Telecom Customer Churn Classification

Achint[1]

[1]Affiliation not available

January 26, 2018

Abstract

Telecom companies need to have a better customer retention. Due to the decrease in abundance of customers, it is becoming less and less to retain their customers, let alone acquire new ones.

Introduction

Today, telecommunication market all over the world is facing a severe loss of revenue due to fierce competition and loss of potential customers. To keep the competitive advantages and acquire as many customers as possible, most operators invest a huge amount of revenue to expand their business in the very beginning . Therefore, it has become vital for the operators to acquire the amount invested and to gain at least a minimum profit within a very short period of time. Because it is very much challenging and tedious issue to keep the customers intact for a long duration due to the competition involved in this business field. To survive in the market, telecom operators usually offer a variety of retention policies to attract new customers. This is the major cause of the subscribers leaving one network and moving to another one which suits their needs. This also proves to be a bane as the already acquired customers get lured into switching to other companies.

Thus, Customer churn reduction is the central concern of most telecom companies as switching costs to the customer are low and acquisition cost to the company is high. Churn reduces profitability as it means potential loss of future revenue and also losing the invested costs of acquisition. So a good deal of marketing budget is allocated to prevent churn by designing new plans and discounts etc. But they want to understand the hidden patterns in the customer behaviour by use of customer behavioural data which lead to construction of purchase decision and also the underlying loyalty hooks. Fortunately, telecom industries generate and maintain a large volume of data. This voluminous amount data ensures the scope for the application of data mining techniques in telecommunication database. As plenty of information is hidden in the data generated by the telecom industries, there is a lot of scope for the researchers to analyze the data in different perspectives and to help the operators to improve their business in various ways.

Problem Statement

In a business environment, the term customer attrition simply refers to the customers leaving one business service to another. Customer churn or subscriber churn is also similar to attrition, which is the process of customers switching from one service provider to another anonymously. From a machine learning perspective, churn prediction is a supervised (i.e. labeled) problem defined as follows: **Given a predefined forecast horizon, the goal is to predict the future churners over that horizon, given the data associated with each subscriber in the network.**

Current Market

In today's market, churning of a customer is generally denoted by **churn rate**. Churn rates are often used to indicate the strength of a company's customer service division and its overall growth prospects. Lower churn rates suggest a company is, or will be, in a better or stronger competitive state. Customer loss impacts carriers significantly as they often make a significant investment to acquire customers.

The ability to predict that a particular customer is at a high risk of churning, while there is still time to do something about it, represents a huge additional potential revenue source for every online business. Besides the direct loss of revenue that results from a customer abandoning the business, the costs of initially acquiring that customer may not have already been covered by the customer's spending to date. (In other words, acquiring that customer may have actually been a losing investment.) Furthermore, it is always more difficult and expensive to acquire a new customer than it is to retain a current paying customer.

Clearly, churn rate is a critical metric for any subscription business. So, there are also a variety of opinions about how to calculate it.

- **CLTV:** Understanding customer lifetime value (LTV) is one of the most complex and important analyses a business can take on. Every part of your organization affects the outcome of the calculation: acquisition costs, revenue, customer service, and returns. It's an accurate approach to customer churn prediction- at the core it has the ability to predict which customers will churn. The approach takes into consideration both micro- segmentation and their behavior pattern. By merging the most exacting micro-segmentation available anywhere with a deep understanding of how customers move from one micro-segment to another over time – including the ability to predict those moves before they occur – an unprecedented degree of accuracy in customer churn prediction is attainable. Figuring out which one will stay for long and will reap how much revenue, helps the service provider to judge whether spending on a customer is worth the effort or not.

- **CVM:** Customer value management (CVM) is a holistic way of evaluating individual subscribers in terms of their overall profitability- now and in the future. CVM has the potential to boost earnings. This measure covers subscribers at every stage of their relationship with the operator. Relying on a combination of tactics, including customer payback period, budget re balancing, tailored customer rewards, and cross- and up-selling campaigns. CVM technique help companies analyze which customers are the most valuable, and why. Indeed, this approach is a key capability in a world where the potential customer base simply isn't getting any bigger.

- **Predictive Churn Modeling**: Predictive technology is a body of tools capable of discovering and analyzing patterns in data so that past behavior can be used to forecast likely future behavior. Predictive technology is increasingly used for forecasting in most of the Telecom companies' balance sheet. The raw data can be processed to get predictions about consumer behavior for future campaigns.

- **Postpaid and blended churn rates:** This churn rate is based upon the losses of both prepaid and contract customer. Post-paid subscribers are a telecom company's one of the biggest revenue segments since they have a significant lifetime value for telecom companies. Their discontinuation of services accounts for a major loss in company's revenue.

- **ARPU:** Average Revenue per User or ARPU or average revenue per unit is an expression of the income generated by a typical subscriber or device per unit time in a telecommunications network. ARPU provides an indication of the effectiveness with which revenue-generating potential is exploited. The ARPU can be broken down according to income-producing categories or according to diverse factors such as geographic location, user age, user occupation, user income and the total time per month each user spends on the system.

- **AMPU:** The Average Margin per User is calculated on the basis of net profit rather than total income. In recent years, some telecommunications carriers have increased their reliance on AMPU rather than ARPU to maximize their returns as niche markets become saturated. By breaking down customer sales by margin rather than by revenue, companies that have lower sales volumes but create larger margins can be considered more efficient and arguably more profitable than their high-volume competitors.

- **Real- Time Data:** Real-time data in customer churn makes the best possible solution today, as it is based on up-to-the-moment information about a subscriber. Achieving real-time data enables the company to immediately adjust it's offers and solutions in response to the reason of dissatisfaction/ discontinuation of services. Deploying analytics and systems that trigger the moment your subscriber is shifting to your competitor, helps process the retention effective and faster.

- **Binary classification method:** This method uses a gain/loss matrix, which incorporates the gain of targeting and retaining the most valuable churners and the cost of incentives to the targeted customers. This approach leads to far more profitable retention campaigns than the traditional churn modeling approaches.

- **General Signs:** Customers today are highly conscious of what they need and what is available in the market. The telecom players should always lookout for signs that the customer may be planning to shift. These can easily be picked up from sales support interaction with them- when he bluntly says he is shifting to a competitor, when he is quoting what other players offer, when he is enquiring about MNP or when he is simply calling competitor's phone line looking for alternatives to his problem.

Method

The churn prediction problem represented here involves 2 phases, namely, i) training phase, ii) test phase. The input for this problem includes the data on past calls for each mobile subscriber, together with all personal and business information that is maintained by the service provider. In addition, for the training phase, labels are provided in the form of a list of churners. After the model is trained with highest accuracy, the model must be able to predict the list of churners from the real data set which does not include any churn label. In the perspective of knowledge discovery process, this problem is categorized as predictive mining or predictive modeling. Churn Prediction is a phenomenon which is used to identify the possible churners in advance before they leave the network.

First, the data needs to be aggregated. The data provided is given in 4 separate datasets - one containing the demographic details of each customer, second containing the account information of each customer. The third set contains the details of the services that each customer has opted for, while the fourth dataset has the details of whether the customer has churned or not. Moreover, separate data has been provided for customers for whom it needs to be predicted whether they will churn in the immediate future or not.

Then the combined data needs to be cleaned and processed by checking out the missing values and outliers. Many alternative methods are present to deal with both missing values and outliers. The records with missing values can be omitted, imputed or can be left as it is. Outliers can also be dealt with by omission, capping or minimizing the effects of those records. New variables can also be derived for better classification of the churn.

Different types of models will be implemented on the readied dataset including naive bayes, logistic regression, decision trees, random forest, gradient boosted trees and extreme gradient boosting. The results will be compared and the final conclusions will be made.

Dataset Description

The data has been provided into 4 different data sets. These data sets contain the demographic information of 5298 customers, along with their account information, the services they opted for, and in the case of training set - whether they will churn or not. The test data set also contains the same information of 1769 other customers. The attribute description is as follows :

Demographics Data :

* HouseholdID : Each Household id
* Country : Country of each customer
* State : State of each customer
* Retired : Whether the customer is retired or not
* HasPartner : Whether the customer has partner or not
* HasDependents : Whether the customer has dependents or not
* Education : Education qualification of each customer
* Gender : Gender of each customer

Account Information :

* CustomerID : Customer ID of each customer

* BaseCharges : Charges for Base plan
* DOC : Date of data collection
* TotalCharges : Total charges of each customer
* DOE : Date of entry as customer
* ElectronicBilling : Whether customer has opted for electronic billing or not
* ContractType : Type of contract opted by the customer
* PaymentMethod : Method of payment opted by the customer

Data of ServicedOptedFor :

* CustomerID : Customer ID
* TypeOfService : Service signed for by the customer
* SeviceDetails : Details of each type of service

Churn Data :

* CustomerID : Customer ID
* Churn : Whether the customer churns (Only on training set)

Analysis

There were some apparent factors involved in churning of the customers which were found in univariate and bi-variate analysis. This analysis was done just to analyze which factors were unusually contributing in churning of a customer.

Base Charges : There are no outliers in the base charges of the customers. However, most of the customers who did not churn had a lower base charge as compared to the people who churned. So, regulation in base charges or cheaper base plans may help in retaining the customers.

Contract Type : Most of the customers chose month-to-month plans as opposed to yearly and two-year plans. But, further observations reveal that the number of people churning were quite less for yearly and two-year plans than for month-to-month users. So, having more attractive yearly and two-year plans might help in making people choose yearly plans.

Device Protection : There were as many customers who did not opt for device protection than who did. But the customers who did opt for device protection were less likely to churn than the ones who do not. Further, many customers did not opt for internet services. They were found to be not likely to churn.

Education : Customers ranged from every level of education facilities. But customers who were educated till high-school or below were more likely to churn than the others.

Internet : There were more number of customers who opted for fiber optic cables rather than DSL connections. But they were the ones who were more likely to churn.

Online Backup : Customers who had not opted for online backup were more likely to churn than the ones who did.

Online Security : Customers who did not opt for online security were also more likely to churn than the ones who did.

Payment Method : Customers who opted for electronic cheques were more likely to churn than the customers who opted for other modes of payment.

Technical Support : Customers who did not or were not able to get technical support were more likely to churn than the others who received it.

Classification

The data set was first aggregated to form a combined data set. The attributes were converted to their respective data types. Two new attributes were made. First is **diff** - number of days between the date of collection and the date of customer entry. The other attribute is **charge** - the difference between base charge and total charge of each customer.

Random Forest gave the best results for classification of customers. Random forests are an ensemble learning method for classification other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set.

Here, random forest was implemented on different versions of the data set. Most notable ones dealt with the problem of imputation of missing values and the problem of class imbalance. For imputation of missing values, missForest package was used while ROSE package was used for dealing with class imbalance.

missForest package uses a random forest trained on the observed values of a data matrix to predict the missing values. It can be used to impute continuous and/or categorical data including complex interactions and non-linear relations. It yields an out-of-bag (OOB) imputation error estimate without the need of a test set or elaborate cross-validation. It can be run in parallel to save computation time.

ROSE (Random Over Sampling Examples) package helps us to generate artificial data based on sampling methods and smoothed bootstrap approach.

The results derived from the random forest model had the best accuracy of 75.635% while the recall of class *yes* was 77.378%.

The other prominent model that gave good results was naive bayes model. Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of independence between every pair of features. Assuming the independence of features, the results got were on par with random forest model. The best accuracy gotten was 76.823% while the recall was 75.044%.

Results

Many other models were also implemented but random forest and naive bayes classifier gave the best all-around results. Although both the models can be considered as a final model and naive bayes seems to give a minutely better result than random forest, random forest can be considered as the best model for this objective.

Although random forest is considered to be a black box and a considerably complex model, and occam’s razor implies that a simpler model should be preferred, there is a very strong assumption in the case of naive bayes - all the features are independent. This is more often than not false in real world. More amount of data can help in clarifying the choice of the model.

Patterns were also generated by implementing C5.0 model and using associative rules on the training set.

Observations from univariate and bivariate analysis can also be very beneficial for retention of customers.

Appendices

```
#Reading file path
setwd(’F:/PHD/Hack/Data’)
getwd()
############# AGGREGATION
#Reading the files
data1<-read.csv(’Train.csv’)
data2<-read.csv(’Train_AccountInfo.csv’)
data3<-read.csv(’Train_Demographics.csv’,na.strings = c(’?’,’‘,’ ‘,’MISSINGVAL’))
data4<-read.csv(’Train_ServicesOptedFor.csv’)
#Reshaping data4
library(reshape2)
data4_reshaped<-dcast(data4,CustomerID~data4$TypeOfService)
#Merging all the files
```

```
data3$HouseholdID<-NULL
data4_reshaped$CustomerID<-NULL
data1$CustomerID<-NULL
train_new<-cbind(data2,data3,data4_reshaped,data1)
#Reading test files
test1<-read.csv(’Test.csv’)
test2<-read.csv(’Test_AccountInfo.csv’)
test3<-read.csv(’Test_Demographics.csv’,na.strings = c(’?’,’‘,’ ‘,’MISSINGVAL’))
test4<-read.csv(’Test_ServicesOptedFor.csv’)
#Reshaping test4
test4_reshaped<-dcast(test4,CustomerID~test4$TypeOfService)
#Merging all test files
test3$HouseholdID<-NULL
test4_reshaped$CustomerID<-NULL
test1$CustomerID<-NULL
test_new<-cbind(test2,test3,test4_reshaped,test1)
rm(data1,data2,data3,data4,data4_reshaped,test1,test2,test3,test4,test4_reshaped)
############### ANALYSIS
#changing date column from factors to dates
train_new$DOC<-as.Date(train_new$DOC, format=“%d/%m/%Y”)
train_new$DOE<-as.Date(as.character(train_new$DOE), format=’%d-%b-%y’)
test_new$DOC<-as.Date(test_new$DOC, format=“%d/%m/%Y”)
test_new$DOE<-as.Date(as.character(test_new$DOE), format=’%d-%b-%y’)
str(train_new)
str(test_new)
#making new variable explaining the number of days the customer had the network
train_new$diff<-train_new$DOC-train_new$DOE
train_new$diff<-as.numeric(train_new$diff)
test_new$diff<-test_new$DOC-test_new$DOE
test_new$diff<-as.numeric(test_new$diff)
#deleting country and state (no variance)
train_new$Country<-NULL
train_new$State<-NULL
test_new$Country<-NULL
```

```
test_new$State<-NULL
#Changing various columns to factor
colnames(train_new)
colnames(test_new)
x<-colnames(train_new[14:22])
x<-append(x,colnames(train_new[9:11]))
train_new[x]<-lapply(train_new[x], factor)
test_new[x]<-lapply(test_new[x], factor)
#Changing total charges to numeric
train_new$TotalCharges<-as.numeric(as.character(train_new$TotalCharges))
test_new$TotalCharges<-as.numeric(as.character(test_new$TotalCharges))
################ MODEL 1
#Using h2o for faster computation
library(h2o)
#Initializing h2o and allowing all cores to run parallely
h2o.init(nthreads = -1)
#Selecting the independent and target variable(s)
colnames(train_new)
colnames(test_new)
y.dep<-23
x.indep<-c(2,4,6:22,24)
#Reading datasets as h2o frames
train.h2o<-as.h2o(train_new)
test.h2o<-as.h2o(test_new)
#Training the model
rforest.model <- h2o.randomForest(y=y.dep, x=x.indep, training_frame = train.h2o, ntrees = 1000, mtries = 3, max_depth = 4, seed = 1122)
#Checking the performance of the model
h2o.performance(rforest.model)
#Predicting the values
predict.rforest <- as.data.frame(h2o.predict(rforest.model, test.h2o))
#Creating a submission
improved_1 <- data.frame(CustomerID = test_new$CustomerID,Churn =  predict.rforest$predict)
write.csv(improved_1,file=’improved_1.csv’,row.names = F)
#Shutting down h2o
```

```
h2o.shutdown()

y

####################### MODEL 2

library(h2o)

h2o.init(nthreads = -1)

colnames(train_new)

colnames(test_new)

y.dep<-23

x.indep<-c(2,4,6:22,24)

train.h2o<-as.h2o(train_new)

test.h2o<-as.h2o(test_new)

rforest.model2 <- h2o.randomForest(y=y.dep, x=x.indep, training_frame = train.h2o, ntrees = 2500, mtries = 5, max_depth = 4, seed = 1124)

#h2o.performance(rforest.model2)

predict.rforest2 <- as.data.frame(h2o.predict(rforest.model2, test.h2o))

improved_2 <- data.frame(CustomerID = test_new$CustomerID,Churn =  predict.rforest2$predict)

write.csv(improved_2,file=’improved_2.csv’,row.names = F)

h2o.shutdown()

y

####################### MODEL 3

library(h2o)

h2o.init(nthreads = -1)

colnames(train_new)

colnames(test_new)

y.dep<-23

x.indep<-c(6:22,24)

train.h2o<-as.h2o(train_new)

test.h2o<-as.h2o(test_new)

naive_model <- h2o.naiveBayes(x=x.indep, y=y.dep, train.h2o)

predict.naive <- as.data.frame(h2o.predict(naive_model, test.h2o))

improved_3 <- data.frame(CustomerID = test_new$CustomerID,Churn =  predict.naive$predict)

write.csv(improved_3,file=’improved_3.csv’,row.names = F)

h2o.shutdown()

y

###################### MODEL 4
```

```
#Making a new dataframe
train_naive<-train_new
str(train_naive)
#Removing numeric columns
train_naive$BaseCharges<-NULL
train_naive$TotalCharges<-NULL
train_naive$DOC<-NULL
train_naive$DOE<-NULL
#Checking histogram of new variable created earlier
hist(train_naive$diff)
#Binning into equal frequencies
library(infotheo)
diff<-discretize(train_naive$diff,’equalfreq’,nbins = 4)
train_naive$diff<-diff$X
train_naive$diff<-as.factor(train_naive$diff)
#Doing same to test
test_naive<-test_new
str(test_naive)
test_naive$BaseCharges<-NULL
test_naive$TotalCharges<-NULL
test_naive$DOC<-NULL
test_naive$DOE<-NULL
hist(test_naive$diff)
library(infotheo)
diff<-discretize(test_naive$diff,’equalfreq’,nbins = 4)
test_naive$diff<-diff$X
test_naive$diff<-as.factor(test_naive$diff)
#implementing naive bayes
h2o.init(nthreads = -1)
colnames(train_naive)
colnames(test_naive)
y.dep<-19
x.indep<-c(2:18,20)
train.h2o<-as.h2o(train_naive)
```

```
test.h2o<-as.h2o(test_naive)
naive_model2 <- h2o.naiveBayes(x=x.indep, y=y.dep, train.h2o)
predict.naive2 <- as.data.frame(h2o.predict(naive_model2, test.h2o))
improved_4 <- data.frame(CustomerID = test_new$CustomerID,Churn =  predict.naive2$predict)
write.csv(improved_4,file=’improved_4.csv’,row.names = F)
h2o.shutdown()
y
############################# MODEL 5
#Reinitializing dataframe
train_naive<-train_new
str(train_naive)
#Removing date columnms
train_naive$DOC<-NULL
train_naive$DOE<-NULL
#Binning base and total charges also
#Binning diff column
library(infotheo)
diff<-discretize(train_naive$diff,’equalfreq’,nbins = 4)
train_naive$diff<-diff$X
train_naive$diff<-as.factor(train_naive$diff)
#Binning base charges
base<-discretize(train_naive$BaseCharges,’equalfreq’,nbins = 4)
train_naive$BaseCharges<-base$X
train_naive$BaseCharges<-as.factor(train_naive$BaseCharges)
#Binning total charges
tot<-discretize(train_naive$TotalCharges,’equalfreq’,nbins = 4)
train_naive$TotalCharges<-tot$X
train_naive$TotalCharges<-as.factor(train_naive$TotalCharges)
#Doing same with test
test_naive<-test_new
str(test_naive)
test_naive$DOC<-NULL
test_naive$DOE<-NULL
diff<-discretize(test_naive$diff,’equalfreq’,nbins = 4)
```

```
test_naive$diff<-diff$X
test_naive$diff<-as.factor(test_naive$diff)
base<-discretize(test_naive$BaseCharges,’equalfreq’,nbins = 4)
test_naive$BaseCharges<-base$X
test_naive$BaseCharges<-as.factor(test_naive$BaseCharges)
tot<-discretize(test_naive$TotalCharges,’equalfreq’,nbins = 4)
test_naive$TotalCharges<-tot$X
test_naive$TotalCharges<-as.factor(test_naive$TotalCharges)
#implementing naive bayes model
h2o.init(nthreads = -1)
colnames(train_naive)
colnames(test_naive)
y.dep<-21
x.indep<-c(2:20,22)
train.h2o<-as.h2o(train_naive)
test.h2o<-as.h2o(test_naive)
naive_model3 <- h2o.naiveBayes(x=x.indep, y=y.dep, train.h2o)
predict.naive3 <- as.data.frame(h2o.predict(naive_model3, test.h2o))
improved_5 <- data.frame(CustomerID = test_new$CustomerID,Churn =  predict.naive3$predict)
write.csv(improved_5,file=’improved_5.csv’,row.names = F)
h2o.shutdown()
y
######################### MODEL 6
h2o.init(nthreads = -1)
colnames(train_naive)
colnames(test_naive)
y.dep<-21
x.indep<-c(2:20,22)
train.h2o<-as.h2o(train_naive)
test.h2o<-as.h2o(test_naive)
#implementing random forest model
rforest.model3 <- h2o.randomForest(y=y.dep, x=x.indep, training_frame = train.h2o, ntrees = 1000, mtries = 5, max_depth = 4, seed = 1127)
h2o.performance(rforest.model3)
h2o.varimp_plot(rforest.model3)
```

```
predict.rforest3 <- as.data.frame(h2o.predict(rforest.model3, test.h2o))
improved_6 <- data.frame(CustomerID = test_new$CustomerID,Churn =  predict.rforest3$predict)
write.csv(improved_6,file=’improved_6.csv’,row.names = F)
h2o.shutdown()
y
############################ MODEL 7
#Reinitializing dataframe
train_naive<-train_new
str(train_naive)
#Removing date columnms
train_naive$DOC<-NULL
train_naive$DOE<-NULL
#Making a new variable charge as difference of base and total charges
train_naive$Charge<-train_naive$TotalCharges-train_naive$BaseCharges
#Removing base and total charges
train_naive$BaseCharges<-NULL
train_naive$TotalCharges<-NULL
diff<-discretize(train_naive$diff,’equalfreq’,nbins = 4)
train_naive$diff<-diff$X
train_naive$diff<-as.factor(train_naive$diff)
#Binning base charges
charge<-discretize(train_naive$Charge,’equalfreq’,nbins = 4)
train_naive$Charge<-charge$X
train_naive$Charge<-as.factor(train_naive$Charge)
#Doing same for test
test_naive<-test_new
str(test_naive)
#Removing date columnms
test_naive$DOC<-NULL
test_naive$DOE<-NULL
#Making a new variable charge as difference of base and total charges
test_naive$Charge<-test_naive$TotalCharges-test_naive$BaseCharges
#Removing base and total charges
test_naive$BaseCharges<-NULL
```

```
test_naive$TotalCharges<-NULL
diff<-discretize(test_naive$diff,’equalfreq’,nbins = 4)
test_naive$diff<-diff$X
test_naive$diff<-as.factor(test_naive$diff)
#Binning base charges
charge<-discretize(test_naive$Charge,’equalfreq’,nbins = 4)
test_naive$Charge<-charge$X
test_naive$Charge<-as.factor(test_naive$Charge)
#implementing naive bayes model
h2o.init(nthreads = -1)
colnames(train_naive)
colnames(test_naive)
y.dep<-19
x.indep<-c(2:18,20,21)
train.h2o<-as.h2o(train_naive)
test.h2o<-as.h2o(test_naive)
naive_model4 <- h2o.naiveBayes(x=x.indep, y=y.dep, train.h2o)
predict.naive4 <- as.data.frame(h2o.predict(naive_model4, test.h2o))
improved_7 <- data.frame(CustomerID = test_new$CustomerID,Churn =  predict.naive4$predict)
write.csv(improved_7,file=’improved_7.csv’,row.names = F)
h2o.shutdown()
y
########################### MODEL 8
#random forest model
colnames(train_naive)
colnames(test_naive)
y.dep<-19
x.indep<-c(2:18,20,21)
train.h2o<-as.h2o(train_naive)
test.h2o<-as.h2o(test_naive)
rforest.model4 <- h2o.randomForest(y=y.dep, x=x.indep, training_frame = train.h2o, ntrees = 1000, mtries
= 5, max_depth = 4, seed = 1127)
h2o.performance(rforest.model4)
h2o.varimp_plot(rforest.model4)
predict.rforest4 <- as.data.frame(h2o.predict(rforest.model4, test.h2o))
```

```
improved_8 <- data.frame(CustomerID = test_new$CustomerID,Churn =  predict.rforest4$predict)
write.csv(improved_8,file=’improved_8.csv’,row.names = F)
h2o.shutdown()
y
######################### MODEL 9
#naive nayes model
h2o.init(nthreads = -1)
colnames(train_naive)
colnames(test_naive)
y.dep<-19
x.indep<-c(3,4,8,10,12,14,15,18,20,21)
train.h2o<-as.h2o(train_naive)
test.h2o<-as.h2o(test_naive)
naive_model5 <- h2o.naiveBayes(x=x.indep, y=y.dep, train.h2o)
predict.naive5 <- as.data.frame(h2o.predict(naive_model5, test.h2o))
improved_9 <- data.frame(CustomerID = test_new$CustomerID,Churn =  predict.naive5$predict)
write.csv(improved_9,file=’improved_9.csv’,row.names = F)
h2o.shutdown()
y
```