Predicting Customer Churn using Machine Learning Algorithms

Uche Kalu

4/4/2022

We are going to Predict Customer Churn using Telecom dataset. We will introduce Logistic Regression, Decision Tree, and Random Forest. Customer churn, also known as *customer attrition*, occurs when customers or subscribers stop doing business with a company or service, .

It is also referred as loss of clients or customers.

One industry in which churn rates are particularly useful is the telecommunications industry, because most customers have multiple options from which to choose within a geographic location.

```
library(plyr)
library(dplyr)
library(corrplot)
library(ggplot2)
library(gridExtra)
library(ggthemes)
library(caret)
library(MASS)
library(randomForest)
library(party)
```

DATA

```
mychurn <- read.csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
str(mychurn)
```

```
## 'data.frame':    7043 obs. of  21 variables:
##  $ customerID      : chr  "7590-VHVEG" "5575-GNVDE" "3668-QPYBK" "7795-CFOCW" ...
##  $ gender          : chr  "Female" "Male" "Male" "Male" ...
##  $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Partner         : chr  "Yes" "No" "No" "No" ...
##  $ Dependents      : chr  "No" "No" "No" "No" ...
##  $ tenure          : int  1 34 2 45 2 8 22 10 28 62 ...
##  $ PhoneService    : chr  "No" "Yes" "Yes" "No" ...
##  $ MultipleLines   : chr  "No phone service" "No" "No" "No phone service" ...
##  $ InternetService : chr  "DSL" "DSL" "DSL" "DSL" ...
##  $ OnlineSecurity  : chr  "No" "Yes" "Yes" "Yes" ...
##  $ OnlineBackup    : chr  "Yes" "No" "Yes" "No" ...
##  $ DeviceProtection: chr  "No" "Yes" "No" "Yes" ...
```

```
##  $ TechSupport     : chr  "No" "No" "No" "Yes" ...
##  $ StreamingTV     : chr  "No" "No" "No" "No" ...
##  $ StreamingMovies : chr  "No" "No" "No" "No" ...
##  $ Contract        : chr  "Month-to-month" "One year" "Month-to-month" "One year" ...
##  $ PaperlessBilling: chr  "Yes" "No" "Yes" "No" ...
##  $ PaymentMethod   : chr  "Electronic check" "Mailed check" "Mailed check" "Bank transfer (automatic)
##  $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
##  $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
##  $ Churn           : chr  "No" "No" "Yes" "No" ...
```

- customerID
- gender (female, male)
- SeniorCitizen (Whether the customer is a senior citizen or not (1, 0))
- Partner (Whether the customer has a partner or not (Yes, No))
- Dependents (Whether the customer has dependents or not (Yes, No))
- tenure (Number of months the customer has stayed with the company)
- PhoneService (Whether the customer has a phone service or not (Yes, No))
- MultipleLines (Whether the customer has multiple lines r not (Yes, No, No phone service)
- InternetService (Customers internet service provider (DSL, Fiber optic, No)
- OnlineSecurity (Whether the customer has online security or not (Yes, No, No internet service)
- OnlineBackup (Whether the customer has online backup or not (Yes, No, No internet service)
- DeviceProtection (Whether the customer has device protection or not (Yes, No, No internet service)
- TechSupport (Whether the customer has tech support or not (Yes, No, No internet service)
- streamingTV (Whether the customer has streaming TV or not (Yes, No, No internet service)
- streamingMovies (Whether the customer has streaming movies or not (Yes, No, No internet service)
- Contract (The contract term of the customer (Month-to-month, One year, Two year)
- PaperlessBilling (Whether the customer has paperless billing or not (Yes, No))
- PaymentMethod (The customers payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic)))
- MonthlyCharges (The amount charged to the customer monthly - numeric)
- TotalCharges (The total amount charged to the customer - numeric)
- Churn (Whether the customer churned or not (Yes or No))

The raw data contains 7043 rows (customers) and 21 columns (features). The **Churn** column is our target.

```
mychurn <- mychurn |> mutate(across(c(customerID, gender, Partner, Dependents,
                                      PhoneService:PaymentMethod, Churn),
                                    as.factor))
```

Checking for Missing Values

lets check the number of missing values in each column in the dataset and remove such missing rows of values

```
sapply(mychurn, function(x) sum(is.na(x)))

mychurn <- mychurn[complete.cases(mychurn),]
```

```
sapply(mychurn, 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                0
##            Churn
##                0
```

Data Wrangling

We will re-code some values in some columns and turn some values into intervals/groups

```
unique(mychurn['OnlineSecurity'])

cols_recode1 <- c(10:15)
for (i in 1:ncol(mychurn[, cols_recode1])) {
  mychurn[, cols_recode1][, i] <- as.factor(
    mapvalues(mychurn[, cols_recode1][, i],                                    from =
}

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

min(mychurn$tenure); max(mychurn$tenure)

group_tenure <- function(tenure){
  if(tenure >= 0 & tenure <= 12){
    return('0 - 12 Month')         # 1st group- 0 to 12 months
  }else if(tenure > 12 & tenure <= 24){
    return('12 - 24 Month')         # 2st group- 12 to 24 months
  }else if(tenure > 24 & tenure <= 48){
    return('24 - 48 Month')         # 3rd group- 24 to 48 months
  }else if(tenure > 48 & tenure <= 60){
    return('48 - 60 Month')        # 4th group- 48 to 60 months
  }else if(tenure > 60){
    return('> 60 Month')            # 5th group- > 60 months
```

```
  }
}

mychurn$tenure_group <- sapply(mychurn$tenure, group_tenure)
head(mychurn$tenure_group, 10)

mychurn$tenure_group <- as.factor(mychurn$tenure_group)

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

str(mychurn)

mychurn$customerID <- NULL
mychurn$tenure <- NULL
```

```
str(mychurn)
```

```
## 'data.frame':    7032 obs. of  20 variables:
##  $ gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
##  $ SeniorCitizen   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ 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 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...
##  $ MultipleLines   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 1 2 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 ...
##  $ tenure_group    : Factor w/ 5 levels "> 60 Month","0 - 12 Month",..: 2 4 2 4 2 2 3 2 4 1 ...
```

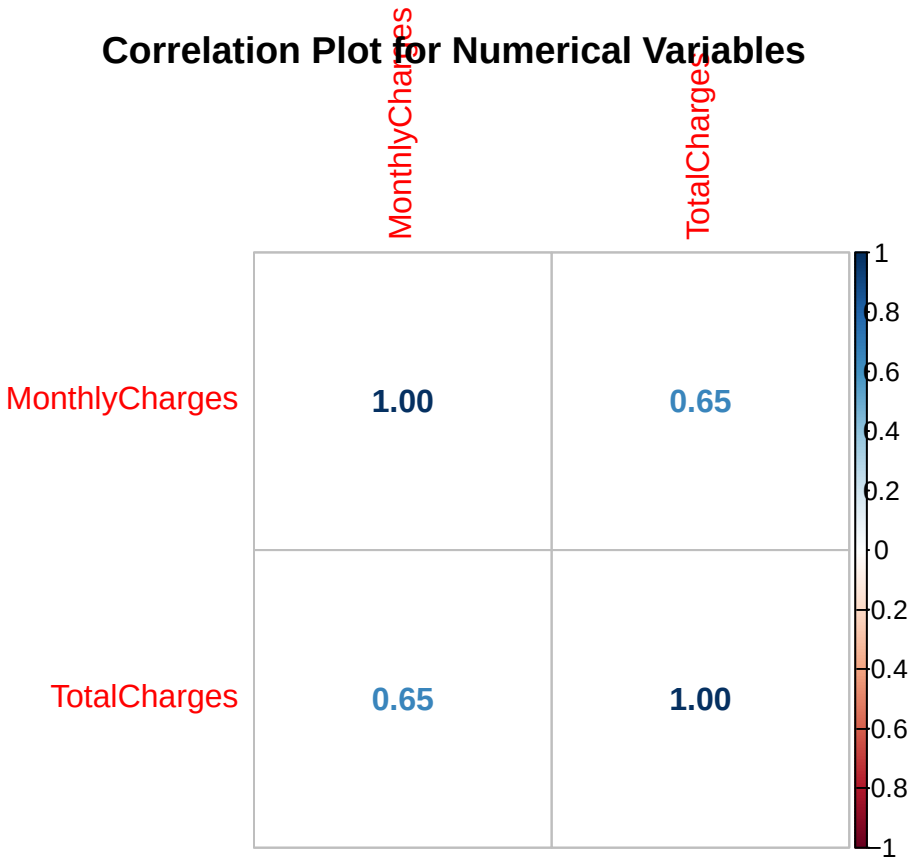
Exploratory Data Analysis and Feature Selection

Lets check for Correlation between the numerical variables/columns in the dataset

```
numeric_var <- sapply(mychurn, is.numeric)
numeric_var

corr_matrix <- cor(mychurn[, numeric_var])
corr_matrix
```

Visualizing the numerical columns and their correlation values

```
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.

```
mychurn$TotalCharges <- NULL
str(mychurn)
```

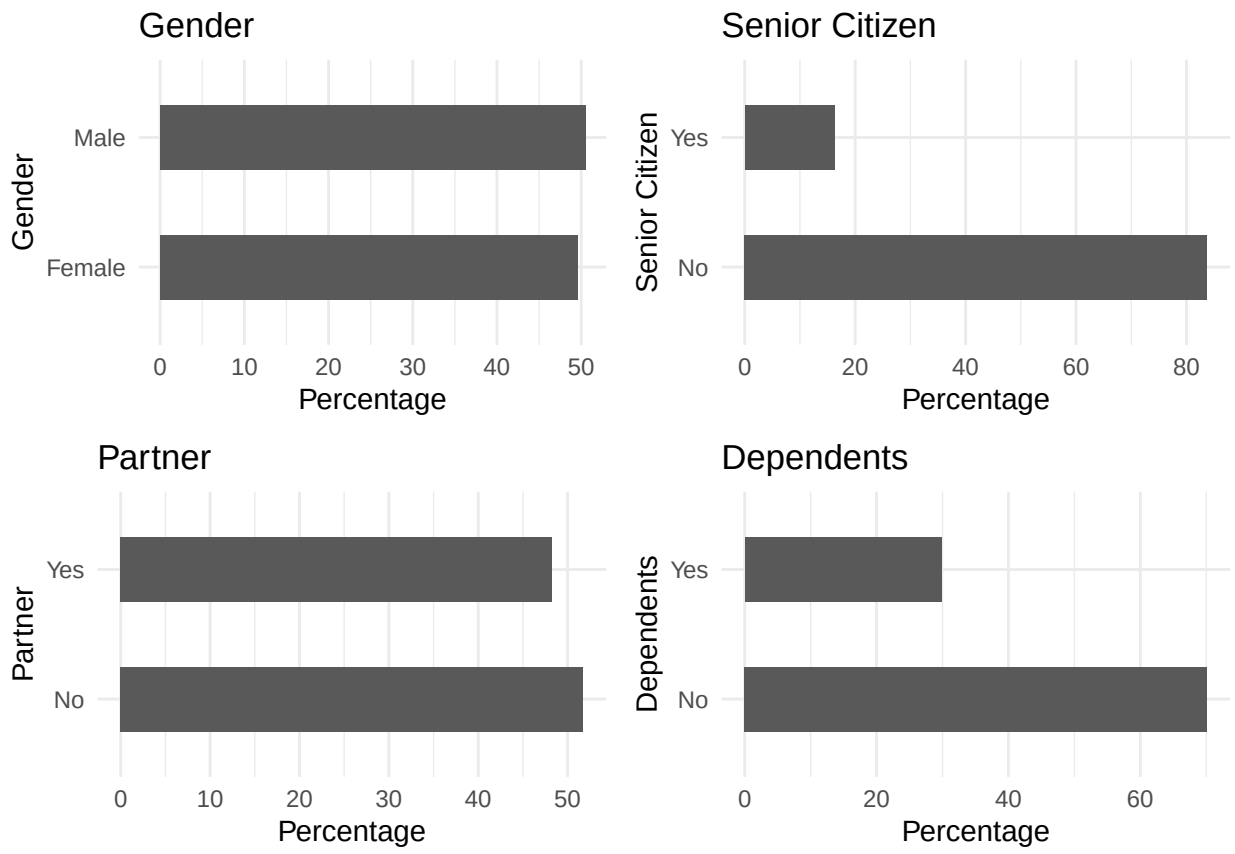
```
## 'data.frame':    7032 obs. of  19 variables:
##  $ gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
##  $ SeniorCitizen   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...
##  $ MultipleLines   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 1 2 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 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1 ...
##  $ tenure_group    : Factor w/ 5 levels "> 60 Month","0 - 12 Month",..: 2 4 2 4 2 2 3 2 4 1 ...
```

Creating Visualizations

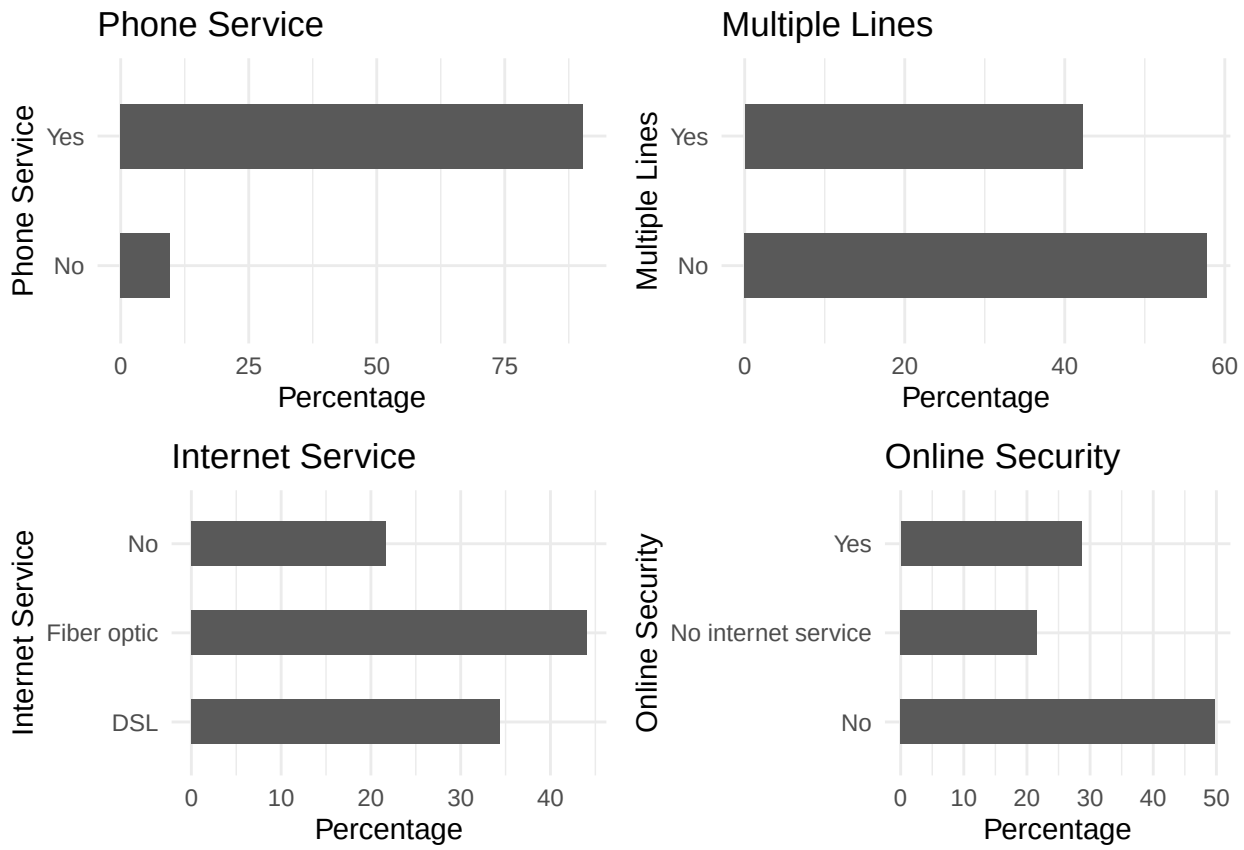
Creating bar charts for each categorical variable/column(all columns that do not contain numerical values)

```
p1 <- ggplot(mychurn, aes(x = gender)) +  # using Gender column
  ggtitle("Gender") + xlab("Gender") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") +
  coord_flip() +
  theme_minimal()


p2 <- ggplot(mychurn, aes(x = SeniorCitizen)) +  # using SeniorCitizen column
  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(mychurn, aes(x = Partner)) +  # using Partner column
  ggtitle("Partner") +
  xlab("Partner") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") +
  coord_flip() +
  theme_minimal()


p4 <- ggplot(mychurn, aes(x = Dependents)) +  # using Dependents column
  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)
```



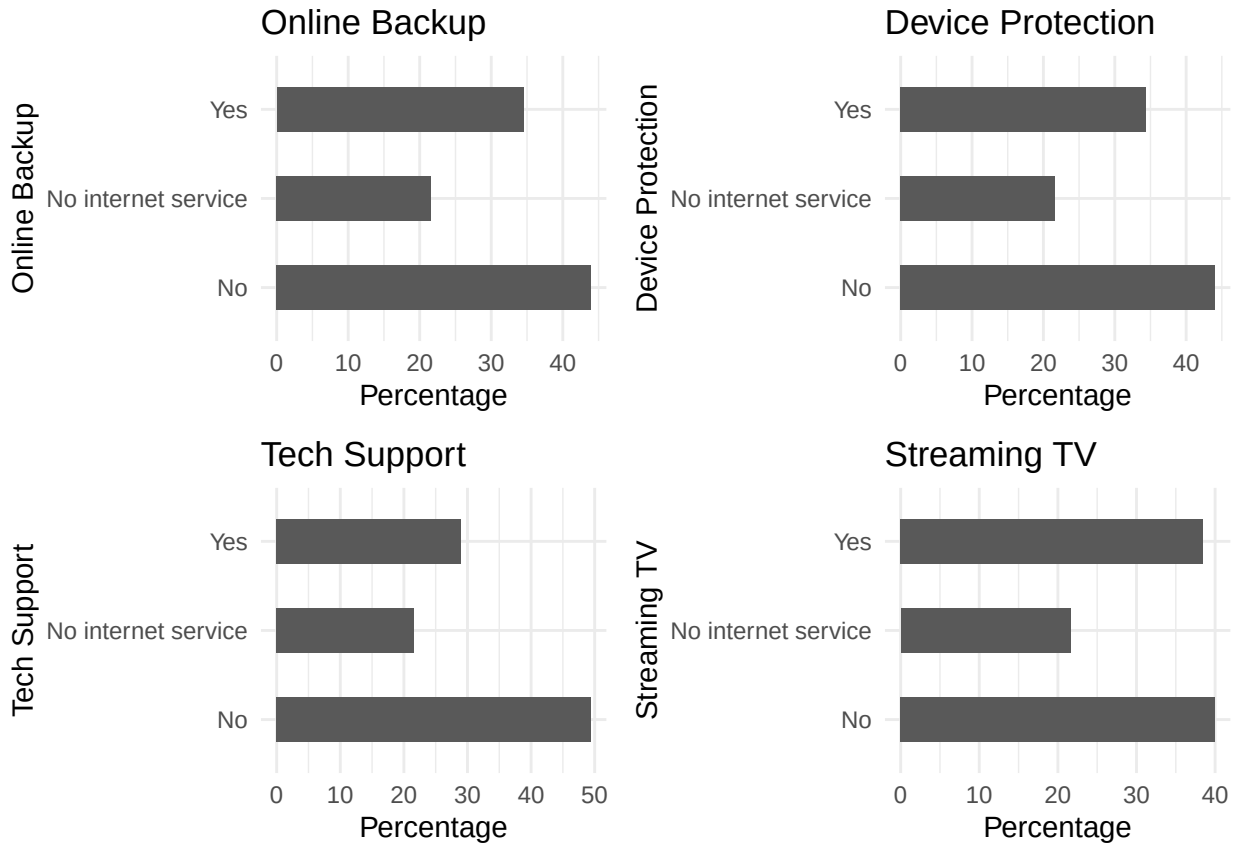

```
p5 <- ggplot(mychurn, aes(x = PhoneService)) +  # using PhoneService column
  ggtitle("Phone Service") +
  xlab("Phone Service") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") +
  coord_flip() +
  theme_minimal()


p6 <- ggplot(mychurn, aes(x = MultipleLines)) +  # using MultipleLines column
  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(mychurn, aes(x = InternetService)) +  # using InternetService column
  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(mychurn, aes(x = OnlineSecurity)) +    # using OnlineSecurity column
  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)
```

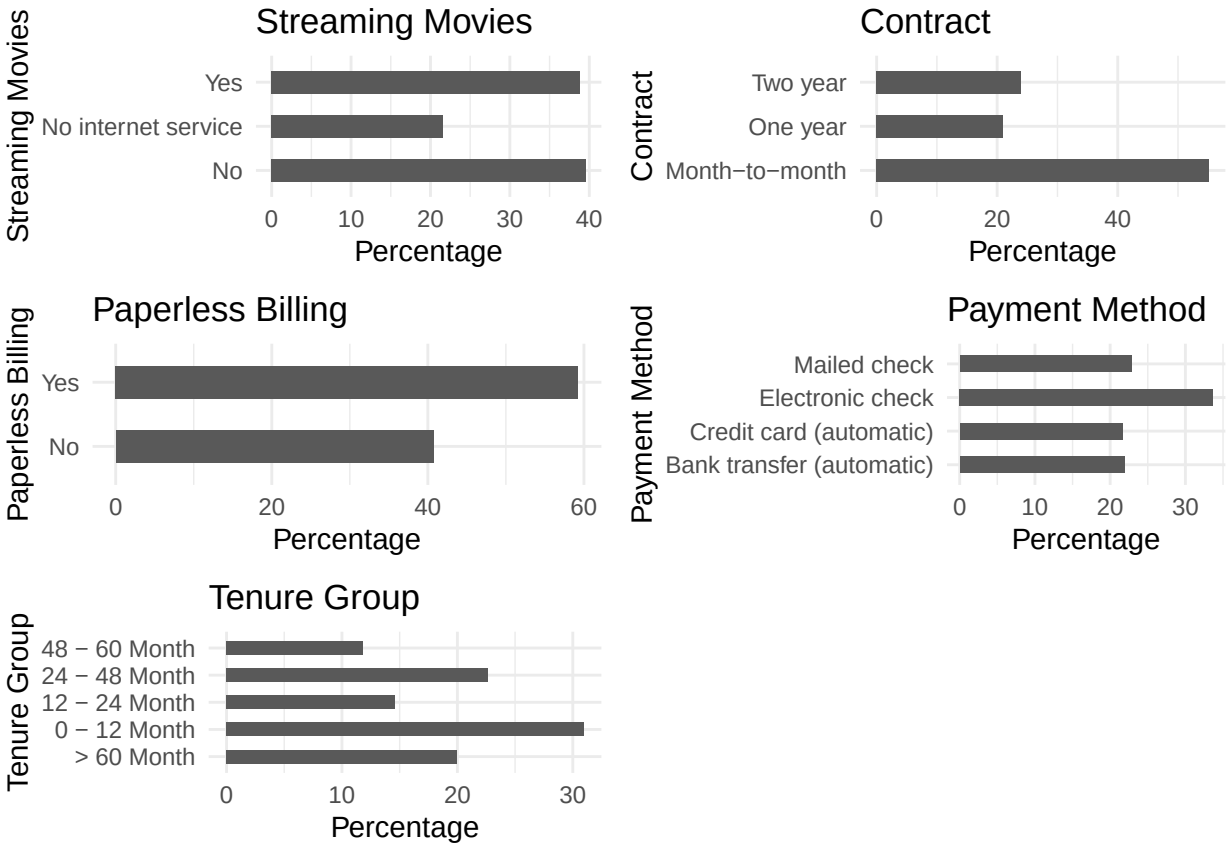



```
p9 <- ggplot(mychurn, aes( x = OnlineBackup)) +  # using OnlineBackup column
  ggtitle("Online Backup") +
  xlab("Online Backup") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") +
  coord_flip() +
  theme_minimal()


p10 <- ggplot(mychurn, aes(x = DeviceProtection)) +  # using 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(mychurn, aes(x = TechSupport)) +  # using TechSupport column
  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(mychurn, aes(x = StreamingTV)) +  # using StreamingTV column
  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(mychurn, aes(x = StreamingMovies)) +  # using StreamingMovies column
  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(mychurn, aes(x = Contract)) +  # using Contract column
  ggtitle("Contract") +
  xlab("Contract") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") +
  coord_flip() +
  theme_minimal()


p15 <- ggplot(mychurn, aes(x = PaperlessBilling)) +  # using 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(mychurn, aes(x = PaymentMethod)) +  # using PaymentMethod column
  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(mychurn, aes(x = tenure_group)) +  # using tenure_group column
  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)
```



Creating Models using Machine Learning algorithms

Logistic Regression

First, we split the data into training and testing sets:

```
intrain <- createDataPartition(mychurn$Churn,
                               p = 0.7, list = FALSE)

set.seed(2018)

training <- mychurn[intrain, ]
testing <- mychurn[- intrain, ]

dim(training); dim(testing)
```

Creating Logistical 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
## -1.9989  -0.6695  -0.3031   0.6712   3.0439
##
## Coefficients: (6 not defined because of singularities)
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -0.8462434  0.9863099  -0.858 0.390898
## genderMale                        -0.0243225  0.0773851  -0.314 0.753290
## SeniorCitizenYes                   0.2199999  0.1002630   2.194 0.028219 *
## PartnerYes                        -0.1035251  0.0930432  -1.113 0.265856
## DependentsYes                     -0.1220652  0.1072782  -1.138 0.255188
## PhoneServiceYes                    0.5319312  0.7709857   0.690 0.490234
## MultipleLinesYes                   0.4476706  0.2089674   2.142 0.032169 *
## InternetServiceFiber optic         1.9143215  0.9486985   2.018 0.043608 *
## InternetServiceNo                 -2.0393830  0.9609061  -2.122 0.033808 *
## OnlineSecurityNo internet service          NA         NA      NA       NA
## OnlineSecurityYes                 -0.1604459  0.2114535  -0.759 0.447986
## OnlineBackupNo internet service            NA         NA      NA       NA
## OnlineBackupYes                    0.0876866  0.2087118   0.420 0.674389
## DeviceProtectionNo internet service        NA         NA      NA       NA
## DeviceProtectionYes                0.2771690  0.2086986   1.328 0.184151
## TechSupportNo internet service             NA         NA      NA       NA
## TechSupportYes                    -0.2349037  0.2140781  -1.097 0.272519
## StreamingTVNo internet service             NA         NA      NA       NA
## StreamingTVYes                     0.7168333  0.3888549   1.843 0.065264 .
## StreamingMoviesNo internet service         NA         NA      NA       NA
## StreamingMoviesYes                 0.6908843  0.3884698   1.778 0.075326 .
## ContractOne year                  -0.8239165  0.1280097  -6.436 1.22e-10 ***
## ContractTwo year                  -1.4058314  0.2037286  -6.901 5.18e-12 ***
## PaperlessBillingYes                0.2986343  0.0891805   3.349 0.000812 ***
## PaymentMethodCredit card (automatic)  0.0363632  0.1349036   0.270 0.787507
## PaymentMethodElectronic check      0.4344086  0.1130654   3.842 0.000122 ***
## PaymentMethodMailed check          0.0008522  0.1374749   0.006 0.995054
## MonthlyCharges                    -0.0445560  0.0377133  -1.181 0.237427
## tenure_group0 - 12 Month           1.6940739  0.1989724   8.514  < 2e-16 ***
## tenure_group12 - 24 Month          0.7687504  0.1955475   3.931 8.45e-05 ***
## tenure_group24 - 48 Month          0.4688615  0.1768537   2.651 0.008022 **
## tenure_group48 - 60 Month          0.2191021  0.1939654   1.130 0.258647
## ---
## 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: 4126.1  on 4898  degrees of freedom
## AIC: 4178.1
##
## Number of Fisher Scoring iterations: 6
```

Feature Analysis: The top three most-relevant features include Contract, tenure_group and PaperlessBillin

```
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     1.18      4922     5701.6 0.2775749
## SeniorCitizen      1    97.40      4921     5604.2 < 2.2e-16 ***
## Partner            1   129.01      4920     5475.2 < 2.2e-16 ***
## Dependents         1    30.05      4919     5445.1 4.203e-08 ***
## PhoneService       1     1.12      4918     5444.0 0.2893103
## MultipleLines      1     6.68      4917     5437.3 0.0097732 **
## InternetService    2   456.70      4915     4980.6 < 2.2e-16 ***
## OnlineSecurity     1   176.58      4914     4804.0 < 2.2e-16 ***
## OnlineBackup       1    62.01      4913     4742.0 3.425e-15 ***
## DeviceProtection   1    35.10      4912     4706.9 3.138e-09 ***
## TechSupport        1   105.84      4911     4601.1 < 2.2e-16 ***
## StreamingTV        1     2.83      4910     4598.3 0.0926369 .
## StreamingMovies    1     0.02      4909     4598.2 0.8875484
## Contract           2   267.61      4907     4330.6 < 2.2e-16 ***
## PaperlessBilling   1    10.83      4906     4319.8 0.0009992 ***
## PaymentMethod      3    40.12      4903     4279.7 1.006e-08 ***
## MonthlyCharges     1     1.69      4902     4278.0 0.1936142
## tenure_group       4   151.93      4898     4126.1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Assessing the predictive ability of the Logistic Regression model

Lets recode some values in some columns

```
testing$Churn <- as.character(testing$Churn)
testing$Churn[testing$Churn == "No"] <- "0"
testing$Churn[testing$Churn == "Yes"] <- "1"
str(testing)

fitted_results <- predict(LogModel, newdata = testing, type = "response")
head(fitted_results)

fitted_results <- ifelse(fitted_results > 0.5, 1, 0)

misClasificError <- mean(fitted_results != testing$Churn)
misClasificError
```

```
print(paste('Logistic Regression Accuracy is: ', 1- misClasificError))
```

Accuracy

```
## [1] "Logistic Regression Accuracy is:  0.804079696394687"
```

```
print("Confusion Matrix for Logistic Regression");
```

Logistic Regression Confusion Matrix

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

```
table(testing$Churn, fitted_results > 0.5)
```

```
##
##     FALSE TRUE
##   0  1395  153
##   1   260  300
```

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)                      0.4290236 0.06198948  2.9641684
## genderMale                       0.9759709 0.83859936  1.1358515
## SeniorCitizenYes                 1.2460767 1.02351189  1.5164459
## PartnerYes                       0.9016534 0.75132814  1.0820983
## DependentsYes                    0.8850906 0.71670936  1.0915446
## PhoneServiceYes                  1.7022165 0.37574227  7.7230834
## MultipleLinesYes                 1.5646632 1.03912828  2.3578659
## InternetServiceFiber optic       6.7823354 1.05857219 43.6767597
## InternetServiceNo                0.1301090 0.01974975  0.8548234
## OnlineSecurityNo internet service        NA         NA         NA
## OnlineSecurityYes                0.8517639 0.56247073  1.2887964
## OnlineBackupNo internet service          NA         NA         NA
## OnlineBackupYes                  1.0916459 0.72512530  1.6437192
## DeviceProtectionNo internet service      NA         NA         NA
## DeviceProtectionYes              1.3193894 0.87658144  1.9869495
## TechSupportNo internet service           NA         NA         NA
## TechSupportYes                   0.7906470 0.51931083  1.2022259
## StreamingTVNo internet service           NA         NA         NA
## StreamingTVYes                   2.0479378 0.95637014  4.3934412
```

```
## StreamingMoviesNo internet service          NA         NA         NA
## StreamingMoviesYes                    1.9954793 0.93251578  4.2773745
## ContractOne year                      0.4387101 0.34033982  0.5623186
## ContractTwo year                      0.2451631 0.16271153  0.3620911
## PaperlessBillingYes                   1.3480166 1.13214833  1.6060768
## PaymentMethodCredit card (automatic) 1.0370324 0.79597517  1.3510756
## PaymentMethodElectronic check         1.5440497 1.23832361  1.9293195
## PaymentMethodMailed check             1.0008526 0.76473879  1.3111247
## MonthlyCharges                        0.9564220 0.88819680  1.0297481
## tenure_group0 - 12 Month              5.4416041 3.69674880  8.0677549
## tenure_group12 - 24 Month             2.1570691 1.47386014  3.1738435
## tenure_group24 - 48 Month             1.5981736 1.13298241  2.2676618
## tenure_group48 - 60 Month             1.2449584 0.85095195  1.8217092
```

Decision Tree Model

We are using only the 3 columns - Contract, tenure_group and PaperlessBilling (that we found out was the most significant to the Logistical Regression model earlier) in this Decision Tree model

```
mytree <- ctree(Churn ~ Contract + tenure_group + PaperlessBilling,
                training)
```

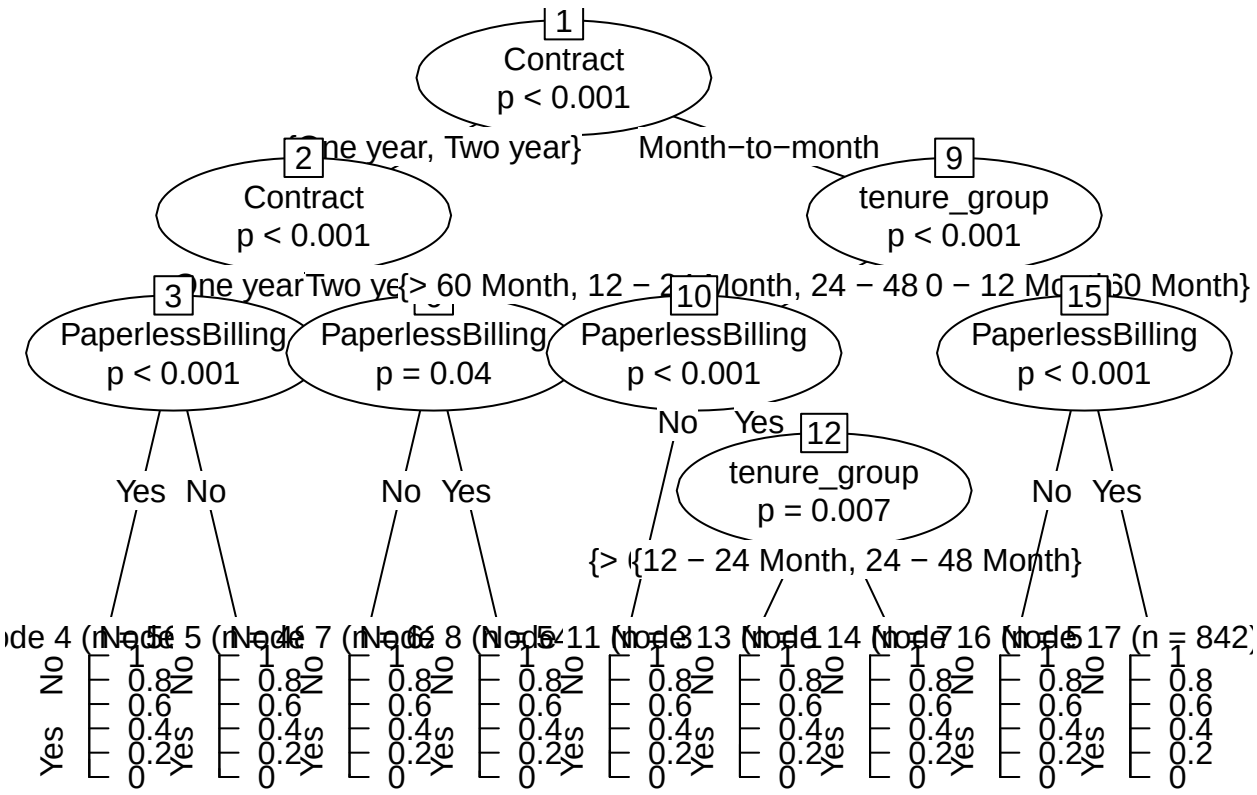
Decision Tree Visualization From the Decision Tree diagram, the 1st/top column in the diagram is the most important column to the model since it determines which decision will occur(ie Contract column as the 1st/top column is most important to the model)

As such Contract column is the most important column to be used to predict Churn column from the three variables we used in the model

From the Decision Tree diagram; If a customer has a one-year or two-year contract, no matter if he (she) has PapelessBilling or not, he (she) is less likely to churn (since No probability is much higher than Yes in the stacked bar chart) .

If a customer has a month-to-month contract, and is in the tenure group of 0 - 12 month, and using PaperlessBilling, then this customer is more likely to churn (since No probability is not that higher than Yes in the stacked bar chart).

```
plot(mytree)
```




```
pred_tree <- predict(mytree, testing)
```

Decision Tree Prediction on testing data

```
print("Confusion Matrix for Decision Tree"); table(Predicted = pred_tree, Actual = testing$Churn)
```

Decision Tree Confusion Matrix

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

##           Actual
## Predicted    0    1
##       No  1389  329
##       Yes  159  231
```

```
tab2 <- table(Predicted = pred_tree, Actual = testing$Churn)
```

Confusion Matrix of testing data

```
print(paste('Decision Tree Accuracy', sum(diag(tab2))/sum(tab2)))
```

Decision Tree Accuracy

```
## [1] "Decision Tree Accuracy 0.768500948766603"
```

```
p1 <- predict(mytree, training)
```

Decision Tree Prediction on training data

```
tab1 <- table(Predicted = p1, Actual = training$Churn)
```

Decision Tree Confusion Matrix

Random Forest 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: 21.57%
## Confusion matrix:
##       No Yes class.error
## No  3224 391   0.1081604
## Yes  671 638   0.5126050
```

```
pred_rf <- predict(rfModel, testing)
```

Random Forest Prediction

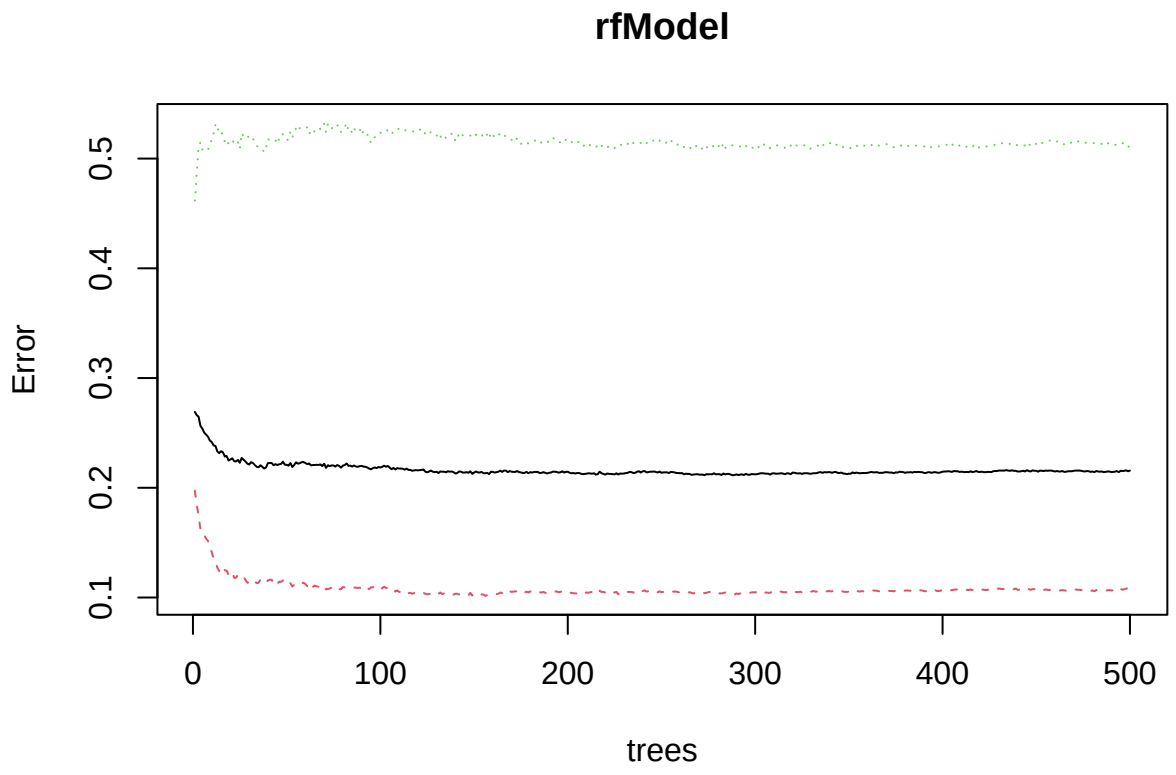
```
table(Predicted = pred_rf, Actual = testing$Churn)
```

Confusion matrix

```
##          Actual
## Predicted    0    1
##       No  1383  278
##       Yes  165  282
```

Random Forest Error Rate We use this plot to help us determine the number of trees (you can see trees parameter in x-axis).

```
plot(rfModel)
```

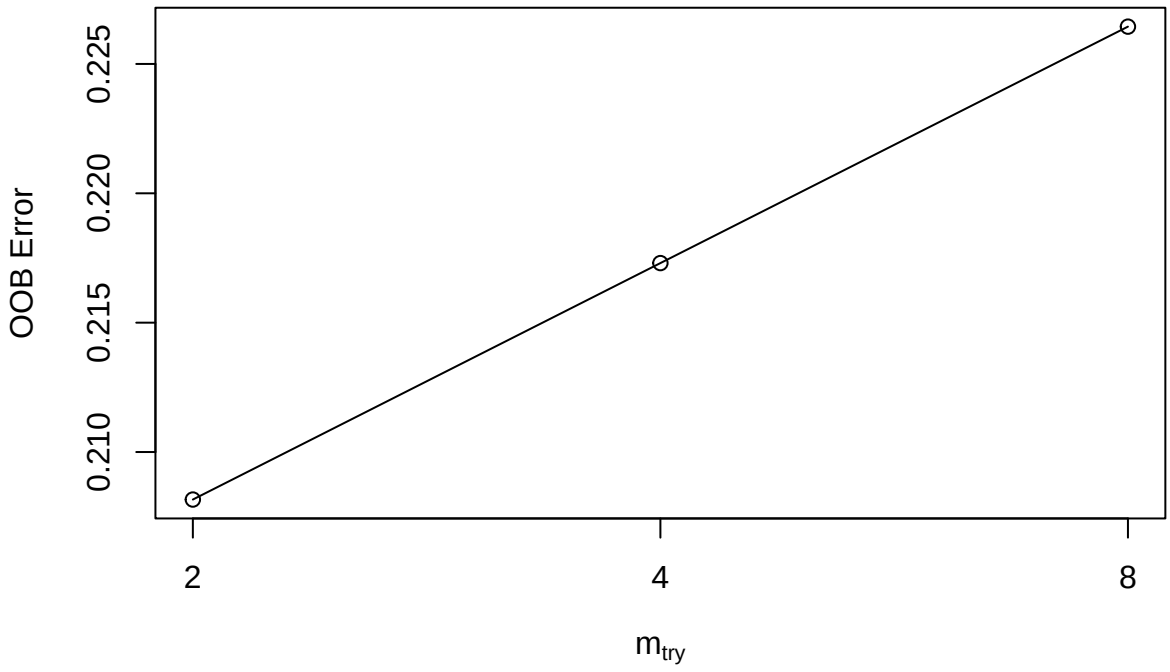


Tune Random Forest Model We use this plot to give us some ideas on the number of mtry to choose. OOB error rate is at the lowest when mtry is 2. Therefore, we choose mtry=2.

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

```
## mtry = 4  OOB error = 21.73%
```

```
## Searching left ...
## mtry = 8     OOB error = 22.64%
## -0.04205607 0.05
## Searching right ...
## mtry = 2     OOB error = 20.82%
## 0.04205607 0.05
```




```
t
```

```
##       mtry  OOBError
## 2.OOB    2 0.2081641
## 4.OOB    4 0.2173030
## 8.OOB    8 0.2264419
```

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

Create the Random Forest Model after Tuning and getting d optimal mtry value

```
##
## Call:
```

```
##  randomForest(formula = Churn ~ ., data = training, ntree = 200,      mtry = 2, importance = TRUE, pr
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 20.82%
## Confusion matrix:
##       No Yes class.error
## No  3277 338  0.09349931
## Yes  687 622  0.52482811
```

```
pred_rf_new <- predict(rfModel_new, testing)
```

Random Forest Predictions after Tuning

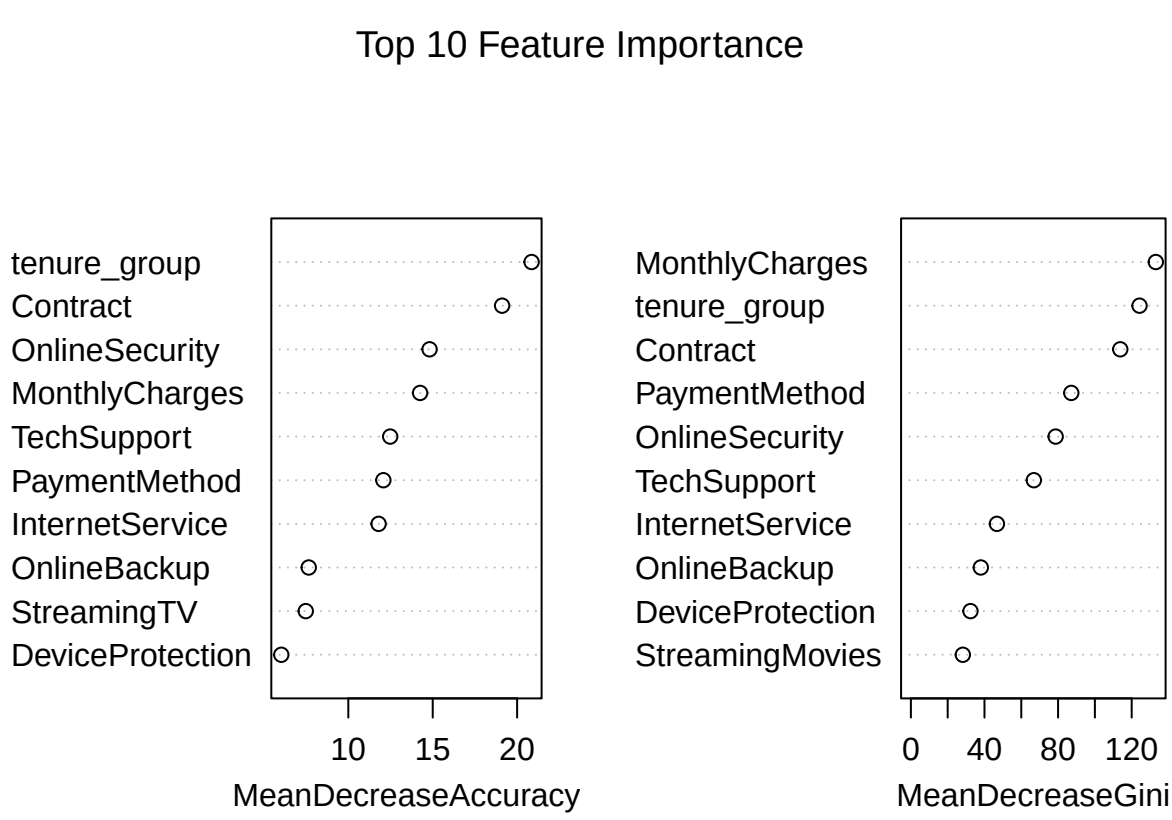
```
table(Predicted = pred_rf_new, Actual = testing$Churn)
```

Random Forest Confusion matrix after Tuning

```
##          Actual
## Predicted    0    1
##       No  1411  294
##       Yes  137  266
```

Random Forest Feature Importance: Here we view the columns/variables in order of importance to the model

```
varImpPlot(rfModel_new, sort=T, n.var = 10, main = 'Top 10 Feature Importance')
```



Summary

From the above example, we can see that Logistic Regression, Decision Tree and Random Forest can be used for customer churn analysis for this particular dataset equally fine.

Throughout the analysis, we have learned several important things:

- Features such as tenure_group, Contract, PaperlessBilling, MonthlyCharges and InternetService appear to play a role in customer churn.

*There does not seem to be a relationship between gender and churn.

- 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.