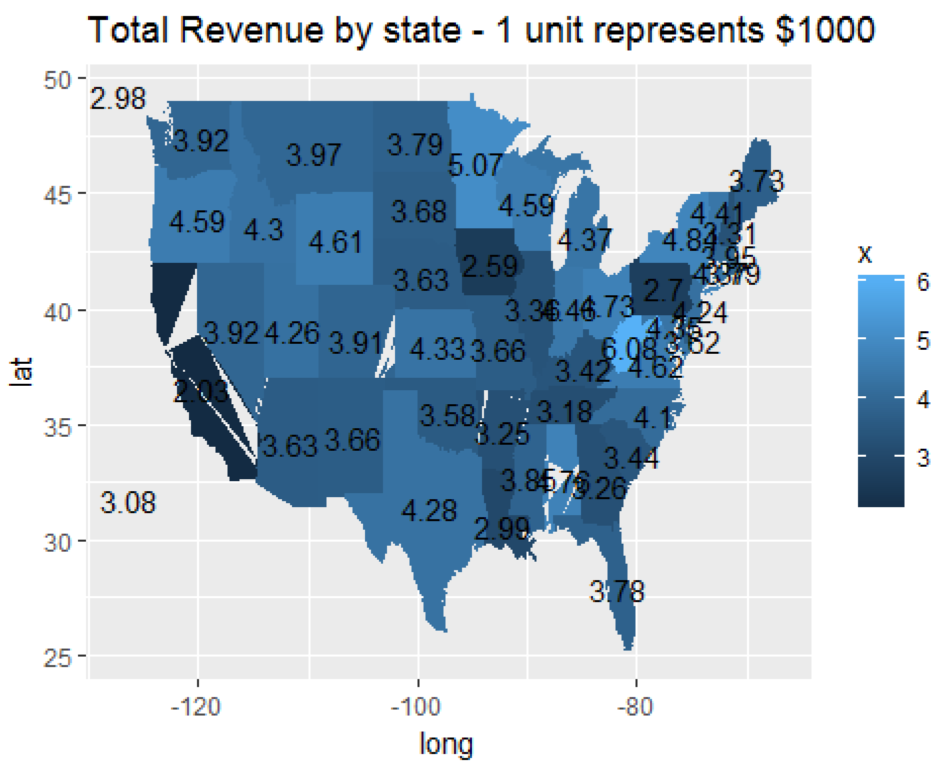
Project

Shivam Tiwari

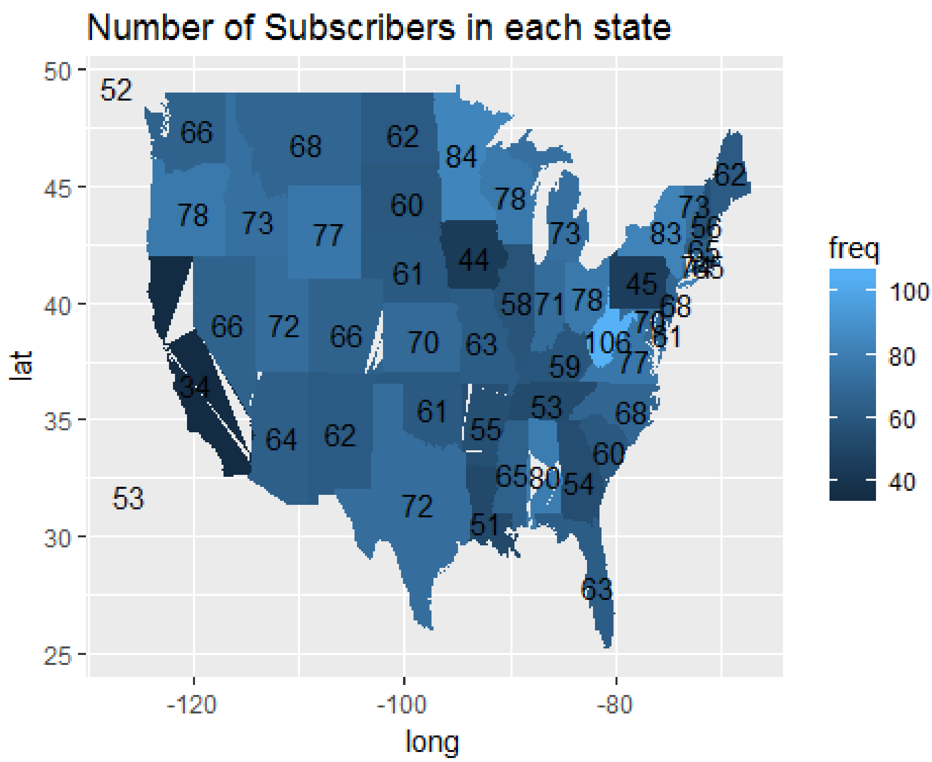
May 6, 2017

```
data(churn)
#Question 1

#The data contains following information about the customer
#1. The State and the area code of the customer
#2. The duration of customer's account with the company (account length)
(most probably in weeks)
#3. Whether the customer has taken an international plan and voice mail plan
or not (one column for each)
#4. The number of voice mail messages the customer has received
#5. Total minutes, total calls and total amount incurred for each customer
overall in the day, evening and night (one column for each)
#6. Total minutes, calls and total amount incurred for each customer in the
international calls
#7. Total number of customer service calls customer has made to the company
#8. Whether the customer churned or not



fullset <- rbind(churnTest, churnTrain)
ch_tot <- sum(fullset$churn == 'yes')
ch_rate <- (ch_tot/nrow(fullset))*100
ch_rate

## [1] 14.14

#Overall Churn rate of the company (including both training and test data) is
14.14%


for (i in 1:nrow(churnTrain)){
  churnTrain$ovrcharge[i] <- sum(churnTrain$total_day_charge[i],
churnTrain$total_eve_charge[i],churnTrain$total_night_charge[i],churnTrain$to
tal_intl_charge[i])
}
state_ovr<- aggregate(churnTrain$ovrcharge, by= list(churnTrain$state), FUN =
'sum')
#US Map for total revenue by state
states <- map_data('state')
state_ovr$x <- state_ovr$x/1000
state_ovr$region <- state.name[match(state_ovr$Group.1,state.abb)]
```

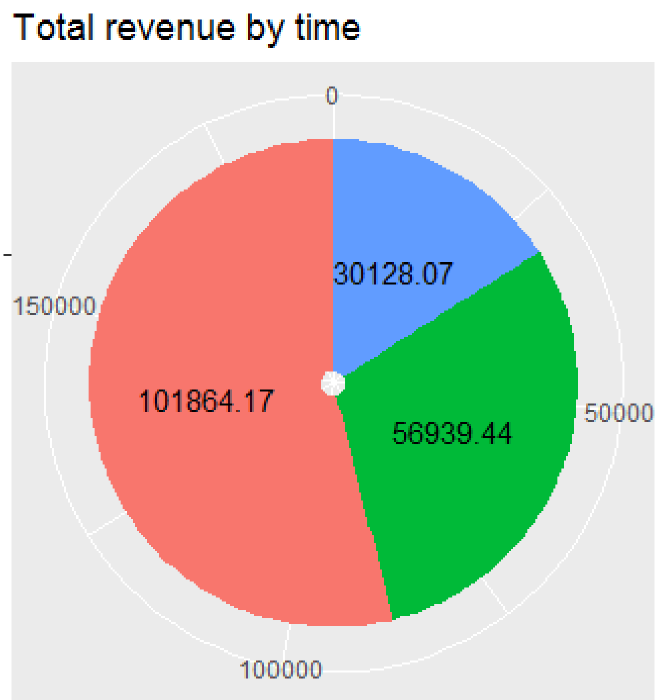
```r
f<- c("region", "x")
usplot <- state_ovr[f]
uss <- na.omit(usplot)
uss$x <- round(uss$x, 2)
uss$region <- tolower(uss$region)
sim_dg <- merge(states,uss, by= 'region')
snames <- data.frame(region=tolower(state.name), long=state.center$x,
lat=state.center$y)
snames <- merge(snames, uss, by='region')
ggplot(sim_dg, aes(long, lat)) + geom_polygon(aes(group=group, fill=x)) +
geom_text(data=snames, aes(long, lat, label=x)) + ggtitle("Total Revenue by
state - 1 unit represents $1000")
```




```r
#US Map for total subscribers by state
subs <- count(churnTrain$state)
subs$x <- state.name[match(subs$x,state.abb)]
subs <- na.omit(subs)
subs$x <- tolower(subs$x)
subs$region <- subs$x
subs <- subs[,-1]
sub_dg <- merge(states,subs, by= 'region')
snames1 <- data.frame(region=tolower(state.name), long=state.center$x,
lat=state.center$y)
snames1 <- merge(snames1, subs, by='region')
ggplot(sub_dg, aes(long, lat)) + geom_polygon(aes(group=group, fill=freq)) +
geom_text(data=snames1, aes(long, lat, label=freq)) + ggtitle("Number of
Subscribers in each state")
```




```
#Pie Chart for total revenue by time (day, evening and night)
day <- sum(churnTrain$total_day_charge)
eve <- sum(churnTrain$total_eve_charge)
night <- sum(churnTrain$total_night_charge)
all<- cbind(day,eve,night)
all <- data.frame(t(all))
all$time <- c('day','eve','night')
ggplot(data = all, aes(x = "", y =t.all., fill = time)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = t.all.), position = position_stack(vjust = 0.5)) +
  coord_polar(theta = "y") + ggtitle("Total revenue by time") +
  labs(x="",y="")
```




```
#Question 2

#For Interpretable model, we use Logit model

#Logit
#checking and removing correlations
churnt <- churnTrain[,-c(1,3,4,5,20,21)]
churnt <- churnt[,-16]
df <- cor(churnt, method = 'pearson')
trainset <- churnTrain[,-c(7,10,13,16,21,22,23)]


#We remove state and area code from the model because they would not
contribute to any strategy that we aim to device for retaining customers
logittrain <- train(x=trainset[,-c(1,3,16,17)], y=trainset$churn, method =
'glm', family = binomial)
summary(logittrain)

##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2626   0.1954   0.3398   0.5120   2.1341
##
```

```
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  8.6558865  0.7242648  11.951  < 2e-16 ***
## account_length              -0.0008334  0.0013913  -0.599 0.549142
## international_planyes       -2.0373143  0.1453553 -14.016  < 2e-16 ***
## voice_mail_planyes           2.0075786  0.5732018   3.502 0.000461 ***
## number_vmail_messages       -0.0353455  0.0179863  -1.965 0.049399 *
## total_day_calls             -0.0032139  0.0027575  -1.166 0.243805
## total_day_charge            -0.0763955  0.0063738 -11.986  < 2e-16 ***
## total_eve_calls             -0.0010730  0.0027805  -0.386 0.699580
## total_eve_charge            -0.0852189  0.0134450  -6.338 2.32e-10 ***
## total_night_calls           -0.0006893  0.0028398  -0.243 0.808206
## total_night_charge          -0.0822072  0.0246791  -3.331 0.000865 ***
## total_intl_calls             0.0920574  0.0250065   3.681 0.000232 ***
## total_intl_charge           -0.3253095  0.0755019  -4.309 1.64e-05 ***
## number_customer_service_calls -0.5137234  0.0392394 -13.092  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2758.3  on 3332  degrees of freedom
## Residual deviance: 2159.7  on 3319  degrees of freedom
## AIC: 2187.7
##
## Number of Fisher Scoring iterations: 6
```

```
logitpredict <- predict(logittrain, churnTest)
confusionMatrix(logitpredict,churnTest$churn)
```

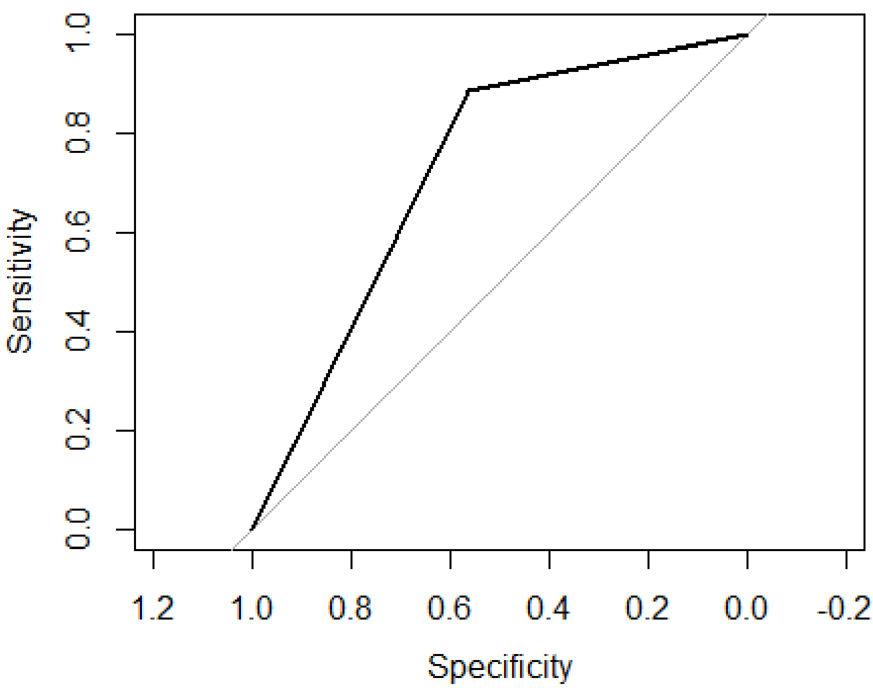
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  yes   no
##        yes   43   33
##        no   181 1410
##
##                Accuracy : 0.8716
##                  95% CI : (0.8546, 0.8873)
##     No Information Rate : 0.8656
##     P-Value [Acc > NIR] : 0.249
##
##                   Kappa : 0.2346
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.19196
##             Specificity : 0.97713
##          Pos Pred Value : 0.56579
##          Neg Pred Value : 0.88624
##              Prevalence : 0.13437
```

```
##          Detection Rate : 0.02579
##    Detection Prevalence : 0.04559
##       Balanced Accuracy : 0.58455
##
##        'Positive' Class : yes
##
```

```r
#Though there was class imbalance, but in our case, a False Positive rate would hurt us more - as we would say that a customer is not churning, but in actual it would.
#Here our specificity is high, so there seems to be no problem
cal_roc <- roc(as.numeric(logitpredict), as.numeric(churnTest$churn))
#Area under curve
auc(cal_roc)
```

```
## Area under the curve: 0.726
```

```r
#ROC plot
plot(cal_roc)
```




```r
#We see there are some significant variables that contribute to churn. Most important of them are:
#1. Those who have taken international plan
#2. Those who have taken a voice mail plan
#3. Total charge incurred by a customer in the day
#4. Total charge incurred by a customer in the evening
#5. Total charge incurred by a customer in the night
```

```
#6. Total International Calls made by each customer
#7. Total charge incurred on international calls
#8. Total number of customer service calls
#9. Total number of voice mail messages

#Looking at the estimates, it seems that if a customer incurs more charge at
the day, evening or night, or if he/she takes our international plan
#he/she is more likely to stay with the customer - This shows that those
customers are LOYAL customers and are happy with our services
#However, if the International Calls (not charge) of a customer are more, he
or she is likely to churn. Maybe then the customer calls and incurs more
charge but finds the rates unreasonable or high (as the customer might have
taken international plan as well - cause it is also a significant variable)
#This suggests that IF we reduce the International rates, we will have a
better chance to retain the customer
#Also, more the number of voice mail messages, more likely the customer is to
churn. This means that the customer has taken voice mail plan but does not
use it. So, he/she might feel that the plan is going waste.
#So, we can ask those specific customers to deactivate the voice mail plan
and increase the chance of retaining them.
#We come up with 2 strategies here -
#1. Decrease International call price
#2. Request those customers who have many voice mails to deactivate the plan



#Question 3

#Random Forest was chosen after running models of Decision trees, Random
Forests and XGBoost
#Random Forest model is provided below


red<- churnTrain
red <- red[,-c(7,10,13,16,20,21,22,23)]
indx <- createFolds(churnTrain$churn, returnTrain = TRUE)
ctrl <- trainControl(method = "cv",summaryFunction = twoClassSummary, index =
indx,classProbs = TRUE, savePredictions = TRUE)
mtryValues <- c(1:5)
set.seed(714)
rfCART <- train(red, churnTrain$churn,
                method = "rf",
                metric = "Kappa",
                ntree = 1000,
                importance = TRUE,
                tuneGrid = data.frame(.mtry = mtryValues),
                trControl = ctrl)

## Loading required package: 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

## Warning in train.default(red, churnTrain$churn, method = "rf", metric =
## "Kappa", : The metric "Kappa" was not in the result set. ROC will be used
## instead.
```

```
summary(rfCART)
```

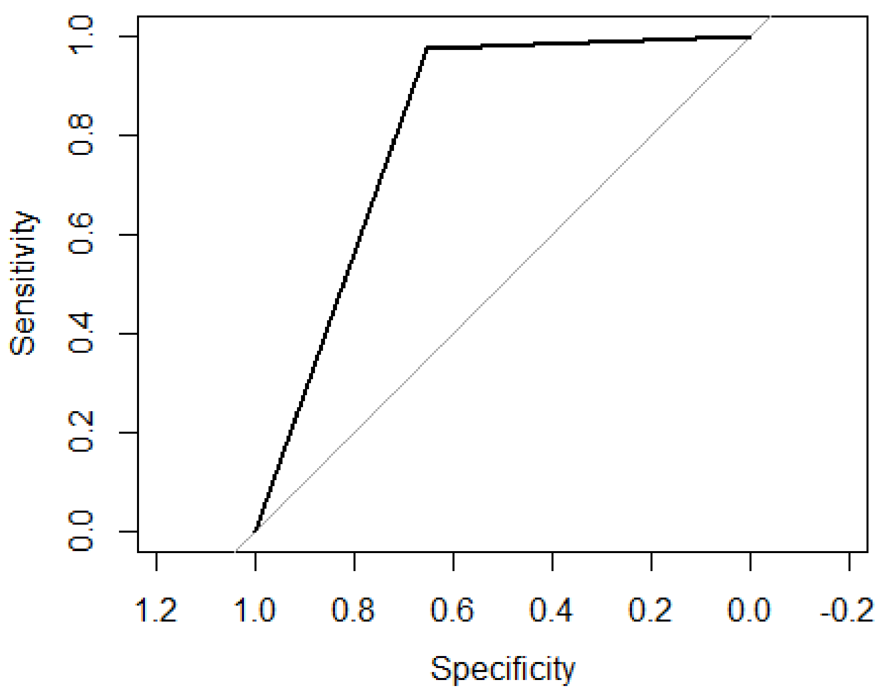
```
##                 Length Class      Mode
## call               6   -none-     call
## type               1   -none-     character
## predicted       3333   factor     numeric
## err.rate        3000   -none-     numeric
## confusion          6   -none-     numeric
## votes           6666   matrix     numeric
## oob.times       3333   -none-     numeric
## classes            2   -none-     character
## importance        60   -none-     numeric
## importanceSD      45   -none-     numeric
## localImportance    0   -none-     NULL
## proximity          0   -none-     NULL
## ntree              1   -none-     numeric
## mtry               1   -none-     numeric
## forest            14   -none-     list
## y               3333   factor     numeric
## test               0   -none-     NULL
## inbag              0   -none-     NULL
## xNames            15   -none-     character
## problemType        1   -none-     character
## tuneValue          1   data.frame list
## obsLevels          2   -none-     character
```

```
rfp <- predict(rfCART,churnTest)
tr <-confusionMatrix(rfp, churnTest$churn)
tr
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  yes   no
##        yes  186   98
```

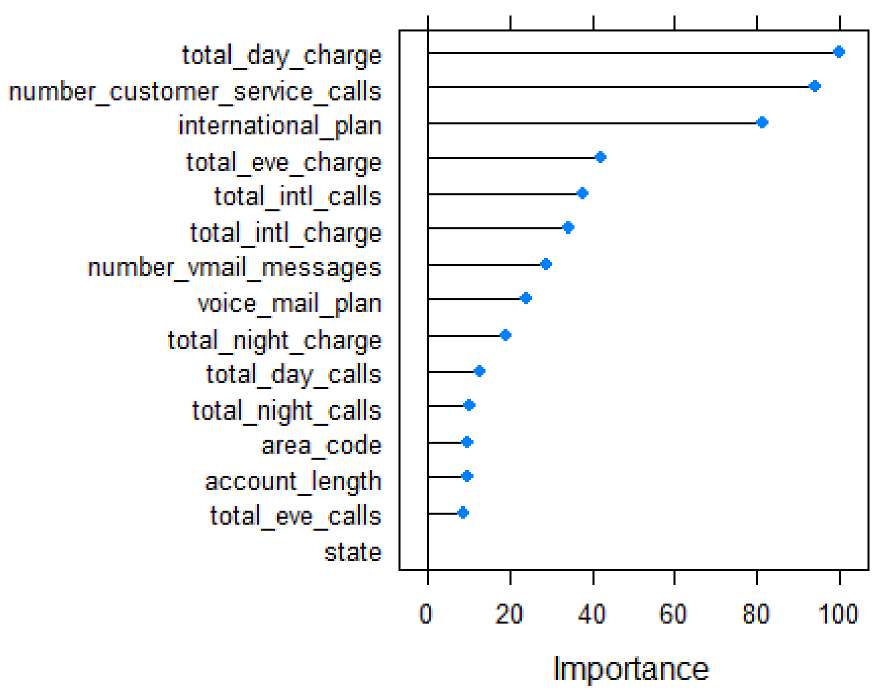
```
##         no    38 1345
##                                           
##                Accuracy : 0.9184          
##                  95% CI : (0.9042, 0.9311)
##     No Information Rate : 0.8656          
##     P-Value [Acc > NIR] : 1.064e-11       
##                                           
##                   Kappa : 0.6849          
##  Mcnemar's Test P-Value : 4.210e-07       
##                                           
##             Sensitivity : 0.8304          
##             Specificity : 0.9321          
##          Pos Pred Value : 0.6549          
##          Neg Pred Value : 0.9725          
##              Prevalence : 0.1344          
##          Detection Rate : 0.1116          
##    Detection Prevalence : 0.1704          
##       Balanced Accuracy : 0.8812          
##                                           
##        'Positive' Class : yes             
##
```

```r
roc_rf <- roc(as.numeric(rfp), sapply(churnTest$churn,as.numeric))
plot(roc_rf)
```




```r
auc(roc_rf)
```

```
## Area under the curve: 0.8137

varImp(rfCART)

## rf variable importance
##
##                               Importance
## total_day_charge                100.000
## number_customer_service_calls    93.878
## international_plan               81.196
## total_eve_charge                 42.309
## total_intl_calls                 37.706
## total_intl_charge                34.160
## number_vmail_messages            28.869
## voice_mail_plan                  24.139
## total_night_charge               19.048
## total_day_calls                  12.557
## total_night_calls                10.561
## area_code                         9.970
## account_length                    9.608
## total_eve_calls                   8.629
## state                             0.000

plot(varImp(rfCART))
```




```
#There are some important variables that we derived from the model -
```

```
#total_day_charge                  100.000

#number_customer_service_calls      89.356

#international_plan                 77.130

#total_eve_charge                   41.712

#total_intl_calls                   36.438

#total_intl_charge                  34.402

#number_vmail_messages              28.609


#Thus, our natural plan would be to focus on those variables.

#We see that total day charge has the most weightage. Also, total evening
charge and total international #charge hold weightage among the charges

#Customer service calls are made if customer faces some issues, so we would
have to work on that – so #that there are lesser issues in future.

#Moreover, as mentioned in answer 2, we would encourage our customers to take
up international #plans because it is important variable and request those
who have many voice mail messages to drop #the voice mail plan.

#As for the financial plan – we can give some rebate on day charges, evening
charges and international #charges.

#Suppose we give a 5% discount on all the above charges - to those customers
that we think will churn.

#The mean revenue of all our customers was $56.72 and the mean of those who
churned was $62.61.

#However, the mean revenue of churn customers for day, evening and
international calls combined was #$56.22 and for those who did not churn had
mean of $49.52.

#Our model has predicted that 284 would churn. Out of them, 186 did actual
churn and 98 actually did not. So, giving discount to those 98 would be a
loss.

#Our model also predicted that 1383 would not churn but out of them, 38 did.
So, those 38 would also #be accounted into loss.

#Calculations:

#Total revenue as by our model – 56.22*284 + 49.52*1383 = 84452.64

#Actual revenue -  56.22*224 + 49.52*1443 = 84050.64

#Anticipated Revenue of churn category After Discount – 0.95*56.22*284 =
15168.56
```

```
#Actual Revenue of the churn category after Discount (as it had non churners too) – 0.95*56.22*186 + #0.95*49.52*98 = 14554.39

#Loss value in churn category –    49.52*98 – 0.95*49.52*98 = 242.65

#Loss from customers who churned and our model couldn’t identify – 38*56.22 = 2136.36

#Thus, our actual profit from our action of giving 5% discount would be – 14554.39-242.65-2136.36 = #$12175.38!!!

#This is considering we will be able to contain all our potential churning customers.

#Out of these, if some still churn at our company churn rate of 14.14%, our profit still would be #$10453.78!!!

#The cost of retaining the customers would be [(56.22*186)+(49.52*98)]-14554.39 = $755.49


#Based on the performance of our model, the plan is very much profitable and can be implemented!!
```