R_Project

Sheldon Sequeira 1948023, Abhishek Kumar

3/2/2020

Prediction of Telecom Customer Churn Using Predictive Regression Models

The crux of this project is to find out the cautionary features that help in telling us whether a customer will churn or no. To do this, we have delved deeper into understanding our dataset by checking the correlation between the variables and plotting graphs to check for linearity between the variables as well as whether the variables follow a statistical distribution or no. This helped us in understanding the relevant features that can be used as regressors in the forthcoming models. We tested our dataset on linear regression and gave valid evidence with the help of statistics why linear regression is not suitable for predicting churn using this dataset and performed logistic regression, decision tree and random forest algorithms on the dataset by fine tuning the parameters respectively.

The dataset consists of 3333 observations with 21 columns and has numerical, factor as well as boolean values. Customers who left within the last month – this column is called Churn. All other variables specify the behaviour of customers with their telecom plans in different states in U.S.A. (1)

The objectives of this project are:

1.To find out whether linear regression technique is suitable for the dataset in predicting the churn rate.

2.To check which variables are highly correlated and can be removed for selecting the optimum predictor variables in our model.

3.To implement the Logistic Regression model and calculate model accuracy in predicting the churn rate.

4.To implement the Decision Tree model and calculate its accuracy in predicting the churn rate.

5.To implement the Random Forest model and fine-tune its parameters so that the model gives optimal accuracy.

```
m=read.csv("C:/Users/HP/Desktop/R program second semester/churn_dataset_ra.cs
v")
head(m)

##   state account.length area.code phone.number international.plan
## 1    KS            128       415     382-4657                 no
## 2    OH            107       415     371-7191                 no
## 3    NJ            137       415     358-1921                 no
## 4    OH             84       408     375-9999                yes
## 5    OK             75       415     330-6626                yes
## 6    AL            118       510     391-8027                yes
##   voice.mail.plan number.vmail.messages total.day.minutes total.day.calls
## 1             yes                    25             265.1             110
## 2             yes                    26             161.6             123
## 3              no                     0             243.4             114
## 4              no                     0             299.4              71
## 5              no                     0             166.7             113
## 6              no                     0             223.4              98
##   total.day.charge total.eve.minutes total.eve.calls total.eve.charge
## 1            45.07             197.4              99            16.78
## 2            27.47             195.5             103            16.62
## 3            41.38             121.2             110            10.30
## 4            50.90              61.9              88             5.26
## 5            28.34             148.3             122            12.61
## 6            37.98             220.6             101            18.75
##   total.night.minutes total.night.calls total.night.charge total.intl.minu
tes
## 1               244.7                91              11.01                1
0.0
## 2               254.4               103              11.45                1
3.7
## 3               162.6               104               7.32                1
2.2
## 4               196.9                89               8.86                 
6.6
## 5               186.9               121               8.41                1
0.1
```

```
## 6                 203.9               118               9.18
6.3
##   total.intl.calls total.intl.charge customer.service.calls churn
## 1                3              2.70                      1 FALSE
## 2                3              3.70                      1 FALSE
## 3                5              3.29                      0 FALSE
## 4                7              1.78                      2 FALSE
## 5                3              2.73                      3 FALSE
## 6                6              1.70                      0 FALSE
```

```r
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.6.1
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.6.3
```

```
## corrplot 0.84 loaded
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.1
```

```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.6.2
```

```r
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 3.6.3
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.6.1
```

```
## Loading required package: lattice
```

```r
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.6.2
```

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.6.2
```

```
## randomForest 4.6-14
```

```
## 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

library(party)

## Warning: package 'party' was built under R version 3.6.3

## 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

## Warning: package 'strucchange' was built under R version 3.6.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.6.3

##
## Attaching package: 'zoo'

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

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 3.6.3

Churn=ifelse(m$churn=='TRUE',1,0)
class(Churn)

## [1] "numeric"

s=m[-21]
p=m[-c(1,4,5,6,21)]
```

```
p=cbind(p,Churn)
head(p)

##   account.length area.code number.vmail.messages total.day.minutes
## 1            128       415                    25             265.1
## 2            107       415                    26             161.6
## 3            137       415                     0             243.4
## 4             84       408                     0             299.4
## 5             75       415                     0             166.7
## 6            118       510                     0             223.4
##   total.day.calls total.day.charge total.eve.minutes total.eve.calls
## 1             110            45.07             197.4              99
## 2             123            27.47             195.5             103
## 3             114            41.38             121.2             110
## 4              71            50.90              61.9              88
## 5             113            28.34             148.3             122
## 6              98            37.98             220.6             101
##   total.eve.charge total.night.minutes total.night.calls total.night.charg
e
## 1            16.78               244.7                91             11.0
1
## 2            16.62               254.4               103             11.4
5
## 3            10.30               162.6               104              7.3
2
## 4             5.26               196.9                89              8.8
6
## 5            12.61               186.9               121              8.4
1
## 6            18.75               203.9               118              9.1
8
##   total.intl.minutes total.intl.calls total.intl.charge customer.service.c
alls
## 1               10.0                3              2.70
1
## 2               13.7                3              3.70
1
## 3               12.2                5              3.29
0
## 4                6.6                7              1.78
2
## 5               10.1                3              2.73
3
## 6                6.3                6              1.70
0
##   Churn
## 1     0
## 2     0
## 3     0
## 4     0
```

```
## 5     0
## 6     0
```

```
sapply(p, function(p) sum(is.na(p)))
```

```
##          account.length              area.code  number.vmail.messages 
##                       0                      0                      0 
##       total.day.minutes        total.day.calls       total.day.charge 
##                       0                      0                      0 
##       total.eve.minutes        total.eve.calls       total.eve.charge 
##                       0                      0                      0 
##     total.night.minutes      total.night.calls     total.night.charge 
##                       0                      0                      0 
##      total.intl.minutes       total.intl.calls      total.intl.charge 
##                       0                      0                      0 
## customer.service.calls                  Churn 
##                       0                      0
```

There are no null values in our dataset

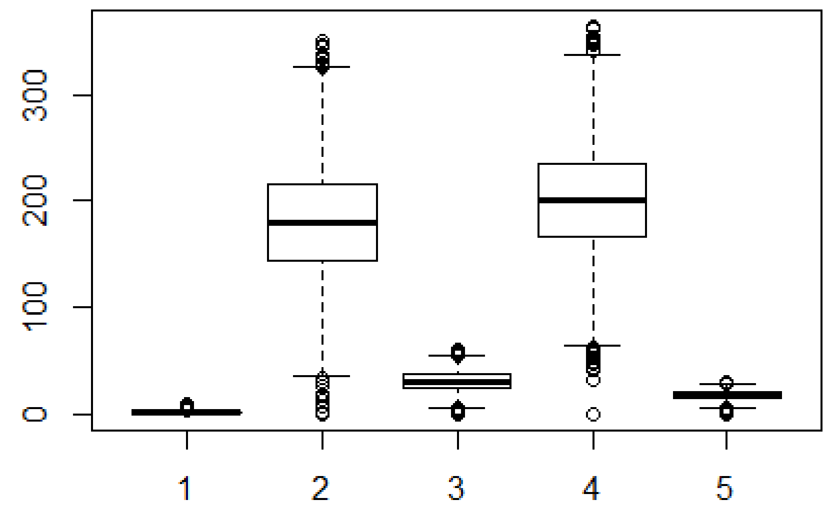
```
head(cor(p))
```

```
##                        account.length    area.code number.vmail.messages
## account.length            1.000000000 -0.012463497         -0.0046278243
## area.code                -0.012463497  1.000000000         -0.0019943701
## number.vmail.messages    -0.004627824 -0.001994370          1.0000000000
## total.day.minutes         0.006216021 -0.008264366          0.0007782741
## total.day.calls           0.038469882 -0.009646044         -0.0095480677
## total.day.charge          0.006214135 -0.008264441          0.0007755235
##                        total.day.minutes total.day.calls total.day.charge
## account.length              0.0062160205     0.038469882     0.0062141347
## area.code                  -0.0082643662    -0.009646044    -0.0082644411
## number.vmail.messages       0.0007782741    -0.009548068     0.0007755235
## total.day.minutes           1.0000000000     0.006750414     0.9999999522
## total.day.calls             0.0067504139     1.000000000     0.0067529620
## total.day.charge            0.9999999522     0.006752962     1.0000000000
##                        total.eve.minutes total.eve.calls total.eve.charge
## account.length              -0.006757142     0.019259967     -0.006745302
## area.code                    0.003580395    -0.011886271      0.003606690
## number.vmail.messages        0.017562034    -0.005864351      0.017577780
## total.day.minutes            0.007042511     0.015768993      0.007029035
## total.day.calls             -0.021451408     0.006462114     -0.021449263
## total.day.charge             0.007049607     0.015769282      0.007036131
##                        total.night.minutes total.night.calls total.night.ch
arge
## account.length                -0.008955192      -0.013176275       -0.00895
9535
## area.code                     -0.005824660       0.016522317       -0.00584
5376
## number.vmail.messages          0.007681136       0.007123063        0.00766
3290
```

```
## total.day.minutes             0.004323367       0.022972456        0.00430
0357
## total.day.calls               0.022937845      -0.019556965        0.02292
6638
## total.day.charge              0.004323879       0.022972420        0.00430
0861
##                      total.intl.minutes total.intl.calls total.intl.charg
e
## account.length              0.009513902      0.020661428       0.00954567
5
## area.code                  -0.018288168     -0.024178589      -0.01839469
6
## number.vmail.messages       0.002856196      0.013957339       0.00288365
8
## total.day.minutes          -0.010154586      0.008033357      -0.01009197
4
## total.day.calls             0.021564794      0.004574268       0.02166609
5
## total.day.charge           -0.010156862      0.008031572      -0.01009425
7
##                      customer.service.calls        Churn
## account.length                 -0.003795939  0.016540742
## area.code                       0.027572226  0.006174233
## number.vmail.messages          -0.013262583 -0.089727970
## total.day.minutes              -0.013423186  0.205150829
## total.day.calls                -0.018941930  0.018459312
## total.day.charge               -0.013426969  0.205150743
```

After correlation analysis we choose the the top five features for churn prediction namely, customer.service.calls, total.day.minutes, total.day.charges,total.eve.minutes, total.eve.charge

```
boxplot(p$customer.service.calls, p$total.day.minutes, p$total.day.charge, p$
total.eve.minutes, p$total.eve.charge)
```



The independant variables have many outliers as can be seen from the above boxplots

```
model = Churn ~ customer.service.calls + total.day.minutes + total.day.charge
+ total.eve.minutes + total.eve.charge
fit=lm(model,p)
summary(fit)

##
## Call:
## lm(formula = model, data = p)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.57442 -0.17748 -0.10538 -0.01906  1.13890
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -0.315787   0.031314 -10.084   <2e-16 ***
## customer.service.calls  0.056929   0.004413  12.900   <2e-16 ***
## total.day.minutes      -0.002820   0.344727  -0.008    0.993
## total.day.charge        0.024463   2.027809   0.012    0.990
## total.eve.minutes       0.098771   0.171035   0.577    0.564
## total.eve.charge       -1.154324   2.012181  -0.574    0.566
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.335 on 3327 degrees of freedom
```

```
## Multiple R-squared:  0.09577,    Adjusted R-squared:  0.09441
## F-statistic: 70.48 on 5 and 3327 DF,  p-value: < 2.2e-16
```

The above table proves that there is a strong positive relationship between customer.service.calls and churn. Also this variable is statistically significant.

Adjusted R-squared: The model explained 9.4% of the variance of churn (response variable)

```
confint(fit, level=0.99)

##                              0.5 %      99.5 %
## (Intercept)            -0.39649424 -0.23508062
## customer.service.calls  0.04555481  0.06830357
## total.day.minutes      -0.89128663  0.88564718
## total.day.charge       -5.20182545  5.25075176
## total.eve.minutes      -0.34203929  0.53958036
## total.eve.charge       -6.34033270  4.03168465
```

The output reports 99% confidence intervals for all co-efficients in our multiple linear regression model.

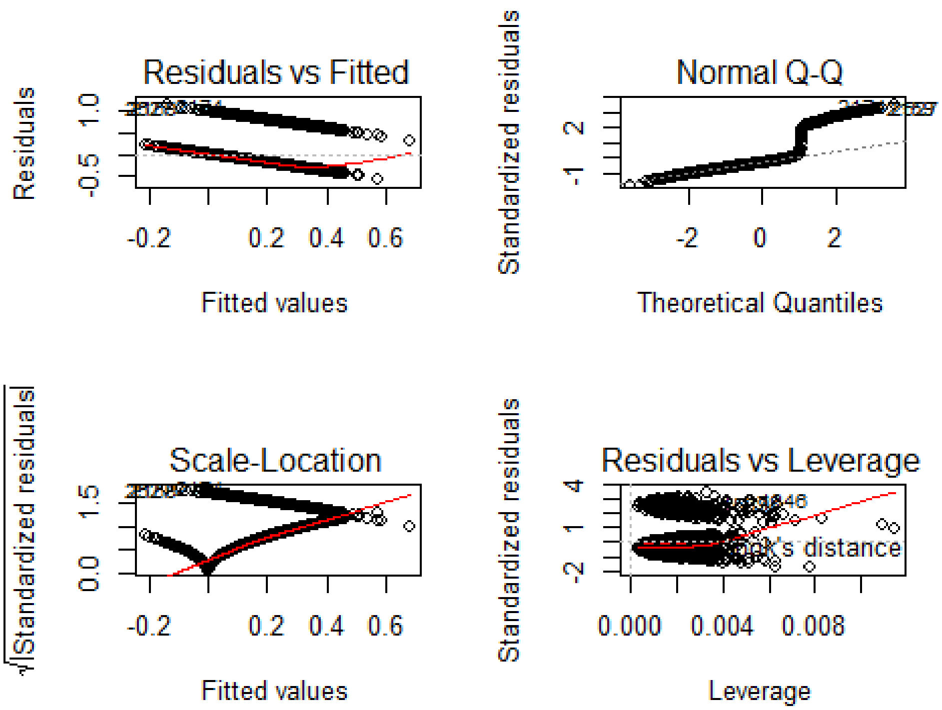
```
newdata=data.frame(customer.service.calls=5,total.day.minutes=220,total.day.c
harge=30,total.eve.minutes=120,total.eve.charge=10)

predict(fit,newdata,interval="confidence")

##         fit       lwr      upr
## 1 0.3916376 -29.04166 29.82493
```

The 95% confidence interval of the response variable(churn) with the given parameters is between -29.04166 and 29.82493.

```
anova(fit)

## Analysis of Variance Table
##
## Response: Churn
##                          Df Sum Sq Mean Sq  F value    Pr(>F)
## customer.service.calls    1  18.00 17.9974 160.3346 < 2.2e-16 ***
## total.day.minutes         1  17.86 17.8634 159.1412 < 2.2e-16 ***
## total.day.charge          1   0.00  0.0023   0.0202    0.8870
## total.eve.minutes         1   3.65  3.6539  32.5518 1.262e-08 ***
## total.eve.charge          1   0.04  0.0369   0.3291    0.5662
## Residuals              3327 373.45  0.1122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here with the help of anova, customer.service.calls, total.day.minutes and total.eve.minutes are significant variables when compared to the reponse variable Churn.

```r
par(mfrow=c(2,2))
plot(fit)
```



Residual analysis among the four plots tells us that there is a lot of variance in our model.

```r
library(alr3)

## Warning: package 'alr3' was built under R version 3.6.2

## Loading required package: car

## Warning: package 'car' was built under R version 3.6.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 3.6.1

##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:modeltools':
##
##     Predict
```

```
##
## Attaching package: 'alr3'
```

```
## The following object is masked from 'package:MASS':
##
##     forbes
```

```r
pureErrorAnova(fit)
```

```
## Warning in anova.lm(lm(mod$model[, 1] ~ mod$model$Lack.of.Fit, weights =
## weights(mod))): ANOVA F-tests on an essentially perfect fit are unreliable
```

```
## Analysis of Variance Table
##
## Response: Churn
##                         Df Sum Sq Mean Sq    F value    Pr(>F)
## customer.service.calls    1  18.00 17.9974 3.8629e+28 < 2.2e-16 ***
## total.day.minutes         1  17.86 17.8634 3.8342e+28 < 2.2e-16 ***
## total.day.charge          1   0.00  0.0023 4.8619e+24 < 2.2e-16 ***
## total.eve.minutes         1   3.65  3.6539 7.8427e+27 < 2.2e-16 ***
## total.eve.charge          1   0.04  0.0369 7.9289e+25 < 2.2e-16 ***
## Residuals              3327 373.45  0.1122
##  Lack of fit           3325 373.45  0.1123 2.4107e+26 < 2.2e-16 ***
##  Pure Error               2   0.00  0.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can't use this model as a predictor of the response as the Pr(>F) is smaller than 0.10

Inference:

1. There are many outliers in our regressor variables and hence we must eliminate them before implementing in our model.

2. From the residual analysis, we can see that the variance is non-constant and the residuals deviate from the mean value greatly.

3. Due to the presence of outliers, residual analysis can't be performed accurately.

4. The response variable seems to give us a numeric output whereas our response variable is binary. Hence we can infer that linear regression is not possible for predicting the churn variable.

Exploratory data analysis and feature selection

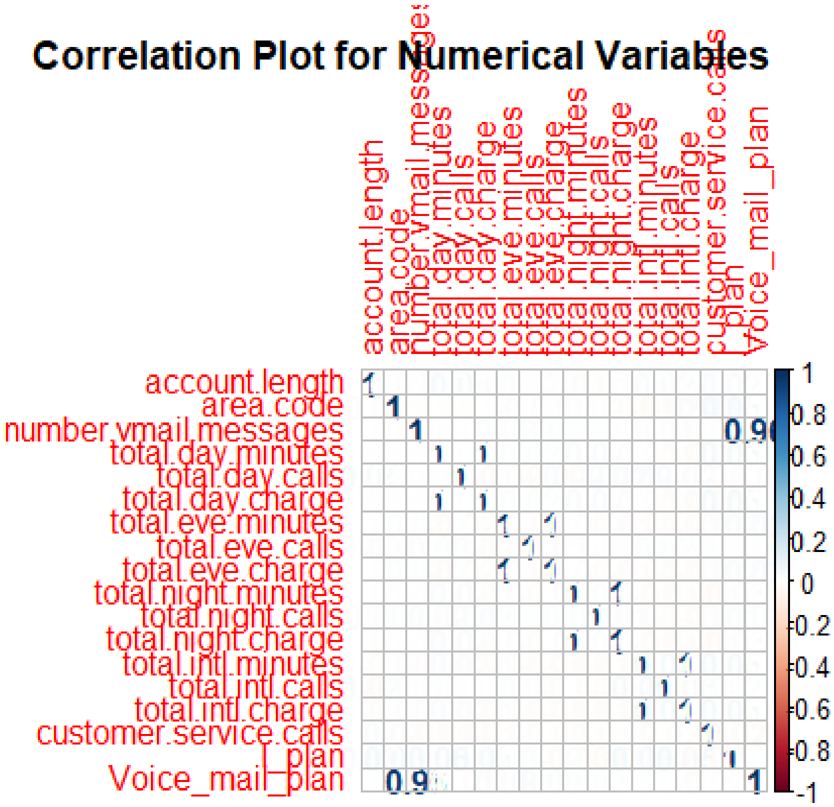
```
Churn=ifelse(m$churn=='TRUE','yes','no')
I_plan=ifelse(m$international.plan=='yes',1,0)
Voice_mail_plan=ifelse(m$voice.mail.plan=='yes',1,0)
t=m[-c(1,4,5,6,21)]
t=cbind(t,I_plan,Voice_mail_plan,Churn)
head(t)
```

```
##   account.length area.code number.vmail.messages total.day.minutes
## 1            128       415                    25             265.1
## 2            107       415                    26             161.6
## 3            137       415                     0             243.4
## 4             84       408                     0             299.4
## 5             75       415                     0             166.7
## 6            118       510                     0             223.4
##   total.day.calls total.day.charge total.eve.minutes total.eve.calls
## 1             110            45.07             197.4              99
## 2             123            27.47             195.5             103
## 3             114            41.38             121.2             110
## 4              71            50.90              61.9              88
## 5             113            28.34             148.3             122
## 6              98            37.98             220.6             101
##   total.eve.charge total.night.minutes total.night.calls total.night.charg
e
## 1            16.78               244.7                91              11.0
```

```
1
## 2           16.62               254.4             103             11.4
5
## 3           10.30               162.6             104              7.3
2
## 4            5.26               196.9              89              8.8
6
## 5           12.61               186.9             121              8.4
1
## 6           18.75               203.9             118              9.1
8
##   total.intl.minutes total.intl.calls total.intl.charge customer.service.c
alls
## 1               10.0                3              2.70
1
## 2               13.7                3              3.70
1
## 3               12.2                5              3.29
0
## 4                6.6                7              1.78
2
## 5               10.1                3              2.73
3
## 6                6.3                6              1.70
0
##   I_plan Voice_mail_plan Churn
## 1      0               1    no
## 2      0               1    no
## 3      0               0    no
## 4      1               0    no
## 5      1               0    no
## 6      1               0    no
```

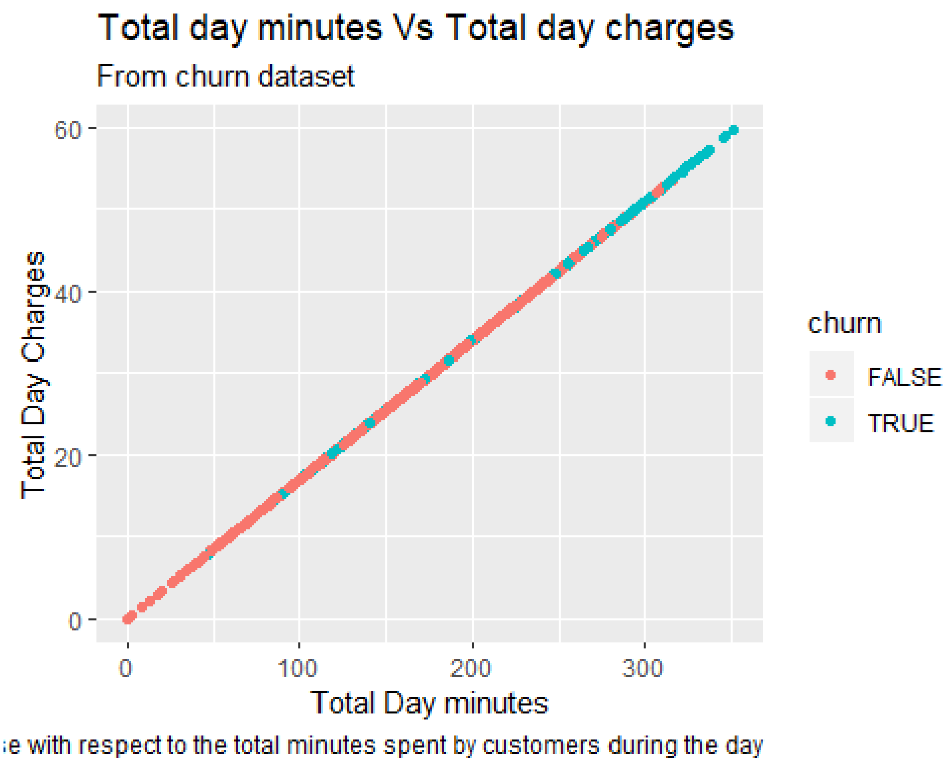
Here we have converted the logical variable Churn of the dataset into factor variable

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



The total.eve.minutes, total.day.minutes and total.night.minutes are highly correlated with total.eve.charge, total.day.charge and total.night.charge. So we can remove one variable each among them.

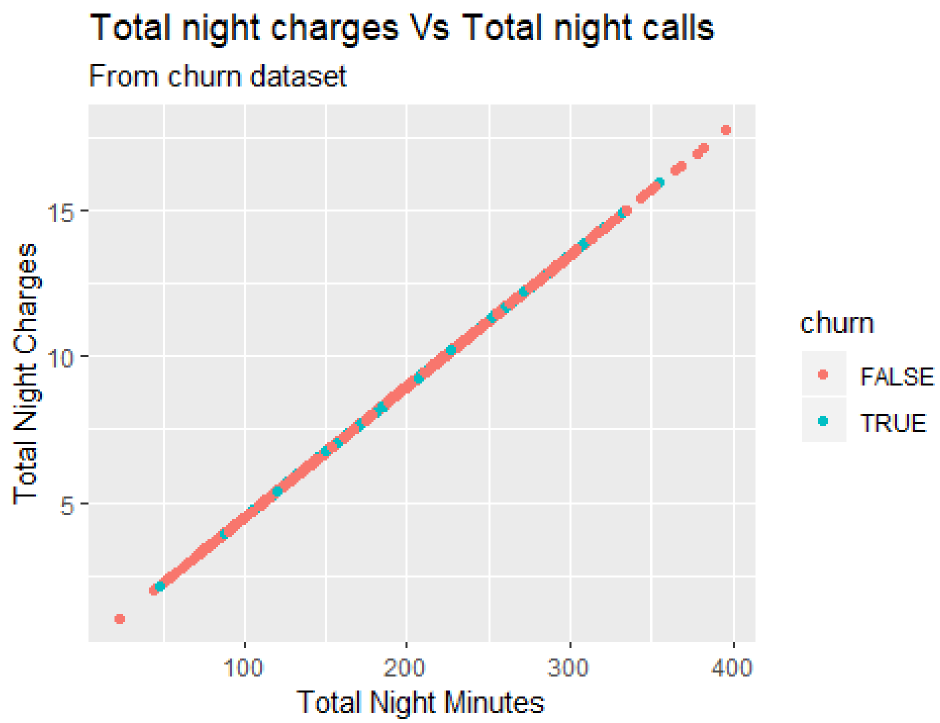
```
t$total.day.charge <- NULL
t$total.eve.charge <- NULL
t$total.night.charge <- NULL

library(ggplot2)

ggplot(m, aes(x=s$total.day.minutes, y=s$total.day.charge))+geom_point(aes(co
l=churn))+labs(title="Total day minutes Vs Total day charges",
subtitle = "From churn dataset",x = "Total Day minutes", y = "Total Day Charg
es",
caption = "Plot shows how the network charges increase with respect to the to
tal minutes spent by customers during the day")
```



The above plot tells us that the customers who spend a lot of time on phone calls are highly likely to churn as compared to the customers who spend fewer minutes on a phone call.

In this dataset, the churn rate can be seen to increase sharply after the customer spends more than 250 minutes.

```
library(ggplot2)
ggplot(m, aes(x=s$total.night.minutes, y=s$total.night.charge))+geom_point(ae
s(col=churn))+labs(title="Total night charges Vs Total night calls",
subtitle = "From churn dataset",x = "Total Night Minutes", y = "Total Night C
harges",
caption = "Plot shows how the network charges increase with respect to the to
tal minutes spent by customers during the night")
```

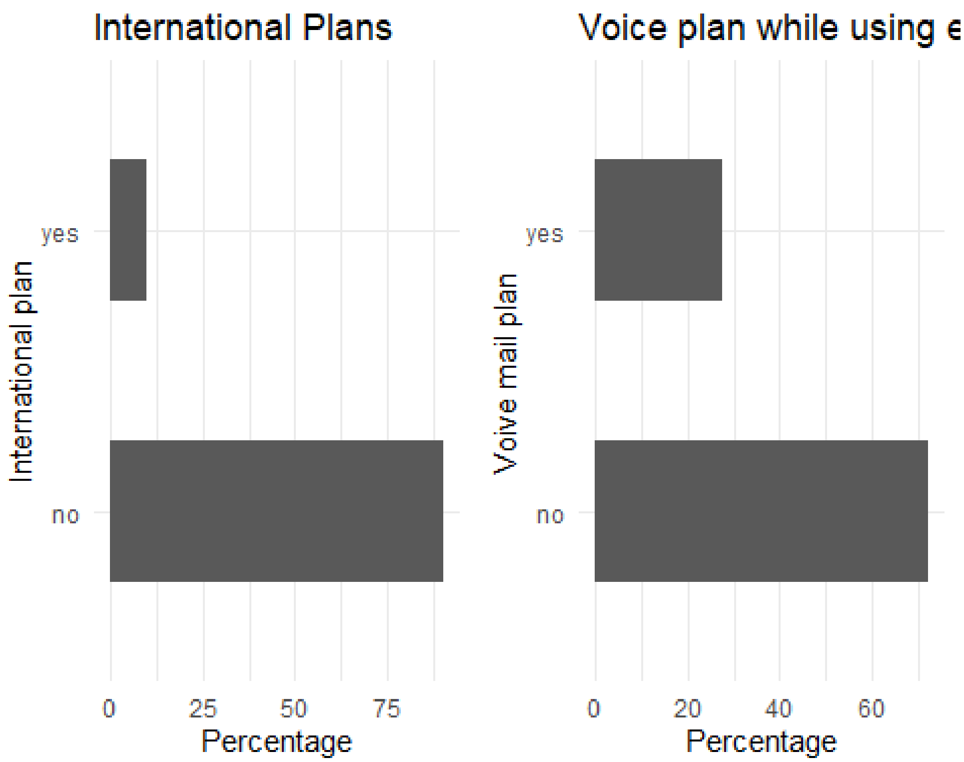


with respect to the total minutes spent by customers during the night

The above plot tells us that there is a linear relationship between the network charges and the total minutes spent by the customer on phone calls.

But we cannot say whether customer will churn from this plot as the churn rate is equally distributed along the line.

```
library(ggplot2)
p1 <- ggplot(m, aes(x=international.plan)) + ggtitle("International Plans") +
xlab("International plan") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Perc
entage") + coord_flip() + theme_minimal()
p2 <- ggplot(m, aes(x=voice.mail.plan)) + ggtitle("Voice plan while using ema
il") + xlab("Voive mail plan") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Perc
entage") + coord_flip() + theme_minimal()

grid.arrange(p1, p2, ncol=2)
```



The two categorical variables seem to have a reasonably broad distribution, hence both of them can be kept for further analysis. (2)

Logistic Regression

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

dim(training); dim(testing)

## [1] 2334   16

## [1] 999  16

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

##
## Call:
## glm(formula = Churn ~ ., family = binomial(link = "logit"), data = trainin
g)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0544  -0.5275  -0.3499  -0.2001   3.2032
```

```
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -8.1126566  1.1059445  -7.336 2.21e-13 ***
## account.length        0.0001092  0.0016423   0.066 0.946991
## area.code            -0.0008403  0.0015849  -0.530 0.595983
## number.vmail.messages  0.0372612  0.0218201   1.708 0.087700 .
## total.day.minutes     0.0126831  0.0012867   9.857  < 2e-16 ***
## total.day.calls       0.0024226  0.0032200   0.752 0.451824
## total.eve.minutes     0.0077560  0.0013721   5.653 1.58e-08 ***
## total.eve.calls       0.0014530  0.0032974   0.441 0.659459
## total.night.minutes   0.0045845  0.0013196   3.474 0.000513 ***
## total.night.calls    -0.0022861  0.0033657  -0.679 0.496991
## total.intl.minutes   -7.4736722  6.2980869  -1.187 0.235363
## total.intl.calls     -0.1044428  0.0300512  -3.475 0.000510 ***
## total.intl.charge    28.0797299 23.3274240   1.204 0.228697
## customer.service.calls  0.4322496  0.0464866   9.298  < 2e-16 ***
## I_plan                2.0754773  0.1747117  11.879  < 2e-16 ***
## Voice_mail_plan      -2.0770227  0.6886856  -3.016 0.002562 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1934.3  on 2333  degrees of freedom
## Residual deviance: 1529.6  on 2318  degrees of freedom
## AIC: 1561.6
##
## Number of Fisher Scoring iterations: 6
```

The top four features in our above model are total.day.minutes, total.eve.minutes, customer.service.calls and I_plan (2)

```
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                                   2333     1934.3
## account.length         1    0.075      2332     1934.2  0.783618
## area.code              1    0.013      2331     1934.2  0.910825
## number.vmail.messages  1   26.350      2330     1907.8 2.848e-07 ***
## total.day.minutes      1   97.206      2329     1810.6 < 2.2e-16 ***
```

```
## total.day.calls        1     1.069       2328      1809.6  0.301216
## total.eve.minutes      1    23.570       2327      1786.0 1.205e-06 ***
## total.eve.calls        1     0.198       2326      1785.8  0.656481
## total.night.minutes    1     5.556       2325      1780.2  0.018420 *
## total.night.calls      1     0.110       2324      1780.1  0.740028
## total.intl.minutes     1    20.258       2323      1759.9 6.767e-06 ***
## total.intl.calls       1    10.266       2322      1749.6  0.001355 **
## total.intl.charge      1     0.958       2321      1748.7  0.327755
## customer.service.calls 1    73.318       2320      1675.3 < 2.2e-16 ***
## I_plan                 1   135.896       2319      1539.4 < 2.2e-16 ***
## Voice_mail_plan        1     9.846       2318      1529.6  0.001702 **
## ---
## 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. (2)

The other variables such as number.vmail.messages and total.intl.minutes seem to improve the model less even though they all have low p-values. (2)

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 <- ifelse(fitted.results > 0.5,1,0)
misClasificError <- mean(fitted.results != testing$Churn)
print(paste('Logistic Regression Accuracy',1-misClasificError))

## [1] "Logistic Regression Accuracy 0.867867867867868"
```

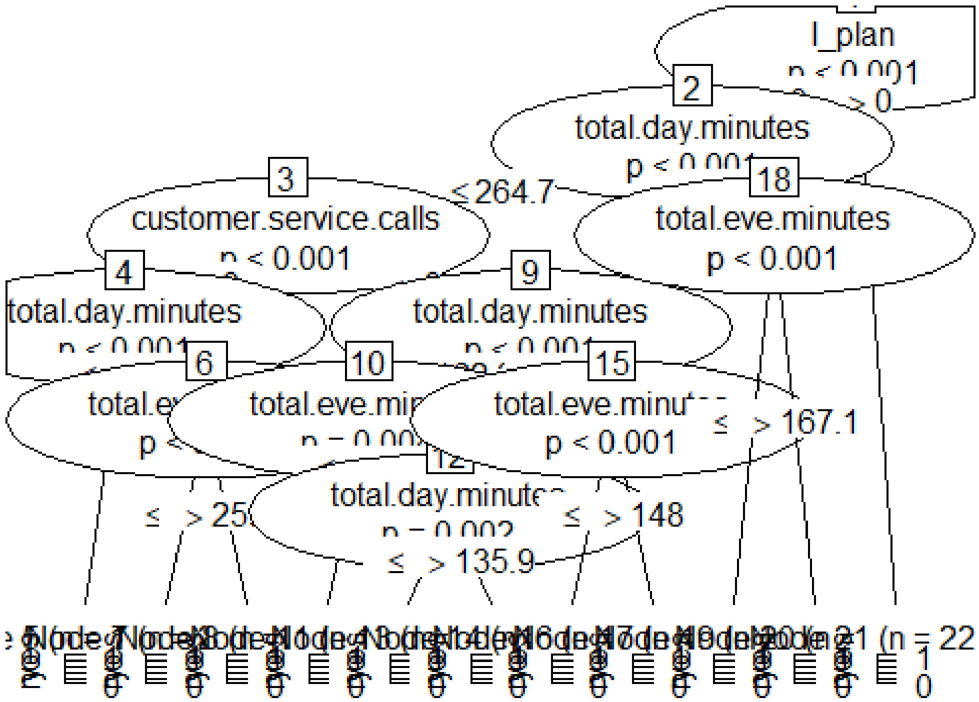
Logistic Regression gives accuracy of 86% (2)

Logistic Regression Confusion Matrix

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

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

##
##     FALSE TRUE
##   0   832   23
##   1   109   35
```

Decision Tree

```
tree <- ctree(Churn~total.day.minutes+ total.eve.minutes+customer.service.cal
ls+ I_plan, training)
plot(tree)
```



1. Out of four variables we use, International plan is the most important variable to predict customer churn or not churn. (2)

2. If a customer receives customer service calls or not, no matter he (she) spends more or less minutes on phone calls, he (she) is less likely to churn. (2)

3. If the customer has an international plan, then this customer is more likely to churn. (2)

Decision Tree Confusion Matrix

```
pred_tree <- predict(tree, testing)
print("Confusion Matrix for Decision Tree"); table(Predicted = pred_tree, Act
ual = testing$Churn)

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

##           Actual
## Predicted   0   1
##       no  828  73
##       yes  27  71
```

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 0.8998998998999"
```

Decision Tree Accuracy is 89% (2)

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: 3
## 
##         OOB estimate of  error rate: 6.04%
## Confusion matrix:
##       no yes class.error
## no  1981  14 0.007017544
## yes  127 212 0.374631268
```

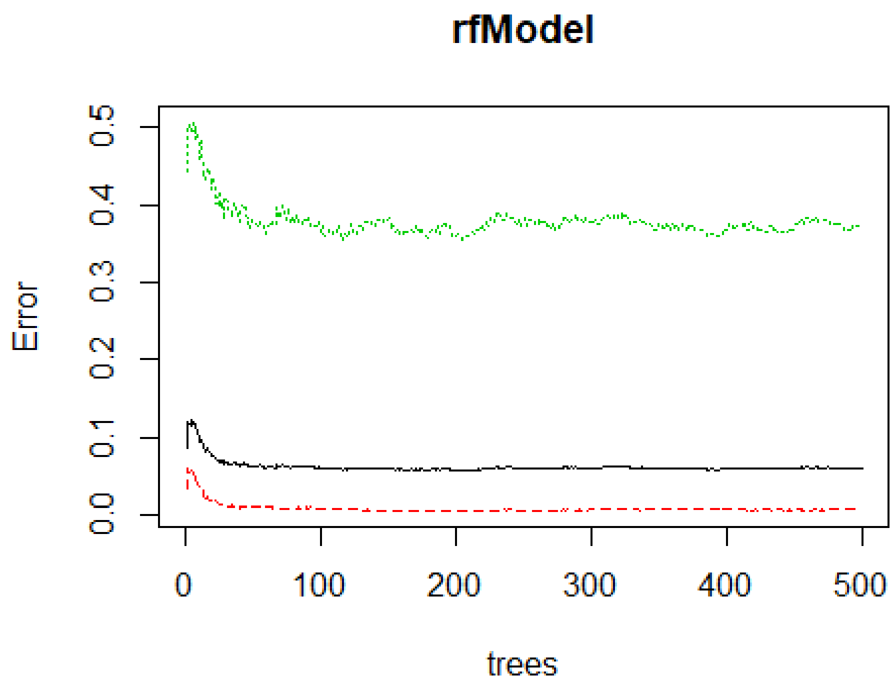
Error rate is pretty low when predicting “no” and much higher when predicting “yes”

Random Forest Prediction and Confusion Matrix

```
pred_rf <- predict(rfModel, testing)
#caret::confusionMatrix(pred_rf, testing$Churn)
```

Since the dataset is not large, overfitting leads to the model giving high accuracy. We try to reduce the OOB error rate for the model abd check its accuracy again.

```
plot(rfModel)
```

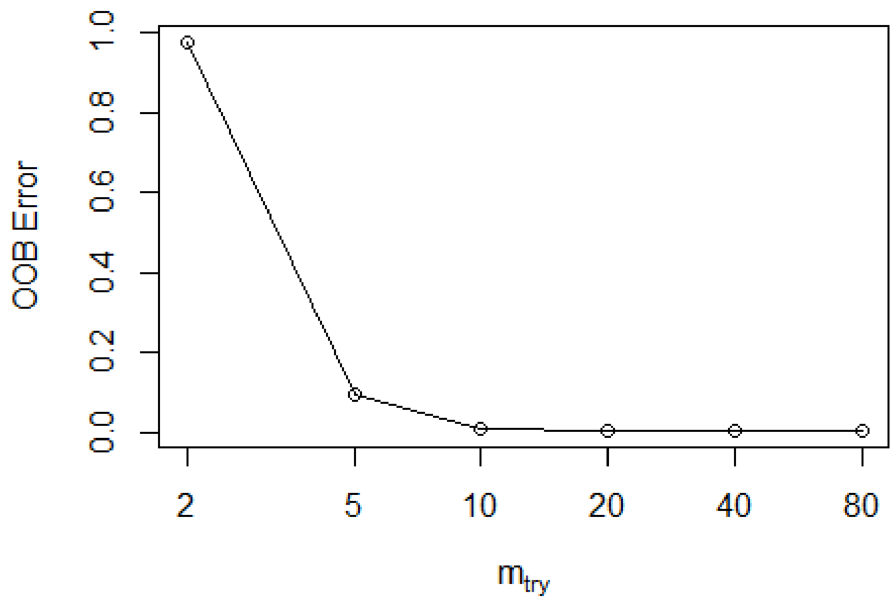


We use this plot to help us determine the number of trees. As the number of trees increases, the OOB error rate decreases, and then becomes almost constant. We are not able to decrease the OOB error rate after about 100 to 200 trees.

Tuning the random forest model

```
l <- tuneRF(training[, -10], training[, 10], stepFactor = 0.5, plot = TRUE, n
treeTry = 200, trace = TRUE, improve = 0.05)

## mtry = 5  OOB error = 0.09254018
## Searching left ...
## mtry = 10    OOB error = 0.006749432
## 0.9270649 0.05

## Warning in randomForest.default(x, y, mtry = mtryCur, ntree = ntreeTry, :
## invalid mtry: reset to within valid range

## mtry = 20    OOB error = 0.004666323
## 0.3086348 0.05

## Warning in randomForest.default(x, y, mtry = mtryCur, ntree = ntreeTry, :
## invalid mtry: reset to within valid range

## mtry = 40    OOB error = 0.0036206
## 0.2241 0.05
```

```
## Warning in randomForest.default(x, y, mtry = mtryCur, ntree = ntreeTry, :
## invalid mtry: reset to within valid range

## mtry = 80    OOB error = 0.003481469
## 0.03842759 0.05
## Searching right ...
## mtry = 2     OOB error = 0.9775629
## -269.0003 0.05
```



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 10. Therefore, we choose mtry=10.

Fitting the Random Forest Model After Tuning

```
rfModel_new <- randomForest(Churn ~., data = training, ntree = 200, mtry = 10
, importance = TRUE, proximity = TRUE)
print(rfModel_new)

##
## Call:
##  randomForest(formula = Churn ~ ., data = training, ntree = 200,      mtry
= 10, importance = TRUE, proximity = TRUE)
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 10
##
##         OOB estimate of  error rate: 4.97%
```

```
## Confusion matrix:
##       no yes class.error
## no  1971  24  0.01203008
## yes   92 247  0.27138643
```

OOB error rate decreased to 5.14% from 6.04% (2)

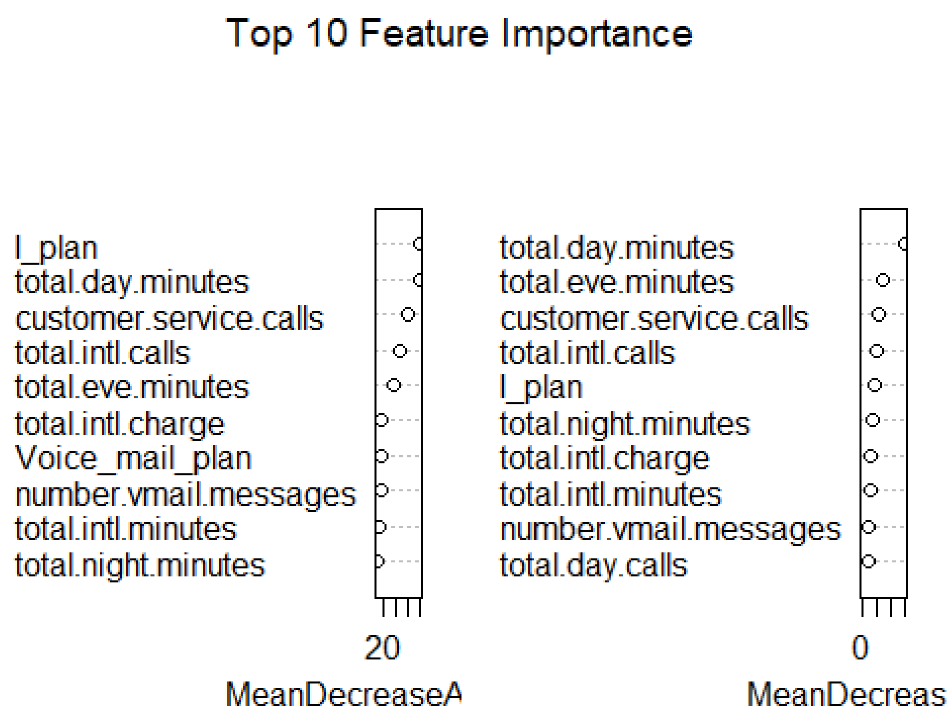
Random Forest Predictions

```
pred_rf_new <- predict(rfModel_new, testing)
#caret::confusionMatrix(pred_rf_new, testing$Churn)
```

The model shows 95% accuracy after reducing its OOB error rate. (2)

Random Forest Feature Importance

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



Conclusion

We can see that Logistic Regression, Decision Tree and Random Forest can be used for customer churn analysis for this particular dataset equally fine.

1. Features such as International Plan, Customer.service.calls, total.day.minutes and total.eve.minutes appear to play a role in customer churn.

2. There does not seem to be a relationship between state variable and churn variable(because we are using prediction).

3. Customers that have an internatonal plan or that get more customer service calls are more likely to churn; On the other hand, customers that do not have an international plan, spend fewer minutes on phone calls throughout the day,evening and night, are less likely to churn.

Bibliography

1. **David_Becks.** Celullar Network Churn analytics. *Kaggle.* [Online] https://www.kaggle.com/becksddf/churn-in-telecoms-dataset.

2. **Li, Susan.** [Online] https://towardsdatascience.com/predict-customer-churn-with-r-9e62357d47b4.