HW2\_Trees.R

Jennifer

Tue Mar 28 14:50:07 2017

setwd("C:/Users/Jennifer/Documents/ADM/HW 2")  
Churn\_Calls<-read.csv("Churn\_Calls.csv")  
  
set.seed(123)  
Churn\_Calls\_rand <- Churn\_Calls[order(runif(5000)), ]   
Churn\_Calls\_train <- Churn\_Calls\_rand[1:4000, ]   
Churn\_Calls\_test <-Churn\_Calls\_rand[4001:5000, ]  
  
library(rpart)  
library(rpart.plot)

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

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

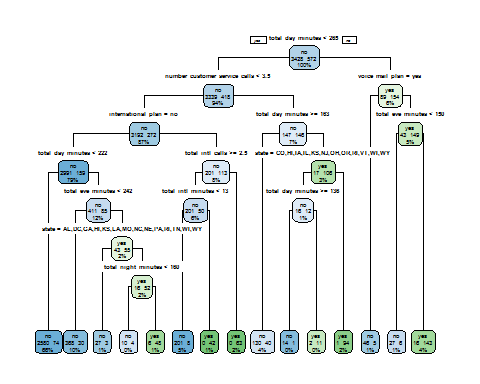
library(partykit)

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

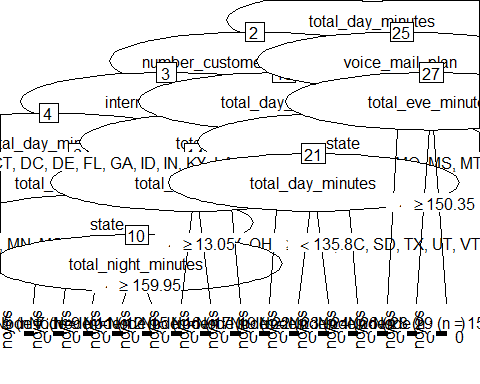
##   
## Attaching package: 'partykit'

## 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

Churn\_Calls\_rpart <- rpart(Churn\_Calls\_train$churn~., method="class", parms = list(split="gini"), data=Churn\_Calls\_train)  
rpart.plot(Churn\_Calls\_rpart, type=1, extra=101)



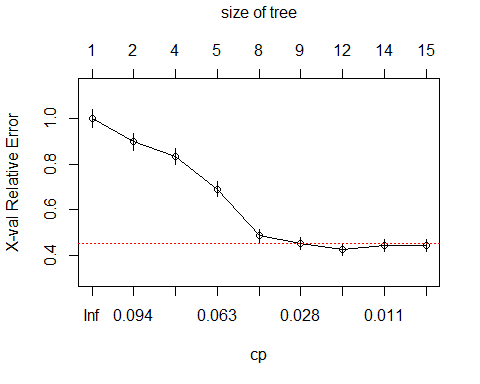
Churn\_Calls\_party<-as.party(Churn\_Calls\_rpart)  
plot(Churn\_Calls\_party)



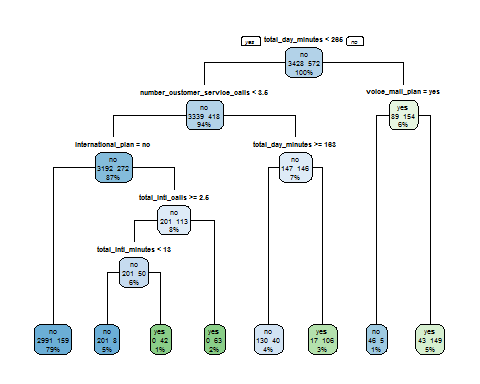
cptable<-printcp(Churn\_Calls\_rpart)

##   
## Classification tree:  
## rpart(formula = Churn\_Calls\_train$churn ~ ., data = Churn\_Calls\_train,   
## method = "class", parms = list(split = "gini"))  
##   
## Variables actually used in tree construction:  
## [1] international\_plan number\_customer\_service\_calls  
## [3] state total\_day\_minutes   
## [5] total\_eve\_minutes total\_intl\_calls   
## [7] total\_intl\_minutes total\_night\_minutes   
## [9] voice\_mail\_plan   
##   
## Root node error: 572/4000 = 0.143  
##   
## n= 4000   
##   
## CP nsplit rel error xerror xstd  
## 1 0.113636 0 1.00000 1.00000 0.038707  
## 2 0.077797 1 0.88636 0.89685 0.036971  
## 3 0.071678 3 0.73077 0.83217 0.035801  
## 4 0.055070 4 0.65909 0.69056 0.032986  
## 5 0.036713 7 0.47552 0.48776 0.028165  
## 6 0.020979 8 0.43881 0.45455 0.027258  
## 7 0.011364 11 0.37587 0.42832 0.026513  
## 8 0.010490 13 0.35315 0.44580 0.027013  
## 9 0.010000 14 0.34266 0.44580 0.027013

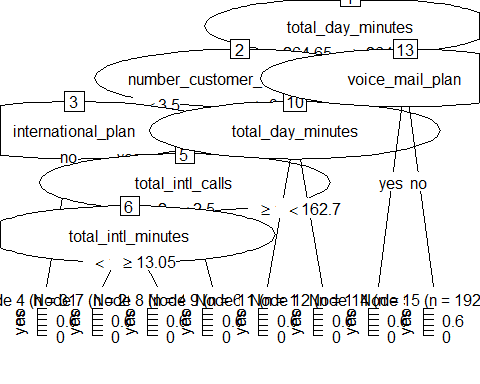
plotcp(Churn\_Calls\_rpart, minline=TRUE, col="red")



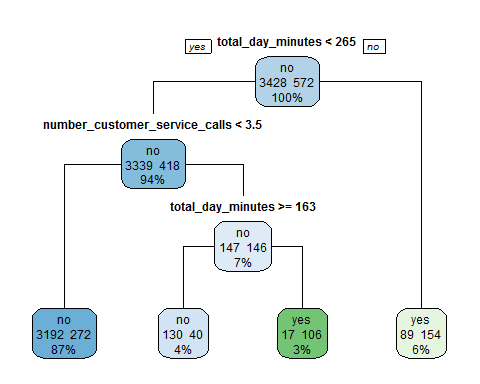
Pruned\_Churn\_Calls\_rpart <-prune(Churn\_Calls\_rpart,cp=.045, minsplit=10, minbucket=round(minsplit/3)) # Going with 4 splits  
rpart.plot(Pruned\_Churn\_Calls\_rpart, type=1, extra=101)



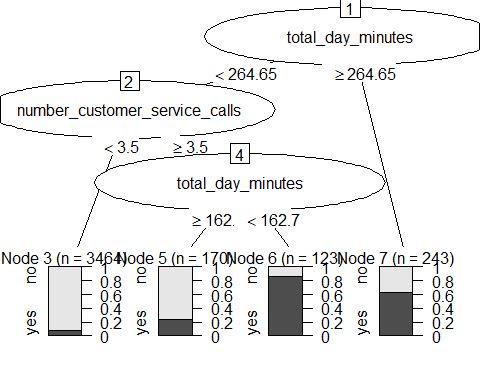
Pruned\_Churn\_Calls\_party<-as.party(Pruned\_Churn\_Calls\_rpart)  
plot(Pruned\_Churn\_Calls\_party)



Pruned\_Churn\_Calls\_rpart2 <-prune(Churn\_Calls\_rpart,cp=.075, minsplit=10, minbucket=round(minsplit/3)) # Going with 4 splits  
rpart.plot(Pruned\_Churn\_Calls\_rpart2, type=1, extra=101)



Pruned\_Churn\_Calls\_party2<-as.party(Pruned\_Churn\_Calls\_rpart2)  
plot(Pruned\_Churn\_Calls\_party2)



library(caret)

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

## Loading required package: lattice

## Loading required package: ggplot2

actual <- Churn\_Calls\_test$churn  
predicted <- predict(Churn\_Calls\_rpart, Churn\_Calls\_test, type="class")  
results.matrix <- confusionMatrix(predicted, actual, positive="yes")  
print(results.matrix)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 844 43  
## yes 21 92  
##   
## Accuracy : 0.936   
## 95% CI : (0.919, 0.9504)  
## No Information Rate : 0.865   
## P-Value [Acc > NIR] : 4.327e-13   
##   
## Kappa : 0.7057   
## Mcnemar's Test P-Value : 0.008665   
##   
## Sensitivity : 0.6815   
## Specificity : 0.9757   
## Pos Pred Value : 0.8142   
## Neg Pred Value : 0.9515   
## Prevalence : 0.1350   
## Detection Rate : 0.0920   
## Detection Prevalence : 0.1130   
## Balanced Accuracy : 0.8286   
##   
## 'Positive' Class : yes   
##

actual <- Churn\_Calls\_test$churn  
predicted <- predict(Pruned\_Churn\_Calls\_rpart, Churn\_Calls\_test, type="class")  
results.matrix <- confusionMatrix(predicted, actual, positive="yes")  
print(results.matrix)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 845 47  
## yes 20 88  
##   
## Accuracy : 0.933   
## 95% CI : (0.9157, 0.9477)  
## No Information Rate : 0.865   
## P-Value [Acc > NIR] : 4.861e-12   
##   
## Kappa : 0.6867   
## Mcnemar's Test P-Value : 0.001491   
##   
## Sensitivity : 0.6519   
## Specificity : 0.9769   
## Pos Pred Value : 0.8148   
## Neg Pred Value : 0.9473   
## Prevalence : 0.1350   
## Detection Rate : 0.0880   
## Detection Prevalence : 0.1080   
## Balanced Accuracy : 0.8144   
##   
## 'Positive' Class : yes   
##

actual <- Churn\_Calls\_test$churn  
predicted <- predict(Pruned\_Churn\_Calls\_rpart2, Churn\_Calls\_test, type="class")  
results.matrix <- confusionMatrix(predicted, actual, positive="yes")  
print(results.matrix)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 823 76  
## yes 42 59  
##   
## Accuracy : 0.882   
## 95% CI : (0.8604, 0.9013)  
## No Information Rate : 0.865   
## P-Value [Acc > NIR] : 0.061408   
##   
## Kappa : 0.4347   
## Mcnemar's Test P-Value : 0.002382   
##   
## Sensitivity : 0.4370   
## Specificity : 0.9514   
## Pos Pred Value : 0.5842   
## Neg Pred Value : 0.9155   
## Prevalence : 0.1350   
## Detection Rate : 0.0590   
## Detection Prevalence : 0.1010   
## Balanced Accuracy : 0.6942   
##   
## 'Positive' Class : yes   
##

HW2\_KNN.R

Jennifer

Wed Mar 29 17:59:44 2017

setwd("C:/Users/Jennifer/Documents/ADM/HW 2")  
Churn\_Calls<-read.csv("Churn\_Calls.csv")  
  
Churn\_Calls<-Churn\_Calls[,c(20,4:5, 2:3, 6:19, 1)]   
dim(Churn\_Calls)

## [1] 5000 20

names(Churn\_Calls)

## [1] "churn" "international\_plan"   
## [3] "voice\_mail\_plan" "account\_length"   
## [5] "area\_code" "number\_vmail\_messages"   
## [7] "total\_day\_minutes" "total\_day\_calls"   
## [9] "total\_day\_charge" "total\_eve\_minutes"   
## [11] "total\_eve\_calls" "total\_eve\_charge"   
## [13] "total\_night\_minutes" "total\_night\_calls"   
## [15] "total\_night\_charge" "total\_intl\_minutes"   
## [17] "total\_intl\_calls" "total\_intl\_charge"   
## [19] "number\_customer\_service\_calls" "state"

table(Churn\_Calls$churn)

##   
## no yes   
## 4293 707

Churn\_Calls$international\_plan <-ifelse(Churn\_Calls$international\_plan=="no",1,0)  
Churn\_Calls$voice\_mail\_plan <-ifelse(Churn\_Calls$voice\_mail\_plan=="no",1,0)   
library(stringr)  
Churn\_Calls$area\_code <- str\_sub(Churn\_Calls$area\_code,-3,-1)  
usage<-Churn\_Calls[,4:19]  
usage$area\_code<- as.numeric(factor(usage$area\_code))  
dim(usage)

## [1] 5000 16

normalize<- function(x){return((x-min(x))/(max(x)-min(x)))}  
usage\_n<-as.data.frame(lapply(usage, normalize))  
dim(usage\_n)

## [1] 5000 16

summary(usage$number\_vmail\_messages)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 0.000 0.000 7.755 17.000 52.000

summary(usage\_n$number\_vmail\_messages)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.1491 0.3269 1.0000

Churn\_Calls\_n<-cbind(Churn\_Calls[,c(1:3)], usage\_n[complete.cases(usage\_n),])  
dim(Churn\_Calls\_n)

## [1] 5000 19

set.seed(123)  
Churn\_Calls\_n\_rand <- Churn\_Calls\_n[order(runif(5000)), ]   
Churn\_Calls\_n\_train <- Churn\_Calls\_n\_rand[1:4000,2:19]  
Churn\_Calls\_n\_test <- Churn\_Calls\_n\_rand[4001:5000,2:19]  
Churn\_Calls\_n\_train\_labels<-Churn\_Calls\_n\_rand[1:4000,1]  
Churn\_Calls\_n\_test\_labels<-Churn\_Calls\_n\_rand[4001:5000,1]  
  
sqrt(5000)

## [1] 70.71068

library(class)  
set.seed(123)  
Churn\_Calls\_pred\_knn<-knn(train=Churn\_Calls\_n\_train, test=Churn\_Calls\_n\_test, cl=Churn\_Calls\_n\_train\_labels, k=71)  
library(gmodels)

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

CrossTable(x=Churn\_Calls\_n\_test\_labels, y=Churn\_Calls\_pred\_knn, prob.chisq=FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | Chi-square contribution |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 1000   
##   
##   
## | Churn\_Calls\_pred\_knn   
## Churn\_Calls\_n\_test\_labels | no | yes | Row Total |   
## --------------------------|-----------|-----------|-----------|  
## no | 865 | 0 | 865 |   
## | 0.056 | 6.920 | |   
## | 1.000 | 0.000 | 0.865 |   
## | 0.872 | 0.000 | |   
## | 0.865 | 0.000 | |   
## --------------------------|-----------|-----------|-----------|  
## yes | 127 | 8 | 135 |   
## | 0.358 | 44.339 | |   
## | 0.941 | 0.059 | 0.135 |   
## | 0.128 | 1.000 | |   
## | 0.127 | 0.008 | |   
## --------------------------|-----------|-----------|-----------|  
## Column Total | 992 | 8 | 1000 |   
## | 0.992 | 0.008 | |   
## --------------------------|-----------|-----------|-----------|  
##   
##

TP = 8  
TN = 865  
FP = 0  
FN = 127  
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 | 8 |
| True Negative | 865 |
| False Positive | 0 |
| False Negative | 127 |
| Sensitivity=TP/(TN+FP) | 0.059 |
| Specificity=TN/(TN+TP) | 1 |
| Precision=TP/(TP+FP) | 1 |
| Accuracy=(TP+TN)/total | 0.873 |

library(class)  
set.seed(123)  
Churn\_Calls\_pred\_knn2<-knn(train=Churn\_Calls\_n\_train, test=Churn\_Calls\_n\_test, cl=Churn\_Calls\_n\_train\_labels, k=11)  
library(gmodels)  
CrossTable(x=Churn\_Calls\_n\_test\_labels, y=Churn\_Calls\_pred\_knn2, prob.chisq=FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | Chi-square contribution |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 1000   
##   
##   
## | Churn\_Calls\_pred\_knn2   
## Churn\_Calls\_n\_test\_labels | no | yes | Row Total |   
## --------------------------|-----------|-----------|-----------|  
## no | 859 | 6 | 865 |   
## | 1.110 | 25.321 | |   
## | 0.993 | 0.007 | 0.865 |   
## | 0.897 | 0.143 | |   
## | 0.859 | 0.006 | |   
## --------------------------|-----------|-----------|-----------|  
## yes | 99 | 36 | 135 |   
## | 7.113 | 162.241 | |   
## | 0.733 | 0.267 | 0.135 |   
## | 0.103 | 0.857 | |   
## | 0.099 | 0.036 | |   
## --------------------------|-----------|-----------|-----------|  
## Column Total | 958 | 42 | 1000 |   
## | 0.958 | 0.042 | |   
## --------------------------|-----------|-----------|-----------|  
##   
##

TP = 36  
TN = 859  
FP = 6  
FN = 99  
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)  
kable(table)

|  |  |
| --- | --- |
| Measure | Value |
| True Positive | 36 |
| True Negative | 859 |
| False Positive | 6 |
| False Negative | 99 |
| Sensitivity=TP/(TN+FP) | 0.267 |
| Specificity=TN/(TN+TP) | 0.993 |
| Precision=TP/(TP+FP) | 0.857 |
| Accuracy=(TP+TN)/total | 0.895 |

HW2\_GLM.R

Jennifer

Wed Mar 29 18:09:08 2017

setwd("C:/Users/Jennifer/Documents/ADM/HW 2")  
Churn\_Calls<-read.csv("Churn\_Calls.csv")  
  
set.seed(123)  
Churn\_Calls\_rand <- Churn\_Calls[order(runif(5000)), ]   
Churn\_Calls\_train <- Churn\_Calls\_rand[1:4000, ]   
Churn\_Calls\_test <-Churn\_Calls\_rand[4001:5000, ]  
  
Churn\_Calls\_train\_logit<-Churn\_Calls\_train[,c(20,7,5,19,4,17,16)]  
Churn\_Calls\_train\_logit$total\_day\_minutes<-ifelse(Churn\_Calls\_train\_logit$total\_day\_minutes<265,"no","yes")  
Churn\_Calls\_train\_logit$number\_customer\_service\_calls<-ifelse(Churn\_Calls\_train\_logit$number\_customer\_service\_calls<3.5,"no","yes")  
Churn\_Calls\_train\_logit$total\_intl\_calls<-ifelse(Churn\_Calls\_train\_logit$total\_intl\_calls<2.5,"no","yes")  
Churn\_Calls\_train\_logit$total\_intl\_minutes<-ifelse(Churn\_Calls\_train\_logit$total\_intl\_minutes<13.5,"no","yes")  
names(Churn\_Calls\_train\_logit)

## [1] "churn" "total\_day\_minutes"   
## [3] "voice\_mail\_plan" "number\_customer\_service\_calls"  
## [5] "international\_plan" "total\_intl\_calls"   
## [7] "total\_intl\_minutes"

Churn\_Calls.logit <- glm(Churn\_Calls\_train\_logit$churn~., data=Churn\_Calls\_train\_logit, family=binomial()) #Fit a logistic regression  
summary(Churn\_Calls.logit) #coefficients are presented as log-odds (probabilities on logit scale)

##   
## Call:  
## glm(formula = Churn\_Calls\_train\_logit$churn ~ ., family = binomial(),   
## data = Churn\_Calls\_train\_logit)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.2161 -0.4581 -0.3173 -0.1885 2.8422   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.1447 0.1164 -18.422 < 2e-16 \*\*\*  
## total\_day\_minutesyes 3.1653 0.1656 19.109 < 2e-16 \*\*\*  
## voice\_mail\_planyes -1.0575 0.1465 -7.217 5.32e-13 \*\*\*  
## number\_customer\_service\_callsyes 2.7687 0.1471 18.823 < 2e-16 \*\*\*  
## international\_planyes 2.1958 0.1414 15.527 < 2e-16 \*\*\*  
## total\_intl\_callsyes -0.8190 0.1240 -6.603 4.04e-11 \*\*\*  
## total\_intl\_minutesyes 0.7621 0.1511 5.043 4.58e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3283 on 3999 degrees of freedom  
## Residual deviance: 2312 on 3993 degrees of freedom  
## AIC: 2326  
##   
## Number of Fisher Scoring iterations: 6

odds<-exp(cbind(Odds\_Ratio=coef(Churn\_Calls.logit))) #Take exponent of log odds gives "odds" ratio.  
odds

## Odds\_Ratio  
## (Intercept) 0.1171066  
## total\_day\_minutesyes 23.6955256  
## voice\_mail\_planyes 0.3473148  
## number\_customer\_service\_callsyes 15.9373861  
## international\_planyes 8.9874757  
## total\_intl\_callsyes 0.4408532  
## total\_intl\_minutesyes 2.1428251

prob<-odds/(1+odds)  
prob

## Odds\_Ratio  
## (Intercept) 0.1048302  
## total\_day\_minutesyes 0.9595068  
## voice\_mail\_planyes 0.2577830  
## number\_customer\_service\_callsyes 0.9409590  
## international\_planyes 0.8998746  
## total\_intl\_callsyes 0.3059668  
## total\_intl\_minutesyes 0.6818149

anova(Churn\_Calls.logit,test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: Churn\_Calls\_train\_logit$churn  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)  
## NULL 3999 3283.0   
## total\_day\_minutes 1 331.63 3998 2951.3 < 2.2e-16  
## voice\_mail\_plan 1 42.49 3997 2908.9 7.110e-11  
## number\_customer\_service\_calls 1 311.59 3996 2597.3 < 2.2e-16  
## international\_plan 1 219.83 3995 2377.4 < 2.2e-16  
## total\_intl\_calls 1 41.48 3994 2336.0 1.191e-10  
## total\_intl\_minutes 1 23.93 3993 2312.0 1.002e-06  
##   
## NULL   
## total\_day\_minutes \*\*\*  
## voice\_mail\_plan \*\*\*  
## number\_customer\_service\_calls \*\*\*  
## international\_plan \*\*\*  
## total\_intl\_calls \*\*\*  
## total\_intl\_minutes \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Churn\_Calls\_test\_logit<-Churn\_Calls\_test[,c(20,7,5,19,4,17,16)]  
Churn\_Calls\_test\_logit$total\_day\_minutes<-ifelse(Churn\_Calls\_test$total\_day\_minutes<265,"no","yes")  
Churn\_Calls\_test\_logit$number\_customer\_service\_calls<-ifelse(Churn\_Calls\_test$number\_customer\_service\_calls<3.5,"no","yes")  
Churn\_Calls\_test\_logit$total\_intl\_calls<-ifelse(Churn\_Calls\_test$total\_intl\_calls<2.5,"no","yes")  
Churn\_Calls\_test\_logit$total\_intl\_minutes<-ifelse(Churn\_Calls\_test$total\_intl\_minutes<13.5,"no","yes")  
  
Churn\_Calls\_test\_logit$predict.churn<-predict(Churn\_Calls.logit, newdata=Churn\_Calls\_test\_logit)  
Churn\_Calls\_test\_logit$predict.churn<-predict(Churn\_Calls.logit, newdata=Churn\_Calls\_test\_logit,type = "response")  
  
Churn\_Calls\_test\_logit\_CI<-cbind(Churn\_Calls\_test\_logit,predict(Churn\_Calls.logit, newdata=Churn\_Calls\_test\_logit,type="link",se=TRUE))  
  
Churn\_Calls\_test\_logit\_CI <- within(Churn\_Calls\_test\_logit\_CI,   
 {  
 PredictedProb <- plogis(fit)  
 LL <- plogis(fit - (1.96 \* se.fit))  
 UL <- plogis(fit + (1.96 \* se.fit))  
 })   
  
summary(Churn\_Calls\_test\_logit\_CI)

## churn total\_day\_minutes voice\_mail\_plan  
## no :865 Length:1000 no :705   
## yes:135 Class :character yes:295   
## Mode :character   
##   
##   
##   
## number\_customer\_service\_calls international\_plan total\_intl\_calls   
## Length:1000 no :899 Length:1000   
## Class :character yes:101 Class :character   
## Mode :character Mode :character   
##   
##   
##   
## total\_intl\_minutes predict.churn fit se.fit   
## Length:1000 Min. :0.01761 Min. :-4.021 Min. :0.09375   
## Class :character 1st Qu.:0.04909 1st Qu.:-2.964 1st Qu.:0.09375   
## Mode :character Median :0.04909 Median :-2.964 Median :0.12930   
## Mean :0.15142 Mean :-2.334 Mean :0.13353   
## 3rd Qu.:0.10483 3rd Qu.:-2.145 3rd Qu.:0.16100   
## Max. :0.98956 Max. : 4.551 Max. :0.28282   
## residual.scale UL LL PredictedProb   
## Min. :1 Min. :0.02399 Min. :0.01291 Min. :0.01761   
## 1st Qu.:1 1st Qu.:0.05842 1st Qu.:0.04119 1st Qu.:0.04909   
## Median :1 Median :0.05842 Median :0.04119 Median :0.04909   
## Mean :1 Mean :0.17654 Mean :0.12852 Mean :0.15142   
## 3rd Qu.:1 3rd Qu.:0.12956 3rd Qu.:0.08527 3rd Qu.:0.10483   
## Max. :1 Max. :0.99397 Max. :0.98196 Max. :0.98956

dim(Churn\_Calls\_test\_logit\_CI)

## [1] 1000 14

Churn\_Calls\_test\_logit\_CI$predict.churn2<-ifelse(Churn\_Calls\_test\_logit\_CI$predict.churn>.5, "yes", "no")  
probs<-Churn\_Calls\_test\_logit\_CI[,c(1,15)]   
  
  
library(gmodels)

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

CrossTable(x=probs$churn, y=probs$predict.churn2, prob.chisq=FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | Chi-square contribution |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 1000   
##   
##   
## | probs$predict.churn2   
## probs$churn | no | yes | Row Total |   
## -------------|-----------|-----------|-----------|  
## no | 839 | 26 | 865 |   
## | 2.644 | 29.211 | |   
## | 0.970 | 0.030 | 0.865 |   
## | 0.915 | 0.313 | |   
## | 0.839 | 0.026 | |   
## -------------|-----------|-----------|-----------|  
## yes | 78 | 57 | 135 |   
## | 16.941 | 187.165 | |   
## | 0.578 | 0.422 | 0.135 |   
## | 0.085 | 0.687 | |   
## | 0.078 | 0.057 | |   
## -------------|-----------|-----------|-----------|  
## Column Total | 917 | 83 | 1000 |   
## | 0.917 | 0.083 | |   
## -------------|-----------|-----------|-----------|  
##   
##

TP = 57  
TN = 839  
FP = 26  
FN = 78  
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 | 57 |
| True Negative | 839 |
| False Positive | 26 |
| False Negative | 78 |
| Sensitivity=TP/(TN+FP) | 0.422 |
| Specificity=TN/(TN+TP) | 0.97 |
| Precision=TP/(TP+FP) | 0.687 |
| Accuracy=(TP+TN)/total | 0.896 |