Assignment\_Week4

Abhimanyu Kumar

2/4/2018

## Customer churn

Customer churn occurs when a customer stop doing business with services or company. In industries like telecommunication or insurance, churn is really useful as customer has option to choose from multiple service providers based on different factor and geographical area. I have used a telecom data set which i have downloaded from IBM sample dataset.

In the below code I have combined all the packages and libraries which I have used in the code below.

install.packages("plyr", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/mn/7jk0pyy53cjbzyt8ryfh3wzc0000gn/T//RtmpZXsGDM/downloaded\_packages

library(plyr)  
install.packages("corrplot", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/mn/7jk0pyy53cjbzyt8ryfh3wzc0000gn/T//RtmpZXsGDM/downloaded\_packages

library(corrplot)

## corrplot 0.84 loaded

install.packages("ggplot2", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/mn/7jk0pyy53cjbzyt8ryfh3wzc0000gn/T//RtmpZXsGDM/downloaded\_packages

library(ggplot2)  
install.packages("gridExtra", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/mn/7jk0pyy53cjbzyt8ryfh3wzc0000gn/T//RtmpZXsGDM/downloaded\_packages

library(gridExtra)  
install.packages("ggthemes", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/mn/7jk0pyy53cjbzyt8ryfh3wzc0000gn/T//RtmpZXsGDM/downloaded\_packages

library(ggthemes)  
install.packages("caret", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/mn/7jk0pyy53cjbzyt8ryfh3wzc0000gn/T//RtmpZXsGDM/downloaded\_packages

library(caret)

## Loading required package: lattice

install.packages("MASS", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/mn/7jk0pyy53cjbzyt8ryfh3wzc0000gn/T//RtmpZXsGDM/downloaded\_packages

library(MASS)  
install.packages("randomForest", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/mn/7jk0pyy53cjbzyt8ryfh3wzc0000gn/T//RtmpZXsGDM/downloaded\_packages

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:gridExtra':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

install.packages("party", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/mn/7jk0pyy53cjbzyt8ryfh3wzc0000gn/T//RtmpZXsGDM/downloaded\_packages

library(party)

## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

##   
## Attaching package: 'modeltools'

## The following object is masked from 'package:plyr':  
##   
## empty

## Loading required package: strucchange

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: sandwich

install.packages("e1071", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/mn/7jk0pyy53cjbzyt8ryfh3wzc0000gn/T//RtmpZXsGDM/downloaded\_packages

library(e1071)  
library(knitr)  
#install.packages("rproject", repos = "http://cran.us.r-project.org")  
#library(rproject)

## Reading the data set

## 'data.frame': 7043 obs. of 21 variables:  
## $ customerID : Factor w/ 7043 levels "0002-ORFBO","0003-MKNFE",..: 5376 3963 2565 5536 6512 6552 1003 4771 5605 4535 ...  
## $ gender : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...  
## $ SeniorCitizen : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Partner : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...  
## $ Dependents : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...  
## $ tenure : int 1 34 2 45 2 8 22 10 28 62 ...  
## $ PhoneService : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...  
## $ MultipleLines : Factor w/ 3 levels "No","No phone service",..: 2 1 1 2 1 3 3 2 3 1 ...  
## $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 1 1 1 2 2 2 1 2 1 ...  
## $ OnlineSecurity : Factor w/ 3 levels "No","No internet service",..: 1 3 3 3 1 1 1 3 1 3 ...  
## $ OnlineBackup : Factor w/ 3 levels "No","No internet service",..: 3 1 3 1 1 1 3 1 1 3 ...  
## $ DeviceProtection: Factor w/ 3 levels "No","No internet service",..: 1 3 1 3 1 3 1 1 3 1 ...  
## $ TechSupport : Factor w/ 3 levels "No","No internet service",..: 1 1 1 3 1 1 1 1 3 1 ...  
## $ StreamingTV : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 3 1 3 1 ...  
## $ StreamingMovies : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 1 1 3 1 ...  
## $ Contract : Factor w/ 3 levels "Month-to-month",..: 1 2 1 2 1 1 1 1 1 2 ...  
## $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1 ...  
## $ PaymentMethod : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 4 4 1 3 3 2 4 3 1 ...  
## $ MonthlyCharges : num 29.9 57 53.9 42.3 70.7 ...  
## $ TotalCharges : num 29.9 1889.5 108.2 1840.8 151.7 ...  
## $ Churn : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1 ...

Here I am hvaing 7043 rows (customers) and 21 columns (features or variables). My main target is the churn column.

I have used sapply below to check the number of missing values in each columns. I found 11 missing values in “TotalCharges” columns and I removed all the rows with missing value.

sapply(churn, function(x) sum(is.na(x)))

## customerID gender SeniorCitizen Partner   
## 0 0 0 0   
## Dependents tenure PhoneService MultipleLines   
## 0 0 0 0   
## InternetService OnlineSecurity OnlineBackup DeviceProtection   
## 0 0 0 0   
## TechSupport StreamingTV StreamingMovies Contract   
## 0 0 0 0   
## PaperlessBilling PaymentMethod MonthlyCharges TotalCharges   
## 0 0 0 11   
## Churn   
## 0

churn <- churn[complete.cases(churn), ]

After closely looking at the data i noticed that some wrangling is needed here.

1.) I substituted “No internet service to”No" for six columns and they are: “OnlineSecurity”, “OnlineBackup”, “DeviceProtection”, “TechSupport”, “streamingTV”, “streamingMovies”.

cols\_recode1 <- c(10:15)  
for(i in 1:ncol(churn[,cols\_recode1])) {  
 churn[,cols\_recode1][,i] <- as.factor(mapvalues  
 (churn[,cols\_recode1][,i], from =c("No internet service"),to=c("No")))  
}

2.) “No phone service” to “No” for column “MultipleLines”

churn$MultipleLines <- as.factor(mapvalues(churn$MultipleLines,   
 from=c("No phone service"),  
 to=c("No")))

1. Since the minimum tenure is 1 month and maximum tenure is 72 months, we can group them into five tenure groups: “0–12 Month”, “12–24 Month”, “24–48 Months”, “48–60 Month”, “> 60 Month”

min(churn$tenure); max(churn$tenure)

## [1] 1

## [1] 72

churn$tenure\_group <-cut(churn$tenure,   
breaks = c(0, 12, 24, 48, 60, Inf),  
labels = c("0–12 Month",  
"12–24 Month",  
"24–48 Months",  
"48–60 Month",  
"> 60 Month"))

4.)Change the values in column “SeniorCitizen” from 0 or 1 to “No” or “Yes”

churn$SeniorCitizen <- as.factor(mapvalues(churn$SeniorCitizen,  
 from=c("0","1"),  
 to=c("No", "Yes")))

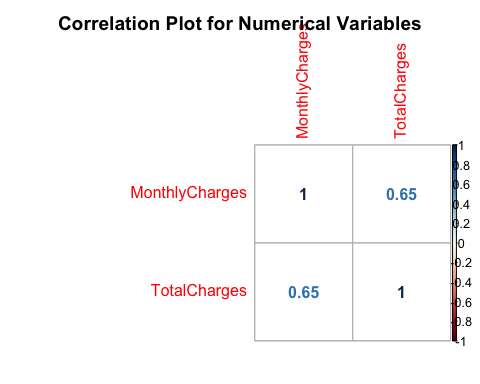
1. Remove the columns we do not need for the analysis.

churn$customerID <- NULL  
churn$tenure <- NULL

## Exploratory data analysis and feature selection

Correlation between numeric variables

numeric.var <- sapply(churn, is.numeric)  
corr.matrix <- cor(churn[,numeric.var])  
corrplot(corr.matrix, main="\n\nCorrelation Plot for Numerical Variables", method="number")

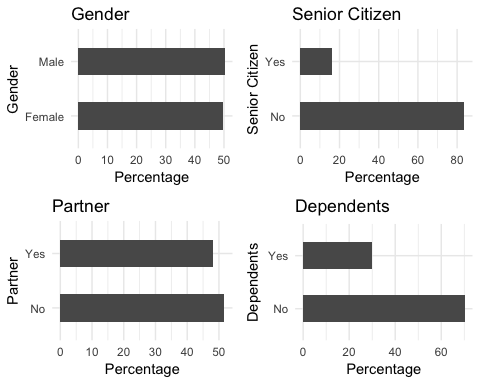


The Monthly Charges and Total Charges are correlated. So one of them will be removed from the model. We remove Total Charges.

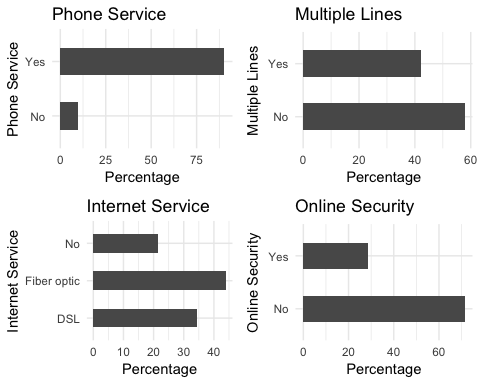
churn$TotalCharges <- NULL

Bar plots of categorical variables

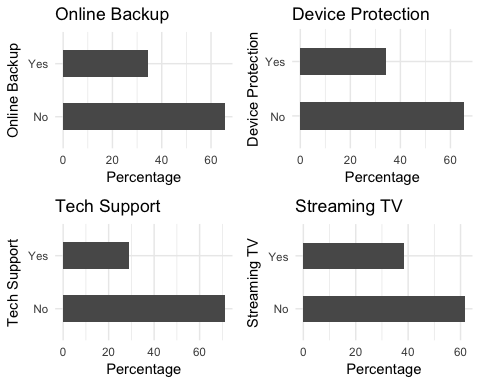
p1 <- ggplot(churn, aes(x=gender)) + ggtitle("Gender") + xlab("Gender") +  
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p2 <- ggplot(churn, aes(x=SeniorCitizen)) + ggtitle("Senior Citizen") + xlab("Senior Citizen") +   
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p3 <- ggplot(churn, aes(x=Partner)) + ggtitle("Partner") + xlab("Partner") +   
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p4 <- ggplot(churn, aes(x=Dependents)) + ggtitle("Dependents") + xlab("Dependents") +  
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
grid.arrange(p1, p2, p3, p4, ncol=2)



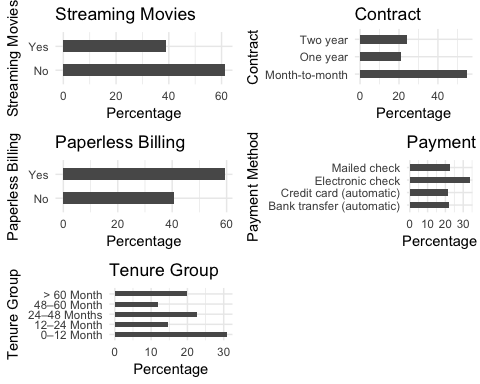
p5 <- ggplot(churn, aes(x=PhoneService)) + ggtitle("Phone Service") + xlab("Phone Service") +  
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p6 <- ggplot(churn, aes(x=MultipleLines)) + ggtitle("Multiple Lines") + xlab("Multiple Lines") +   
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p7 <- ggplot(churn, aes(x=InternetService)) + ggtitle("Internet Service") + xlab("Internet Service") +   
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p8 <- ggplot(churn, aes(x=OnlineSecurity)) + ggtitle("Online Security") + xlab("Online Security") +  
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
grid.arrange(p5, p6, p7, p8, ncol=2)



p9 <- ggplot(churn, aes(x=OnlineBackup)) + ggtitle("Online Backup") + xlab("Online Backup") +  
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p10 <- ggplot(churn, aes(x=DeviceProtection)) + ggtitle("Device Protection") + xlab("Device Protection") +   
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p11 <- ggplot(churn, aes(x=TechSupport)) + ggtitle("Tech Support") + xlab("Tech Support") +   
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p12 <- ggplot(churn, aes(x=StreamingTV)) + ggtitle("Streaming TV") + xlab("Streaming TV") +  
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
grid.arrange(p9, p10, p11, p12, ncol=2)



p13 <- ggplot(churn, aes(x=StreamingMovies)) + ggtitle("Streaming Movies") + xlab("Streaming Movies") +  
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p14 <- ggplot(churn, aes(x=Contract)) + ggtitle("Contract") + xlab("Contract") +   
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p15 <- ggplot(churn, aes(x=PaperlessBilling)) + ggtitle("Paperless Billing") + xlab("Paperless Billing") +   
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p16 <- ggplot(churn, aes(x=PaymentMethod)) + ggtitle("Payment Method") + xlab("Payment Method") +  
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
p17 <- ggplot(churn, aes(x=tenure\_group)) + ggtitle("Tenure Group") + xlab("Tenure Group") +  
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()  
grid.arrange(p13, p14, p15, p16, p17, ncol=2)



As I can see all the categorical variables have a broad distribution, So, I will keep all of them for further analysis.

## Logistic Regression

First, we split the data into training and testing sets

intrain<- createDataPartition(churn$Churn,p=0.7,list=FALSE)  
set.seed(2017)  
training<- churn[intrain,]  
testing<- churn[-intrain,]

After that I will Confirm, if the splitting is correct or not??

dim(training); dim(testing)

## [1] 4924 19

## [1] 2108 19

Once after confirming I Fitted the Logistic Regression Model

LogModel <- glm(Churn ~ .,family=binomial(link="logit"),data=training)  
print(summary(LogModel))

##   
## Call:  
## glm(formula = Churn ~ ., family = binomial(link = "logit"), data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9711 -0.6764 -0.3058 0.6662 3.0263   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 0.84852 0.95995 0.884 0.37674  
## genderMale -0.06488 0.07740 -0.838 0.40192  
## SeniorCitizenYes 0.16957 0.10071 1.684 0.09222  
## PartnerYes -0.05241 0.09219 -0.569 0.56968  
## DependentsYes -0.16384 0.10752 -1.524 0.12755  
## PhoneServiceYes 0.52014 0.77157 0.674 0.50023  
## MultipleLinesYes 0.41106 0.21105 1.948 0.05145  
## InternetServiceFiber optic 1.91319 0.95078 2.012 0.04420  
## InternetServiceNo -1.99662 0.95935 -2.081 0.03741  
## OnlineSecurityYes -0.23261 0.21454 -1.084 0.27826  
## OnlineBackupYes 0.09597 0.20761 0.462 0.64390  
## DeviceProtectionYes 0.17985 0.21026 0.855 0.39235  
## TechSupportYes -0.10619 0.21571 -0.492 0.62252  
## StreamingTVYes 0.70726 0.38932 1.817 0.06927  
## StreamingMoviesYes 0.65079 0.38975 1.670 0.09496  
## ContractOne year -0.62291 0.12497 -4.984 6.22e-07  
## ContractTwo year -1.37891 0.20543 -6.712 1.92e-11  
## PaperlessBillingYes 0.27333 0.08927 3.062 0.00220  
## PaymentMethodCredit card (automatic) -0.09930 0.13469 -0.737 0.46097  
## PaymentMethodElectronic check 0.34139 0.11235 3.039 0.00238  
## PaymentMethodMailed check -0.02989 0.13725 -0.218 0.82761  
## MonthlyCharges -0.04199 0.03777 -1.112 0.26633  
## tenure\_group12–24 Month -0.99701 0.11732 -8.498 < 2e-16  
## tenure\_group24–48 Months -1.30773 0.11895 -10.994 < 2e-16  
## tenure\_group48–60 Month -1.61834 0.16980 -9.531 < 2e-16  
## tenure\_group> 60 Month -1.77896 0.20250 -8.785 < 2e-16  
##   
## (Intercept)   
## genderMale   
## SeniorCitizenYes .   
## PartnerYes   
## DependentsYes   
## PhoneServiceYes   
## MultipleLinesYes .   
## InternetServiceFiber optic \*   
## InternetServiceNo \*   
## OnlineSecurityYes   
## OnlineBackupYes   
## DeviceProtectionYes   
## TechSupportYes   
## StreamingTVYes .   
## StreamingMoviesYes .   
## ContractOne year \*\*\*  
## ContractTwo year \*\*\*  
## PaperlessBillingYes \*\*   
## PaymentMethodCredit card (automatic)   
## PaymentMethodElectronic check \*\*   
## PaymentMethodMailed check   
## MonthlyCharges   
## tenure\_group12–24 Month \*\*\*  
## tenure\_group24–48 Months \*\*\*  
## tenure\_group48–60 Month \*\*\*  
## tenure\_group> 60 Month \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 5702.8 on 4923 degrees of freedom  
## Residual deviance: 4135.7 on 4898 degrees of freedom  
## AIC: 4187.7  
##   
## Number of Fisher Scoring iterations: 6

## Feature Analysis

The top three most-relevant features include Contract, tenure\_group and PaperlessBilling.

anova(LogModel, test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: Churn  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 4923 5702.8   
## gender 1 1.45 4922 5701.3 0.228180   
## SeniorCitizen 1 83.80 4921 5617.5 < 2.2e-16 \*\*\*  
## Partner 1 123.35 4920 5494.2 < 2.2e-16 \*\*\*  
## Dependents 1 38.49 4919 5455.7 5.508e-10 \*\*\*  
## PhoneService 1 2.52 4918 5453.2 0.112553   
## MultipleLines 1 2.82 4917 5450.3 0.093319 .   
## InternetService 2 466.90 4915 4983.4 < 2.2e-16 \*\*\*  
## OnlineSecurity 1 187.63 4914 4795.8 < 2.2e-16 \*\*\*  
## OnlineBackup 1 65.18 4913 4730.6 6.838e-16 \*\*\*  
## DeviceProtection 1 44.59 4912 4686.0 2.428e-11 \*\*\*  
## TechSupport 1 71.10 4911 4614.9 < 2.2e-16 \*\*\*  
## StreamingTV 1 2.79 4910 4612.1 0.094644 .   
## StreamingMovies 1 0.03 4909 4612.1 0.857086   
## Contract 2 248.67 4907 4363.4 < 2.2e-16 \*\*\*  
## PaperlessBilling 1 10.52 4906 4352.9 0.001178 \*\*   
## PaymentMethod 3 40.90 4903 4312.0 6.850e-09 \*\*\*  
## MonthlyCharges 1 1.39 4902 4310.6 0.238003   
## tenure\_group 4 174.95 4898 4135.7 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Analyzing the deviance table we can see the drop in deviance when adding each variable one at a time. Adding InternetService, Contract and tenure\_group significantly reduces the residual deviance. The other variables such as PaymentMethod and Dependents seem to improve the model less even though they all have low p-values.

Assessing the predictive ability of the Logistic Regression model

testing$Churn <- as.character(testing$Churn)  
testing$Churn[testing$Churn=="No"] <- "0"  
testing$Churn[testing$Churn=="Yes"] <- "1"  
fitted.results <- predict(LogModel,newdata=testing,type='response')  
#fitted.results 0.5,1,0)  
fitted.results <- round(fitted.results, 0)  
fitted.results <- as.character(fitted.results)  
misClasificError <- mean(fitted.results != testing$Churn)  
print(paste('Logistic Regression Accuracy',1-misClasificError))

## [1] "Logistic Regression Accuracy 0.796015180265655"

After that I created a Logistic Regression Confusion Matrix

print("Confusion Matrix for Logistic Regression"); table(testing$Churn, fitted.results > 0.5)

## [1] "Confusion Matrix for Logistic Regression"

##   
## FALSE TRUE  
## 0 1406 142  
## 1 288 272

## Odds Ratio

One of the interesting performance measurements in logistic regression is Odds Ratio.Basically, Odds ratio is what the odds of an event is happening.

exp(cbind(OR=coef(LogModel), confint(LogModel)))

## Waiting for profiling to be done...

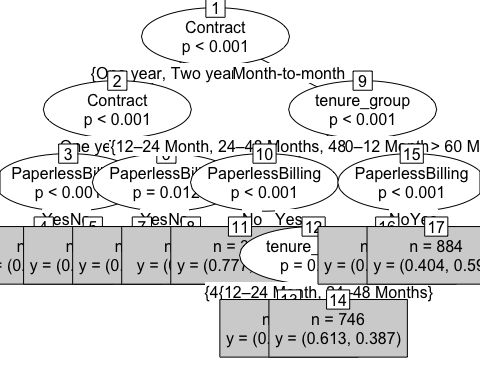
## OR 2.5 % 97.5 %  
## (Intercept) 2.3361915 0.35660274 15.3782288  
## genderMale 0.9371829 0.80523675 1.0907301  
## SeniorCitizenYes 1.1847939 0.97226577 1.4430271  
## PartnerYes 0.9489382 0.79208396 1.1369772  
## DependentsYes 0.8488741 0.68701225 1.0473178  
## PhoneServiceYes 1.6822571 0.37109135 7.6452445  
## MultipleLinesYes 1.5084141 0.99772101 2.2824961  
## InternetServiceFiber optic 6.7746989 1.05371557 43.8344431  
## InternetServiceNo 0.1357936 0.02066086 0.8888560  
## OnlineSecurityYes 0.7924598 0.52015097 1.2063440  
## OnlineBackupYes 1.1007252 0.73283743 1.6540407  
## DeviceProtectionYes 1.1970397 0.79286130 1.8082545  
## TechSupportYes 0.8992525 0.58895806 1.3722441  
## StreamingTVYes 2.0284220 0.94662391 4.3566497  
## StreamingMoviesYes 1.9170553 0.89382374 4.1205114  
## ContractOne year 0.5363828 0.41884934 0.6838179  
## ContractTwo year 0.2518539 0.16660319 0.3732259  
## PaperlessBillingYes 1.3143283 1.10364518 1.5661993  
## PaymentMethodCredit card (automatic) 0.9054725 0.69512195 1.1788970  
## PaymentMethodElectronic check 1.4069007 1.12975360 1.7552041  
## PaymentMethodMailed check 0.9705542 0.74186010 1.2707638  
## MonthlyCharges 0.9588833 0.89035520 1.0324915  
## tenure\_group12–24 Month 0.3689816 0.29271463 0.4637052  
## tenure\_group24–48 Months 0.2704324 0.21384271 0.3409186  
## tenure\_group48–60 Month 0.1982277 0.14154490 0.2755217  
## tenure\_group> 60 Month 0.1688141 0.11306533 0.2501875

## Decision Tree

## Decision Tree visualization

For illustration purpose,I use only three variables for plotting Decision Trees, they are “Contract”, “tenure\_group” and “PaperlessBilling”.

tree <- ctree(Churn~Contract+tenure\_group+PaperlessBilling, training)  
plot(tree, type='simple')



1. Out of three variables which I used, Contract is the most important variable to predict customer churn or not churn.
2. If a customer in a one-year or two-year contract, no matter he (she) has PapelessBilling or not, he (she) is less likely to churn.
3. On the other hand, if a customer is in a month-to-month contract, and in the tenure group of 0–12 month, and using PaperlessBilling, then this customer is more likely to churn.

## Decision Tree Confusion Matrix

I am using all the variables to product confusion matrix table and make predictions.

pred\_tree <- predict(tree, testing)  
print("Confusion Matrix for Decision Tree"); table(Predicted = pred\_tree, Actual = testing$Churn)

## [1] "Confusion Matrix for Decision Tree"

## Actual  
## Predicted 0 1  
## No 1406 354  
## Yes 142 206

## Decision Tree Accuracy

p1 <- predict(tree, training)  
tab1 <- table(Predicted = p1, Actual = training$Churn)  
tab2 <- table(Predicted = pred\_tree, Actual = testing$Churn)  
print(paste('Decision Tree Accuracy',sum(diag(tab2))/sum(tab2)))

## [1] "Decision Tree Accuracy 0.764705882352941"

The accuracy for Decision Tree has hardly improved. So, I am going to try Random Foret to see if I can do it any better using Random forest.

## Random Forest

Random Forest Initial Model

rfModel <- randomForest(Churn ~., data = training)  
print(rfModel)

##   
## Call:  
## randomForest(formula = Churn ~ ., data = training)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 21.49%  
## Confusion matrix:  
## No Yes class.error  
## No 3226 389 0.1076072  
## Yes 669 640 0.5110772

After implementing the random forest I observed that the error rate is relatively low when predicting “No”, and the error rate is much higher when predicting “Yes”.

## Random Forest Prediction and Confusion Matrix

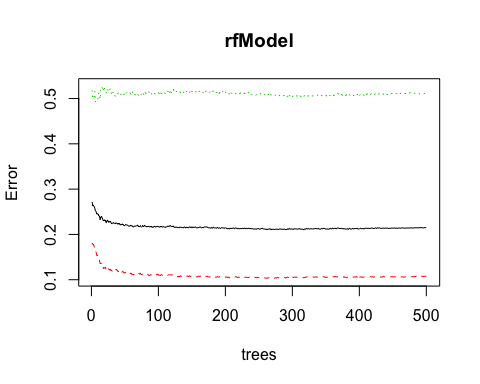
testing$Churn[testing$Churn == 0] <- "No"  
testing$Churn[testing$Churn == 1] <- "Yes"

pred\_rf <- predict(rfModel, testing)  
caret::confusionMatrix(pred\_rf, testing$Churn)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 1389 290  
## Yes 159 270  
##   
## Accuracy : 0.787   
## 95% CI : (0.7689, 0.8043)  
## No Information Rate : 0.7343   
## P-Value [Acc > NIR] : 1.252e-08   
##   
## Kappa : 0.41   
## Mcnemar's Test P-Value : 8.512e-10   
##   
## Sensitivity : 0.8973   
## Specificity : 0.4821   
## Pos Pred Value : 0.8273   
## Neg Pred Value : 0.6294   
## Prevalence : 0.7343   
## Detection Rate : 0.6589   
## Detection Prevalence : 0.7965   
## Balanced Accuracy : 0.6897   
##   
## 'Positive' Class : No   
##

After that I have plotted the Random Forest Error Rate below

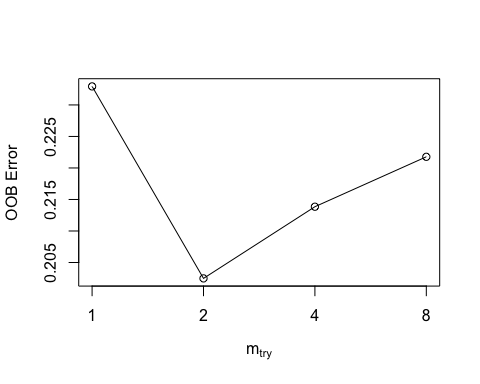
plot(rfModel)

 I used this plot to determine the number of trees. I noticed that as the number of trees increases, OOB error rate decreases, and then becomes almost constant.I am not able to decrease the OOB error even after 100 to 200 trees.

## Tune Random Forest Model

t <- tuneRF(training[, -18], training[, 18], stepFactor = 0.5, plot = TRUE, ntreeTry = 200, trace = TRUE, improve = 0.05)

## mtry = 4 OOB error = 21.39%   
## Searching left ...  
## mtry = 8 OOB error = 22.18%   
## -0.03703704 0.05   
## Searching right ...  
## mtry = 2 OOB error = 20.25%   
## 0.05318139 0.05   
## mtry = 1 OOB error = 23.29%   
## -0.1504514 0.05



I use the plot to get some idea on the number of mtry to choose. As the OOB error is lowest when mtry is 2 and so i choose the value oif mtry as 2.

Fit the Random Forest Model After Tuning

rfModel\_new <- randomForest(Churn ~., data = training, ntree = 200, mtry = 2, importance = TRUE, proximity = TRUE)  
print(rfModel\_new)

##   
## Call:  
## randomForest(formula = Churn ~ ., data = training, ntree = 200, mtry = 2, importance = TRUE, proximity = TRUE)   
## Type of random forest: classification  
## Number of trees: 200  
## No. of variables tried at each split: 2  
##   
## OOB estimate of error rate: 20.25%  
## Confusion matrix:  
## No Yes class.error  
## No 3308 307 0.08492393  
## Yes 690 619 0.52711994

Now i noticed that the OOB error rate decreased to 19.7% from 20.65% earlier.

Random Forest Predictions and Confusion Matrix After Tuning

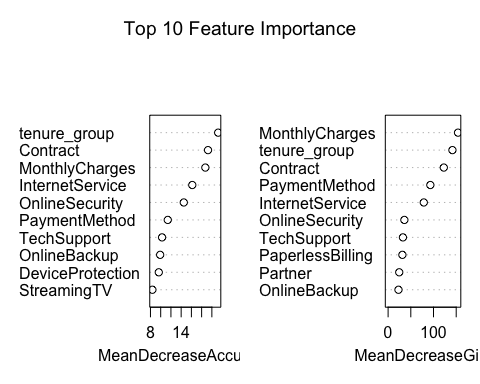
pred\_rf\_new <- predict(rfModel\_new, testing)  
caret::confusionMatrix(pred\_rf\_new, testing$Churn)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 1416 311  
## Yes 132 249  
##   
## Accuracy : 0.7898   
## 95% CI : (0.7718, 0.8071)  
## No Information Rate : 0.7343   
## P-Value [Acc > NIR] : 2e-09   
##   
## Kappa : 0.4002   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.9147   
## Specificity : 0.4446   
## Pos Pred Value : 0.8199   
## Neg Pred Value : 0.6535   
## Prevalence : 0.7343   
## Detection Rate : 0.6717   
## Detection Prevalence : 0.8193   
## Balanced Accuracy : 0.6797   
##   
## 'Positive' Class : No   
##

Here what i have noticed that the accuracy did not increase but the sensitivity improved, compare with the initial Random Forest model.

Random Forest Feature Importance

varImpPlot(rfModel\_new, sort=T, n.var = 10, main = 'Top 10 Feature Importance')



From the above example, I have noticed that logistic regression and random fores perfoms better than decission tree for churn analysis for this particular data set.

Throughout the analysis, I noticed and learned several things: 1. Features such as tenure\_group, Contract, PaperlessBilling, MonthlyCharges and InternetService appear to play a role in customer churn. 2. I didnt noticed any relationship between churn and gender. 3. Customers in a month-to-month contract, with PaperlessBilling and are within 12 months tenure, are more likely to churn; On the other hand, customers with one or two year contract, with longer than 12 months tenure, that are not using PaperlessBilling, are less likely to churn.

References:- <https://datascienceplus.com/predict-customer-churn-logistic-regression-decision-tree-and-random-forest/>