## month..category 11 1418.36 36144 24029 < 2.2e-16 \*\*\*  
## poutcome..category 3 1499.96 36141 22529 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

bankfull\_test\_logit<-bankfull\_test2  
  
bankfull\_test\_logit$predict.y<-predict(bankfull.logit, newdata=bankfull\_test\_logit)  
bankfull\_test\_logit$predict.y<-predict(bankfull.logit, newdata=bankfull\_test\_logit,type = "response")  
  
bankfull\_test\_logit\_CI<-cbind(bankfull\_test\_logit,predict(bankfull.logit, newdata=bankfull\_test\_logit,type="link",se=TRUE))  
  
bankfull\_test\_logit\_CI <- within(bankfull\_test\_logit\_CI,   
 {  
 PredictedProb <- plogis(fit)  
 LL <- plogis(fit - (1.96 \* se.fit))  
 UL <- plogis(fit + (1.96 \* se.fit))  
 })   
  
summary(bankfull\_test\_logit\_CI)

## y..category age..number job..category day..number   
## no :7990 Min. :18.00 blue-collar:1967 Min. : 1.00   
## yes:1052 1st Qu.:33.00 management :1855 1st Qu.: 9.00   
## Median :39.00 technician :1470 Median :16.00   
## Mean :41.05 admin. :1045 Mean :15.85   
## 3rd Qu.:48.00 services : 861 3rd Qu.:21.00   
## Max. :95.00 retired : 449 Max. :31.00   
## (Other) :1395   
## month..category poutcome..category predict.y fit   
## may :2731 failure: 995 Min. :0.03911 Min. :-3.201   
## jul :1378 other : 350 1st Qu.:0.06161 1st Qu.:-2.723   
## aug :1312 success: 292 Median :0.08198 Median :-2.416   
## jun :1053 unknown:7405 Mean :0.11509 Mean :-2.263   
## nov : 805 3rd Qu.:0.10633 3rd Qu.:-2.129   
## apr : 564 Max. :0.93798 Max. : 2.716   
## (Other):1199   
## se.fit residual.scale UL LL   
## Min. :0.05306 Min. :1 Min. :0.04481 Min. :0.03088   
## 1st Qu.:0.06605 1st Qu.:1 1st Qu.:0.07166 1st Qu.:0.05321   
## Median :0.07581 Median :1 Median :0.09425 Median :0.07056   
## Mean :0.08433 Mean :1 Mean :0.13120 Mean :0.10066   
## 3rd Qu.:0.09631 3rd Qu.:1 3rd Qu.:0.12406 3rd Qu.:0.09342   
## Max. :0.23878 Max. :1 Max. :0.95204 Max. :0.92013   
##   
## PredictedProb   
## Min. :0.03911   
## 1st Qu.:0.06161   
## Median :0.08198   
## Mean :0.11509   
## 3rd Qu.:0.10633   
## Max. :0.93798   
##

dim(bankfull\_test\_logit\_CI)

## [1] 9042 13

bankfull\_test\_logit\_CI$predict.y2<-ifelse(bankfull\_test\_logit\_CI$predict.y>.5, "yes", "no")  
probs<-bankfull\_test\_logit\_CI[,c(1,14)]   
  
  
library(gmodels)

## Warning: package 'gmodels' was built under R version 3.3.3

CrossTable(x=probs$y, y=probs$predict.y2, prob.chisq=FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | Chi-square contribution |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 9042   
##   
##   
## | probs$predict.y2   
## probs$y | no | yes | Row Total |   
## -------------|-----------|-----------|-----------|  
## no | 7887 | 103 | 7990 |   
## | 2.530 | 80.662 | |   
## | 0.987 | 0.013 | 0.884 |   
## | 0.900 | 0.375 | |   
## | 0.872 | 0.011 | |   
## -------------|-----------|-----------|-----------|  
## yes | 880 | 172 | 1052 |   
## | 19.217 | 612.636 | |   
## | 0.837 | 0.163 | 0.116 |   
## | 0.100 | 0.625 | |   
## | 0.097 | 0.019 | |   
## -------------|-----------|-----------|-----------|  
## Column Total | 8767 | 275 | 9042 |   
## | 0.970 | 0.030 | |   
## -------------|-----------|-----------|-----------|  
##   
##

TP = 170  
TN = 7884  
FP = 106  
FN = 882  
Sensitivity = TP/(TP+FN) #true positive rate; recall; TP/(TP+FN)  
Specificity = TN/(TN+FP) #how often is the prediction negative when actual is negative?  
Precision = TP/(TP+FP) #how often is prediction positive when actual is positive?  
Accuracy = (TP+TN)/(TP+TN+FP+FN) #how often is classifier correct  
Value<-round(c(TP,TN,FP,FN,Sensitivity,Specificity,Precision,Accuracy),digits=3)  
Measure<-c("True Positive","True Negative","False Positive","False Negative","Sensitivity=TP/(TN+FP)",  
 "Specificity=TN/(TN+TP)","Precision=TP/(TP+FP)","Accuracy=(TP+TN)/total")  
table<-as.data.frame(cbind(Measure,Value))  
library(knitr)

## Warning: package 'knitr' was built under R version 3.3.3

kable(table)

|  |  |
| --- | --- |
| Measure | Value |
| True Positive | 170 |
| True Negative | 7884 |
| False Positive | 106 |
| False Negative | 882 |
| Sensitivity=TP/(TN+FP) | 0.162 |
| Specificity=TN/(TN+TP) | 0.987 |
| Precision=TP/(TP+FP) | 0.616 |
| Accuracy=(TP+TN)/total | 0.891 |