Final_Project

Saksham Dixit

May 3, 2017

PART 1

```
# This part conatins data exploration, calculations and graphs

data(churn)
train <- churnTrain
test <- churnTest

churnTot <- rbind(churnTrain,churnTest)
churnYes <- churnTot[churnTot$churn == 'yes',]
churnNo <-  churnTot[churnTot$churn == 'no',]

churnRate <- (length(churnTot[churnTot$churn == 'yes',20])
                                /length(churnTot$churn))

df <- data.frame(c(length(churnYes$churn),length(churnNo$churn)),
                 c('Churn','Continuing'))
colnames(df) <- c('number','class')

#pie chart of customer churn

slices <- df$number
lbls <- df$class
prct <- round(slices/sum(slices)*100)
lbls <- paste(lbls, prct)
lbls <- paste(lbls,"%",sep="")

acLengthNO <- mean(churnNo$account_length)
acLengthYes <- mean(churnYes$account_length)

# percentage churn by state

states <- levels(churnYes$state)
yesCount <- NA
noCount <- NA
churnPerc <- NA
for(i in seq_len(length(states))){
        yes <- NA
        no <- NA

        yes <- ifelse (churnYes$state == states[i],1,0)
        no <- ifelse (churnNo$state == states[i],1,0)
        yesCount[i] <- sum(yes)
        noCount[i] <- sum(no)
        churnPerc[i] <- yesCount[i]/(sum(yesCount[i],noCount[i]))
}
```

```
churnState <- data.frame(states,churnPerc)
colnames(churnState) <- c('State','Churn_Perc')

gChurnstatePerc <- ggplot(churnState,aes(churnState$State,churnState$Churn_Perc))+
        geom_bar(stat = 'identity',size = churnState$Churn_Perc)+
        labs(title ='Percentage Churn By State', x = 'State', y = 'Percentage Churn')




# percentage of customer churning with international plan

intPlanCont <- ifelse(churnNo$international_plan == 'yes',1,0)
intPlanContPerc <- sum(intPlanCont)/length(churnTot$international_plan)

intPlanChurn <- ifelse(churnYes$international_plan == 'yes',1,0)
intPlanChurnPerc <- sum(intPlanChurn)/length(churnTot$international_plan)

intPlanTot <- ifelse(churnTot$international_plan == 'yes',1,0)
intPlanPerc <- sum(intPlanTot)/length(churnTot$international_plan)

intPlanDf <- data.frame(c(intPlanPerc,intPlanContPerc,intPlanChurnPerc),
                        c('Total % customers With International Plan',
                          '% customers chruning','% customers continuing'))
colnames(intPlanDf) <- c('a','b')

gIntPlan <- ggplot(intPlanDf,aes(intPlanDf$b,intPlanDf$a))+
        geom_bar(stat = 'identity')+
        labs(title ='Percentage Churn By International Plan', x = 'Customers with International Plan', y



# Customer Service calls

ChurnCCC <- sum(churnYes$number_customer_service_calls)/length(churnYes$churn)
ChurnNoCCC <- sum(churnNo$number_customer_service_calls)/length(churnNo$churn)
dfCCC <- data.frame(c(ChurnCCC,ChurnNoCCC),c('Churning Customers','Continuing Customer'))

gChurnCCC <- ggplot(dfCCC,aes(dfCCC[,2],dfCCC[,1]))+
        geom_bar(stat = 'identity')+
        labs(title ='Average Customer Service calls',
             x = 'Customers', y = 'Average Customer Service Calls')

#Voice Mail Messages

churnVM <- sum(churnYes$number_vmail_messages)/
        length(churnYes[churnYes$voice_mail_plan == 'yes',5])
churnNoVM <- sum(churnNo$number_vmail_messages)/
        length(churnNo[churnNo$voice_mail_plan == 'yes',5])

dfVM <- data.frame(c(churnVM,churnNoVM),c('Churning Customers','Continuing Customer'))
```

```r
gChurnVM <- ggplot(dfVM,aes(dfVM[,2],dfVM[,1]))+
        geom_bar(stat = 'identity')+
        labs(title ='Average Number of Voice Mail Messages ',
             x = 'Customers', y = 'Number of Voice Mails')


#earnings from customers
earnings <- sum(churnTot$total_day_charge+churnTot$total_eve_charge+churnTot$total_night_charge
                +churnTot$total_intl_charge)
earningsPerCust <- earnings/length(churnTot$churn)

#international charges from customers
 intEarnAvg <- sum(churnTot$total_intl_charge)/length(churnTot$churn)
 intEarnNoAvg <- sum(churnNo$total_intl_charge)/length(churnNo$churn)
 intEarnChurnAvg <- sum(churnYes$total_intl_charge)/length(churnYes$churn)

 #day charges
 dayEarnAvg <- sum(churnTot$total_day_charge)/length(churnTot$churn)
 dayEarnNoAvg <- sum(churnNo$total_day_charge)/length(churnNo$churn)
 dayEarnChurnAvg <- sum(churnYes$total_day_charge)/length(churnYes$churn)

 #evening charges
 eveEarnAvg <- sum(churnTot$total_eve_charge)/length(churnTot$churn)
 eveEarnNoAvg <- sum(churnNo$total_eve_charge)/length(churnNo$churn)
 eveEarnChurnAvg <- sum(churnYes$total_eve_charge)/length(churnYes$churn)

 #night charges
 nightEarnAvg <- sum(churnTot$total_night_charge)/length(churnTot$churn)
 nightEarnNoAvg <- sum(churnNo$total_night_charge)/length(churnNo$churn)
 nightEarnChurnAvg <- sum(churnYes$total_night_charge)/length(churnYes$churn)

#day minutes
 dayMntsAvg <- sum(churnTot$total_day_minutes)/length(churnTot$churn)
 dayMntsNoAvg<- sum(churnNo$total_day_minutes)/length(churnNo$churn)
 daysMntsChurnAvg <- sum(churnYes$total_day_minutes)/length(churnYes$churn)

#international minutes

 intMntsAvg <- sum(churnTot$total_intl_minutes)/length(churnTot$churn)
 intMntsNoAvg<- sum(churnNo$total_intl_minutes)/length(churnNo$churn)
 intsMntsChurnAvg <- sum(churnYes$total_intl_minutes)/length(churnYes$churn)
```

The dataset provided contains 19 Predictors and a dependent variable that tells whether the customer would churn or not

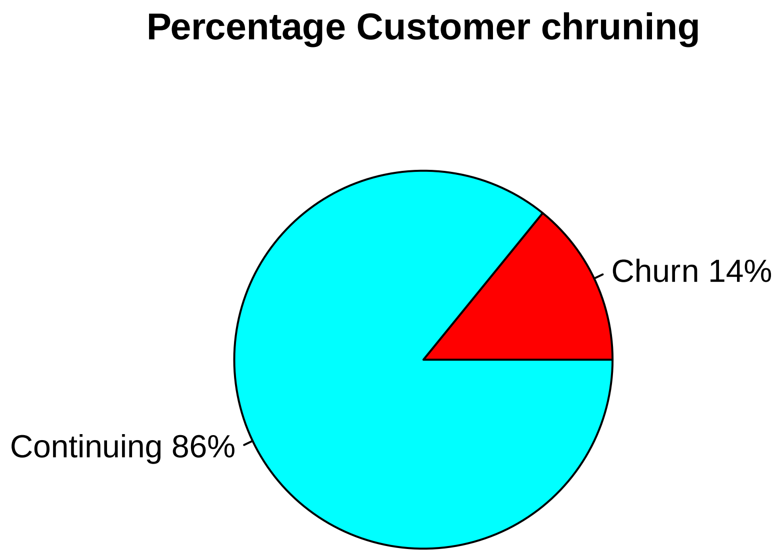
The categorical predictors are: State of the customer, the area code internationalplan, voice mail plan

The nummerical predictors are: account_length, area_cod, number_vmail_messages total_day_minutes, total_day_calls, total_day_charge, total_eve_minutes, total_eve_calls, total_eve_charge, total_night_minutes, total_night_calls, total_night_charge, total_intl_minutes, total_intl_calls, total_intl_charge, number_customer_service_calls,

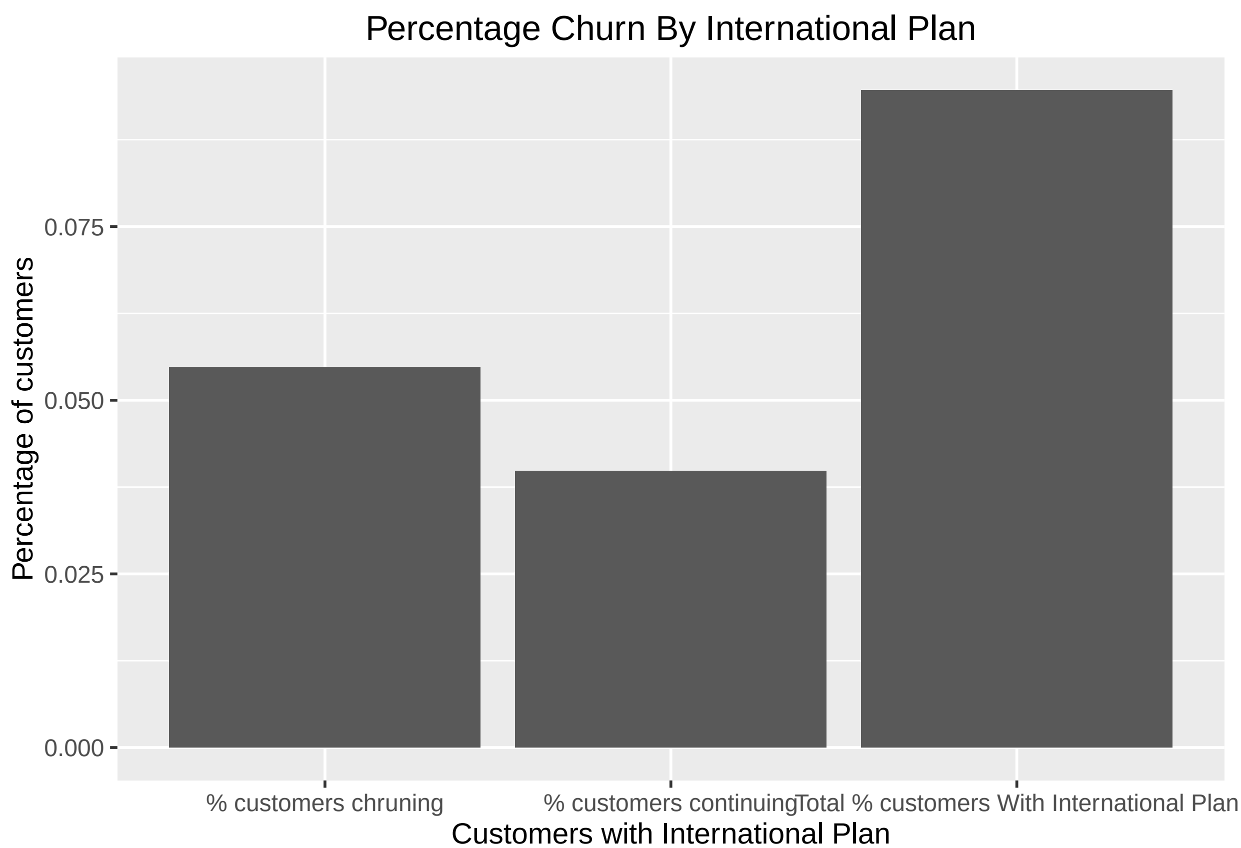
Each record contains customer level details.

The overall churn rate is 0.1414

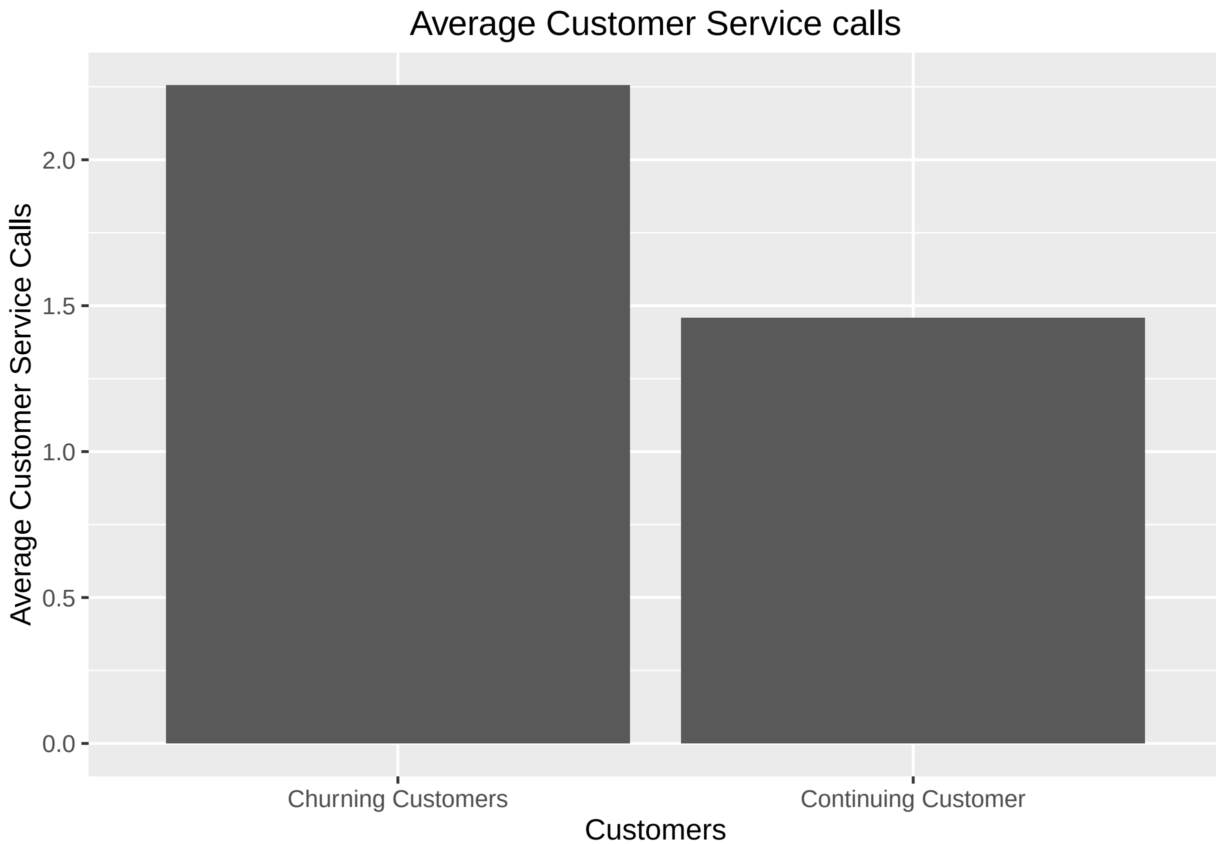
```
#churn Rate
pieChurn <- pie(slices,labels = lbls, col=rainbow(length(lbls)),
        main="Percentage Customer chruning")
```




```
print(gIntPlan)
```

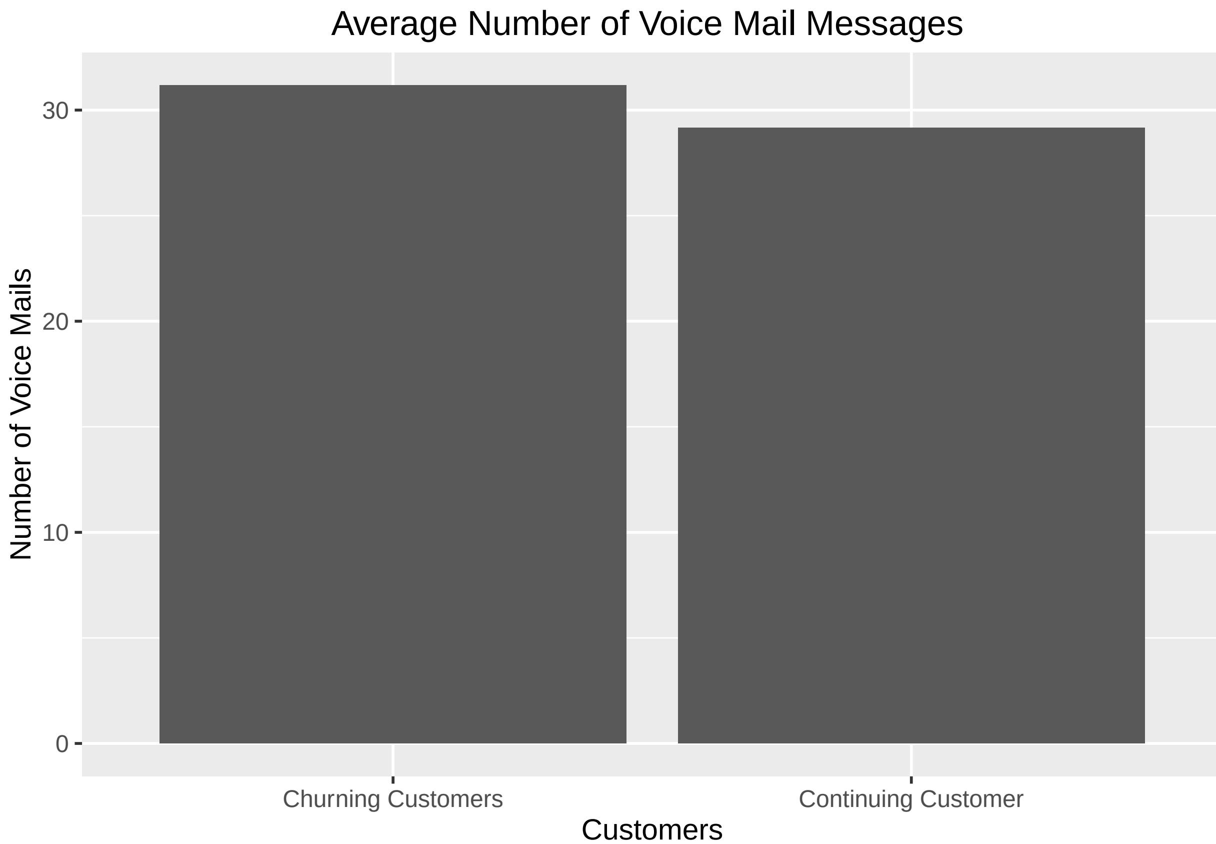



```
print(gChurnCCC)
```



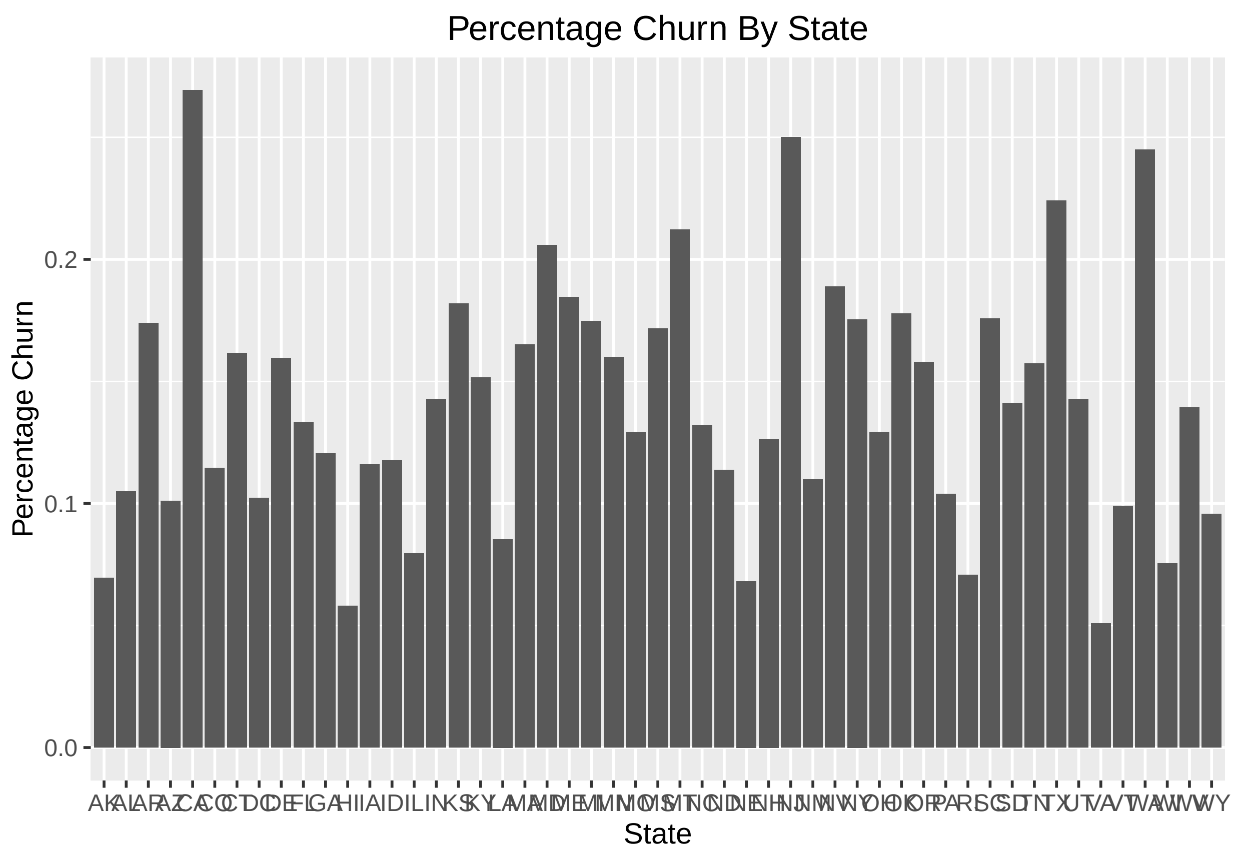
We note that on churning customers have made almost twice the number of calls to customer service

```
print(gChurnVM)
```



The diffrence in the number of voice mail messages of churning and continuing customers is almost negligible

```
print(gChurnstatePerc)
```



We note that California, New Jersey, Texas and Washington have the highest Churn percentage

Creating a Logistic Regression MOdel to predict customers who are going to churn

```
#Convert the levels of factors from string to numeric

train$state <- as.numeric(train$state)
train$state <- as.factor(train$state)

test$state <- as.numeric(test$state)
test$state <- as.factor(test$state)

#New levels area_code_408 = 1, area_code_415 =2 area_code_510 =3
train$area_code <- as.numeric(train$area_code )
train$area_code <- as.factor(train$area_code)

test$area_code <- as.numeric(test$area_code )
test$area_code <- as.factor(test$area_code)

#new levels 'yes' = 1, 'no' = 0
train$international_plan <- as.factor(ifelse(train$international_plan == 'yes',1,0))
train$voice_mail_plan <- as.factor(ifelse(train$voice_mail_plan == 'yes',1,0))
train$churn <- ifelse <- as.factor(ifelse(train$churn == 'yes',1,0))
```

```r
test$international_plan <- as.factor(ifelse(test$international_plan == 'yes',1,0))
test$voice_mail_plan <- as.factor(ifelse(test$voice_mail_plan == 'yes',1,0))

test$churn <- ifelse <- as.factor(ifelse(test$churn == 'yes',1,0))

#creating Logistic Model

logModel <- glm(churn ~ ., family = binomial(),
                        data = train, model = 'TRUE')

# We give 1 to the probability of 0.4 and above to compensate for class imbalance

predictLogProb <- predict(logModel,test[,-20],type = 'response')
predictLogClass  <- as.factor(ifelse( predictLogProb >= 0.4,'1','0'))
logCnm <- confusionMatrix(data = predictLogClass,
                           reference = test$churn)
print(logCnm$overall[1])
```

```
##  Accuracy
## 0.8674265
```

```r
print(logCnm$table)
```

```
##           Reference
## Prediction    0    1
##          0 1365  143
##          1   78   81
```

Observations:

From the Logistic Regression model we created above we the get the significance Values of the predictors. The predictors which are resposible in determing 'churn' are :

Inernational Plan : According to our model the customers with an International Plans are more likely to churn. This suggests that perhaps our international plan is not as competitive enough.

Total internation calls: As a corollary to the above observation, the customers who use our serivce to make international calls are also more likely to move away. This confirms our assumption that our International Plan is failing us.

Number of customer service calls : The customer who make contact our customer service more also ten to churn more. This may suggest that either they are facing issues repeatedly or their serivce/request are not being fullfilled by our representatives, Which is causing them to leave our service

Voice Mail Plan: Our customers who have voice mail plan are also more likely to churn

Number Of Voice Mail Messages: Our customers who get more voice mails seem to churn more This may suggest network availability issues or that those customers have some other provider too, which makes them less available on our service.

Suggestions

Based on these observations I would suggest the following strategies

Our international plan needs a complete overhaul. We need to analyze the plans offered by our competitors and come up with an extremly competitive and attractive one.

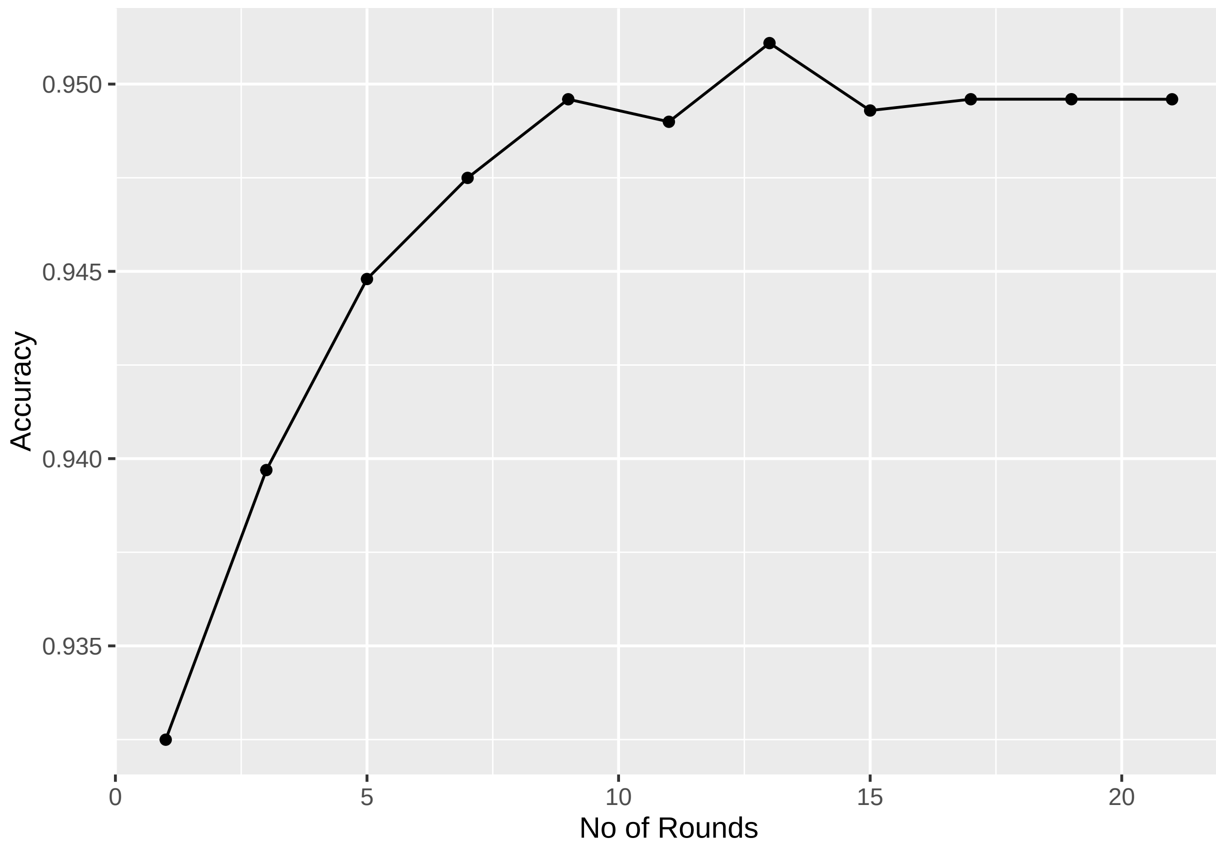
We need to survey our customer service representatives regarding the services and requests most made by the customers and introduce any changes if they seem feasible. We can further train our customer service representative so that they are able to solve our customer's issues in one go.

Or customers with more number of voice mail messages also seem to churn more, we can survey our customers to understand whether there are connectivity issues at some places and rectify these issues if they exist.

Creating a gradient boosting model using XGBoost

```
#function to create boosted trees.
createBT <- function(trainData,testData,n){

        target <- as.numeric(levels(trainData$churn))[trainData$churn]
        boosTree <- xgboost(data = data.matrix(trainData[,-20]), label = target,
                            max.depth = 6,nrounds = n,verbose = 0,
                            objective = 'binary:logistic',eta = 0.3)

        predictBTProb <- predict(boosTree,data.matrix(testData[,-20]),
                                 type = 'response')
        predictBTClass <- as.factor(ifelse( predictBTProb >= 0.5,1,0))

        btCnm <- confusionMatrix(data = predictBTClass,
                                reference = testData$churn)

        acc <- (btCnm$table)
        return(btCnm$overall[1])
}


#3 Fold Cross Validation
set.seed(111)
cv_train_index <- createFolds(seq_len(nrow(train)), k = 3, list = TRUE,
                                                            returnTrain = TRUE)
#Tuning the model
avgAcc <- NA
acc <- NA
k <- 1
for(i in seq(from = 1, to = 21 , by = 2)){
        acc <- NA

        for(j in 1:3){

                cvTrain <- train[cv_train_index[[j]],]
                cvTest <- train[-cv_train_index[[j]],]

                acc[j] <- createBT(cvTrain,cvTest,i)
                cnm <- NA

        }
         avgAcc[k] <- mean(acc)
```

```
        k <- k+1
}

ledger_mat <- data.frame(seq(from = 1, to = 21 , by = 2),avgAcc)

#plotting Accuracy Vs. Number of boosting rounds
ggplot(ledger_mat, aes(ledger_mat[,1],ledger_mat[,2]))+
geom_point()+ geom_line()+xlab('No of Rounds')+ylab('Accuracy')
```




```
bestRounds <- ledger_mat[which.max(ledger_mat[,2]),1]
bestAcc <-  ledger_mat[which.max(ledger_mat[,2]),2]

#build the final model using the tuned number of rounds

target <- as.numeric(levels(train$churn))[train$churn]
finalBoosTree <- xgboost(data = data.matrix(train[,-20]), label = target,
                         max.depth = 6,nrounds = bestRounds,verbose = 0,
                         objective = 'binary:logistic',eta = 0.3)


predictBTProb <- predict(finalBoosTree,data.matrix(test[,-20]), type = 'response')
predictBTClass <- as.factor(ifelse( predictBTProb >= 0.5,1,0))

btCnm <- confusionMatrix(data = predictBTClass,
                               reference = test$churn)
```

```r
print(btCnm$overall[1])
```

```
##  Accuracy
## 0.9586083
```

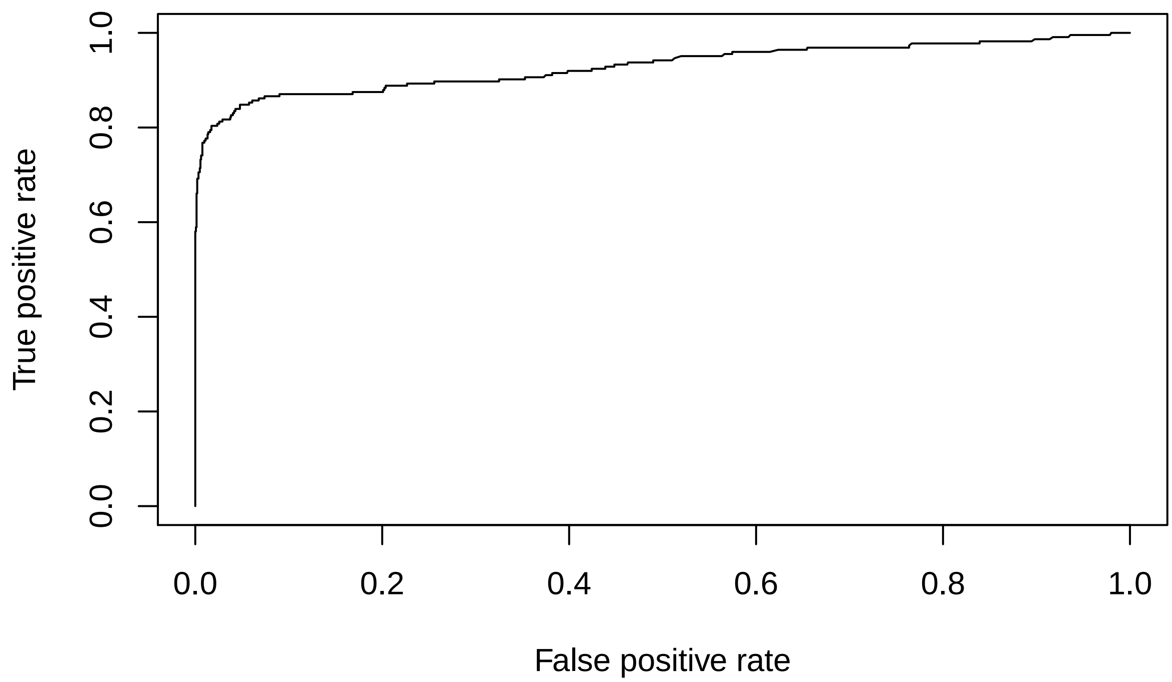
```r
print(btCnm$table)
```

```
##           Reference
## Prediction    0    1
##          0 1435   61
##          1    8  163
```

```r
featureImp <- xgb.importance( feature_names = names(train),model = finalBoosTree)
print(featureImp)
```

```
##                           Feature         Gain        Cover   Frequency
##  1:             total_day_minutes 0.2811607162 0.2960397685 0.194570136
##  2:             total_eve_minutes 0.1604501359 0.1671516993 0.219457014
##  3: number_customer_service_calls 0.1318204415 0.1430118775 0.038461538
##  4:            total_intl_minutes 0.0986518190 0.0246610903 0.074660633
##  5:            international_plan 0.0833215277 0.1380041939 0.054298643
##  6:              total_intl_calls 0.0740491429 0.0139756574 0.040723982
##  7:               voice_mail_plan 0.0703204773 0.0221394977 0.047511312
##  8:           total_night_minutes 0.0525368478 0.0570482489 0.133484163
##  9:             total_night_calls 0.0106752345 0.0183553095 0.040723982
## 10:                account_length 0.0100338552 0.0323077814 0.049773756
## 11:               total_day_calls 0.0091934413 0.0223656781 0.031674208
## 12:                         state 0.0079932011 0.0105503330 0.033936652
## 13:               total_eve_calls 0.0052316368 0.0072392471 0.020361991
## 14:          number_vmail_messages 0.0044565545 0.0468993307 0.018099548
## 15:                     area_code 0.0001049683 0.0002502867 0.002262443
```

```r
prefBT <- performance(prediction(predictBTProb,test$churn),
                measure = "tpr",x.measure = "fpr")
plot(prefBT)
```




```r
#auc
aucBT <- performance(prediction(predictBTProb,test$churn),measure = "auc")
print(c("Area Under the Curve: ", aucBT@y.values[[1]]),quote = FALSE)
```

```
## [1] Area Under the Curve:  0.929180897930896
```

Observations

Average day charges incurred by all the customers = 30.64967 Average daytime minutes consumed by all customers = 180.2889

Average day charges incurred by the customers who will not churn = 29.87749 Average daytime minutes consumed by customers who will not churn = 175.7466

Average day charges incurred by cuatomers who churn = 35.33842 Average daytime minutes consumed by customers who churn = 207.8706

It appears that our customers who churn tend to call more during daytime and are obviously unsatistfied by our rates hence they churn.

*Our current daytime rate is : $0.17 per minute.

58% of our customers who have international plan churn. So wee need to do something about this too.

```r
print("Taking a look at the confusion matrix")
```

```
## [1] "Taking a look at the confusion matrix"
```

```r
print(btCnm$table)
```

```
##           Reference
## Prediction    0    1
##          0 1435   61
##          1    8  163
```

No. of churning customers predicted = 163

No. of churning customers not predicted = 61

No. of continuing customers wrongly predicted as churning = 61

My plan is to give the customers predicted as churning 300 mintues of free daytime calling and 20$ of Google Voice amount
for a contract of 6 months

Assumptions:

Monthy incomming earnings from a customer = $60.44 Monthly Cost of providing service to a customer = .925*60.44 = $55.907 (92.5% of incoming cost)

Monthly earninngs form a single customer = $60.44-$55.907 = 4.533 Targeted Customers wont churn for 6 months due to contract

Note:

Average day minutes consumed by targeted customer per month = 208 Average evening minutes consumed by targeted customer per month = 210 Average day minutes consumed by targeted customer per month = 207

Our investment per targeted customer is 20$ for 6 months and 300 daytime call minutes.

We assume that customer will use up 208 free daytime minutes in the first month and the remaining 92 minutes in the second month. We assume that the customer uses the google voice amount provided for international calls

Total no of customers we targeted are 171 but the calculations below are for the 163 customer who were surely going to churn.

Calculations

1st Month cost for targeted customer: daytime call cost = 0*0.17 = $0.00 evening call cost = 210*0.08 = $16.8 night call cost = 207*0.04 = $8.04

Total incoming from targeted customer in 1st month = 16.8+8.28 = $25.08 Operating costs = $23.19 Total earnings from 1 customer for 1st month = 25.08-23.19 = $1.89 No of targeted Customers = 163 Total earning for 1st month = 163*1.89 = $308.07

2nd Month cost for targeted customer: daytime call cost = 92*0.17 = $15.64 evening call cost = 210*0.08 = $16.8 night call cost = 207*0.04 = $8.28

Total incoming from targeted customer in 2nd month = 16.8+8.28 = $40.72 Operating costs = $37.66 Total earnings from 1 customer for 3rd month = 25.08-23.19 = $3.05 No of targeted Customers = 163 Total earning for 1st month = 163*3.05 = $495.17

3rd Month cost for targeted customer: daytime call cost = 208*0.17 = $35.56 evening call cost = 210*0.08 = $16.8 night call cost = 207*0.04 = $8.04

Total incoming from targeted customer in 1st month = 16.8+8.28 = $60.44 Operating costs = $55.90 Total earnings from 1 customer for 1st month = 25.08-23.19 = $4.533 No of targeted Customers = 163 Total earning for 3rd month = 163*4.533 = $738.879

For the 4th, 5th and 6th month earnings from each customer is the same as that of 3rd month. Earning for 4th month = $738.879 Earning for 5th month = $738.879 Earning for 6th month = $738.879

Monthly Dollar value gained by retaining a customer is $4.533

Total revenue lost on targeting the 8 customers who were not going to churn = $68.52

Total earnings from targeted customer who were going to churn at the end of 5 months is $3021.72.

Total earnings from targeted customer who were going to churn at the end of 6 months is $3760.00

Total revenue lost on targeting the 8 customers who were not going to churn = $68.52.

Total investment is $20*163 = $3420..The 300 free mins are included in calculations. We are guaranteed to meet our breakeven point by the end of sixth month

As I was able to predict 163 out of the 224 customers who were going to churn and was able to comeup with a plan that is profiatable, I conclude that the model is good