Untitled 9

Gangadhar Epili

April 22, 2019

library(readr)

## Warning: package 'readr' was built under R version 3.5.2

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.2

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.5.2

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)

## Warning: package 'tidyr' was built under R version 3.5.2

library(corrplot)

## Warning: package 'corrplot' was built under R version 3.5.3

## corrplot 0.84 loaded

library(caret)

## Warning: package 'caret' was built under R version 3.5.3

## Loading required package: lattice

#library(rms)  
library(MASS)

## Warning: package 'MASS' was built under R version 3.5.2

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(e1071)

## Warning: package 'e1071' was built under R version 3.5.2

library(ROCR)

## Warning: package 'ROCR' was built under R version 3.5.3

## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.5.3

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

library(gplots)  
library(pROC)

## Warning: package 'pROC' was built under R version 3.5.2

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

##   
## Attaching package: 'pROC'

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

library(rpart)  
library(randomForest)

## Warning: package 'randomForest' was built under R version 3.5.3

## randomForest 4.6-14

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

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

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

library(ggpubr)

## Warning: package 'ggpubr' was built under R version 3.5.3

## Loading required package: magrittr

## Warning: package 'magrittr' was built under R version 3.5.2

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:tidyr':  
##   
## extract

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.5.2

## -- Attaching packages ---------------------------------------------------------------- tidyverse 1.2.1 --

## v tibble 2.0.1 v stringr 1.4.0  
## v purrr 0.3.0 v forcats 0.3.0

## Warning: package 'tibble' was built under R version 3.5.2

## Warning: package 'purrr' was built under R version 3.5.2

## Warning: package 'stringr' was built under R version 3.5.2

## Warning: package 'forcats' was built under R version 3.5.2

## -- Conflicts ------------------------------------------------------------------- tidyverse\_conflicts() --  
## x randomForest::combine() masks dplyr::combine()  
## x magrittr::extract() masks tidyr::extract()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x purrr::lift() masks caret::lift()  
## x randomForest::margin() masks ggplot2::margin()  
## x MASS::select() masks dplyr::select()  
## x purrr::set\_names() masks magrittr::set\_names()

library(magrittr)  
library(data.table)

## Warning: package 'data.table' was built under R version 3.5.2

##   
## Attaching package: 'data.table'

## The following object is masked from 'package:purrr':  
##   
## transpose

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

library(reshape2)

## Warning: package 'reshape2' was built under R version 3.5.3

##   
## Attaching package: 'reshape2'

## The following objects are masked from 'package:data.table':  
##   
## dcast, melt

## The following object is masked from 'package:tidyr':  
##   
## smiths

library(stats)

telcoCust <- read.csv("C:/Users/epili/Desktop/MVA/Telecom-Customer-Churn-prediction-master/Telco-Customer-Churn.csv")  
str(telcoCust)

## '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 ...

# Change Senior Citizen into factor  
telcoCust$SeniorCitizen <- factor(telcoCust$SeniorCitizen)  
  
# Factor the response variable into 0, 1  
telcoCust$Churn <- factor(ifelse(telcoCust$Churn == 'No', 0, 1))  
  
# replace missing data with mean values  
telcoCust$TotalCharges <- ifelse(is.na(telcoCust$TotalCharges),   
 ave(telcoCust$TotalCharges, FUN = function(x) mean(x, na.rm=TRUE)),  
 telcoCust$TotalCharges)

Data Visualization

createplot <- function(dst, column, name) {  
 plt <- ggplot(dst, aes(x=column, fill=(Churn))) +   
 ggtitle(name) +   
 xlab(name) +  
 ylab("Percentage") +  
 geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.7) +   
 theme\_minimal() +  
 theme(legend.position="none", axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_fill\_manual(values=c("#d0d0d0", "#E69F00"))  
 return(plt)  
}  
  
# Plot 1 by gender   
p1 <- createplot(telcoCust, telcoCust$gender, "Gender")   
# plot 2 by Senior Citizen  
p2 <- createplot(telcoCust, telcoCust$SeniorCitizen, "Senior Citizen")  
# plot 3 by Partner  
p3 <- createplot(telcoCust, telcoCust$Partner, "Partner")  
# plot 4 by Dependents  
p4 <- createplot(telcoCust, telcoCust$Dependents, "Dependents")  
# plot 5 by Phone Service  
p5 <- createplot(telcoCust, telcoCust$PhoneService, "Phone Service")  
# plot 6 by Multiple Lines  
p6 <- createplot(telcoCust, telcoCust$MultipleLines, "Multiple Lines")  
# plot 7 by Internet Service  
p7 <- createplot(telcoCust, telcoCust$InternetService, "Internet Service")  
# plot 8 by Online Security  
p8 <- createplot(telcoCust, telcoCust$OnlineSecurity, "Online Security")  
  
# draw the plot grid  
library(gridExtra)

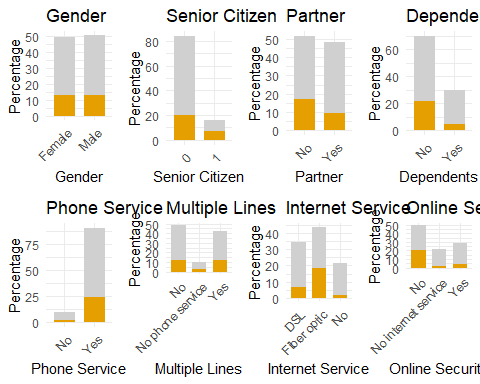
## Warning: package 'gridExtra' was built under R version 3.5.2

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:randomForest':  
##   
## combine

## The following object is masked from 'package:dplyr':  
##   
## combine

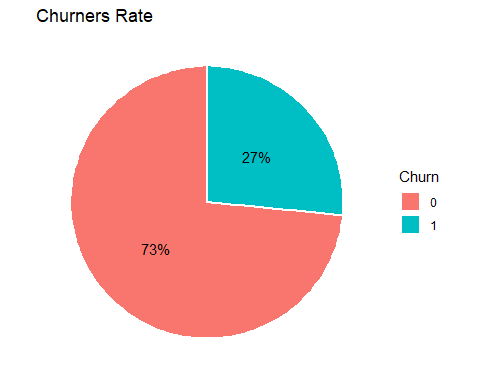
grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, ncol=4)



# Churn Rate  
telcoCust1 <- telcoCust %>% group\_by(Churn) %>%  
 summarise(Count = length(Churn)) %>%  
 mutate(Rate = Count / sum(Count)\*100.0)

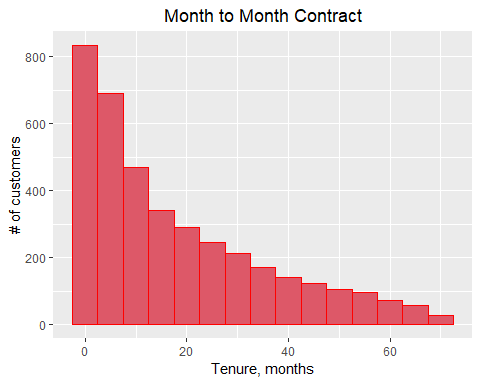
## Warning: package 'bindrcpp' was built under R version 3.5.3

ggplot(telcoCust1, aes(x = '', y = Rate, fill = Churn)) +  
 geom\_bar(width = 1, size = 1, color = 'white', stat = 'identity') +  
 coord\_polar('y') +  
 geom\_text(aes(label = paste0(round(Rate), '%')),  
 position = position\_stack(vjust = 0.5)) +  
 labs(title = 'Churners Rate') +  
 theme\_classic() +  
 theme(axis.line = element\_blank(),axis.title.x = element\_blank(),axis.title.y = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text = element\_blank())

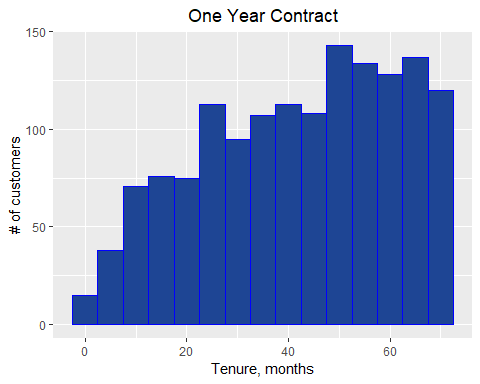


Tenure for different types of contract histograms

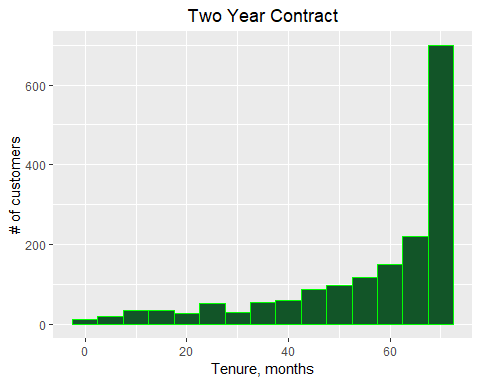
#Tenure for month-to month option  
mtm <- subset(telcoCust, telcoCust$Contract=="Month-to-month")  
mtmplot1 <- ggplot(mtm, aes(x=tenure)) + geom\_histogram(binwidth=5, color="red", fill="#DD5868") +  
 ggtitle("Month to Month Contract") + theme(plot.title = element\_text(hjust = 0.5)) + xlab("Tenure, months") + ylab("# of customers")  
mtmplot1



#Tenure for one year contract option  
oneyear <- subset(telcoCust, telcoCust$Contract=="One year")  
mtmplot2 <- ggplot(oneyear, aes(x=tenure)) + geom\_histogram(binwidth=5, color="blue", fill="#1E4594") +  
 ggtitle("One Year Contract") + theme(plot.title = element\_text(hjust = 0.5)) + xlab("Tenure, months") + ylab("# of customers")  
mtmplot2



#Tenure for two year contract option  
twoyear <- subset(telcoCust, telcoCust$Contract=="Two year")  
mtmplot3 <- ggplot(twoyear, aes(x=tenure)) + geom\_histogram(binwidth=5, color="green", fill="#125528") +  
 ggtitle("Two Year Contract") + theme(plot.title = element\_text(hjust = 0.5)) + xlab("Tenure, months") + ylab("# of customers")  
mtmplot3



#Converting Monthly Charges to Factor variable  
telcoCust$MonthlyChargesBin <- NA  
telcoCust$MonthlyChargesBin[telcoCust$MonthlyCharges > 0 & telcoCust$MonthlyCharges <= 20] <- '20'  
telcoCust$MonthlyChargesBin[telcoCust$MonthlyCharges > 20 & telcoCust$MonthlyCharges <= 40] <- '40'  
telcoCust$MonthlyChargesBin[telcoCust$MonthlyCharges > 40 & telcoCust$MonthlyCharges <= 60] <- '60'  
telcoCust$MonthlyChargesBin[telcoCust$MonthlyCharges > 60 & telcoCust$MonthlyCharges <= 80] <- '80'  
telcoCust$MonthlyChargesBin[telcoCust$MonthlyCharges > 80 & telcoCust$MonthlyCharges <= 100] <- '100'  
telcoCust$MonthlyChargesBin[telcoCust$MonthlyCharges > 100 & telcoCust$MonthlyCharges <= 120] <- '120'  
telcoCust$MonthlyChargesBin[telcoCust$MonthlyCharges > 120 & telcoCust$MonthlyCharges <= 140] <- '140'  
telcoCust$MonthlyChargesBin[telcoCust$MonthlyCharges > 140 & telcoCust$MonthlyCharges <= 160] <- '160'  
telcoCust$MonthlyChargesBin[telcoCust$MonthlyCharges > 160 & telcoCust$MonthlyCharges <= 180] <- '180'  
telcoCust$MonthlyChargesBin <- factor(telcoCust$MonthlyChargesBin,   
 levels = c('20', '40', '60', '80', '100', '120', '140', '160', '180'))  
  
  
# Remove PhoneService & InternetService  
telcoCust1 <- telcoCust[,-c(7,9)]  
  
# Remove Customer ID, Phone Service, Internet Service & monthly charges column  
telco <- telcoCust[, -c(1, 7, 9, 19)]   
  
# Replace No Phone/Internet Service of all service columns  
telco$MultipleLines <- replace(telco$MultipleLines, telco$MultipleLines == 'No phone service', 'No')  
telco$OnlineSecurity <- replace(telco$OnlineSecurity, telco$OnlineSecurity == 'No internet service', 'No')  
telco$OnlineBackup <- replace(telco$OnlineBackup, telco$OnlineBackup == 'No internet service', 'No')  
telco$DeviceProtection <- replace(telco$DeviceProtection, telco$DeviceProtection == 'No internet service', 'No')  
telco$TechSupport <- replace(telco$TechSupport, telco$TechSupport == 'No internet service', 'No')  
telco$StreamingTV <- replace(telco$StreamingTV, telco$StreamingTV == 'No internet service', 'No')  
telco$StreamingMovies <- replace(telco$StreamingMovies, telco$StreamingMovies == 'No internet service', 'No')  
  
# Churn Rate of top-tier customers  
topTierCust <- telcoCust1[which(telcoCust1$MultipleLines == 'Yes' & telcoCust1$OnlineSecurity == 'Yes' &   
 telcoCust1$OnlineBackup == 'Yes' & telcoCust1$DeviceProtection == 'Yes' &   
 telcoCust1$TechSupport == 'Yes' & telcoCust1$StreamingTV == 'Yes' &   
 telcoCust1$StreamingMovies == 'Yes'), ]  
paste0('Top-tier customers: ', round((length(which(topTierCust$Churn == 1)) / nrow(topTierCust))\*100, 2), '%')

## [1] "Top-tier customers: 5.29%"

topTier = topTierCust[, c(2, 3, 4, 5, 6, 14, 15, 16, 20, 19)]  
names(topTier)

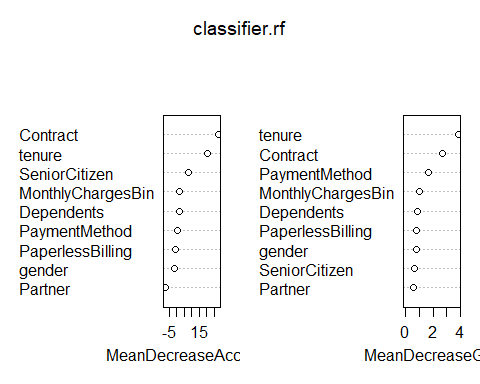
## [1] "gender" "SeniorCitizen" "Partner"   
## [4] "Dependents" "tenure" "Contract"   
## [7] "PaperlessBilling" "PaymentMethod" "MonthlyChargesBin"  
## [10] "Churn"

library(caTools)

## Warning: package 'caTools' was built under R version 3.5.2

set.seed(234)  
split = sample.split(topTier$Churn, SplitRatio = 0.7)  
trainingSet\_topTier = subset(topTier, split == TRUE)  
testSet\_topTier = subset(topTier, split == FALSE)

#Random Forest   
library(randomForest)  
set.seed(456)  
classifier.rf <- randomForest(as.factor(Churn) ~ gender + SeniorCitizen + Partner + Dependents + tenure + Contract + PaperlessBilling + PaymentMethod + MonthlyChargesBin, data = trainingSet\_topTier,importance = TRUE,ntree = 3000)  
  
varImpPlot(classifier.rf)



y\_Pred <- predict(classifier.rf, testSet\_topTier)  
confusionMatrix(testSet\_topTier$Churn, y\_Pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 58 1  
## 1 3 0  
##   
## Accuracy : 0.9355   
## 95% CI : (0.843, 0.9821)  
## No Information Rate : 0.9839   
## P-Value [Acc > NIR] : 0.9967   
##   
## Kappa : -0.0248   
##   
## Mcnemar's Test P-Value : 0.6171   
##   
## Sensitivity : 0.9508   
## Specificity : 0.0000   
## Pos Pred Value : 0.9831   
## Neg Pred Value : 0.0000   
## Prevalence : 0.9839   
## Detection Rate : 0.9355   
## Detection Prevalence : 0.9516   
## Balanced Accuracy : 0.4754   
##   
## 'Positive' Class : 0   
##

# Contract term  
paste0('The churn rate of month-to-month customers is ',  
 round(length(which(telcoCust$Contract == 'Month-to-month' & telcoCust$Churn == 1)) /   
 length(telcoCust$Contract == 'Month-to-month')\*100, 2), '%')

## [1] "The churn rate of month-to-month customers is 23.5%"

paste0('The churn rate of one year customers is ',  
 round(length(which(telcoCust$Contract == 'One year' & telcoCust$Churn == 1)) /   
 length(telcoCust$Contract == 'One year')\*100, 2), '%')

## [1] "The churn rate of one year customers is 2.36%"

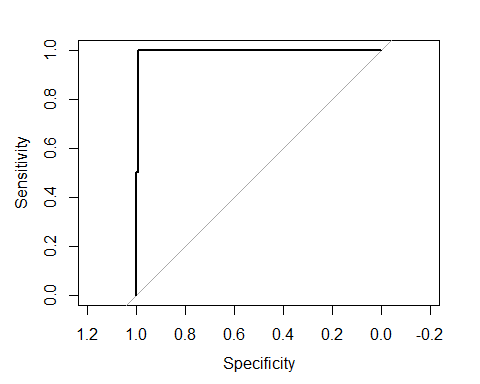
paste0('The churn rate of two year customers is ',  
 round(length(which(telcoCust$Contract == 'Two year' & telcoCust$Churn == 1)) /   
 length(telcoCust$Contract == 'Two year')\*100, 2), '%')

## [1] "The churn rate of two year customers is 0.68%"

#SVM  
  
svm\_fit <- svm(Churn ~ ., probability = TRUE, data = trainingSet\_topTier )  
svm\_fit

##   
## Call:  
## svm(formula = Churn ~ ., data = trainingSet\_topTier, probability = TRUE)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: radial   
## cost: 1   
## gamma: 0.05   
##   
## Number of Support Vectors: 34

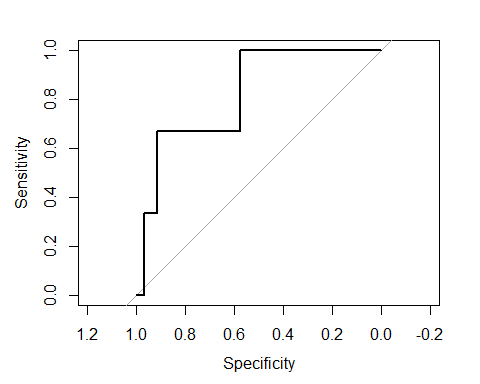
svm\_preds\_train <- predict(svm\_fit, trainingSet\_topTier, probability = TRUE)  
svm\_train\_probs <- data.frame(attr(svm\_preds\_train, "probabilities"))  
plot(roc(trainingSet\_topTier$Churn, svm\_train\_probs$X1))



svmroc\_train\_simple <- roc(trainingSet\_topTier$Churn, svm\_train\_probs$X1)  
svmauc\_train\_simple <- auc(svmroc\_train\_simple)  
svmauc\_train\_simple

## Area under the curve: 0.9964

svm\_preds\_test <- predict(svm\_fit, testSet\_topTier, probability = TRUE)  
svm\_test\_probs <- data.frame(attr(svm\_preds\_test, "probabilities"))  
plot(roc(testSet\_topTier$Churn, svm\_test\_probs$X1))



svmroc\_test <- roc(testSet\_topTier$Churn, svm\_test\_probs$X1)  
svmauc\_test <- auc(svmroc\_test)  
svmauc\_test

## Area under the curve: 0.8192

confusionMatrix(svm\_preds\_test,testSet\_topTier$Churn)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 59 3  
## 1 0 0  
##   
## Accuracy : 0.95165   
## 95% CI : (0.865, 0.9899)  
## No Information Rate : 0.9516   
## P-Value [Acc > NIR] : 0.6473   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : 0.2482   
##   
## Sensitivity : 1.0000   
## Specificity : 0.0000   
## Pos Pred Value : 0.9516   
## Neg Pred Value : NaN   
## Prevalence : 0.9516   
## Detection Rate : 0.9516   
## Detection Prevalence : 1.0000   
## Balanced Accuracy : 0.5000   
##   
## 'Positive' Class : 0   
##

We observe that the model is a good fit because the AUC values for training and test are high and similar. We may consider improving the SVM model by performing a grid search for values of C, gamma and degree of the kernel that gives higher accuracy using k-fold cross-validation.

#Logistic Regression  
  
# Remove Customer ID, Phone Service, Internet Service & monthly charges column  
telco <- telcoCust[, -c(1, 7, 9, 19)]   
  
# Replace No Phone/Internet Service of all service columns  
telco$MultipleLines <- replace(telco$MultipleLines, telco$MultipleLines == 'No phone service', 'No')  
telco$OnlineSecurity <- replace(telco$OnlineSecurity, telco$OnlineSecurity == 'No internet service', 'No')  
telco$OnlineBackup <- replace(telco$OnlineBackup, telco$OnlineBackup == 'No internet service', 'No')  
telco$DeviceProtection <- replace(telco$DeviceProtection, telco$DeviceProtection == 'No internet service', 'No')  
telco$TechSupport <- replace(telco$TechSupport, telco$TechSupport == 'No internet service', 'No')  
telco$StreamingTV <- replace(telco$StreamingTV, telco$StreamingTV == 'No internet service', 'No')  
telco$StreamingMovies <- replace(telco$StreamingMovies, telco$StreamingMovies == 'No internet service', 'No')

# Building the optimal model using backward elimination  
library(stats)  
  
# Fit classifier to the dataset  
classifier <- glm(formula = Churn ~ .,  
 family = 'binomial',  
 data = telco)  
  
# identify variables using backward elimination  
classifier <- step(classifier, direction = 'backward')

## Start: AIC=5974.15  
## Churn ~ gender + SeniorCitizen + Partner + Dependents + tenure +   
## MultipleLines + OnlineSecurity + OnlineBackup + DeviceProtection +   
## TechSupport + StreamingTV + StreamingMovies + Contract +   
## PaperlessBilling + PaymentMethod + TotalCharges + MonthlyChargesBin  
##   
## Df Deviance AIC  
## - Partner 1 5924.2 5972.2  
## - gender 1 5924.4 5972.4  
## - DeviceProtection 1 5925.1 5973.1  
## <none> 5924.1 5974.1  
## - StreamingTV 1 5926.2 5974.2  
## - MultipleLines 1 5926.6 5974.6  
## - StreamingMovies 1 5926.8 5974.8  
## - Dependents 1 5928.8 5976.8  
## - OnlineBackup 1 5930.0 5978.0  
## - SeniorCitizen 1 5935.2 5983.2  
## - TotalCharges 1 5942.3 5990.3  
## - TechSupport 1 5952.1 6000.1  
## - OnlineSecurity 1 5953.6 6001.6  
## - PaperlessBilling 1 5956.0 6004.0  
## - PaymentMethod 3 5968.5 6012.5  
## - MonthlyChargesBin 5 5987.0 6027.0  
## - tenure 1 6024.6 6072.6  
## - Contract 2 6038.1 6084.1  
##   
## Step: AIC=5972.17  
## Churn ~ gender + SeniorCitizen + Dependents + tenure + MultipleLines +   
## OnlineSecurity + OnlineBackup + DeviceProtection + TechSupport +   
## StreamingTV + StreamingMovies + Contract + PaperlessBilling +   
## PaymentMethod + TotalCharges + MonthlyChargesBin  
##   
## Df Deviance AIC  
## - gender 1 5924.4 5970.4  
## - DeviceProtection 1 5925.1 5971.1  
## <none> 5924.2 5972.2  
## - StreamingTV 1 5926.2 5972.2  
## - MultipleLines 1 5926.6 5972.6  
## - StreamingMovies 1 5926.8 5972.8  
## - Dependents 1 5929.5 5975.5  
## - OnlineBackup 1 5930.0 5976.0  
## - SeniorCitizen 1 5935.5 5981.5  
## - TotalCharges 1 5942.4 5988.4  
## - TechSupport 1 5952.1 5998.1  
## - OnlineSecurity 1 5953.6 5999.6  
## - PaperlessBilling 1 5956.0 6002.0  
## - PaymentMethod 3 5968.6 6010.6  
## - MonthlyChargesBin 5 5987.0 6025.0  
## - tenure 1 6025.0 6071.0  
## - Contract 2 6038.1 6082.1  
##   
## Step: AIC=5970.39  
## Churn ~ SeniorCitizen + Dependents + tenure + MultipleLines +   
## OnlineSecurity + OnlineBackup + DeviceProtection + TechSupport +   
## StreamingTV + StreamingMovies + Contract + PaperlessBilling +   
## PaymentMethod + TotalCharges + MonthlyChargesBin  
##   
## Df Deviance AIC  
## - DeviceProtection 1 5925.3 5969.3  
## <none> 5924.4 5970.4  
## - StreamingTV 1 5926.4 5970.4  
## - MultipleLines 1 5926.8 5970.8  
## - StreamingMovies 1 5927.1 5971.1  
## - Dependents 1 5929.7 5973.7  
## - OnlineBackup 1 5930.2 5974.2  
## - SeniorCitizen 1 5935.8 5979.8  
## - TotalCharges 1 5942.5 5986.5  
## - TechSupport 1 5952.3 5996.3  
## - OnlineSecurity 1 5953.7 5997.7  
## - PaperlessBilling 1 5956.2 6000.2  
## - PaymentMethod 3 5968.9 6008.9  
## - MonthlyChargesBin 5 5987.3 6023.3  
## - tenure 1 6025.2 6069.2  
## - Contract 2 6038.3 6080.3  
##   
## Step: AIC=5969.3  
## Churn ~ SeniorCitizen + Dependents + tenure + MultipleLines +   
## OnlineSecurity + OnlineBackup + TechSupport + StreamingTV +   
## StreamingMovies + Contract + PaperlessBilling + PaymentMethod +   
## TotalCharges + MonthlyChargesBin  
##   
## Df Deviance AIC  
## - StreamingTV 1 5927.3 5969.3  
## <none> 5925.3 5969.3  
## - StreamingMovies 1 5927.9 5969.9  
## - MultipleLines 1 5928.0 5970.0  
## - Dependents 1 5930.7 5972.7  
## - OnlineBackup 1 5931.0 5973.0  
## - SeniorCitizen 1 5936.7 5978.7  
## - TotalCharges 1 5943.2 5985.2  
## - TechSupport 1 5953.7 5995.7  
## - OnlineSecurity 1 5954.4 5996.4  
## - PaperlessBilling 1 5957.4 5999.4  
## - PaymentMethod 3 5970.3 6008.3  
## - MonthlyChargesBin 5 5987.4 6021.4  
## - tenure 1 6026.4 6068.4  
## - Contract 2 6042.0 6082.0  
##   
## Step: AIC=5969.28  
## Churn ~ SeniorCitizen + Dependents + tenure + MultipleLines +   
## OnlineSecurity + OnlineBackup + TechSupport + StreamingMovies +   
## Contract + PaperlessBilling + PaymentMethod + TotalCharges +   
## MonthlyChargesBin  
##   
## Df Deviance AIC  
## <none> 5927.3 5969.3  
## - MultipleLines 1 5929.6 5969.6  
## - StreamingMovies 1 5930.4 5970.4  
## - Dependents 1 5932.5 5972.5  
## - OnlineBackup 1 5933.5 5973.5  
## - SeniorCitizen 1 5938.5 5978.5  
## - TotalCharges 1 5945.4 5985.4  
## - TechSupport 1 5955.3 5995.3  
## - OnlineSecurity 1 5957.4 5997.4  
## - PaperlessBilling 1 5960.2 6000.2  
## - PaymentMethod 3 5973.2 6009.2  
## - MonthlyChargesBin 5 5999.7 6031.7  
## - tenure 1 6028.1 6068.1  
## - Contract 2 6042.3 6080.3

summary(classifier)

##   
## Call:  
## glm(formula = Churn ~ SeniorCitizen + Dependents + tenure + MultipleLines +   
## OnlineSecurity + OnlineBackup + TechSupport + StreamingMovies +   
## Contract + PaperlessBilling + PaymentMethod + TotalCharges +   
## MonthlyChargesBin, family = "binomial", data = telco)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8692 -0.7025 -0.2905 0.7326 3.4256   
##   
## Coefficients:  
## Estimate Std. Error z value  
## (Intercept) -1.1584264 0.1800693 -6.433  
## SeniorCitizen1 0.2792302 0.0832956 3.352  
## DependentsYes -0.1835786 0.0805872 -2.278  
## tenure -0.0566821 0.0061309 -9.245  
## MultipleLinesYes 0.1210102 0.0800674 1.511  
## OnlineSecurityYes -0.4563750 0.0839345 -5.437  
## OnlineBackupYes -0.1935930 0.0778352 -2.487  
## TechSupportYes -0.4459391 0.0849453 -5.250  
## StreamingMoviesYes 0.1517842 0.0853702 1.778  
## ContractOne year -0.7769263 0.1060737 -7.324  
## ContractTwo year -1.5398925 0.1739701 -8.851  
## PaperlessBillingYes 0.4185645 0.0732658 5.713  
## PaymentMethodCredit card (automatic) -0.0734528 0.1131169 -0.649  
## PaymentMethodElectronic check 0.3784082 0.0933542 4.053  
## PaymentMethodMailed check -0.1491074 0.1127946 -1.322  
## TotalCharges 0.0002976 0.0000716 4.157  
## MonthlyChargesBin40 0.4602844 0.1754231 2.624  
## MonthlyChargesBin60 0.9323477 0.1735501 5.372  
## MonthlyChargesBin80 1.1741521 0.1756034 6.686  
## MonthlyChargesBin100 1.4229276 0.1942970 7.323  
## MonthlyChargesBin120 1.7837095 0.2529982 7.050  
## Pr(>|z|)   
## (Intercept) 1.25e-10 \*\*\*  
## SeniorCitizen1 0.000801 \*\*\*  
## DependentsYes 0.022726 \*   
## tenure < 2e-16 \*\*\*  
## MultipleLinesYes 0.130698   
## OnlineSecurityYes 5.41e-08 \*\*\*  
## OnlineBackupYes 0.012875 \*   
## TechSupportYes 1.52e-07 \*\*\*  
## StreamingMoviesYes 0.075412 .   
## ContractOne year 2.40e-13 \*\*\*  
## ContractTwo year < 2e-16 \*\*\*  
## PaperlessBillingYes 1.11e-08 \*\*\*  
## PaymentMethodCredit card (automatic) 0.516111   
## PaymentMethodElectronic check 5.05e-05 \*\*\*  
## PaymentMethodMailed check 0.186189   
## TotalCharges 3.22e-05 \*\*\*  
## MonthlyChargesBin40 0.008694 \*\*   
## MonthlyChargesBin60 7.78e-08 \*\*\*  
## MonthlyChargesBin80 2.29e-11 \*\*\*  
## MonthlyChargesBin100 2.42e-13 \*\*\*  
## MonthlyChargesBin120 1.79e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 8150.1 on 7042 degrees of freedom  
## Residual deviance: 5927.3 on 7022 degrees of freedom  
## AIC: 5969.3  
##   
## Number of Fisher Scoring iterations: 6

# Remove the variables from the dataset  
telco1 <- telco[, c(2, 4, 5, 7, 8, 10, 12, 13, 14, 15, 16, 17, 18)]  
  
# split the dataset into training set & test set  
library(caTools)  
set.seed(123)  
split = sample.split(telco1$Churn, SplitRatio = .7)  
trainingSet <- subset(telco1, split == TRUE)  
testSet <- subset(telco1, split == FALSE)  
  
# Feature Scaling  
trainingSet[c(3,11)] = scale(trainingSet[c(3,11)])  
testSet[c(3,11)] = scale(testSet[c(3,11)])

# Fitting classifier to the Training set  
classifier <- glm(formula = Churn ~ .,  
 family = 'binomial',  
 data = trainingSet)  
summary(classifier)

##   
## Call:  
## glm(formula = Churn ~ ., family = "binomial", data = trainingSet)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8609 -0.6868 -0.2754 0.7251 3.4307   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -2.27568 0.21533 -10.568 < 2e-16  
## SeniorCitizen1 0.27672 0.10038 2.757 0.00584  
## DependentsYes -0.19817 0.09815 -2.019 0.04349  
## tenure -1.47296 0.18660 -7.894 2.93e-15  
## OnlineSecurityYes -0.46817 0.10187 -4.596 4.31e-06  
## OnlineBackupYes -0.22614 0.09314 -2.428 0.01518  
## TechSupportYes -0.52479 0.10338 -5.076 3.85e-07  
## StreamingMoviesYes 0.09907 0.10206 0.971 0.33166  
## ContractOne year -0.82374 0.13090 -6.293 3.12e-10  
## ContractTwo year -1.51401 0.21028 -7.200 6.03e-13  
## PaperlessBillingYes 0.42495 0.08849 4.802 1.57e-06  
## PaymentMethodCredit card (automatic) 0.01335 0.13700 0.097 0.92234  
## PaymentMethodElectronic check 0.45124 0.11363 3.971 7.15e-05  
## PaymentMethodMailed check -0.11702 0.13753 -0.851 0.39484  
## TotalCharges 0.77288 0.19814 3.901 9.59e-05  
## MonthlyChargesBin40 0.30301 0.20812 1.456 0.14541  
## MonthlyChargesBin60 0.94515 0.20159 4.688 2.75e-06  
## MonthlyChargesBin80 1.12879 0.20176 5.595 2.21e-08  
## MonthlyChargesBin100 1.44525 0.22046 6.556 5.54e-11  
## MonthlyChargesBin120 1.83800 0.28749 6.393 1.62e-10  
##   
## (Intercept) \*\*\*  
## SeniorCitizen1 \*\*   
## DependentsYes \*   
## tenure \*\*\*  
## OnlineSecurityYes \*\*\*  
## OnlineBackupYes \*   
## TechSupportYes \*\*\*  
## StreamingMoviesYes   
## ContractOne year \*\*\*  
## ContractTwo year \*\*\*  
## PaperlessBillingYes \*\*\*  
## PaymentMethodCredit card (automatic)   
## PaymentMethodElectronic check \*\*\*  
## PaymentMethodMailed check   
## TotalCharges \*\*\*  
## MonthlyChargesBin40   
## MonthlyChargesBin60 \*\*\*  
## MonthlyChargesBin80 \*\*\*  
## MonthlyChargesBin100 \*\*\*  
## MonthlyChargesBin120 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 5704.4 on 4929 degrees of freedom  
## Residual deviance: 4069.0 on 4910 degrees of freedom  
## AIC: 4109  
##   
## Number of Fisher Scoring iterations: 6

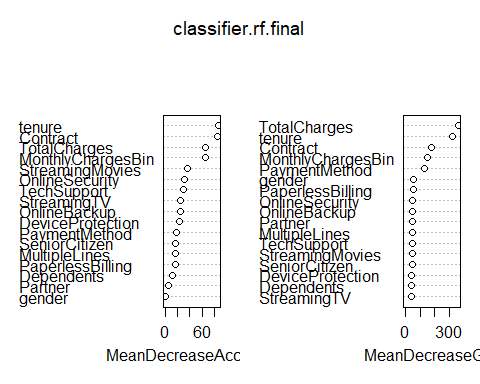
# Predicting the Test set results  
threshold = 0.5  
probPred = predict(classifier, type = 'response', newdata = testSet[-12])  
y\_Pred <- ifelse(probPred > threshold, 1, 0)  
  
  
# Confusion Matrix - simplest but relies on 1 testSet, not relevant  
confusionMatrix(table(testSet$Churn, y\_Pred))

## Confusion Matrix and Statistics  
##   
## y\_Pred  
## 0 1  
## 0 1389 163  
## 1 282 279  
##   
## Accuracy : 0.7894   
## 95% CI : (0.7714, 0.8066)  
## No Information Rate : 0.7908   
## P-Value [Acc > NIR] : 0.5762   
##   
## Kappa : 0.4208   
##   
## Mcnemar's Test P-Value : 2.222e-08   
##   
## Sensitivity : 0.8312   
## Specificity : 0.6312   
## Pos Pred Value : 0.8950   
## Neg Pred Value : 0.4973   
## Prevalence : 0.7908   
## Detection Rate : 0.6574   
## Detection Prevalence : 0.7345   
## Balanced Accuracy : 0.7312   
##   
## 'Positive' Class : 0   
##

# K-folds cross validation  
library(caret)  
set.seed(456)  
folds <- createFolds(trainingSet$Churn, k=10)  
cv <- lapply(folds, function(x) {  
 training\_fold <- trainingSet[-x, ]  
 test\_fold <- trainingSet[x, ]  
 classifier <- glm(formula = Churn ~ .,  
 family = 'binomial',  
 data = training\_fold)  
 threshold = 0.5  
 probPred = predict(classifier, type = 'response', newdata = test\_fold[-12])  
 y\_Pred <- ifelse(probPred > threshold, 1, 0)  
 cm = table(test\_fold[, 12], y\_Pred)  
 accuracy = (cm[1,1] + cm[2,2]) / (cm[1,1] + cm[2,2] + cm[1,2] + cm[2,1])  
 return(accuracy)  
})  
  
#Printing the Mean of Accuracy  
accuracy\_logistic = mean(as.numeric(cv))  
accuracydf <- NULL  
accuracydf$logistic = accuracy\_logistic  
accuracydf

## $logistic  
## [1] 0.801827

#Random Forest   
library(randomForest)  
split = sample.split(telco$Churn, SplitRatio = .7)  
trainingSet <- subset(telco, split == TRUE)  
testSet <- subset(telco, split == FALSE)  
set.seed(456)  
classifier.rf.final <- randomForest(as.factor(Churn)~., data = trainingSet,importance = TRUE,ntree = 1500)  
   
varImpPlot(classifier.rf.final)



names(testSet)

## [1] "gender" "SeniorCitizen" "Partner"   
## [4] "Dependents" "tenure" "MultipleLines"   
## [7] "OnlineSecurity" "OnlineBackup" "DeviceProtection"   
## [10] "TechSupport" "StreamingTV" "StreamingMovies"   
## [13] "Contract" "PaperlessBilling" "PaymentMethod"   
## [16] "TotalCharges" "Churn" "MonthlyChargesBin"

y\_Pred <- predict(classifier.rf.final, testSet[-17])  
confusionMatrix(testSet$Churn, y\_Pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1401 151  
## 1 266 295  
##   
## Accuracy : 0.8027   
## 95% CI : (0.785, 0.8194)  
## No Information Rate : 0.7889   
## P-Value [Acc > NIR] : 0.06346   
##   
## Kappa : 0.4586   
##   
## Mcnemar's Test P-Value : 2.369e-08   
##   
## Sensitivity : 0.8404   
## Specificity : 0.6614   
## Pos Pred Value : 0.9027   
## Neg Pred Value : 0.5258   
## Prevalence : 0.7889   
## Detection Rate : 0.6630   
## Detection Prevalence : 0.7345   
## Balanced Accuracy : 0.7509   
##   
## 'Positive' Class : 0   
##

names(trainingSet)

## [1] "gender" "SeniorCitizen" "Partner"   
## [4] "Dependents" "tenure" "MultipleLines"   
## [7] "OnlineSecurity" "OnlineBackup" "DeviceProtection"   
## [10] "TechSupport" "StreamingTV" "StreamingMovies"   
## [13] "Contract" "PaperlessBilling" "PaymentMethod"   
## [16] "TotalCharges" "Churn" "MonthlyChargesBin"

# K-folds cross validation  
folds <- createFolds(trainingSet$Churn, k=10)  
cv <- lapply(folds, function(x) {  
 training\_fold <- trainingSet[-x, ]  
 test\_fold <- trainingSet[x, ]  
 classifier.rf.final <- randomForest(as.factor(Churn)~., data = training\_fold,importance = TRUE,ntree = 50)  
 y\_Pred <- predict(classifier.rf.final, test\_fold[-17])  
 #probPred = predict(classifier, type = 'response', newdata = test\_fold[-17])  
 #y\_Pred <- ifelse(probPred > threshold, 1, 0)  
 cm = table(test\_fold$Churn, y\_Pred)  
 accuracy = (cm[1,1] + cm[2,2]) / (cm[1,1] + cm[2,2] + cm[1,2] + cm[2,1])  
 return(accuracy)  
})  
  
#Printing the Mean of Accuracy  
accuracy\_randomforest = mean(as.numeric(cv))  
accuracydf$randomforest <- accuracy\_randomforest  
accuracydf

## $logistic  
## [1] 0.801827  
##   
## $randomforest  
## [1] 0.7829566