

```

paste("Title: Will the Telcom Customer Churn?- A Classification Analysis")
## [1] "Title: Will the Telcom Customer Churn?- A Classification Analysis"
paste("Authors: Rajdeep Saha & Soumik Karmakar")
## [1] "Authors: Rajdeep Saha & Soumik Karmakar"

rm(list=ls())
set.seed(1)
library(ggplot2)
library(leaps)
library(caret)

## Loading required package: lattice

library(car)

## Loading required package: carData

library(corrplot)

## corrplot 0.89 loaded

library(tree)
library(MASS)
library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

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

library(pROC)

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

#data access
df <- read.csv('C:/Users/user/OneDrive/Desktop/Self Project/WA_Fn-UseC_-
Telco-Customer-Churn.csv')
head(df)

```

```

##   customerID gender SeniorCitizen Partner Dependents tenure PhoneService
## 1 7590-VHVEG Female           0     Yes         No         1           No
## 2 5575-GNVDE  Male           0     No          No        34           Yes
## 3 3668-QPYBK  Male           0     No          No         2           Yes
## 4 7795-CFOCW  Male           0     No          No        45           No
## 5 9237-HQITU Female           0     No          No         2           Yes
## 6 9305-CDSKC Female           0     No          No         8           Yes
##      MultipleLines InternetService OnlineSecurity OnlineBackup
DeviceProtection
## 1 No phone service          DSL              No              Yes
No
## 2                      No          DSL              Yes              No
Yes
## 3                      No          DSL              Yes              Yes
No
## 4 No phone service          DSL              Yes              No
Yes
## 5                      No      Fiber optic          No              No
No
## 6                      Yes      Fiber optic          No              No
Yes
##      TechSupport StreamingTV StreamingMovies      Contract PaperlessBilling
## 1          No          No              No Month-to-month          Yes
## 2          No          No              No   One year              No
## 3          No          No              No Month-to-month          Yes
## 4          Yes          No              No   One year              No
## 5          No          No              No Month-to-month          Yes
## 6          No          Yes              Yes Month-to-month          Yes
##      PaymentMethod MonthlyCharges TotalCharges Churn
## 1      Electronic check          29.85          29.85   No
## 2          Mailed check          56.95         1889.50   No
## 3          Mailed check          53.85          108.15  Yes
## 4 Bank transfer (automatic)         42.30         1840.75   No
## 5      Electronic check          70.70          151.65  Yes
## 6      Electronic check          99.65          820.50  Yes

dim(df)

## [1] 7043   21

str(df)

## '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" ...
```

```
n <- nrow(df)
```

```
#id column remove
colnames(df)
```

```
## [1] "customerID" "gender" "SeniorCitizen" "Partner"
## [5] "Dependents" "tenure" "PhoneService"
"MultipleLines"
## [9] "InternetService" "OnlineSecurity" "OnlineBackup"
"DeviceProtection"
## [13] "TechSupport" "StreamingTV" "StreamingMovies" "Contract"
## [17] "PaperlessBilling" "PaymentMethod" "MonthlyCharges"
"TotalCharges"
## [21] "Churn"
```

```
df <- df[-which(colnames(df) == 'customerID')]
head(df)
```

```
## gender SeniorCitizen Partner Dependents tenure PhoneService
MultipleLines
## 1 Female 0 Yes No 1 No No phone
service
## 2 Male 0 No No 34 Yes
No
## 3 Male 0 No No 2 Yes
No
## 4 Male 0 No No 45 No No phone
service
## 5 Female 0 No No 2 Yes
No
## 6 Female 0 No No 8 Yes
Yes
## InternetService OnlineSecurity OnlineBackup DeviceProtection TechSupport
```

```
## 1          DSL          No          Yes          No          No
## 2          DSL          Yes          No          Yes          No
## 3          DSL          Yes          Yes          No          No
## 4          DSL          Yes          No          Yes          Yes
## 5      Fiber optic          No          No          No          No
## 6      Fiber optic          No          No          Yes          No
##      StreamingTV StreamingMovies      Contract PaperlessBilling
## 1          No          No Month-to-month          Yes
## 2          No          No      One year          No
## 3          No          No Month-to-month          Yes
## 4          No          No      One year          No
## 5          No          No Month-to-month          Yes
## 6          Yes          Yes Month-to-month          Yes
##      PaymentMethod MonthlyCharges TotalCharges Churn
## 1      Electronic check          29.85          29.85    No
## 2          Mailed check          56.95         1889.50    No
## 3          Mailed check          53.85          108.15   Yes
## 4 Bank transfer (automatic)          42.30         1840.75    No
## 5      Electronic check          70.70          151.65   Yes
## 6      Electronic check          99.65          820.50   Yes
```

#### *#missing value imputation*

```
df$TotalCharges <- as.numeric(df$TotalCharges)
miss = which(is.na(df$TotalCharges) == TRUE)
df$TotalCharges[miss] <- median(df$TotalCharges, na.rm = TRUE)
str(df)
```

```
## 'data.frame':    7043 obs. of  20 variables:
## $ 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" ...

#No Service to No
for(i in (which(colnames(df) == 'OnlineSecurity') : which(colnames(df) ==
'StreamingMovies'))){
  df[i] <- as.factor(ifelse(df[i] != 'Yes', 'No', 'Yes'))
}
df$InternetService <- as.factor(ifelse(df$InternetService != 'No', 'Yes',
'No'))
df$MultipleLines <- as.factor(ifelse(df$MultipleLines != 'Yes', 'No', 'Yes'))
df$SeniorCitizen <- as.factor(df$SeniorCitizen)

for(i in 1:ncol(df)){
  if(class(df[,i]) == 'character'){
    df[,i] <- as.factor(df[,i])
  }
}
str(df)

## 'data.frame':    7043 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 "0","1": 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
...
## $ 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/ 2 levels "No","Yes": 1 1 1 1 1 2 2 1 2 1
...
## $ InternetService : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2
...
## $ OnlineSecurity  : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 1 1 2 1 2
...
## $ OnlineBackup    : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 2 1 1 2
...
## $ DeviceProtection: Factor w/ 2 levels "No","Yes": 1 2 1 2 1 2 1 1 2 1
...
## $ TechSupport     : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 1 1 2 1
...
## $ StreamingTV     : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 1 2 1
...
## $ StreamingMovies : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 2 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

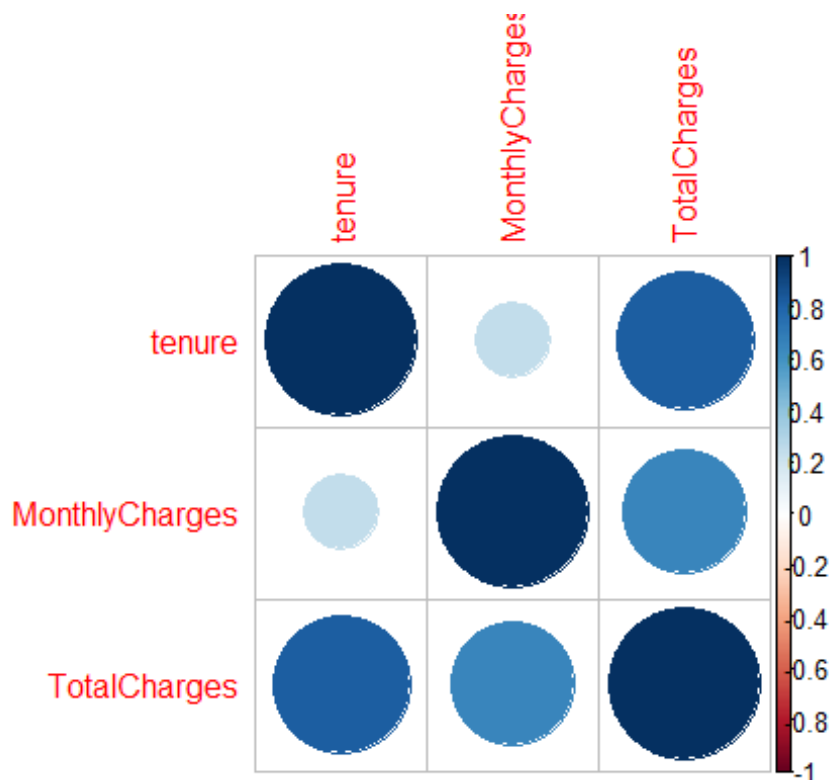
```

```

...
## $ 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
...

#Correlation between numeric variables
cr <-cor(df[,c(5,18,19)])
corrplot(cr, method="circle")

```

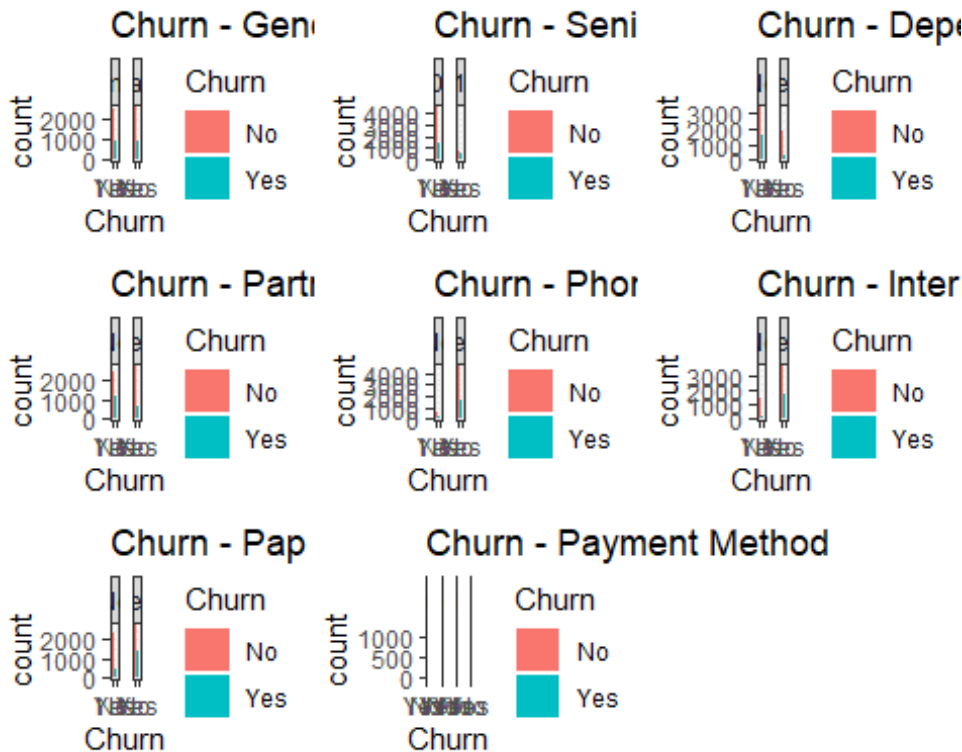


```

#EDA
p1 <- ggplot(df, aes(x = Churn, fill = Churn)) +facet_grid(~gender)+
geom_bar() +ggtitle("Churn - Gender") + theme_bw()
p2 <- ggplot(df, aes(x = Churn, fill = Churn)) +facet_grid(~SeniorCitizen)+
geom_bar() + ggtitle("Churn - SeniorCitizen") + theme_bw()
p3 <- ggplot(df, aes(x = Churn, fill = Churn)) +facet_grid(~Dependents)+
geom_bar() + ggtitle("Churn - Dependents") + theme_bw()
p4 <- ggplot(df, aes(x = Churn, fill = Churn)) +facet_grid(~Partner)+
geom_bar() + ggtitle("Churn - Partner") + theme_bw()
p5 <- ggplot(df, aes(x = Churn, fill = Churn)) +facet_grid(~PhoneService)+
geom_bar() + ggtitle("Churn - PhoneService")+ theme_bw()
p6 <- ggplot(df, aes(x = Churn, fill = Churn)) +facet_grid(~InternetService)+
geom_bar() + ggtitle("Churn - InternetService") + theme_bw()
p7 <- ggplot(df, aes(x = Churn, fill = Churn))
+facet_grid(~PaperlessBilling)+ geom_bar() + ggtitle("Churn -

```

```
PaperlessBilling")) + theme_bw()
p8 <- ggplot(df, aes(x = Churn, fill = Churn)) + facet_grid(~PaymentMethod) +
geom_bar() + ggtitle("Churn - Payment Method") + theme_bw()
ggpubr::ggarrange(p1,p2,p3,p4,p5,p6,p7,p8, nrow = 3, ncol = 3)
```



```
#dummification
attach(df)
to_dummy <- data.frame(Contract,PaymentMethod)
dmy <- dummyVars(" ~ .", data = to_dummy)
df2 <- data.frame(predict(dmy, newdata = to_dummy))
df2 <- df2[, !(colnames(df2) %in% c("Contract.Month.to.month",
"PaymentMethod.Bank.transfer..automatic."))]
df <- df[,!(colnames(df) %in% c("Contract","PaymentMethod","TotalCharges"))]
df <- cbind(df, df2)
head(df)
```

```
##   gender SeniorCitizen Partner Dependents tenure PhoneService
MultipleLines
## 1 Female              0      Yes         No         1          No
No
## 2  Male              0      No          No        34          Yes
No
## 3  Male              0      No          No         2          Yes
No
## 4  Male              0      No          No        45          No
No
## 5 Female              0      No          No         2          Yes
```

```

No
## 6 Female          0      No      No      8      Yes
Yes
##  InternetService OnlineSecurity OnlineBackup DeviceProtection TechSupport
## 1          Yes          No          Yes          No          No
## 2          Yes          Yes          No          Yes          No
## 3          Yes          Yes          Yes          No          No
## 4          Yes          Yes          No          Yes          Yes
## 5          Yes          No          No          No          No
## 6          Yes          No          No          Yes          No
##  StreamingTV StreamingMovies PaperlessBilling MonthlyCharges Churn
## 1          No          No          Yes          29.85      No
## 2          No          No          No          56.95      No
## 3          No          No          Yes          53.85      Yes
## 4          No          No          No          42.30      No
## 5          No          No          Yes          70.70      Yes
## 6          Yes          Yes          Yes          99.65      Yes
##  Contract.One.year Contract.Two.year
PaymentMethod.Credit.card..automatic.
## 1          0          0
0
## 2          1          0
0
## 3          0          0
0
## 4          1          0
0
## 5          0          0
0
## 6          0          0
0
##  PaymentMethod.Electronic.check PaymentMethod.Mailed.check
## 1          1          0
## 2          0          1
## 3          0          1
## 4          0          0
## 5          1          0
## 6          1          0

attach(df)

## The following objects are masked from df (pos = 3):
##
##  Churn, Dependents, DeviceProtection, gender, InternetService,
##  MonthlyCharges, MultipleLines, OnlineBackup, OnlineSecurity,
##  PaperlessBilling, Partner, PhoneService, SeniorCitizen,
##  StreamingMovies, StreamingTV, TechSupport, tenure

dim(df)

## [1] 7043  22

```



### #Feature Selection

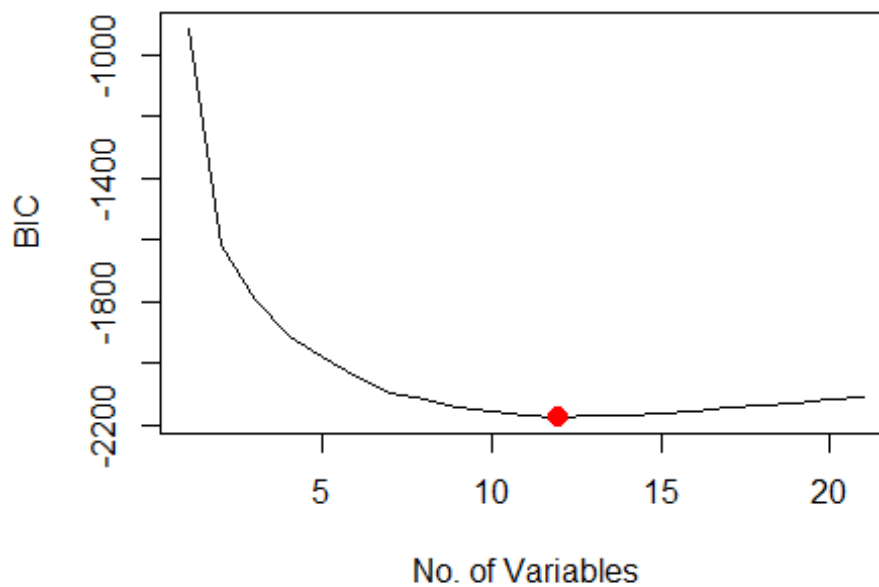
```
regfit.full=regsubsets(Churn~.,data=df,nvmax=21)
reg.summary=summary(regfit.full)
names(reg.summary)

## [1] "which" "rsq" "rss" "adjr2" "cp" "bic" "outmat" "obj"

which.min(reg.summary$bic)

## [1] 12

plot(reg.summary$bic,xlab="No. of
Variables",ylab=expression(paste("BIC")),type="l")
points(12,reg.summary$bic[12],col="red",cex=2,pch=20)
```



```
names(coef(regfit.full,12))[-1]

## [1] "SeniorCitizen1" "tenure"
## [3] "PhoneServiceYes" "OnlineSecurityYes"
## [5] "OnlineBackupYes" "DeviceProtectionYes"
## [7] "TechSupportYes" "PaperlessBillingYes"
## [9] "MonthlyCharges" "Contract.One.year"
## [11] "Contract.Two.year" "PaymentMethod.Electronic.check"
```

### #Final Dataset

```
data=df[, -c(1,3,4,7,8,13,14,20,22)]
dim(data)
```

```
## [1] 7043 13
```

#### *#train-test split:*

```
index1=sample(1:nrow(data),floor(0.7*nrow(data)))
train=data[index1,]
remaining=data[-index1,]
index2=sample(1:nrow(remaining),floor(2/3*nrow(remaining)))
crossval=remaining[index2,]
test=remaining[-index2,]
actual_churn=crossval$Churn
```

#### *#logistic regression*

```
logistic.fit=glm(Churn~.,data=train,family="binomial")
logistic.predict=rep("No",nrow(crossval))
predicted_prob=predict(logistic.fit,newdata=crossval,type="response")
logistic.predict[predicted_prob>0.5]="Yes"
table(logistic.predict,actual_churn)
```

```
##           actual_churn
## logistic.predict No Yes
##           No  938 169
##           Yes 100 201
```

```
mean(logistic.predict==actual_churn)
```

```
## [1] 0.8089489
```

#### *#lda fit*

```
lda.fit=lda(Churn~.,data=train)
lda.predict=predict(lda.fit,crossval)$class
table(lda.predict,actual_churn)
```

```
##           actual_churn
## lda.predict  No Yes
##           No  930 169
##           Yes 108 201
```

```
mean(lda.predict==actual_churn)
```

```
## [1] 0.803267
```

#### *#Classification Tree*

```
tree.fit=tree(Churn~.,train,method="class")
summary(tree.fit)
```

```
##
```

```
## Classification tree:
```

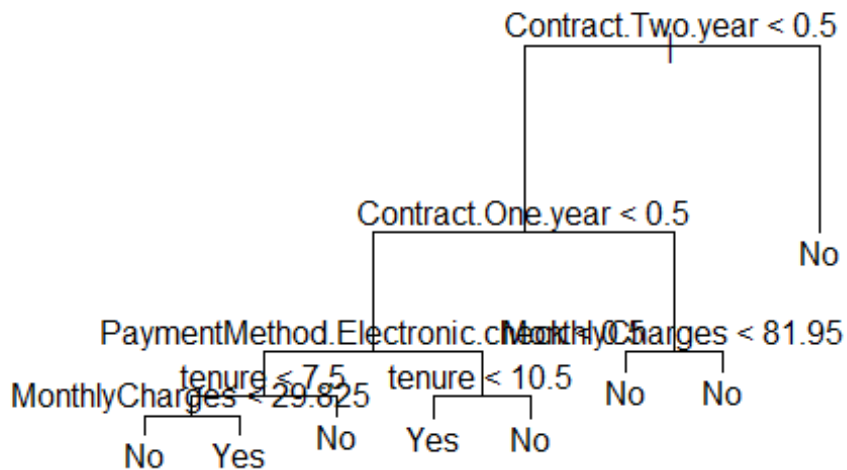
```
## tree(formula = Churn ~ ., data = train, method = "class")
```

```
## Variables actually used in tree construction:
```

```
## [1] "Contract.Two.year" "Contract.One.year"
```

```
## [3] "PaymentMethod.Electronic.check" "tenure"
## [5] "MonthlyCharges"
## Number of terminal nodes: 8
## Residual mean deviance: 0.8856 = 4359 / 4922
## Misclassification error rate: 0.2152 = 1061 / 4930

plot(tree.fit)
text(tree.fit,pretty=0,cex=1)
```



```
tree.predict=predict(tree.fit,crossval,type="class")
table(predicted_churn=tree.predict,actual_churn)

##           actual_churn
## predicted_churn  No  Yes
##           No   920 226
##           Yes  118 144

mean(tree.predict==actual_churn)

## [1] 0.7556818

#Random Forest

rf.fit=randomForest(Churn~.,data=train,ntree=200,mtry=4)
rf.predict=predict(rf.fit,crossval)
table(predicted_churn=rf.predict,actual_churn)
```

```

##                actual_churn
## predicted_churn  No Yes
##                No  917 178
##                Yes  121 192

mean(rf.predict==actual_churn)

## [1] 0.787642

misclassification_rate_logistic=(mean(logistic.predict!=actual_churn))*100
misclassification_rate_lda=(mean(lda.predict!=actual_churn))*100
misclassification_rate_tree=(mean(tree.predict!=actual_churn))*100
misclassification_rate_forest=(mean(rf.predict!=actual_churn))*100
paste("Misclassification Error Rate for Logistic Regression
is",misclassification_rate_logistic,"%")

## [1] "Misclassification Error Rate for Logistic Regression is
19.1051136363636 %"

paste("Misclassification Error Rate for Linear Discriminant Analysis is
is",misclassification_rate_lda,"%")

## [1] "Misclassification Error Rate for Linear Discriminant Analysis is is
19.6732954545455 %"

paste("Misclassification Error Rate for Decision Tree
is",misclassification_rate_tree,"%")

## [1] "Misclassification Error Rate for Decision Tree is 24.4318181818182 %"

paste("Misclassification Error Rate for Random Forest
is",misclassification_rate_forest,"%")

## [1] "Misclassification Error Rate for Random Forest is 21.2357954545455 %"

#Choice is Logistic Regression
#Fit on test dataset
predicted_prob=predict(logistic.fit,newdata=test,type="response")
logistic.predict.test=rep("No",nrow(test))
logistic.predict.test[predicted_prob>0.5]="Yes"
actual.churn.test=test$Churn
table(logistic.predict.test,actual.churn.test)

##                actual.churn.test
## logistic.predict.test  No Yes
##                No  474  72
##                Yes  56 103

misclassification.final=mean(logistic.predict.test!=actual.churn.test)*100
paste("Misclassification Error Rate for final model
is",misclassification.final,"%")

## [1] "Misclassification Error Rate for final model is 18.1560283687943 %"

```

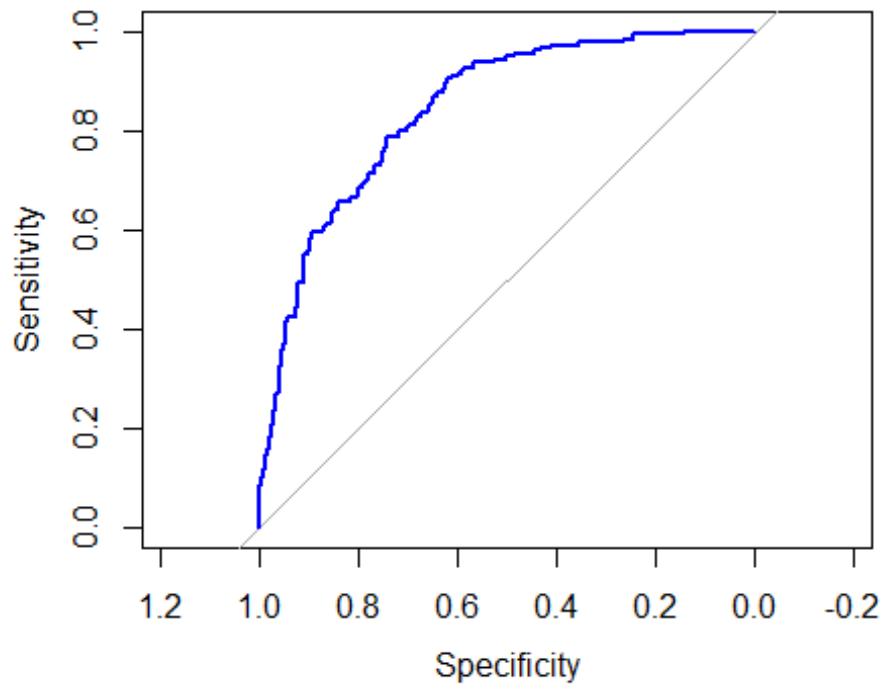
```
#Assessing final model accuracy via ROC curve
```

```
ROC=roc(actual.churn.test,predicted_prob)
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
plot(ROC,col="blue")
```



```
auc(ROC)
```

```
## Area under the curve: 0.8427
```