Churn Analysis project report

Task1

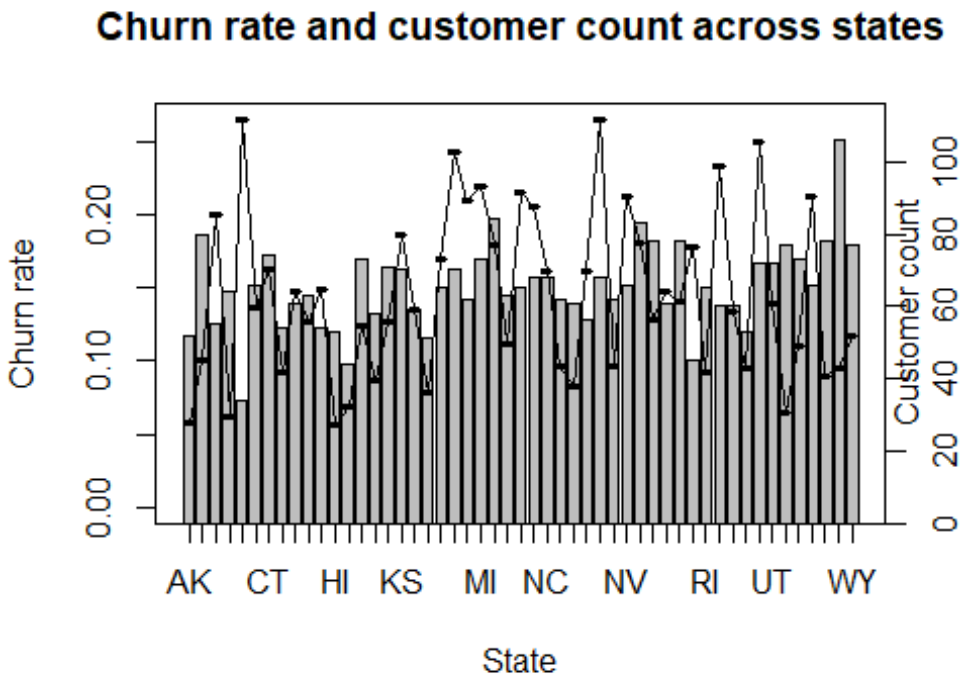
```
# Loading churn data from "C50" package
# install.packages('C50')
library(C50)
data(churn)

# Overall churn rate
table(churnTrain$churn)

##
##  yes   no
##  483 2850

# Descriptive analysis using ggplot
# install.packages('ggplot2')
library(ggplot2)

churnTrain$churn_flag <- ifelse(churnTrain$churn == 'yes',1,0)
data1 <- aggregate(churnTrain$churn_flag,by=list(state=churnTrain$state),FUN
= mean)
data1 <- data.frame(state=data1$state,churnrate=data1$x)

data2 <- aggregate(churnTrain$churn,by=list(state=churnTrain$state),FUN =
length)
data2 <- data.frame(state=data2$state,count=data2$x)
```

The overall churn rate is 14.49%

Descriptive Analysis:

Graph 1:

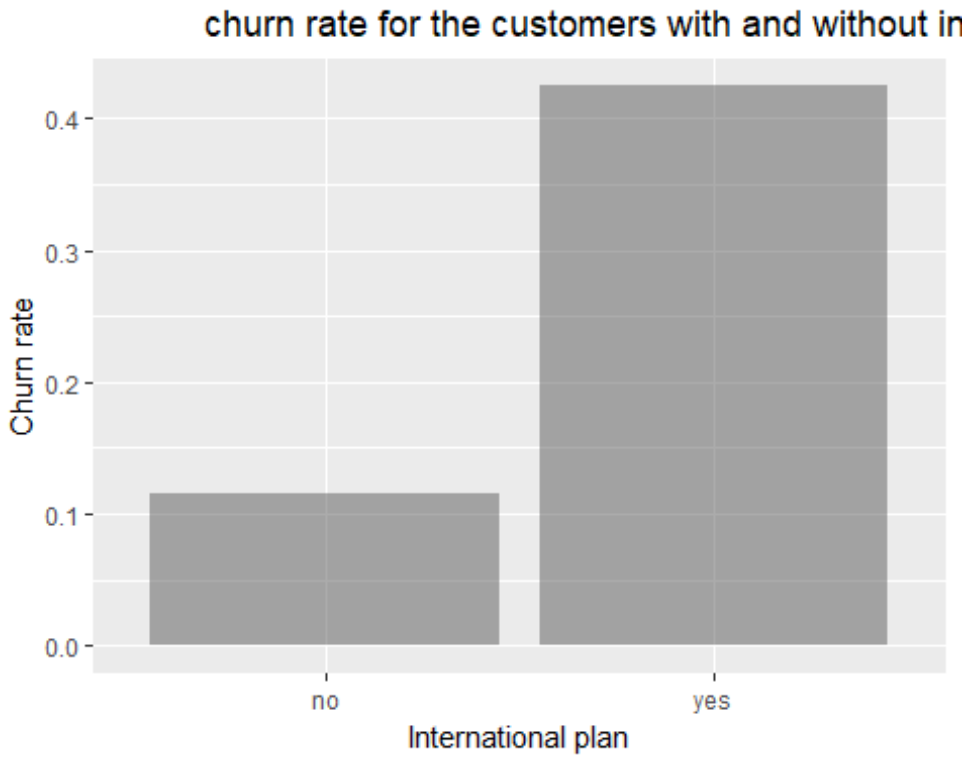
```
# Graph1
# plot.new()
barplot(data2$count, axes=F,ylim=c(0,max(data2$count+10)))
axis(side=4,ylim=c(0,max(data2$count)))
mtext("Customer count",side=4)
par(new=TRUE)
plot(x=data1$state, y=data1$churnrate,main="Churn rate and customer count
across states",xlab="State",ylab="Churn
rate",axes=T,ylim=c(0,max(data1$churnrate)))
lines(x=data1$state, y=data1$churnrate,axes=T,ylim=c(0,max(data1$churnrate)))
```



The above graph shows the churn rate on the primary axis and number of customers on the secondary axis for each state. We can see fromt he graph that there are states like CA,NJ etc whose curn rate is greater than 20%. These shouls be the areas of focus while implementing a churn analysis plan.

Graph 2:

```
# Graph2
# plot.new()
data1 <- aggregate(x =
churnTrain$churn_flag,by=list(Interntional_plan=churnTrain$international_plan
),FUN = mean)
data1 <- data.frame(Int_plan = data1$Interntional_plan,churnrate=data1$x)

ggplot(data1, aes(x=data1$Int_plan,
y=data1$churnrate))+geom_bar(stat="identity", alpha=0.5) +
xlab("International plan") + ylab("Churn rate") + labs(title= "
churn rate for the customers with and without international plan")
```



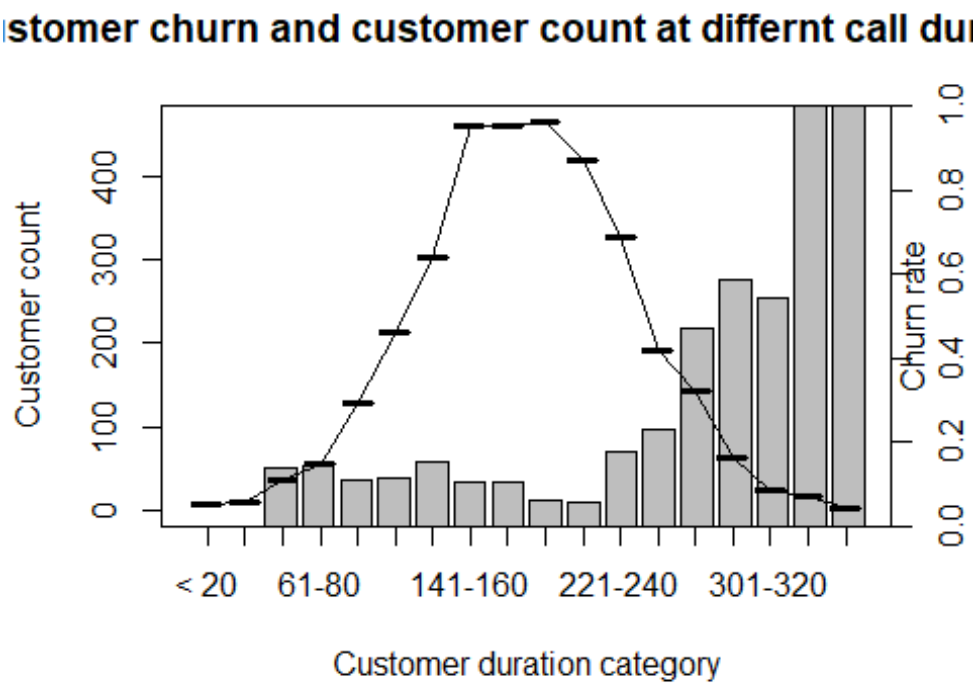
The above graph shows a comparision of churn rates between two segments. The left barplot is the churn rate among the customers who donot have interntional plan. The second bar is for the customers who have international plan. 1. We can see from the graph that the churn rate for the customers with internationla plan is very high(42.4%) when compared to the customers with no international plan. Also the churn in this segment is 28.3% percentage out of total churn which is a significant portion.

2. Also in Graph 1 we see there is high churn rate int he states like CA, TX, NJ where there is a high percentage of foreign citizens. The reason for these customers to churn could be that they are not satisfied with the international calling charges or services

Graph 3

```
# Graph3
# plot.new()
df1<-NULL
df1 <- transform(churnTrain, group=cut(churnTrain$total_day_minutes,
        breaks=seq(0,360,20),
        labels=c('< 20','21-40','41-60','61-80','81-100','101-120','121-
140','141-160','161-180','181-200',
                 '201-220','221-240','241-260','261-280','281-300','301-
320','321-340','341-360')))

data1 <- aggregate(df1$churn_flag,by=list(group=df1$group),FUN = mean)
data1 <- data.frame(group=data1$group,churnrate=data1$x)
```

```
data2 <- aggregate(df1$churn,by=list(group=df1$group),FUN = length)
data2 <- data.frame(group=data2$group,count=data2$x)


barplot(data1$churnrate, axes=F,ylim=c(0,max(data1$churnrate)))
axis(side=4,ylim=c(0,max(data2$count)))
mtext("Churn rate",side=4)
par(new=TRUE)
plot(x=data2$group, y=data2$count,main="Customer churn and customer count at
differnt call durations",xlab="Customer duration category",ylab="Customer
count",axes=T,ylim=c(0,max(data2$count)))
lines(x=data2$group, y=data2$count,axes=T,ylim=c(0,max(data2$count)))
```

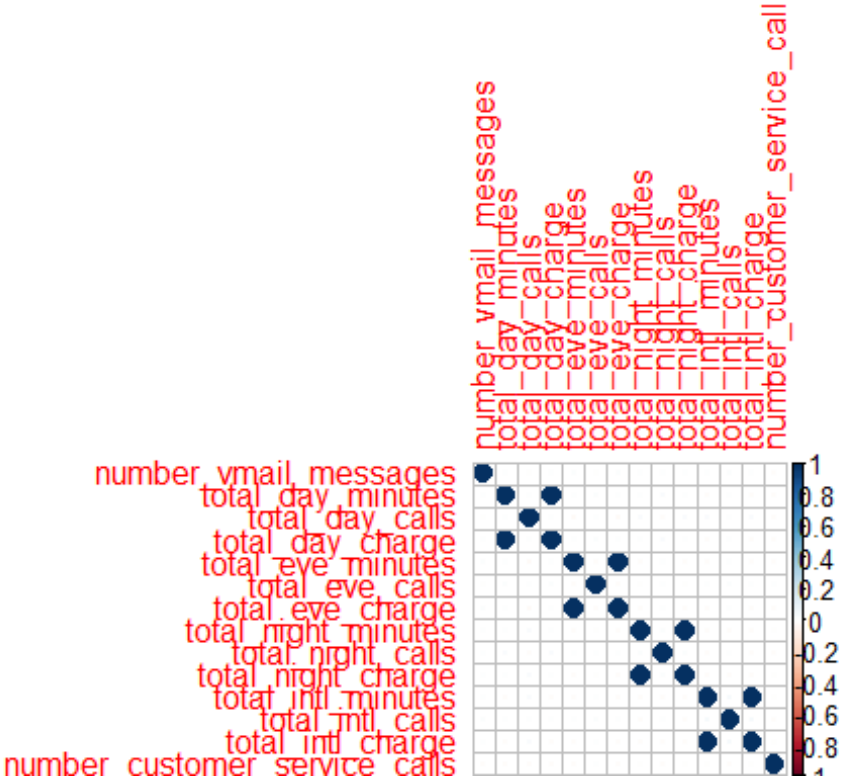


The above plot shows the distribution plot of the customer total day call minutes on the primary axis. The bar plot on the secondary axis shows the churn rate of the customers based on their total duration in day calls.

1. This graph shows that the customers who have "higher duration for the day calls" have higher percentage of churn rate when compared to the other duration buckets. This can be seen at the extreme right distribution in the graph. 8% of the customers fall into this high day calls duration segment. There is a possibility these customers might chhurn in future

Task 2:

```
# install.packages('corrplot')
library(corrplot)
my_data <- churnTrain[,seq(6,19,1)]
z <- cor(my_data)
corrplot(z,method='circle')
```



The above correlation matrix shows that there are some variables which are correlated as high as 0.99. This might cause multicollinearity in the models built further. This should be take caren of when the prediction models are built

Sice we want to understand the driveres that cause customer churn, doing a logistic regression would be appropriate on this data. To avoid Variace Inflation Factor, certain variables are removed fromt the model.

Logistic regression on this data gave the following results:

```
model_data <- churnTrain
model_data[
,c('total_day_charge','total_eve_charge','total_night_charge','total_intl_cha
rge',

'total_day_calls','total_eve_calls','total_night_calls','churn')] <- NULL
model <- glm(churn_flag ~.,family=binomial(link='logit'),data=model_data)
summary(model)
```

```
##
## Call:
## glm(formula = churn_flag ~ ., family = binomial(link = "logit"),
##     data = model_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9210  -0.5012  -0.3126  -0.1679   3.0905
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -9.188843   0.848940 -10.824  < 2e-16 ***
## stateAL                       0.356152   0.762271   0.467 0.640339
## stateAR                       0.939554   0.751170   1.251 0.211012
## stateAZ                       0.115059   0.844047   0.136 0.891569
## stateCA                       1.867943   0.781768   2.389 0.016877 *
## stateCO                       0.664295   0.761330   0.873 0.382910
## stateCT                       1.041330   0.723838   1.439 0.150257
## stateDC                       0.733022   0.806362   0.909 0.363325
## stateDE                       0.772569   0.748477   1.032 0.301984
## stateFL                       0.636844   0.758662   0.839 0.401228
## stateGA                       0.720139   0.774934   0.929 0.352739
## stateHI                      -0.200420   0.896872  -0.223 0.823173
## stateIA                       0.265974   0.901440   0.295 0.767953
## stateID                       0.931957   0.744955   1.251 0.210925
## stateIL                      -0.160534   0.829200  -0.194 0.846489
## stateIN                       0.501593   0.750440   0.668 0.503879
## stateKS                       1.101061   0.728491   1.511 0.130679
## stateKY                       0.831268   0.764070   1.088 0.276619
## stateLA                       0.599701   0.833962   0.719 0.472081
## stateMA                       1.204365   0.740875   1.626 0.104035
## stateMD                       1.165800   0.715228   1.630 0.103108
## stateME                       1.376924   0.725885   1.897 0.057842 .
## stateMI                       1.421392   0.712082   1.996 0.045922 *
## stateMN                       1.192801   0.713850   1.671 0.094733 .
## stateMO                       0.625165   0.773775   0.808 0.419124
## stateMS                       1.387093   0.726811   1.908 0.056331 .
## stateMT                       1.871177   0.716002   2.613 0.008965 **
## stateNC                       0.654210   0.750613   0.872 0.383444
## stateND                       0.181105   0.795585   0.228 0.819928
## stateNE                       0.346832   0.802832   0.432 0.665733
## stateNH                       1.198745   0.766536   1.564 0.117853
## stateNJ                       1.621456   0.707788   2.291 0.021970 *
## stateNM                       0.505542   0.786317   0.643 0.520274
## stateNV                       1.280299   0.723361   1.770 0.076739 .
## stateNY                       1.180274   0.719073   1.641 0.100718
## stateOH                       0.713585   0.744780   0.958 0.338005
## stateOK                       0.907734   0.751833   1.207 0.227293
## stateOR                       0.781286   0.735098   1.063 0.287857
## statePA                       1.167777   0.777498   1.502 0.133105
```

```
## stateRI                         -0.097641   0.820986  -0.119 0.905329
## stateSC                          1.820717   0.734326   2.479 0.013159 *
## stateSD                          0.848726   0.759252   1.118 0.263633
## stateTN                          0.311143   0.819062   0.380 0.704036
## stateTX                          1.679783   0.706766   2.377 0.017467 *
## stateUT                          1.065901   0.742382   1.436 0.151063
## stateVA                         -0.373455   0.820582  -0.455 0.649030
## stateVT                          0.123157   0.775628   0.159 0.873840
## stateWA                          1.444746   0.722300   2.000 0.045478 *
## stateWI                          0.295013   0.778538   0.379 0.704738
## stateWV                          0.632658   0.730759   0.866 0.386625
## stateWY                          0.347665   0.752633   0.462 0.644131
## account_length                   0.001005   0.001434   0.701 0.483414
## area_codearea_code_415          -0.078805   0.141577  -0.557 0.577783
## area_codearea_code_510          -0.106350   0.162633  -0.654 0.513160
## international_planyes            2.187851   0.153241  14.277  < 2e-16 ***
## voice_mail_planyes              -2.106768   0.591432  -3.562 0.000368 ***
## number_vmail_messages            0.037489   0.018557   2.020 0.043361 *
## total_day_minutes                0.013129   0.001109  11.843  < 2e-16 ***
## total_eve_minutes                0.007700   0.001183   6.509 7.55e-11 ***
## total_night_minutes              0.003957   0.001150   3.441 0.000580 ***
## total_intl_minutes               0.083999   0.021080   3.985 6.75e-05 ***
## total_intl_calls                -0.090466   0.025674  -3.524 0.000426 ***
## number_customer_service_calls    0.534971   0.040861  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: 2073.9  on 3270  degrees of freedom
## AIC: 2199.9
##
## Number of Fisher Scoring iterations: 6

# install.packages('car')
library(car)
vif(model)

##                                     GVIF Df GVIF^(1/(2*Df))
## state                           1.358026 50        1.003065
## account_length                  1.030047  1        1.014912
## area_code                       1.061229  2        1.014968
## international_plan              1.147291  1        1.071116
## voice_mail_plan                16.316978  1        4.039428
## number_vmail_messages          16.246702  1        4.030720
## total_day_minutes               1.074450  1        1.036557
## total_eve_minutes               1.055865  1        1.027553
## total_night_minutes             1.035570  1        1.017630
## total_intl_minutes              1.029865  1        1.014823
```

```
## total_intl_calls               1.029270  1        1.014529
## number_customer_service_calls  1.134656  1        1.065202

prediction <- predict(model,churnTest,type='response')
Final_pred <- data.frame(pred=prediction,
actual=as.character(churnTest$churn))
Final_pred$bin_pred <- ifelse(Final_pred$pred>0.5,'yes','no')

Final_pred$comp <- ifelse(Final_pred$actual ==
Final_pred$bin_pred,"Match","Dont Match")
```

Insights from the results above: 1. As shown in the graph 1, most of the states with high churn rate turned out to be significant variables in the model. 2. The major factors that seem to drive the churn rate is a.the call duration ie, churn rate is high for customers with high call duration b.the international flag ie, customers with international plan churn at higher rate c.the voice mail flag ie customers with no voice mail sevice churn at higher rate when compared to the customers with voice mail service d.Customers with more customer service calls also churn at higher rate. This implies that the customer calls are not addressed properly and this is effecting the customer satisfaction.

Mitigations: 1.Special offers should be given to the customers who have high call durations irrespective of the time of day 2.International calling services ahould include more benifits as per the customer needs 3.The customer service calls should be taken more seriously and their issues should be given the best possible solution

Task 3

While the logistic regression model results revealed some drivers which might be causing the churn, the tree based models will give the best possible predictions about the customers who can churn.

In order to ensure the best possible predictions, three tree based models are fit and model with best forecast accuracy is considered to predict the churn and futher address the solution. The models used for comparision are CART, Bagged model and Randome Forest Model. The following code gives the model predictions and accuracy for all the three models

```
# Function to find accuracy of a model
Acc_fun <- function(model_output)
{
  pred <- predict(model_output,churnTest)
  Final_pred <- data.frame(pred, actual=as.character(churnTest$churn))
  Final_pred$bin_pred <- ifelse(Final_pred$pred>0.5,'yes','no')
  Final_pred$compare <- ifelse(Final_pred$actual ==
Final_pred$bin_pred,"Match","Dont Match")
  Accuracy <- table(Final_pred$compare)[2]/nrow(Final_pred)

  return(list(Accuracy,Final_pred))
```

```
}

# Classification tree
# install.packages('rpart')
library(rpart)
cartModel <- rpart(churn_flag~., data = model_data)
Accuracy_CT <- Acc_fun(model_output = cartModel)


# Bagged decision tree
# install.packages('ipred')
library(ipred)
bag_model <- bagging(churn_flag~., data = model_data)
Accuracy_Bag <- Acc_fun(model_output = bag_model)


# Random Forest
# install.packages('randomForest')
library(randomForest)
```

```
## 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
```

```
RF_model <- randomForest(churn_flag~., data = model_data)
RF_output <- Acc_fun(model_output = RF_model)

Accuracy_CT[1]
```

```
## [[1]]
##     Match
## 0.9376125
```

```
Accuracy_Bag[1]
```

```
## [[1]]
##     Match
## 0.9370126
```

```
RF_output[1]
```

```
## [[1]]
##     Match
## 0.9454109
```

Out of all the three models, Randome Forest method has given the best accuracy. So RF model is considered for further analysis.

Solution to control the churn: Offer all the customers who are predicted to churn, a discount of $10 on their monthly charge.

Financials of the plan: Customer should be made aware of this offer to stop him from leaving. For this, they should should be targeted through e mails and postal mails. The company should also design the content of the mails, which contains the message to be comunicated to the customers.The mails are sent to the customers once in two weeks, to make sure, they are aware of this offer.

Assumptions: 1. The response rate for this campaign will be 60% 2. This campaign will stop the customers from churing for atleast 6 months 3. The revenue from each customer is the average value of the charges from all the call types(day calls, eve calls, nigh calls) in the data

Estimated costs are as follows:

Content cost : $2,500 Mail cost : $0.5 per customer Mail letter: $2 per customer Discount on the monthly bill : $10 per customer

When the model is right in predicting the churn customer, the company will have a profit of $43.5 from every correct prediction. If the model predicts a churn customer as "no churn", the company loses $56 on each false negative prediction When the model incorrectly predicts a customer to churn, the company will lose $22.5 on each customer who are incorrectly predicted

With the threshold 0.5, the customers with the churn probability above 0.5 predicted to churn and the customers below the threshold(prob=0.5), are predicted not to churn. As per these predictions the total cost is : $3832 The revenue from retaining the customers : $4401 Profit in a month : $569

Based on the performance of the model, this solution is profitable at the probabilty threshold. But we might be able to get the maximum profit for this model at another threshold. The graph below plots the estimated profits at differnt threshold values using this model. As shown in the graph the maximum profit is seen at

It is most profitable for the company to target all the customers whose probability is above 0.27

The estimated profit for six months is atleast $1819.4*6 = $10,916

```
RF_pred_data <- RF_output[[2]]

#
sequence <- seq(0,1,0.001)
# i <- 5
breakeven_data <- data.frame()
# colnames(breakeven_data) <- c("cutoff","profit")
```

```r
for (i in 1:1000)
{
  check <- RF_pred_data[RF_pred_data$pred>sequence[i],]
  Churn_pred_count <- nrow(check)
  True_negative_count <- table(check$actual)[[2]]

  Content_cost <- 2500
  Mail_cost <- Churn_pred_count*0.5
  postal_cost <- Churn_pred_count*2
  Discount <- Churn_pred_count*5
  total_cost <- Content_cost + Mail_cost + postal_cost + Discount
  Revenue <- 0.6*True_negative_count*56
  profit <- Revenue-total_cost
  a1 <- c(sequence[i],total_cost,Revenue,profit)

  breakeven_data <- rbind(breakeven_data,a1)

}
colnames(breakeven_data) <- c("cutoff","total_cost","Revenue","profit")
# plot(breakeven_data$cutoff,breakeven_data$total_cost)
# plot(breakeven_data$cutoff,breakeven_data$Revenue)
plot(breakeven_data$cutoff,breakeven_data$profit,ylab="profit(in
$)",xlab="cutoff on the probability - Value above the a cutoff value is
customer churn")
```