MIS 6357 - Final Project

Achintya Sen

May 6, 2017

Churn Analysis

Introduction

In many industries it is more expensive to find a new customer then to entice an existing one to stay. Looking forward with the motivation to predict customer behavior of a telecom major we are trying to predict the churn rate based on statistical analysis and suggest strategies to improve the services to lower the rate.

Descriptive Statistics

The dataset is collected from the MLC++ machine learning software for modeling customer churn. There are 19 predictors, mostly numeric:

```
##  [1] "state"                         "account_length"
##  [3] "area_code"                     "international_plan"
##  [5] "voice_mail_plan"               "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"
```

As you can see the dataset consist of few factor variables:

```
## 'data.frame':    3333 obs. of  4 variables:
##  $ state             : Factor w/ 51 levels "AK","AL","AR",..: 17 36 32 36 37 2 20 25 19 50 ...
##  $ area_code         : Factor w/ 3 levels "area_code_408",..: 2 2 2 1 2 3 3 2 1 2 ...
##  $ international_plan: Factor w/ 2 levels "no","yes": 1 1 1 2 2 2 1 2 1 2 ...
##  $ voice_mail_plan   : Factor w/ 2 levels "no","yes": 2 2 1 1 1 1 2 1 1 2 ...
```

The first step is to have a look at the balance of the outcomes. In this case its binary, either the client has an existing contract with our telecommunications company or they have cancelled it.

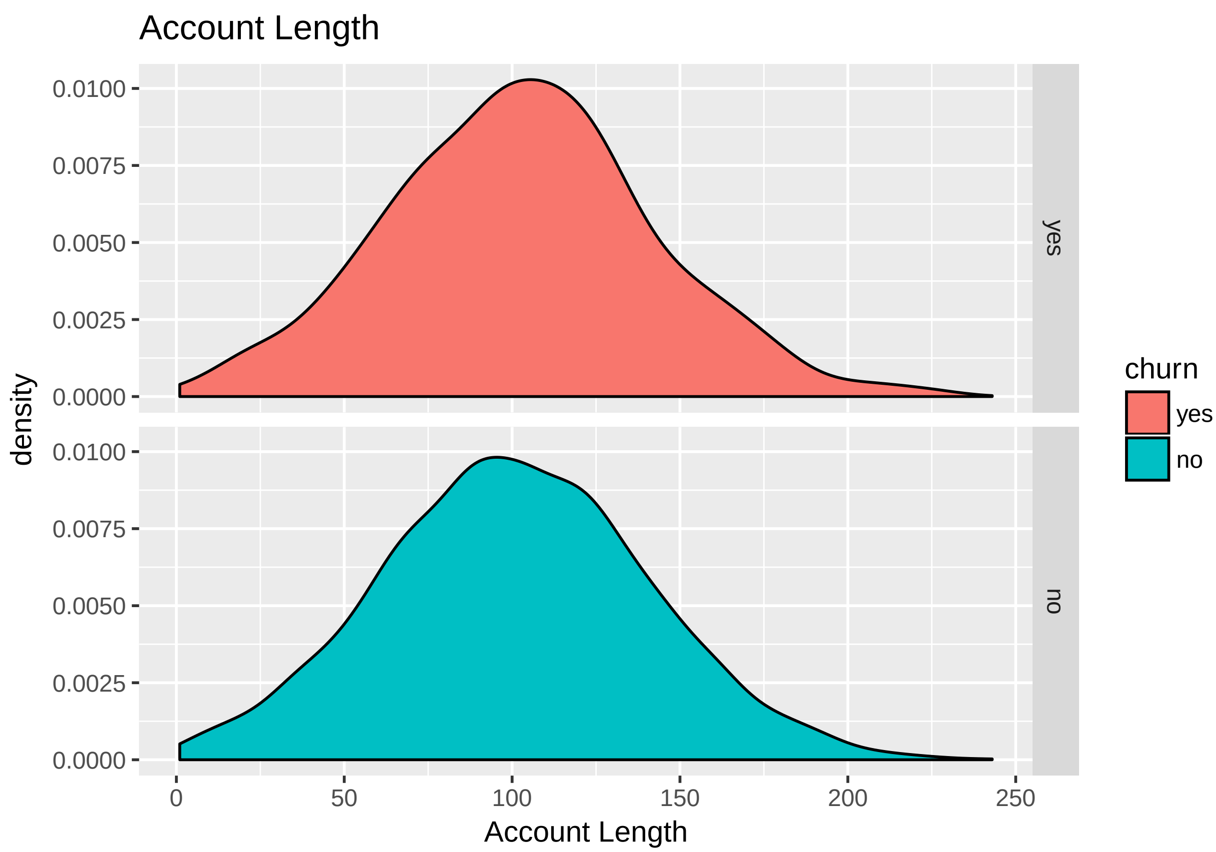
```
##
##  yes   no
##  483 2850
```

The overall churn rate is approximately ~15% i.e. 15% of the customers left the service and ~85% decided to continue with the service.The churn rate in individual state is as shown in the table whci has the list of the top 10.

```
##   states Churned Not Churned Percentage
## 1     CA       9          25      36.00
## 2     NJ      18          50      36.00
## 3     TX      18          54      33.33
## 4     MD      17          53      32.08
## 5     SC      14          46      30.43
```

We can also start to form testable ideas about relationships. For example does the "Account Length" field have an impact on if they churn?

```
ggplot(churnTrain, aes(x=account_length, fill=churn))+geom_density()+ facet_grid(churn~.) + labs(title=
```



Preprocessing

We are going to perform some feature selection on the list of factors and the continuous variables. Starting with the continous variables:
1. **Check for any degenerative function**: Degenerative variables are those variables which has very little variance. And removing the variable.

```
## [1] "number_vmail_messages"
```

2. **Remove degenerative Factors**:Degenrative factors are those factor variable which doesnt contribute much to the churn. Among the list of factor variables in the dataset *area_code* is not that significant.

Let us look at the distribution of state variable and check if the distribution is even.

```
##
##  AK  AL  AR  AZ  CA  CO  CT  DC  DE  FL  GA  HI  IA  ID  IL  IN  KS  KY
##  52  80  55  64  34  66  74  54  61  63  54  53  44  73  58  71  70  59
##  LA  MA  MD  ME  MI  MN  MO  MS  MT  NC  ND  NE  NH  NJ  NM  NV  NY  OH
##  51  65  70  62  73  84  63  65  68  68  62  61  56  68  62  66  83  78
##  OK  OR  PA  RI  SC  SD  TN  TX  UT  VA  VT  WA  WI  WV  WY
##  61  78  45  65  60  60  53  72  72  77  73  66  78 106  77
```

The churn rate seems to vary from state to state. We will keep this as a factor variable.

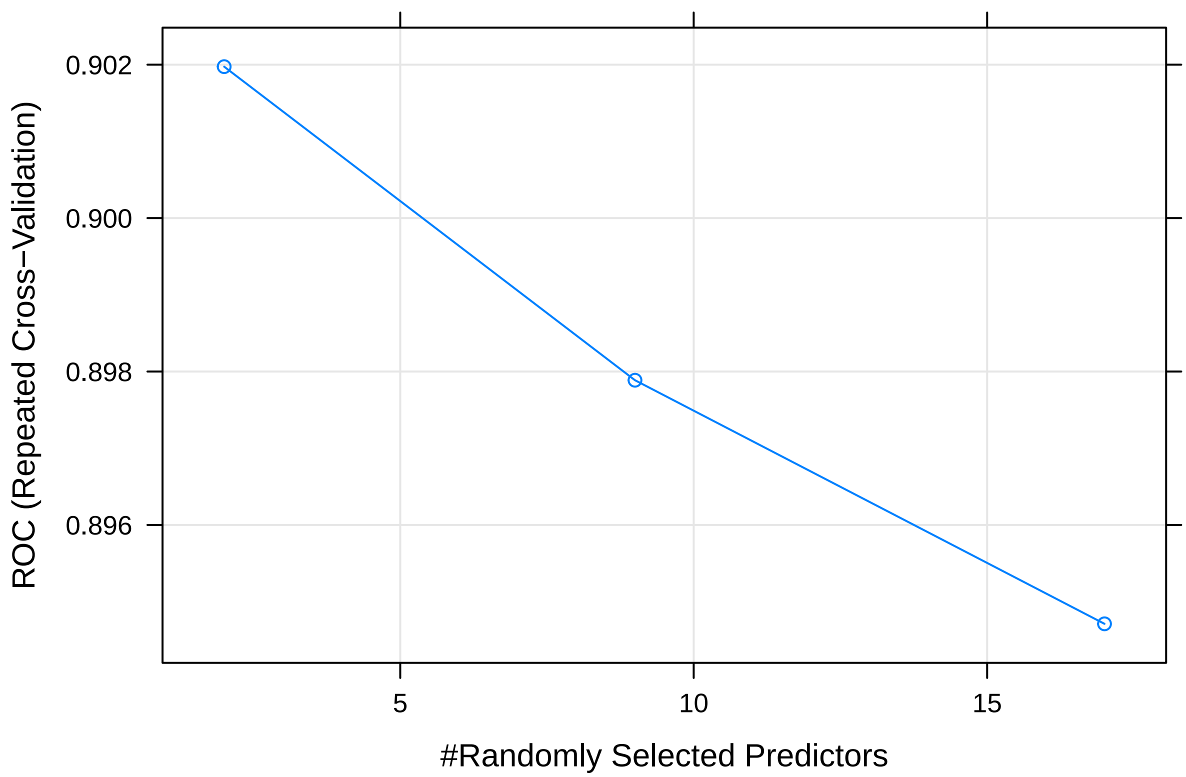
Build Models

We will start the process of building an actual model. As there are some imbalance in the number of churned customers and the fact that we really want to predict who will be a churned customer mean we are intrested in sensitivity in our models rather then specificity.

```
#
source("model_train.R")
#
all_models = build_AUC_models( X, y, churn.test )
```

```
## [1] "AUC Performance"
```

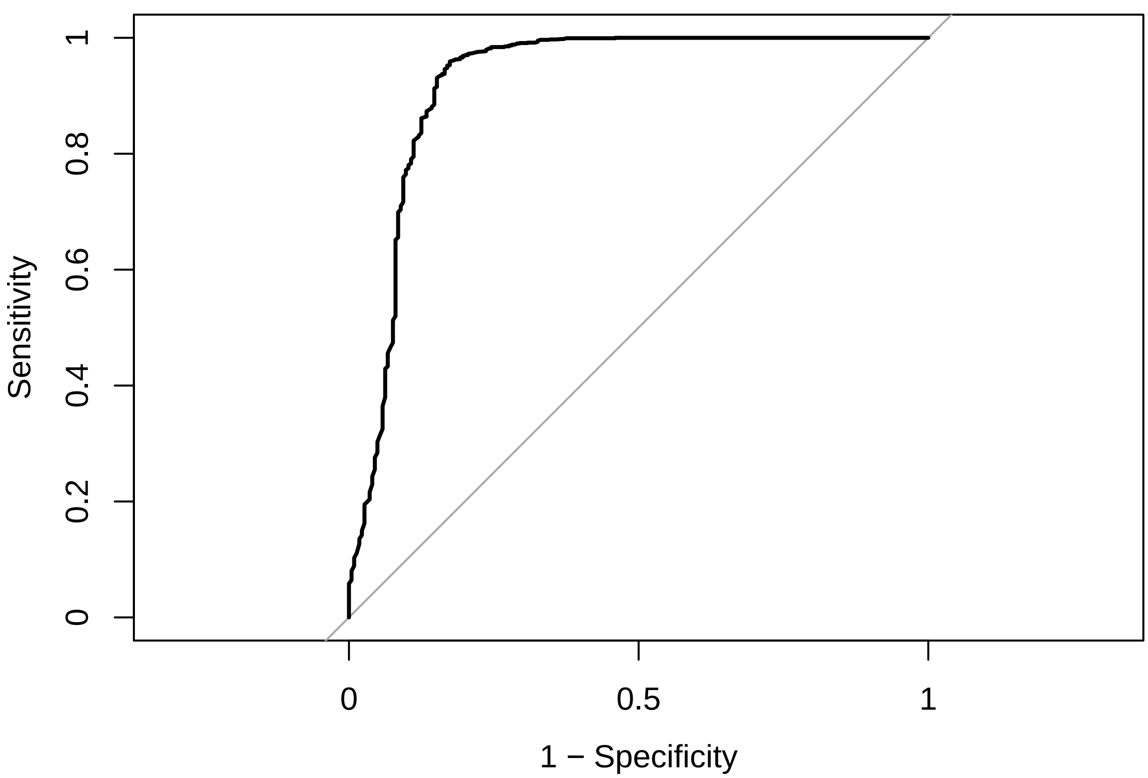
```
##                  Model     auc
## 1 Logistic Regression 83.36 %
## 2       Decision Tree 37.13 %
## 3       Random Forest 92.24 %
## 4        Bagged Trees 89.02 %
```



We will build our confusion matrix and the ROC curve based on the random forest:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  yes   no
##        yes  170   26
##        no    54 1417
##
```

```
##                Accuracy : 0.952
##                  95% CI : (0.9406, 0.9618)
##     No Information Rate : 0.8656
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7822
##  Mcnemar's Test P-Value : 0.002539
##
##             Sensitivity : 0.7589
##             Specificity : 0.9820
##          Pos Pred Value : 0.8673
##          Neg Pred Value : 0.9633
##              Prevalence : 0.1344
##          Detection Rate : 0.1020
##    Detection Prevalence : 0.1176
##       Balanced Accuracy : 0.8705
##
##        'Positive' Class : yes
##
```



Let us find the factors that seem to be driving customer churn. Based on our model the variables driving churn based on the mean decrease in the gini index:

```
##                                  MeanDecreaseGini
## state                                   109.61936
## account_length                           28.03863
## international_plan                       48.82689
```

```
## voice_mail_plan                        16.99535
## total_day_minutes                      99.04701
## total_day_calls                        29.29444
## total_day_charge                      100.53276
## total_eve_minutes                      44.75724
## total_eve_calls                        26.89263
## total_eve_charge                       44.49217
## total_night_minutes                    32.26265
## total_night_calls                      27.16321
## total_night_charge                     31.49161
## total_intl_minutes                     34.60629
## total_intl_calls                       31.66367
## total_intl_charge                      34.50422
## number_customer_service_calls          84.26769
```

As random forest is an ensemble technique encapsulating multiple trees, interpretation on individual features and splits to define strategies is a complex technique. Instead based on the important variables found by the random forest we will try to build a decision tree and optimize it to gain further insight.
We can set an arbitary threshold of *30.00* to select the important features which contrubutes the maximum in defining the customer churn behavior. Further optimization can be performed to select the threshold to fine tune the tree.

```
##                          variable MeanDecreaseGini
## 1                           state        109.61936
## 2                total_day_charge        100.53276
## 3               total_day_minutes         99.04701
## 4   number_customer_service_calls         84.26769
## 5              international_plan         48.82689
## 6               total_eve_minutes         44.75724
## 7                total_eve_charge         44.49217
## 8              total_intl_minutes         34.60629
## 9               total_intl_charge         34.50422
## 10            total_night_minutes         32.26265
## 11               total_intl_calls         31.66367
## 12             total_night_charge         31.49161
```

At this intersection, we can start build strategies on the characteritics that is driving cutomer churn. The major questions we can derive are:
1.**state** - Customers belonging to selected states are more susceptible to churn. Which are those states?
2. **total__day__charge** - What is the charge per day of the customers? At what threshold should we target the customers with better strategies?
3. **total__day__minutes** - After how much time the customer can think of changing the network provider?
4. **number__customer__service__calls** - On an average after how many service call make the customer frustrated?

Build the best tree-based predictive model

We will start by building our decision tree and optimise it to get the best result, utilizng which we will build our assumptions and actions.

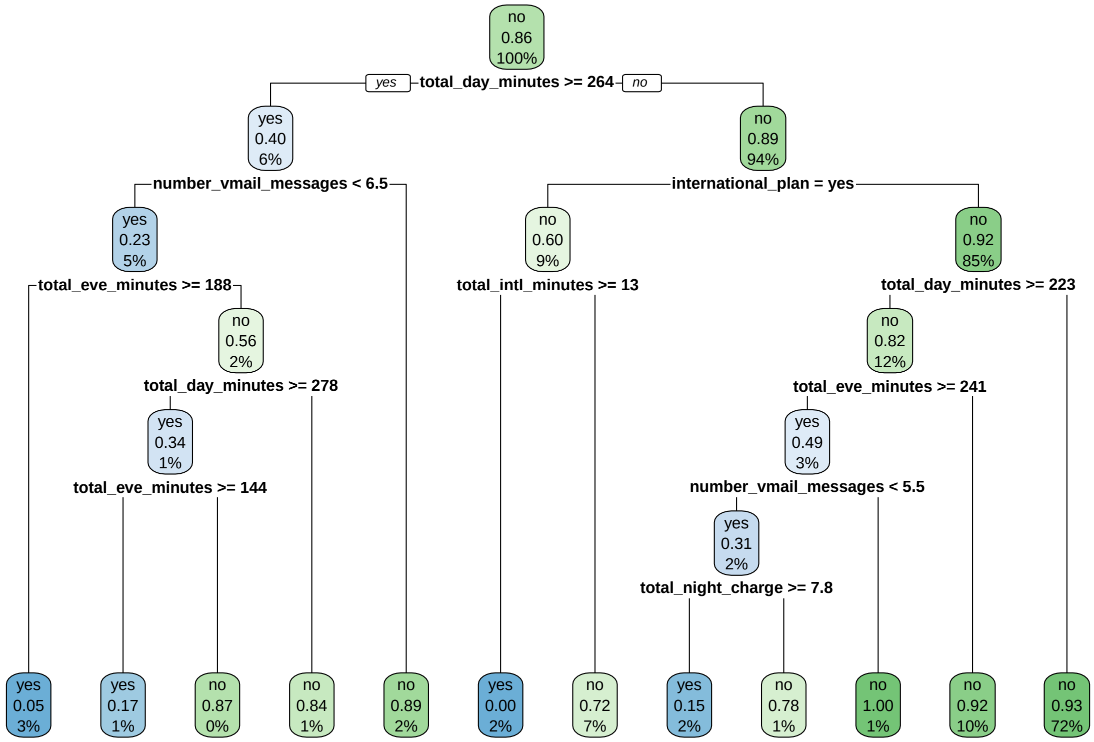
```
source("build_decision_tree.R")
build_decision_tree
```

```
## function (X, y, test)
## {
##     set.seed(12345)
##     tctrl2 <- trainControl(method = "adaptive_cv", repeats = 5,
##         classProbs = TRUE, summaryFunction = twoClassSummary)
##     dtree_fit <- train(X, y, method = "rpart2", parms = list(split = "information"),
##         trControl = tctrl2, metric = "ROC", tuneLength = 10)
##     pred.dtree <- predict(dtree_fit, newdata = test[, -ncol(test)],
##         type = "prob")
##     pred.dtree.res <- predict(dtree_fit, newdata = test[, -ncol(test)])
##     dtree.roc = pROC::roc(response = test[, ncol(test)], predictor = pred.dtree[,
##         1])
##     dtree.auc = dtree.roc$auc[1]
##     dtree = list(classifier = dtree_fit, pred.prob = pred.dtree,
##         pred.result = pred.dtree.res, roc = dtree.roc, auc = dtree.auc)
##     return(dtree)
## }
```

We will build our confusion matrix and the ROC curve based on the decision tree:

```
confusionMatrix(best.model$pred.result,sel.test[,ncol(sel.test)])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  yes   no
##        yes   87    5
##        no   137 1438
##
##                Accuracy : 0.9148
##                  95% CI : (0.9004, 0.9278)
##     No Information Rate : 0.8656
##     P-Value [Acc > NIR] : 2.748e-10
##
##                   Kappa : 0.5125
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.38839
##             Specificity : 0.99653
##          Pos Pred Value : 0.94565
##          Neg Pred Value : 0.91302
##              Prevalence : 0.13437
##          Detection Rate : 0.05219
##    Detection Prevalence : 0.05519
##       Balanced Accuracy : 0.69246
##
##        'Positive' Class : yes
##
```


no
0.86
100%
yes
total_day_minutes >= 264
no
yes
0.40
6%
no
0.89
94%
number_vmail_messages < 6.5
international_plan = yes
yes
0.23
5%
no
0.60
9%
no
0.92
85%
total_eve_minutes >= 188
total_intl_minutes >= 13
total_day_minutes >= 223
no
0.56
2%
no
0.82
12%
total_day_minutes >= 278
total_eve_minutes >= 241
yes
0.34
1%
yes
0.49
3%
total_eve_minutes >= 144
number_vmail_messages < 5.5
yes
0.31
2%
total_night_charge >= 7.8
yes
0.05
3%
yes
0.17
1%
no
0.87
0%
no
0.84
1%
no
0.89
2%
yes
0.00
2%
no
0.72
7%
yes
0.15
2%
no
0.78
1%
no
1.00
1%
no
0.92
10%
no
0.93
72%

Interpretation and Recommendation

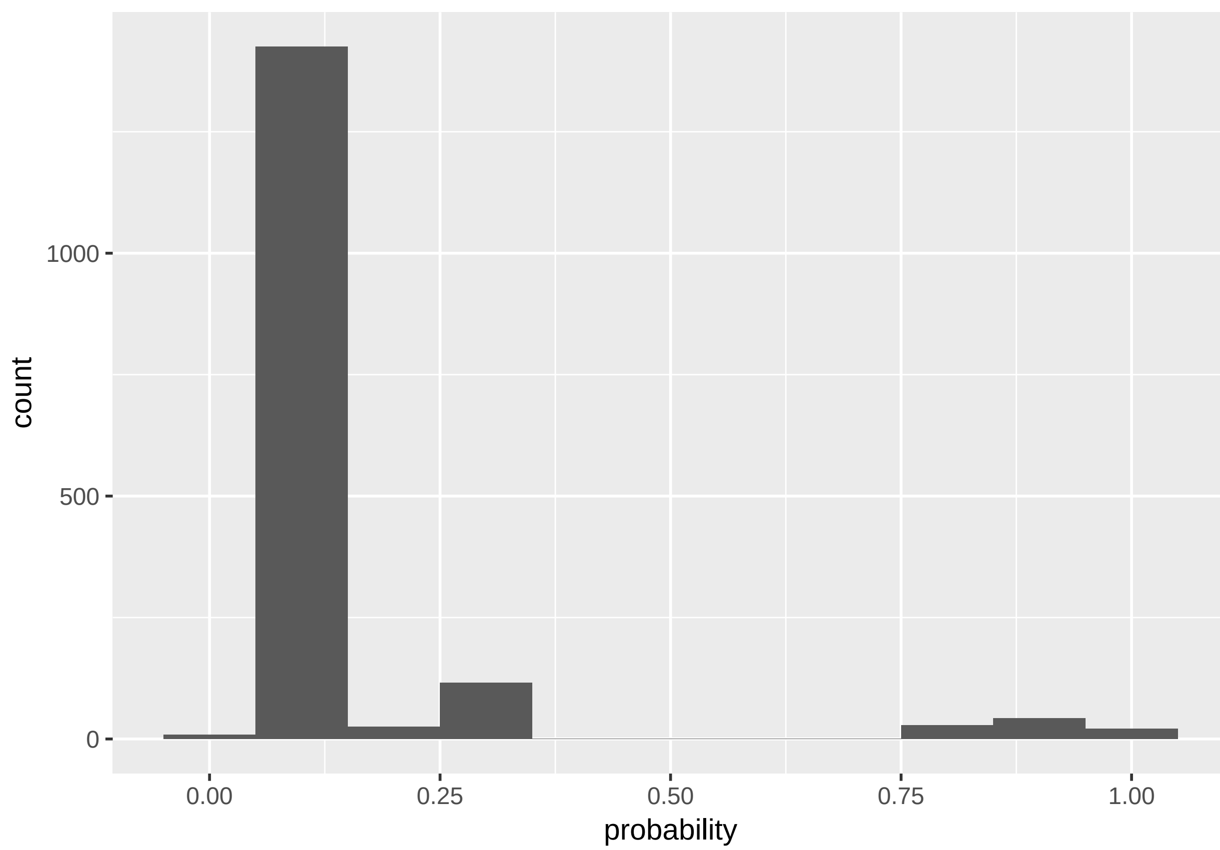
We re-calculate our important variables contributing towards the customer behavior.

```
## Loading required package: rpart

##                 Variable    Overall
## 1      total_day_minutes 202.160878
## 2       total_day_charge 195.125087
## 3     international_plan 180.281658
## 4      total_eve_minutes 134.668180
## 5       total_eve_charge 124.252345
## 6  number_vmail_messages  93.744411
## 7     total_intl_minutes  92.583109
## 8     total_night_charge  56.656237
## 9    total_night_minutes  43.516040
## 10        account_length   5.342492
## 11       total_eve_calls   0.000000
## 12             area_code   0.000000
```

We can see the major criteria for churn is contributed by the *total_day_charge*, *total_eve_minutes*, *international_plan*. As we are proceeding towards a targetted approach towards the customers we would like to gain and undertsanding which customers to target.

```
cust <- cbind(churn.test[,-ncol(churn.test)],probability=round(best.model$pred.prob[,1],2))
ggplot(cust,aes(probability))+geom_histogram(bins = 30,binwidth = 0.1)
```



We can device our targeted customers by choosing a threshold for the estimated probability. We need to keep in mind the cutomers who are at the lower end of the distributed who are most likely to churn irrespective of

the approah we follow to stop them.
Let delve deep into the various startegies that can be built on the model.
1. *total_day_charge* which is the major classification factor in defining the tree can be used to set threshold for customers whose *total_day_time* reduces a certain limit. We can target these customers with lesser talk plans so that we can reach a break-even. Charges are a mjor factor in customer churn as higher price will lead to customer behavior change. These customers need to be targetted with proper plans to optimize the bills.
2. *total_eve_minutes* is a significant criteria wherein the customers with less frequency in the evening is more likely to churn. The value estimated by the model can be used to generate the population parameter confidence interval.
3. *international_plan* is particularly significant in situations where in the customer has not opted for the internation plan but has internation outgoing calls. As international call plans are expensive and most customers prefer to not use, the targetted approach is to judiciously send to customers who will be crossing a threshold.