Advance BA with R Project

Akash Chandrakar

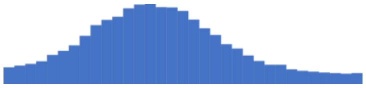
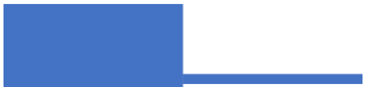
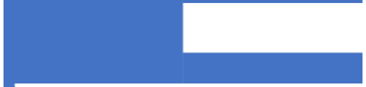
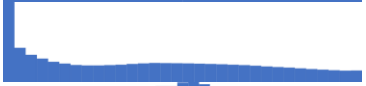
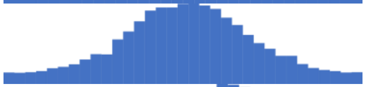
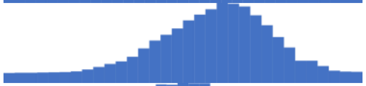
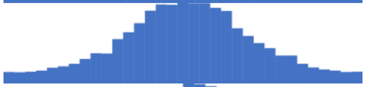
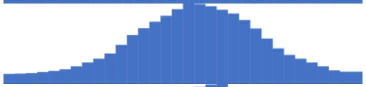
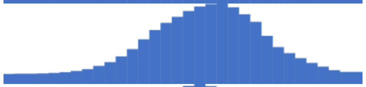
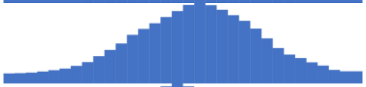
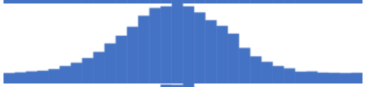
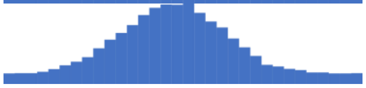
May 6, 2017

Libraries

```
library(C50)
library(tree)
library(caret)
library(AppliedPredictiveModeling)
library(pROC)
library(MASS)
library(glmnet)
library(nnet)
library(ggplot2)
```

Task 1: Explore the data. Describe what the rows and columns of the data represent.

```
data(churn)\
str(churnTrain)
total <- rbind(churnTrain, churnTest)
```

Variable	Type	Description	Frequency Distribution
state	categorical	States in USA	
account_length	Integer	Account length	
area_code	Categorical	Locality	
international_plan	Categorical	Customers having international plan 1- Yes, 2-no	
voice_mail_plan	Categorical	Customers having a voice mail plan	
number_vmail_messages	Integer	Number of voice mail messages	
total_day_minutes	Numerical	Total minutes of usage during day	
total_day_calls	Integer	Total number of calls during day	
total_day_charge	Numerical	Total charges faced during day	
total_eve_minutes	Numerical	Total minutes of usage during evening	
total_eve_calls	Integer	Total number of calls during evening	
total_eve_charge	Numerical	Total charges faced during evening	
total_night_minutes	Numerical	Total minutes of usage during night	
total_night_calls	Integer	Total number of calls during night	

total_night_charge	Numerical	Total charges faced during night	
total_intl_minutes	Numerical	Total minutes during international calls	
total_intl_calls	Integer	Total number of international calls	
total_intl_charge	Numerical	Total charges faced in international calls	
number_customer_service_calls	Integer	Customer service calls	
churn	Categorical	Churn, 1-yes, 2- no	

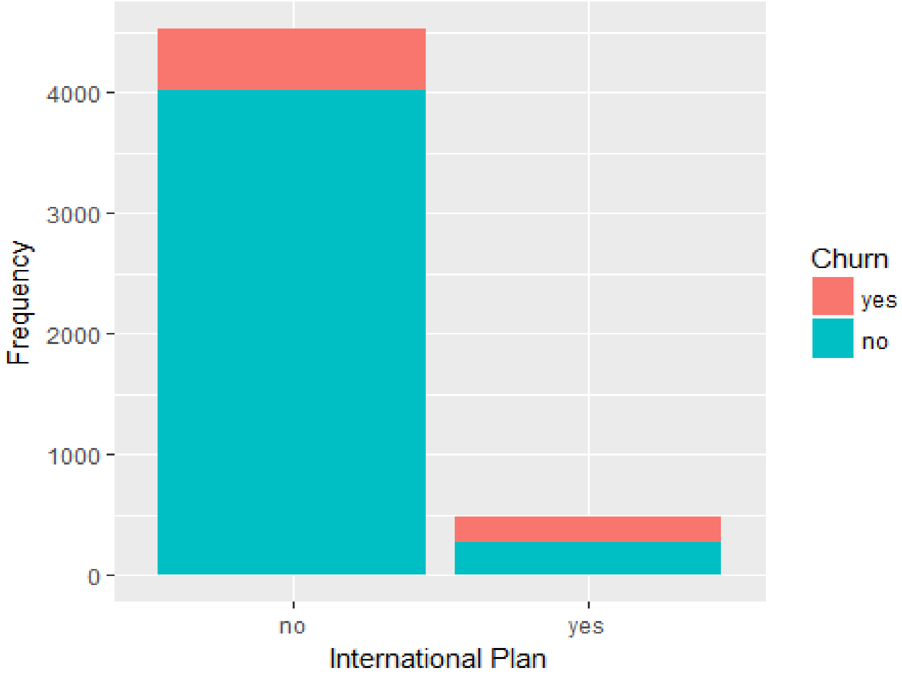
. What is the overall churn rate?

- From the table below it can be seen that the overall churn rate is **14.14%**.

```
churn.table <- as.data.frame(table(total$churn))
churn.table$percentfreq <- churn.table$Freq/sum(churn.table$Freq)
churn.table
##   Var1 Freq percentfreq
## 1  yes  707      0.1414
## 2   no 4293      0.8586
```

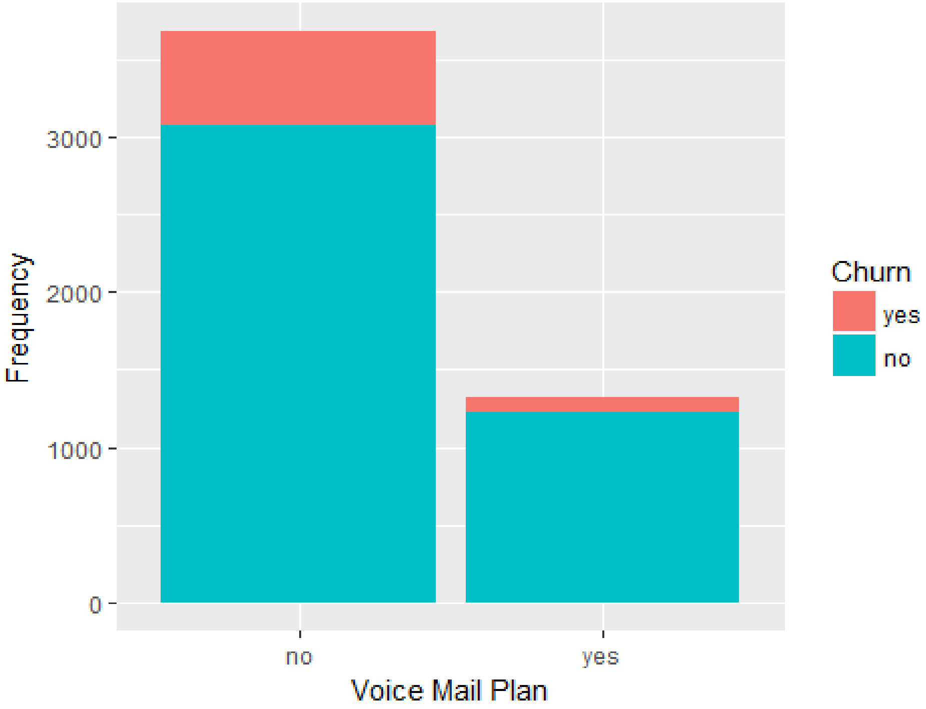
. Show some useful or interesting plots using ggplot.

```
ggplot() + geom_bar(aes(y = ..count..,x =as.factor(international_plan),fill = as.factor(c
hurn)),data=total)+labs(fill="Churn",x="International Plan",y="Frequency")
```



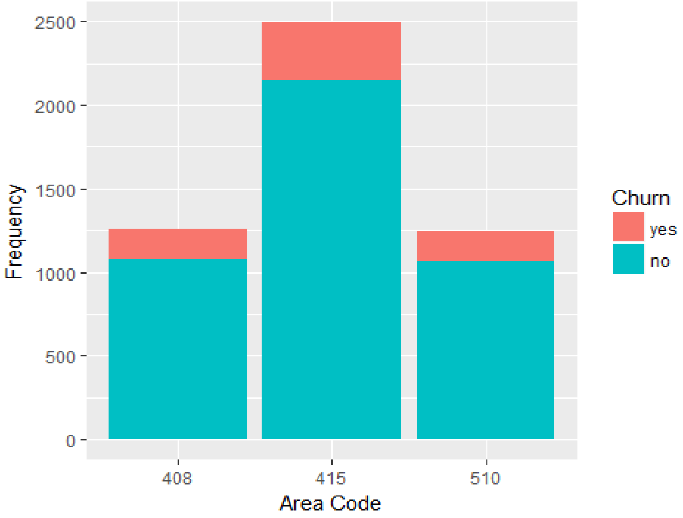
From the above plot it can be inferred that there is a huge number of people who does not have an international plan with their account, also people not having an international plan are more likely to churn as compared to the people who have an international plan.

```
ggplot() + geom_bar(aes(y = ..count..,x =as.factor(voice_mail_plan),fill = as.factor(chur
n)),data=total)+labs(fill="Churn",x="Voice Mail Plan",y="Frequency")
```



From the plot above it can be observed that people who take a voice mail plan are less likely to churn as compared people who do not take a voice mail plan. Almost 20% of people who does not have a voice mail plan are likely to churn. Out of the group data almost 40% of the people have a voice mail plan, out of which only 5% are likely to churn.

```
ggplot() + geom_bar(aes(y = ..count..,x =as.factor(area_code),fill = as.factor(churn)),da
ta=total)+labs(fill="Churn",x="Area Code",y="Frequency")+scale_x_discrete(labels=c("area_
code_408" = "408","area_code_415" = "415","area_code_510" = "510"))
```



From the above plot, it can be observed that out of the three area codes 415 has the maximum number of people who does not churn. And has more number of people churning as compared to the other two respective area codes. The area code 408 and 510 has almost same number of people who churn and does not churn.

Task 2: Build an interpretable model and measure its performance

NOTE:

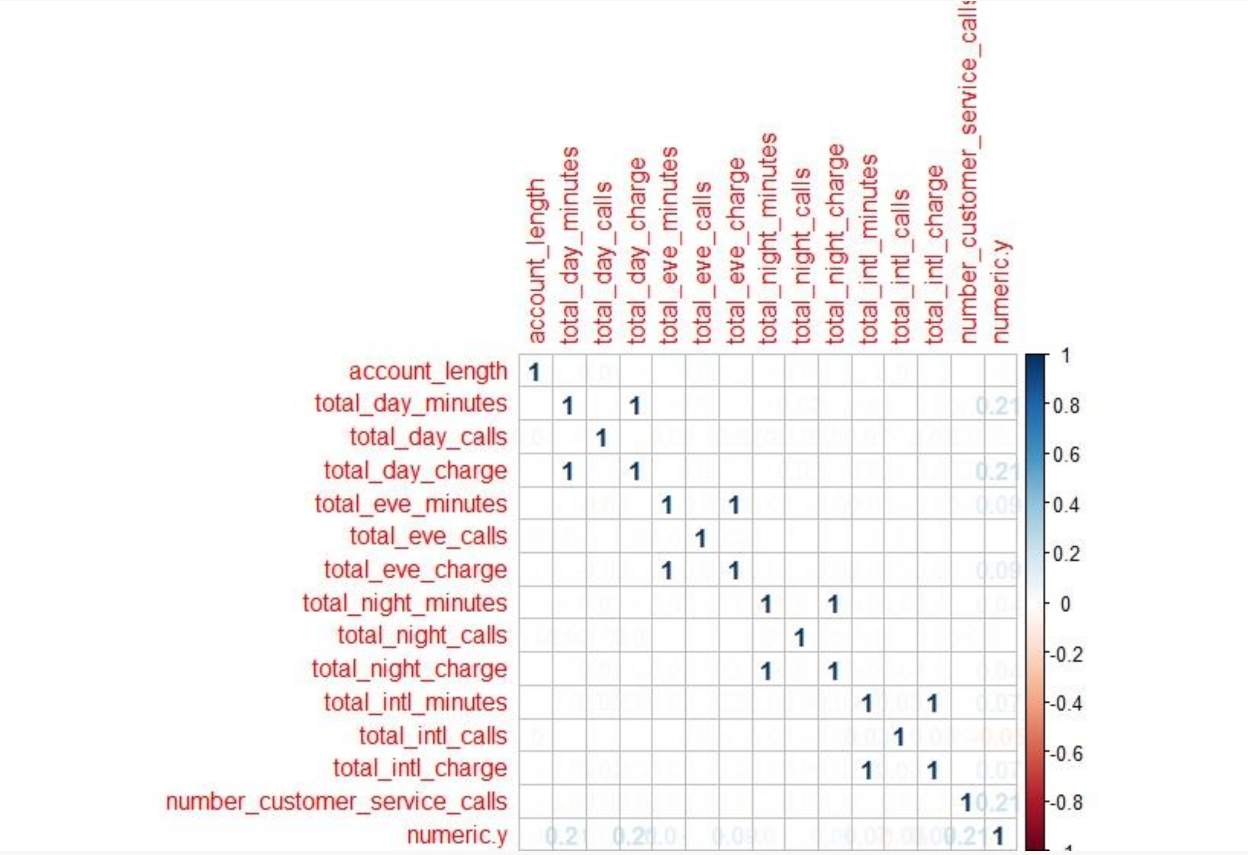
- The four categorical variables are removed from the data set to perform a regression analysis.
- Logistic regression analysis is been performed on the dataset to observe the effect of variables affecting the churn.
- The “glm” function treats the second factor level(no) as the event of interest.

Data Preprocessing:
The categorical variables were removed from the data set since regression requires only numerical variables. Any presence of degenerate variables was checked. There were none. The correlation between different variables were found out and is shown in the correlation matrix below. The correlation of various variables with the dependent variable, i.e. churn was also found. The table is shown below. It can be seen that number of customer service calls and total day minutes are the most correlated with the churn rate. All the variables except total international calls are positively correlated with the churn.

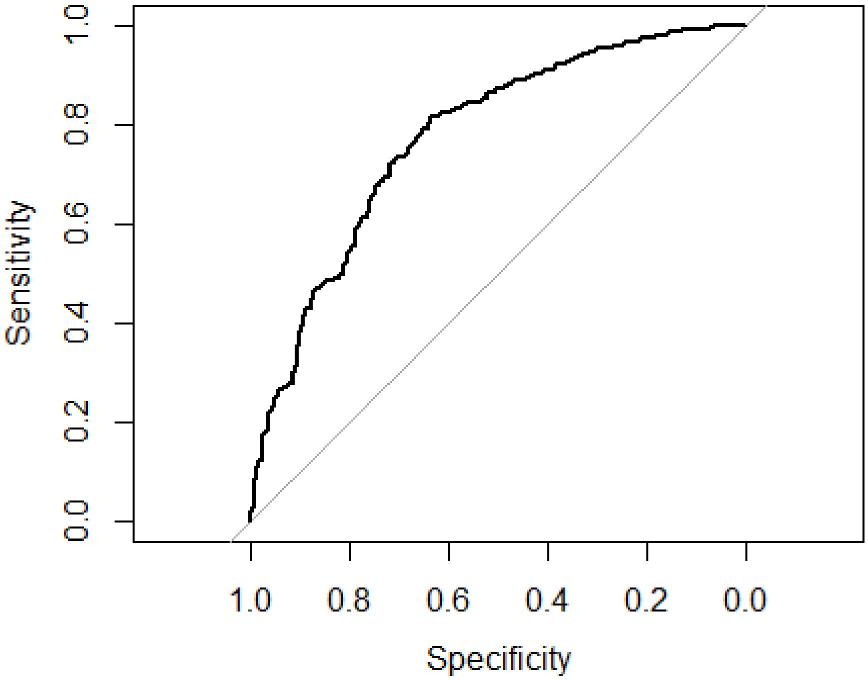
Modeling:
Firstly, a logistic regression with churn as the dependent variable and all other numerical variables as the explanatory variables was run. The output showed that the AUC in the ROC curve vas 67%, but was giving results which were not expected. Only three variables- number of voice mail messages, total number of international calls and total customer service calls were significant and rest all other variables were insignificant.

Model 2: With the help of correlation plot it came under observation that total day charge and total day minutes were very highly correlated with each other, same was the case with total eve charge and total eve minutes, total night charge and total night minutes. Since the variable almost represent the same things to remove the collinearity only one variable from each respective group was taken in account for the next model. The results are as follows

```
churnTrain1 = churnTrain[,-c(1,3,4,5)]
churnTest1 = churnTest[,-c(1,3,4,5)]
churnTrain.x = churnTrain1[,-16]
churnTrain.y = churnTrain1[,16]

zerovar.cols = nearZeroVar(churnTrain.x)
churnTrain.x = churnTrain.x[,-zerovar.cols]

numeric.y = rep( +1, length(churnTrain.y) )
numeric.y[churnTrain.y=="no"] = 0
correlation.matrix = cor( cbind( churnTrain.x, numeric.y ))
corr.name = names( correlation.matrix[,15] )
corr.values = as.double( correlation.matrix[,15])
corr.values.df = data.frame(corr.values)
rownames(corr.values.df) = corr.name
colnames(corr.values.df) = c("correlation")
print( corr.values.df[ order(abs(corr.values.df$correlation)), , drop=FALSE ] )
##                                correlation
## total_night_calls              0.006141203
## total_eve_calls                0.009233132
## account_length                 0.016540742
## total_day_calls                0.018459312
## total_night_minutes            0.035492853
## total_night_charge             0.035495556
## total_intl_calls              -0.052844336
## total_intl_minutes             0.068238776
## total_intl_charge              0.068258632
## total_eve_charge                 0.092786039
## total_eve_minutes                0.092795790
## total_day_charge                 0.205150743
## total_day_minutes                0.205150829
## number_customer_service_calls    0.208749999
## numeric.y                        1.000000000
corrplot(correlation.matrix, method = "number")
```



```
ctrl = trainControl( summaryFunction=twoClassSummary, classProbs=TRUE )
xx =churnTrain1[,-c(3,6,9,12,16)]
y= churnTrain1[,16]
xx1 <- churnTest1[,-c(3,6,9,12,16)]
y1 <- churnTest1[,16]

set.seed(1001)
early.model = train( churnTrain1[,-16], y, method="glm", metric="ROC", family = "binomial
", trControl=ctrl )
summary(early.model)
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0143   0.2504   0.4010   0.5686   1.7641
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    8.0194391  0.6829678  11.742  < 2e-16 ***
## account_length                -0.0014122  0.0013329  -1.060  0.28936
## number_vmail_messages          0.0246141  0.0045121   5.455 4.89e-08 ***
## total_day_minutes              0.6236691  3.1355848   0.199  0.84234
## total_day_calls               -0.0029075  0.0026459  -1.099  0.27183    
## total_day_charge              -3.7430456 18.4447412  -0.203  0.83919    
## total_eve_minutes             -0.4209563  1.5635207  -0.269  0.78775    
## total_eve_calls               -0.0008723  0.0026313  -0.332  0.74026    
## total_eve_charge               4.8742828 18.3943009   0.265  0.79102    
## total_night_minutes           -0.0119276  0.8342215  -0.014  0.98859    
## total_night_calls             -0.0009581  0.0027190  -0.352  0.72457    
## total_night_charge             0.2044409 18.5377138   0.011  0.99120    
## total_intl_minutes             1.4413457  5.0284283   0.287  0.77439    
## total_intl_calls               0.0800674  0.0238065   3.363  0.00077 ***
## total_intl_charge             -5.6814577 18.6229945  -0.305  0.76031    
## number_customer_service_calls -0.4549897  0.0371831 -12.236  < 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: 2361.7  on 3317  degrees of freedom
## AIC: 2393.7
## 
## Number of Fisher Scoring iterations: 5
logistic.model = train( xx, y, method="glm", metric="ROC", family = "binomial", trControl
=ctrl )
logistic.predictions = 1-predict( logistic.model, xx1, type="prob" )
logistic.roc = roc( response=y1, predictor=logistic.predictions[,1] )
summary(logistic.model)
## 
## Call:
## NULL
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.0099   0.2512   0.3999   0.5661   1.7696  
## 
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                    8.0192192  0.6825584  11.749  < 2e-16 ***
## account_length                -0.0014004  0.0013327  -1.051 0.293351    
## number_vmail_messages          0.0246188  0.0045086   5.460 4.75e-08 ***
## total_day_calls               -0.0029161  0.0026444  -1.103 0.270149    
## total_day_charge              -0.0744661  0.0061078 -12.192  < 2e-16 ***
## total_eve_calls               -0.0008759  0.0026299  -0.333 0.739090    
## total_eve_charge              -0.0780904  0.0127898  -6.106 1.02e-09 ***
## total_night_calls             -0.0009562  0.0027174  -0.352 0.724942    
## total_night_charge            -0.0606539  0.0233582  -2.597 0.009413 ** 
## total_intl_calls               0.0796131  0.0237631   3.350 0.000807 ***
## total_intl_charge             -0.3433523  0.0721444  -4.759 1.94e-06 ***
## number_customer_service_calls -0.4548859  0.0371476 -12.245  < 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: 2361.9  on 3321  degrees of freedom
## AIC: 2385.9
##
## Number of Fisher Scoring iterations: 5
plot(logistic.roc)
```



```
##
## Call:
## roc.default(response = y1, predictor = logistic.predictions[,     1])
##
## Data: logistic.predictions[, 1] in 224 controls (y1 yes) < 1443 cases (y1 no).
## Area under the curve: 0.7733
```

. What factors seem to be driving customer churn?

The most important factors driving the customer churn are **number of customer service calls** and **total international charge.**

Other factors that also play a role in driving the customer churn are: number of voice mail messages, total day charges, total evening charges, total night charges, total international calls.

From the above logistic regression the following results can be interpreted.

- With increase in number of voice mail messages by a unit the odds a customer not churning increases by 2.49%.
- With every unit increase in a total day charge the odds of a customer moving to other brand increases by 7.17%.
- With every unit increase in the evening charges the odds of customer churning increases by 7.51%.
- With a unit increase in night charges there is an increase in odds of customer churning by 5.88%.
- With an increase in international calls the odds of customer getting churn reduces by 8.2%.
- With every unit increase in the international charges the odds of customer getting churned increases by 29%.
- If a customer has more number of customer services calls then then with every unit increase in customer service calls the odds of customer moving to other brand increases by 36.54%.

. Give an idea to mitigate churn based on this model? This can be a general policy or strategy proposal.

The above model shows an accuracy of 77.33%. The following measures can be taken to mitigate churn:

- Since international charges and customer service callas are the most sensitive variables for the customer churn the company could focus more on these factors.

- The companies can keep a record of the person's customer service calls and could set a threshold, above which they can give customers some offers which would make them earn a customer's loyalty for some more time, since customers contacting the customer care more are more likely to leave.

- The most active time are evening and day. The company can reduce the evening charges and the day charges a little to get the customers loyalty.

Task 3: Build the best tree-based predictive model you can and measure its performance

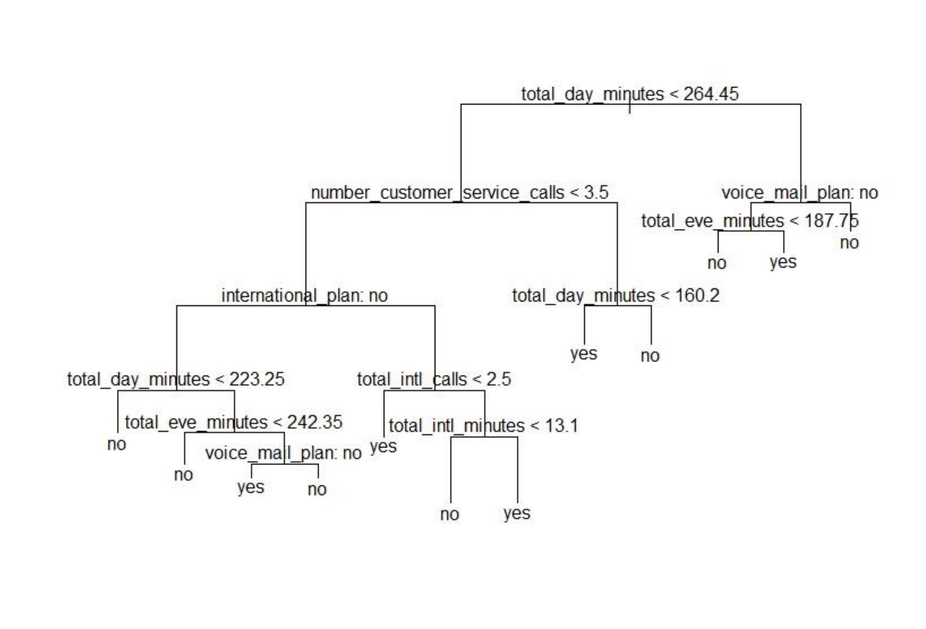
Preprocessing:
The factor variable state was removed from the data set since it contained 51 factors and was causing problems.

Modeling:

A tree based model was created with churn as the target variable and all other variables as categorical variables. The model gave an accuracy of 94%, a sensitivity of 62% and a specificity of 99%. The model was good enough to predict customer not churning. To improve the sensitivity the model was pruned with the help of complexity parameter. The ideal model obtained was with a subtree size of 10. With decision trees with many terminal nodes, we run the risk of overfitting the data. A tree with fewer branches will reduce model variance at the cost of some bias. Ideally, we want to build a tree where every node is determined based on the lowest error rate.

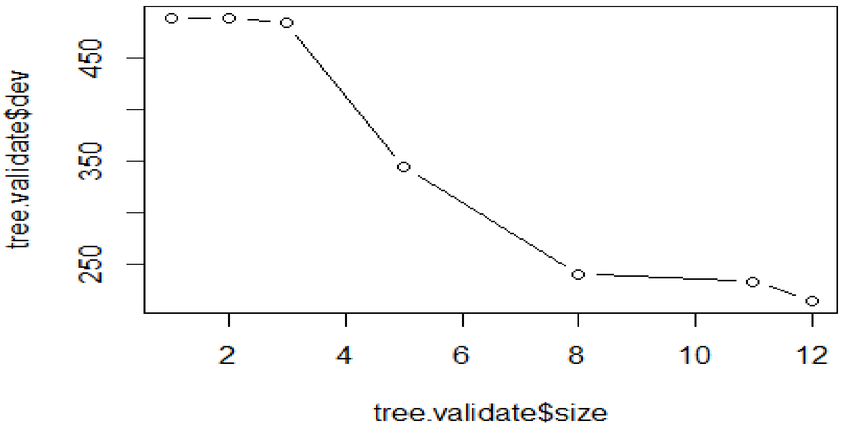
The function has considered 7 trees (12, 11, 8, 5, 3, 2 and 1 terminal nodes) as shown in the size parameter. The dev parameter lists the deviance or MSE for each tree. Pruning the tree to 10 nodes resulted in an improvement to sensitivity at a slight cost to specificity.

```
churnTest.tree = churnTest[,-1]
churnTrain.tree = churnTrain[,-1]
tree.model <- tree(churn ~., data = churnTrain.tree)
summary(tree.model)
##
## Classification tree:
## tree(formula = churn ~ ., data = churnTrain.tree)
## Variables actually used in tree construction:
## [1] "total_day_minutes"             "number_customer_service_calls"
## [3] "international_plan"            "total_eve_minutes"
## [5] "voice_mail_plan"               "total_intl_calls"
## [7] "total_intl_minutes"
## Number of terminal nodes:  12
## Residual mean deviance:  0.3772 = 1253 / 3321
## Misclassification error rate: 0.05911 = 197 / 3333
tree.predicted <- predict(tree.model, newdata = churnTest.tree, type = "class")
tree.cm <- print(confusionMatrix(tree.predicted, churnTest.tree$churn, dnn = c("Predicted
", "Reference")))
plot(tree.model)
text(tree.model, pretty = 0)
```

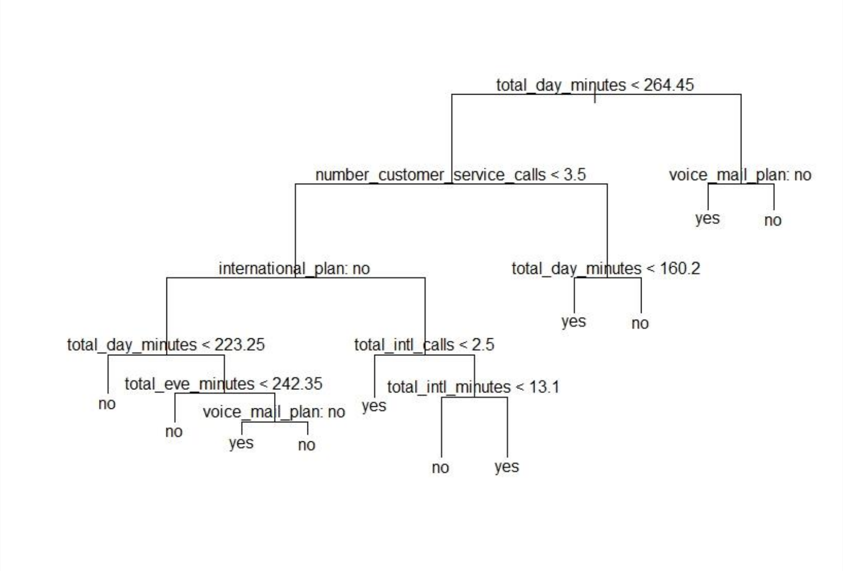

total_day_minutes < 264.45
number_customer_service_calls < 3.5
voice_mail_plan: no
total_eve_minutes < 187.75
no
no
yes
international_plan: no
total_day_minutes < 160.2
yes
no
total_day_minutes < 223.25
total_intl_calls < 2.5
total_eve_minutes < 242.35
total_intl_minutes < 13.1
no
voice_mail_plan: no
yes
no
yes
no
no
yes

```
## Confusion Matrix and Statistics
##
##          Reference
## Predicted  yes   no
##       yes   139  10
##       no    85 1433
##
##                 Accuracy : 0.943
##                   95% CI : (0.9308, 0.9537)
##      No Information Rate : 0.8656
##      P-Value [Acc > NIR] : < 2.2e-16
##
##                    Kappa : 0.7147
##   Mcnemar's Test P-Value : 3.144e-14
##
##              Sensitivity : 0.62054
##              Specificity : 0.99307
##           Pos Pred Value : 0.93289
##           Neg Pred Value : 0.94401
##               Prevalence : 0.13437
##           Detection Rate : 0.08338
##     Detection Prevalence : 0.08938
##        Balanced Accuracy : 0.80680
##
##         'Positive' Class : yes
##
set.seed(10001)
tree.validate <- cv.tree(object = tree.model, FUN = prune.misclass )
tree.validate
## $size
## [1] 12 11  8  5  3  2  1
##
## $dev
## [1] 215 234 241 345 484 489 489
##
## $k
## [1]      -Inf  7.000000  8.333333 31.333333 38.000000 41.000000 43.000000
```

```
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
plot(x=tree.validate$size, y=tree.validate$dev, type="b")
```



```
tree.model.pruned <- prune.misclass(tree.model, best = 10)
tree.pruned.predicted <- predict(tree.model.pruned, newdata = churnTest.tree, type ="clas
s" )
tree.cm2 <- print(confusionMatrix(tree.pruned.predicted, churnTest.tree$churn, dnn = c("P
redicted", "Reference")))
plot(tree.model.pruned)
text(tree.model.pruned, pretty = 0)
```


total_day_minutes < 264.45
number_customer_service_calls < 3.5
voice_mail_plan: no
yes
no
international_plan: no
total_day_minutes < 160.2
yes
no
total_day_minutes < 223.25
total_intl_calls < 2.5
no
total_eve_minutes < 242.35
no
voice_mail_plan: no
yes
no
yes
total_intl_minutes < 13.1
no
yes

```
## Confusion Matrix and Statistics
##
##           Reference
## Predicted  yes   no
##       yes  158   32
##       no    66 1411
##
##                Accuracy : 0.9412
##                  95% CI : (0.9288, 0.952)
##     No Information Rate : 0.8656
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.73
##  Mcnemar's Test P-Value : 0.0008576
##
##             Sensitivity : 0.70536
##             Specificity : 0.97782
##          Pos Pred Value : 0.83158
##          Neg Pred Value : 0.95531
##              Prevalence : 0.13437
##          Detection Rate : 0.09478
##    Detection Prevalence : 0.11398
##       Balanced Accuracy : 0.84159
##
##        'Positive' Class : yes
##
```

• Porpose a way to use this model to reduce churn.

Measures that can be taken to reduce customer churn are as follows:

- The company can observe the loyal customers and can release special offers for the top rated customers.
- If a person is spending more than 265 minutes total on day calls then the company could offer him a special voice mail plan, that would reduce his day charges and increase usage.
- If the customer does not spend too much time during day calls and also have a high frequency of customer service calls, then the company can focus on the customer service part and give some offers on the international call, because if a customer is spending a lot of time in international calls, low charges or time to time offers can keep him going.
- A voice mail plan and evening plans for the youngsters will be effective in sense of reducing customer churn. Since, youngsters are the most fluctuating customers, by giving some perks the company can earn the person's loyalty.

• Describe in detail the financials of your plan. You will have make assumptions here on dollar amounts and how much longer a customer will stay as a result of your intervention. Think about things like:
• What is the cost of your action?
• What is the dollar value gained by retaining a customer?
• What do these mean when your model is right/wrong?
• Based on the performance of your model, is this plan profitable? If not, where is the breakeven point? ie what could you change to make it profitable? The answer here can't be to build a better model.

The company can predict when is the customer is about to churn and can target them. The company can send offers like various voice mail plans, can try to retain them by investing some cost in mailing, marketing, promotions, etc. Take the following example. An average loyal customer is worth up to almost 10 times the dollar amount they spend on first purchase. If the first purchase is $50, just multiply that number by ten to get the estimated MAXIMUM dollar value of the loyal customer – in this case, the customer value would be up to $500. If the amount
If the lifetime value of an average customer is 8000$ and the LTV of a good customer is 10000$ then the company can spend 2000$ more to acquire good customers or retain the existing customers.

From the model it can be seen that, company can invest more on customer services and reduce prices of international calls. Suppose if the customer acquisition cost was $250 and average customer revenue increases $50 every year.
If the average customer cost increases $25 every year. If the customer loyalty rate reduces every year by 25%, average profit retained by each retained customer reduces, and each year the company has 10% discount rate increase and

discounted average profit contribution per retained customer reduces gradually every year then at the end of 10 years the total average profit contribution per customer or the customer life time value is almost $600.

The model above has an accuracy of 77.33%. If the model above is right then the company would make a profit of around 600$ per customer in 10 years, which might vary as per the respective parameters. But company might not want to spend more than the revenue earned per customer on the discounts and promotions because that could lead to loss.

Based on the performance of the above model, yes the plan is profitable, since the cost of acquisition is more than the cost of retention. The model is able to predict the churn very efficiently. The sensitivity is 70%. The model could be used to target the customers.