Customer-churn-logistic-regression in R

Customer churn refers to when a customer ceases his or her relationship with a company.It is also refered as a loss of client or customer in business prespective. If a company has 70% of loyality rate, then churn rate would be 30%. As 80/20 profitability rule 20% of Customers are generating 80% of revenue. So it's really important to know factors affecting users to take this decision.

In this markdown report i am going to show how logistic regression model using R can be used to identify customer churn in telecom dataset.

```
# read the telecom dataset input file
telecomDataframe <- read_csv("~/customer_churn/Telecom.csv")
```

```
# print the structure of the dataframe
print(str(telecomDataframe))
```

```
## Classes 'tbl_df', 'tbl' and '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" ...
##  - attr(*, "spec")=List of 2
##   ..$ cols   :List of 21
##   .. ..$ customerID      : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ gender          : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ SeniorCitizen   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Partner         : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ Dependents      : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ tenure          : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ PhoneService    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
```

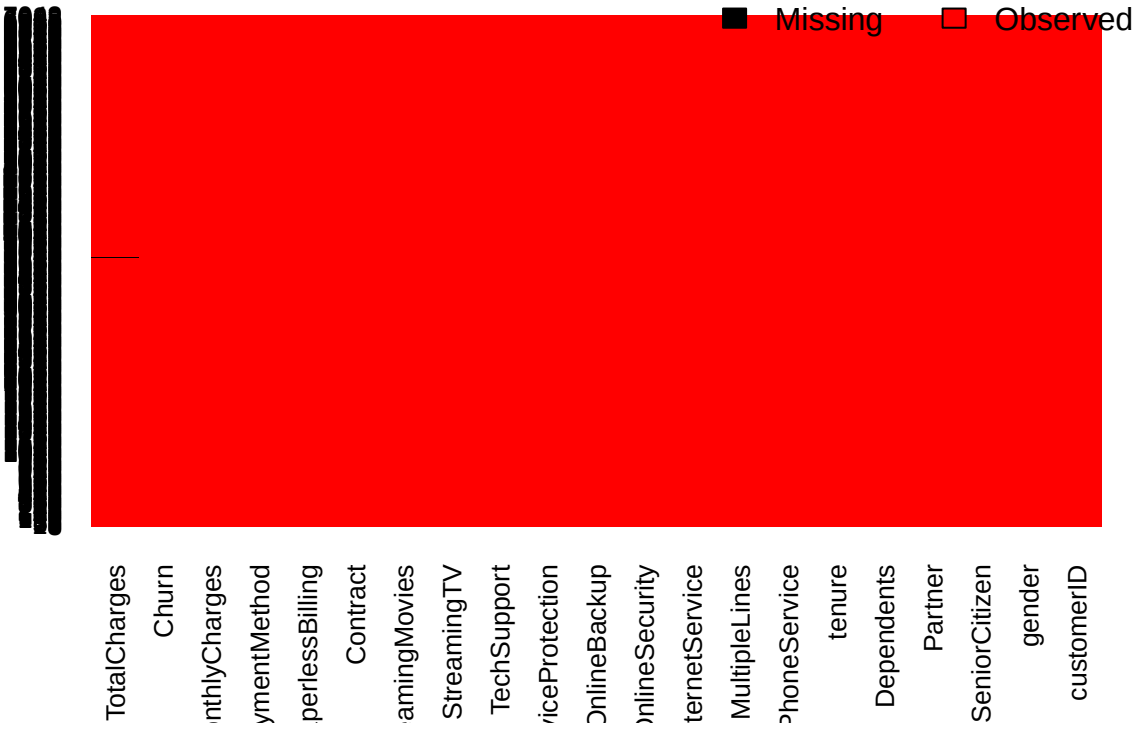
```
##   .. ..$ MultipleLines   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ InternetService : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ OnlineSecurity  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ OnlineBackup    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ DeviceProtection: list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ TechSupport     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ StreamingTV     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ StreamingMovies : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ Contract        : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ PaperlessBilling: list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ PaymentMethod   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ MonthlyCharges  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ TotalCharges    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ Churn           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   ..$ default: list()
##   .. ..- attr(*, "class")= chr  "collector_guess" "collector"
##   ..- attr(*, "class")= chr "col_spec"
## NULL
```

```
# check for the NA values
any(is.na(telecomDataframe))
```

```
## [1] TRUE
```

```
# visualize the missing values using the missing map from the Amelia package
missmap(telecomDataframe,col=c("black","red"))
```

Missingness Map




```
#five num summary
fivenum(telecomDataframe$tenure)
```

```
## [1]  0  9 29 55 72
```

```
# create new column "tenure_interval" from the tenure column
group_tenure <- function(tenure){
  if (tenure >= 0 && tenure <= 5){
    return('0-5 Month')
  }else if(tenure > 5 && tenure <= 10){
    return('6-10 Month')
  }else if (tenure > 10 && tenure <= 15){
    return('11-15 Month')
  }else if (tenure > 15 && tenure <=20){
    return('16-20 Month')
  }else if (tenure > 20 && tenure <=25){
    return('21-25 Month')
  }else if (tenure > 25 && tenure <= 30){
    return('26-30 Month')
  }else if (tenure > 30 && tenure <=35){
    return('31-35 Month')
  }else if (tenure > 35 && tenure <=40){
    return('36-40 Month')
  }else if (tenure > 40 && tenure <=45) {
    return('41-45 Month')
  }else if (tenure > 45 && tenure <=50) {
    return('46-50 Month')
  }else if (tenure > 50 && tenure <=55) {
    return('51-55 Month')
```

```
  }else if(tenure > 55 && tenure <= 60) {
    return('56-60 Month')
  } else if(tenure > 60 && tenure <=65) {
    return('61-66 Month')
  }else if(tenure > 65 && tenure <=70)  {
    return('67-70 Month')
  } else if(tenure > 70){
    return('70+')
  }
}
```

```
# apply group_tenure function on each row of dataframe
telecomDataframe$tenure_interval <- sapply(telecomDataframe$tenure,group_tenure)
telecomDataframe$tenure_interval <- as.factor(telecomDataframe$tenure_interval)

# Ignore the variables with more levels while predicting the model
# Columns "customerID" and "tenure" having more levels
telecomDataframe <- select(telecomDataframe,-customerID,-tenure)
lapply(telecomDataframe, class)
```

```
## $gender
## [1] "character"
##
## $SeniorCitizen
## [1] "integer"
##
## $Partner
## [1] "character"
##
## $Dependents
## [1] "character"
##
## $PhoneService
## [1] "character"
##
## $MultipleLines
## [1] "character"
##
## $InternetService
## [1] "character"
##
## $OnlineSecurity
## [1] "character"
##
## $OnlineBackup
## [1] "character"
##
## $DeviceProtection
## [1] "character"
##
## $TechSupport
## [1] "character"
##
## $StreamingTV
```

```
## [1] "character"
##
## $StreamingMovies
## [1] "character"
##
## $Contract
## [1] "character"
##
## $PaperlessBilling
## [1] "character"
##
## $PaymentMethod
## [1] "character"
##
## $MonthlyCharges
## [1] "numeric"
##
## $TotalCharges
## [1] "numeric"
##
## $Churn
## [1] "character"
##
## $tenure_interval
## [1] "factor"
```

```
# The value of the following columns affecting the model and introducing the NA value for "No phone ser
telecomDataframe$MultipleLines <- as.character(telecomDataframe$MultipleLines)
telecomDataframe$OnlineSecurity <- as.character(telecomDataframe$OnlineSecurity)
telecomDataframe$OnlineBackup <- as.character(telecomDataframe$OnlineBackup)
telecomDataframe$DeviceProtection <- as.character(telecomDataframe$DeviceProtection)
telecomDataframe$TechSupport <- as.character(telecomDataframe$TechSupport)
telecomDataframe$StreamingTV <- as.character(telecomDataframe$StreamingTV)
telecomDataframe$StreamingMovies <- as.character(telecomDataframe$StreamingMovies)
telecomDataframe$InternetService <- as.character(telecomDataframe$InternetService)
```

```
#Replacing using gsub
telecomDataframe$MultipleLines <-gsub("No phone service","No",telecomDataframe$MultipleLines)
telecomDataframe$OnlineSecurity <-gsub("No internet service","No",telecomDataframe$OnlineSecurity)
telecomDataframe$OnlineBackup <-gsub("No internet service","No",telecomDataframe$OnlineBackup)
telecomDataframe$DeviceProtection <-gsub("No internet service","No",telecomDataframe$DeviceProtection)
telecomDataframe$TechSupport <-gsub("No internet service","No",telecomDataframe$TechSupport)
telecomDataframe$StreamingTV <-gsub("No internet service","No",telecomDataframe$StreamingTV)
telecomDataframe$StreamingMovies <-gsub("No internet service","No",telecomDataframe$StreamingMovies)
telecomDataframe$InternetService <-gsub("Fiber optic","Fiber_optic",telecomDataframe$InternetService)
```

```
# converting character variables into factor variables
telecomDataframe$MultipleLines <- as.factor(telecomDataframe$MultipleLines)
telecomDataframe$OnlineSecurity <- as.factor(telecomDataframe$OnlineSecurity)
telecomDataframe$OnlineBackup <- as.factor(telecomDataframe$OnlineBackup)
telecomDataframe$DeviceProtection <- as.factor(telecomDataframe$DeviceProtection)
telecomDataframe$TechSupport <- as.factor(telecomDataframe$TechSupport)
telecomDataframe$StreamingTV <- as.factor(telecomDataframe$StreamingTV)
telecomDataframe$StreamingMovies <- as.factor(telecomDataframe$StreamingMovies)
telecomDataframe$InternetService <- as.factor(telecomDataframe$InternetService)
```

```r
# check the number of NA rows if it is relatively small in number then ignore those rows from the analy
any(is.na(telecomDataframe))
```

```
## [1] TRUE
```

```r
telecomDataframe <- na.omit(telecomDataframe)
```

```r
# set the seed it will output same output when ever the model is executed
set.seed(123)

#spliting train and test
 rows <- sample(nrow(telecomDataframe))
 telecomDataframe <- telecomDataframe[rows,]

 split <- round(nrow(telecomDataframe)*.70)
 trainData <- telecomDataframe[1:split,]
 testData <- telecomDataframe[(split+1):nrow(telecomDataframe),]
 nrow(trainData)/nrow(telecomDataframe)
```

```
## [1] 0.6999431
```

```r
# train glm with custom trainControl
myControl <- trainControl(
  method = "repeatedcv",
  number = 10,
  repeats = 5,
  summaryFunction = twoClassSummary,
  classProbs = TRUE

)
```

```r
##Model
model <- train(Churn~., data=telecomDataframe,method="glm",metric="ROC",
                trControl=myControl)
```

```r
summary(model)
```

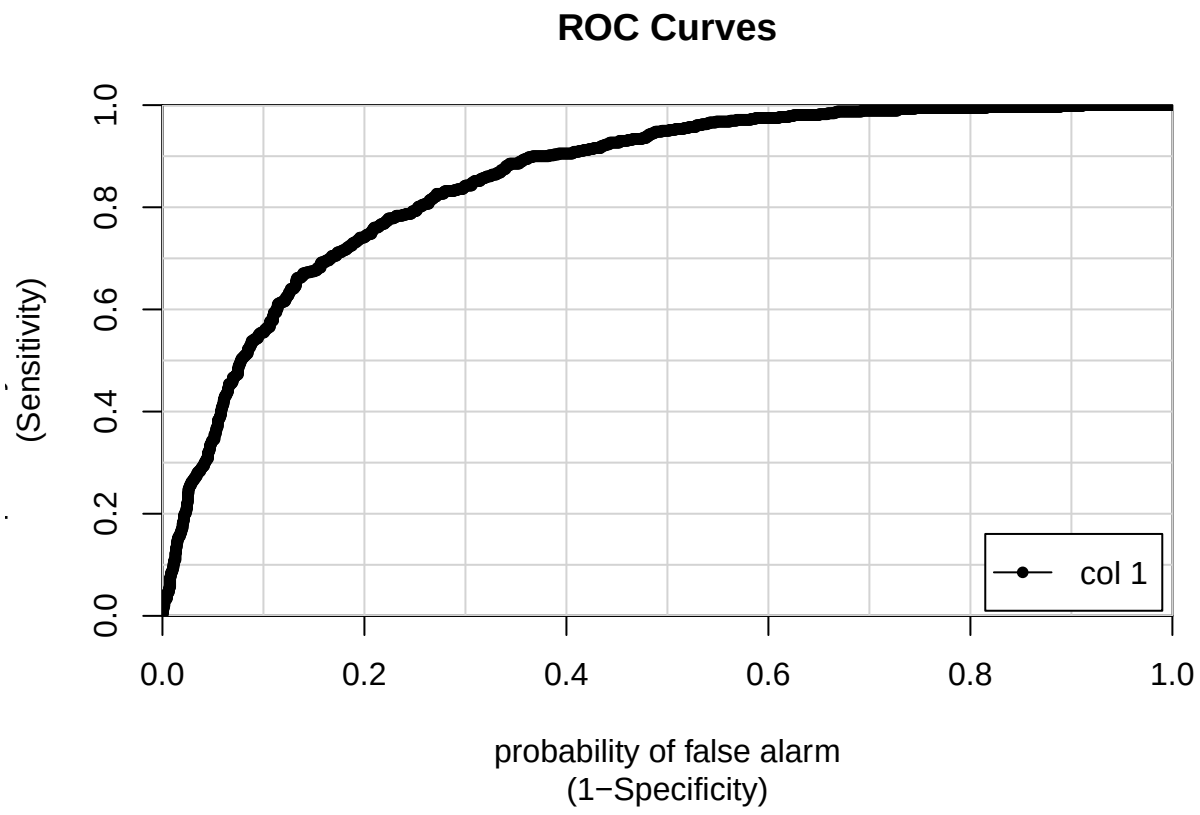
```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1779  -0.6819  -0.2775   0.6307   3.2028
##
## Coefficients:
##                                         Estimate Std. Error z value
## (Intercept)                            1.086e+00  8.239e-01   1.318
## genderMale                            -1.828e-02  6.564e-02  -0.279
## SeniorCitizen                          2.260e-01  8.533e-02   2.649
## PartnerYes                             2.374e-02  7.870e-02   0.302
## DependentsYes                         -1.366e-01  9.063e-02  -1.508
## PhoneServiceYes                        2.494e-01  6.571e-01   0.380
## MultipleLinesYes                       5.240e-01  1.796e-01   2.917
## InternetServiceFiber_optic             1.863e+00  8.082e-01   2.306
## InternetServiceNo                     -1.765e+00  8.177e-01  -2.159
```

```
## OnlineSecurityYes                      -1.345e-01  1.811e-01  -0.743
## OnlineBackupYes                         5.011e-02  1.776e-01   0.282
## DeviceProtectionYes                     2.060e-01  1.785e-01   1.154
## TechSupportYes                         -1.258e-01  1.825e-01  -0.689
## StreamingTVYes                          6.660e-01  3.308e-01   2.013
## StreamingMoviesYes                      6.866e-01  3.310e-01   2.074
## `ContractOne year`                     -7.148e-01  1.095e-01  -6.526
## `ContractTwo year`                     -1.493e+00  1.878e-01  -7.951
## PaperlessBillingYes                     3.568e-01  7.566e-02   4.716
## `PaymentMethodCredit card (automatic)` -1.048e-01  1.142e-01  -0.917
## `PaymentMethodElectronic check`         2.618e-01  9.520e-02   2.750
## `PaymentMethodMailed check`            -1.056e-01  1.169e-01  -0.904
## MonthlyCharges                         -3.837e-02  3.217e-02  -1.193
## TotalCharges                            3.370e-05  7.154e-05   0.471
## `tenure_interval11-15 Month`           -1.046e+00  1.328e-01  -7.874
## `tenure_interval16-20 Month`           -1.387e+00  1.538e-01  -9.017
## `tenure_interval21-25 Month`           -1.494e+00  1.710e-01  -8.736
## `tenure_interval26-30 Month`           -1.886e+00  1.997e-01  -9.444
## `tenure_interval31-35 Month`           -1.846e+00  2.170e-01  -8.507
## `tenure_interval36-40 Month`           -1.637e+00  2.467e-01  -6.636
## `tenure_interval41-45 Month`           -1.976e+00  2.747e-01  -7.193
## `tenure_interval46-50 Month`           -1.883e+00  2.991e-01  -6.297
## `tenure_interval51-55 Month`           -2.038e+00  3.252e-01  -6.267
## `tenure_interval56-60 Month`           -2.371e+00  3.646e-01  -6.503
## `tenure_interval6-10 Month`            -8.924e-01  1.181e-01  -7.556
## `tenure_interval61-66 Month`           -2.710e+00  4.226e-01  -6.414
## `tenure_interval67-70 Month`           -2.040e+00  4.291e-01  -4.754
## `tenure_interval70+`                   -3.325e+00  5.449e-01  -6.102
##                                        Pr(>|z|)
## (Intercept)                             0.18742
## genderMale                              0.78062
## SeniorCitizen                           0.00808 **
## PartnerYes                              0.76294
## DependentsYes                           0.13164
## PhoneServiceYes                         0.70430
## MultipleLinesYes                        0.00354 **
## InternetServiceFiber_optic              0.02113 *
## InternetServiceNo                       0.03088 *
## OnlineSecurityYes                       0.45760
## OnlineBackupYes                         0.77782
## DeviceProtectionYes                     0.24843
## TechSupportYes                          0.49063
## StreamingTVYes                          0.04408 *
## StreamingMoviesYes                      0.03807 *
## `ContractOne year`                     6.75e-11 ***
## `ContractTwo year`                     1.85e-15 ***
## PaperlessBillingYes                    2.41e-06 ***
## `PaymentMethodCredit card (automatic)`  0.35891
## `PaymentMethodElectronic check`         0.00596 **
## `PaymentMethodMailed check`             0.36612
## MonthlyCharges                          0.23290
## TotalCharges                            0.63754
## `tenure_interval11-15 Month`           3.42e-15 ***
## `tenure_interval16-20 Month`            < 2e-16 ***
```

```
## `tenure_interval21-25 Month`            < 2e-16 ***
## `tenure_interval26-30 Month`            < 2e-16 ***
## `tenure_interval31-35 Month`            < 2e-16 ***
## `tenure_interval36-40 Month`           3.23e-11 ***
## `tenure_interval41-45 Month`           6.32e-13 ***
## `tenure_interval46-50 Month`           3.04e-10 ***
## `tenure_interval51-55 Month`           3.68e-10 ***
## `tenure_interval56-60 Month`           7.89e-11 ***
## `tenure_interval6-10 Month`            4.15e-14 ***
## `tenure_interval61-66 Month`           1.42e-10 ***
## `tenure_interval67-70 Month`           2.00e-06 ***
## `tenure_interval70+`                   1.05e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8143.4  on 7031  degrees of freedom
## Residual deviance: 5739.7  on 6995  degrees of freedom
## AIC: 5813.7
##
## Number of Fisher Scoring iterations: 6
```

```r
#predict
pred.glmModel <- as.vector(predict(model, newdata=testData,
                                   type="prob")[,"Yes"])
```

```r
#ROC Curve
colAUC(pred.glmModel,testData$Churn,plotROC = TRUE)
```

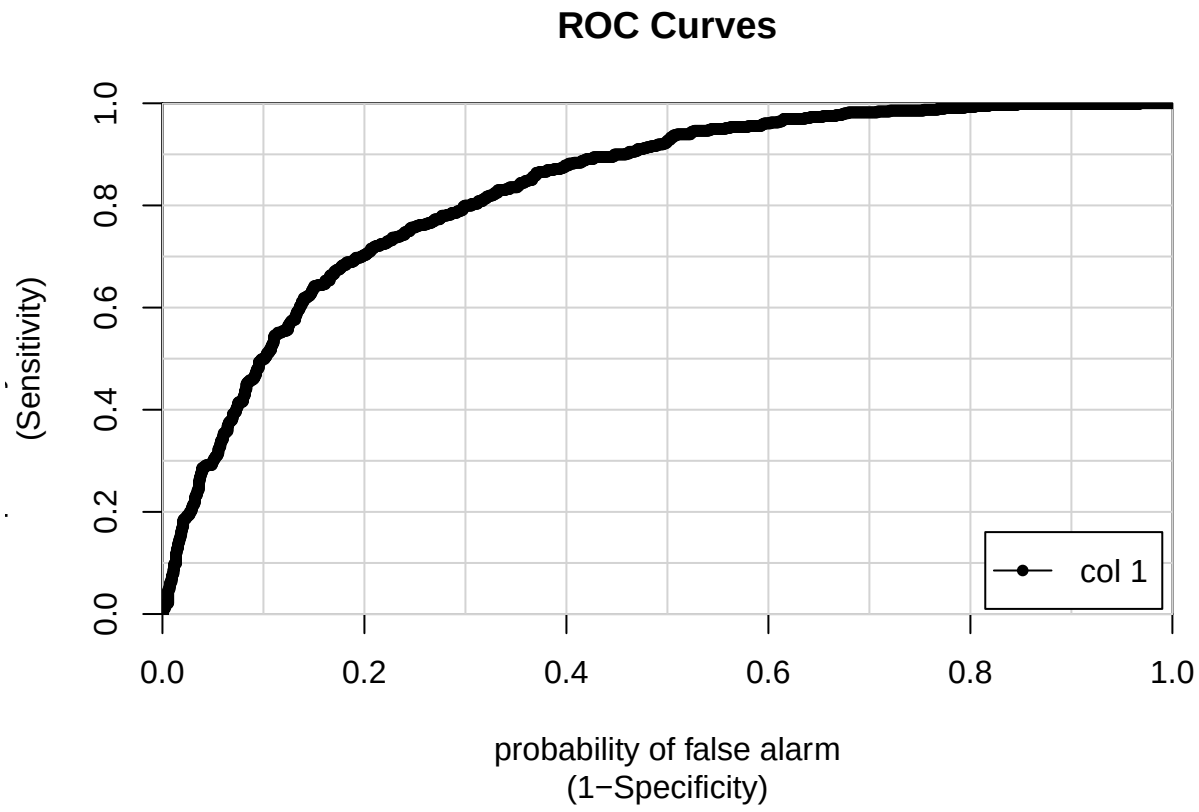


```
##                  [,1]
## No vs. Yes 0.8546947
```

```
###Model2
model2 <- train(Churn ~., data=telecomDataframe,method="glmnet",metric="ROC",
                tuneGrid=expand.grid(alpha=0:1,lambda=seq(0.10/10)),trControl=myControl)
```

```
pred.glmnetModel <- as.vector(predict(model2, newdata=testData,
                                      type="prob")[,"Yes"])
```

```
colAUC(pred.glmnetModel,testData$Churn,plotROC=TRUE)
```

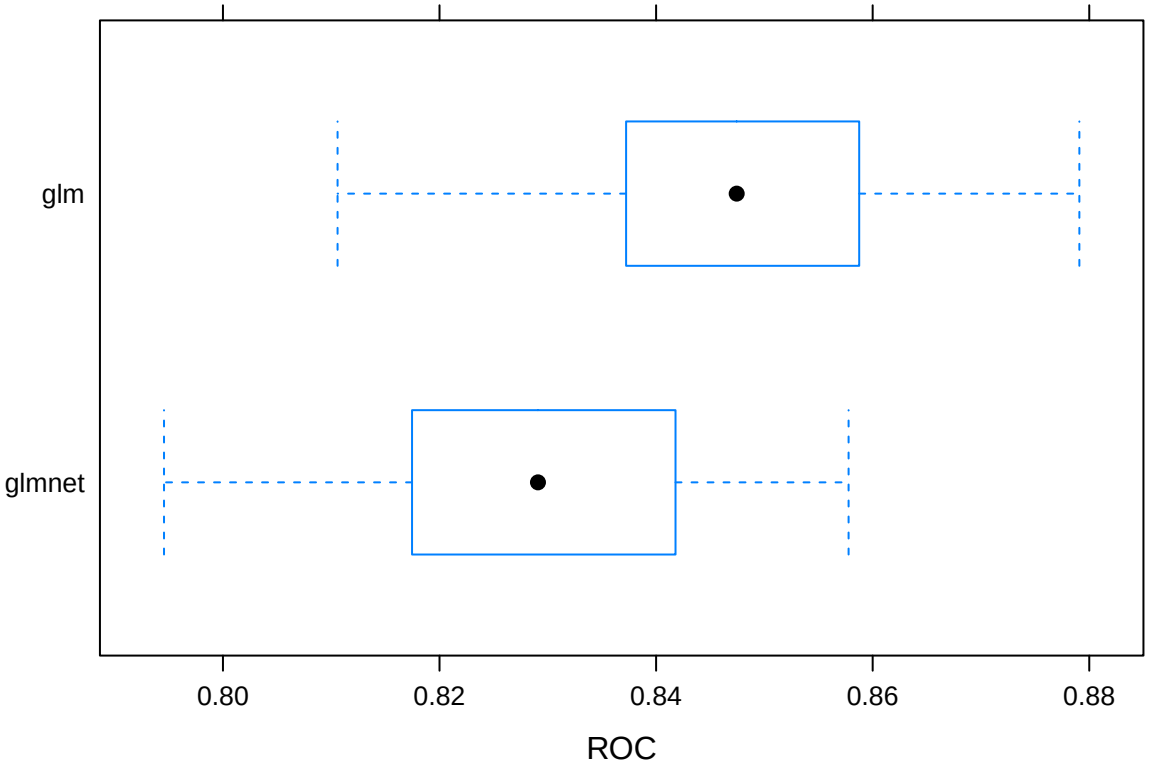



```
##                  [,1]
## No vs. Yes 0.8309741
```

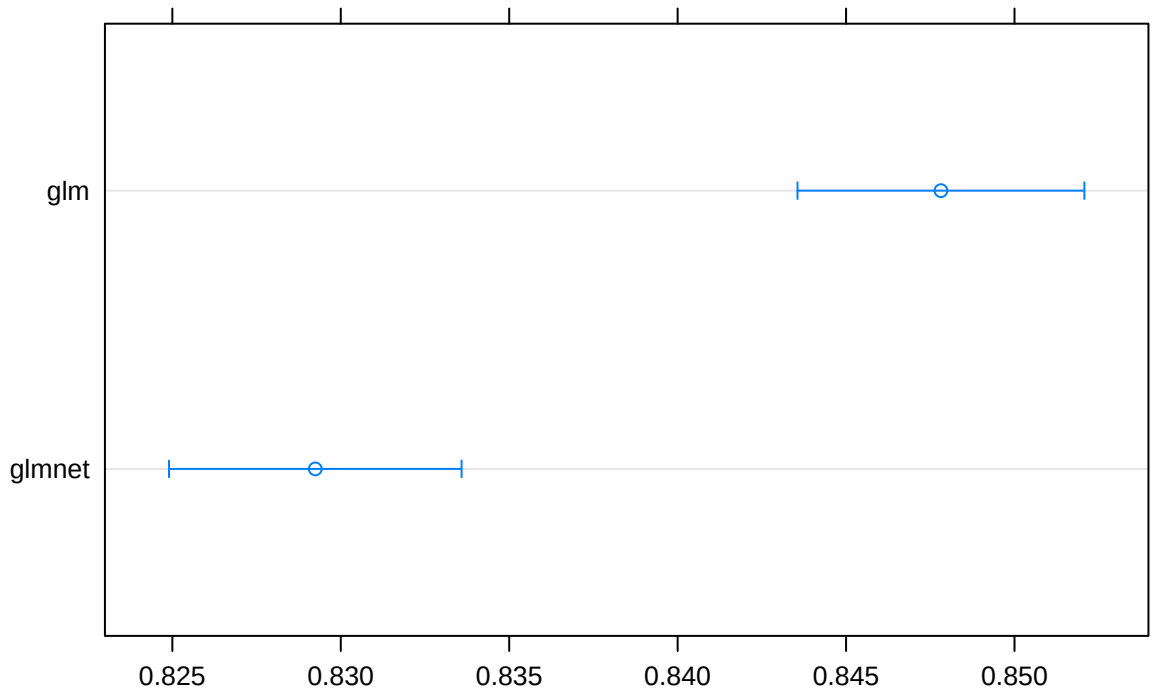
```
#Models comparision
models <- list(glm=model,glmnet=model2)
models <- resamples(models)

models
```

```
##
## Call:
## resamples.default(x = models)
##
## Models: glm, glmnet
## Number of resamples: 50
## Performance metrics: ROC, Sens, Spec
## Time estimates for: everything, final model fit
```

```
##BWplot
bwplot(models,metric = "ROC")
```

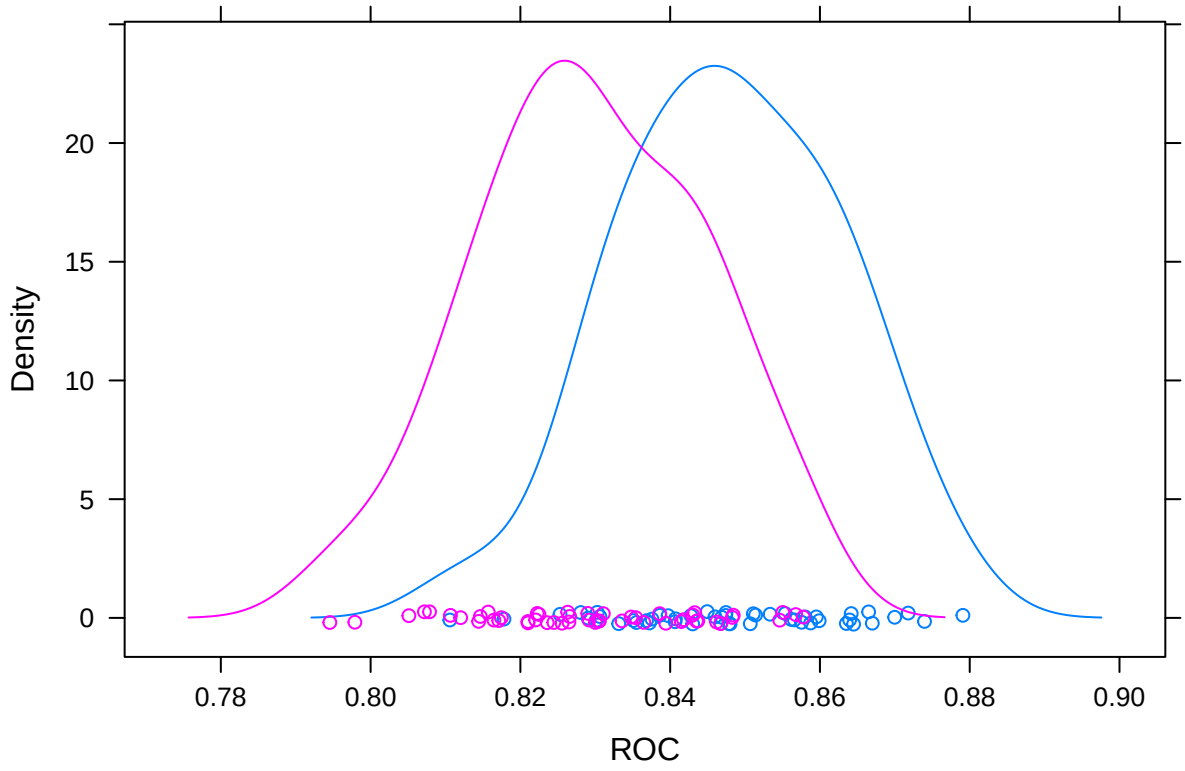



```
##dotplot
dotplot(models,metric = "ROC")
```

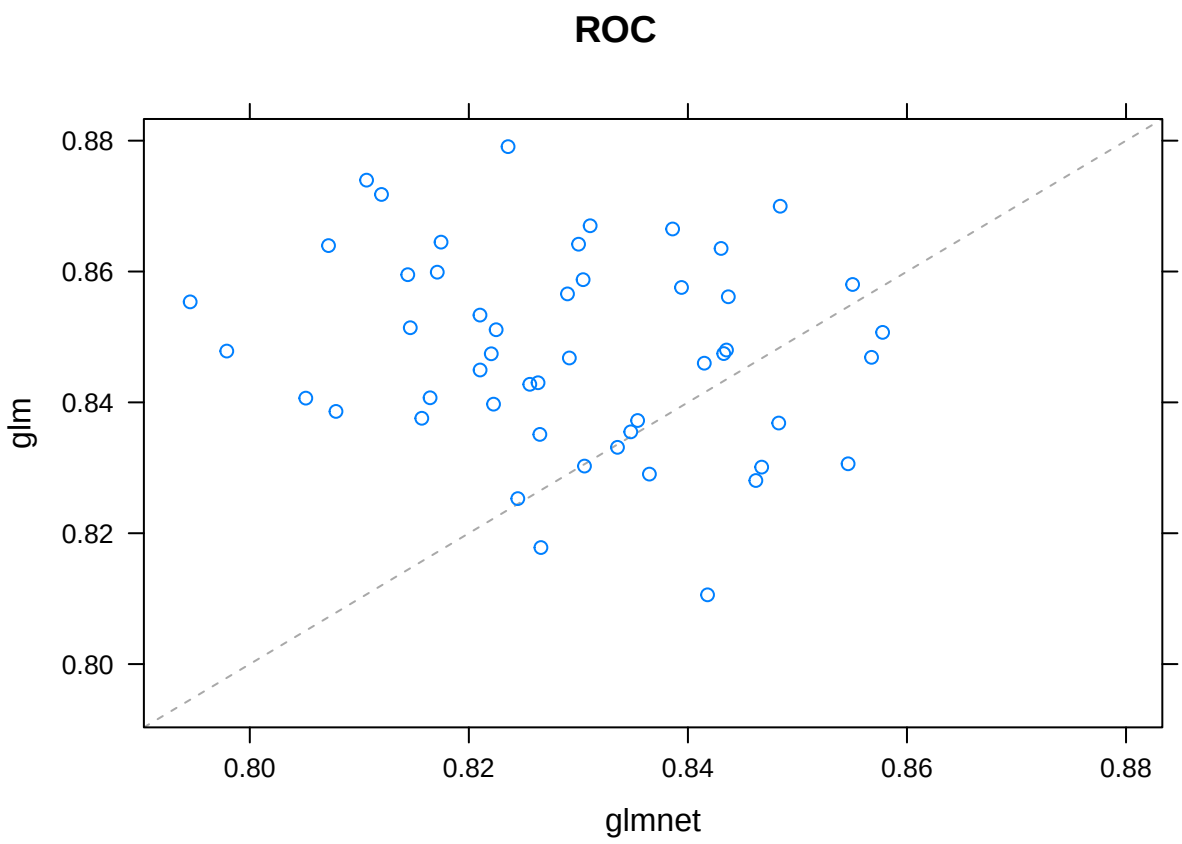


Confidence Level: 0.95

```
##density plot
densityplot(models,metric = "ROC")
```




```
#scatter plot
xyplot(models,metric = "ROC")
```




```
#Result
f.results <- ifelse(pred.glmModel > 0.5,1,0)
```

```
#Converting testData churn into character to convert replace them
testData$Churn <- as.character(testData$Churn)
testData$Churn[testData$Churn=="No"] <- "0"
testData$Churn[testData$Churn=="Yes"] <- "1"

#Misclassification error
misClasificationError <- mean(f.results!=testData$Churn)
print(misClasificationError)
```

```
## [1] 0.1890995
```

```
# calculating the accuracy rate
accuracyRate <- 1-misClasificationError
print(accuracyRate)
```

```
## [1] 0.8109005
```

```
#Confusion matrix
confusionMatrix(f.results,testData$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1408  256
```

```
##          1  143  303
##
##                Accuracy : 0.8109
##                  95% CI : (0.7935, 0.8274)
##     No Information Rate : 0.7351
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4809
##  Mcnemar's Test P-Value : 2.058e-08
##
##             Sensitivity : 0.9078
##             Specificity : 0.5420
##          Pos Pred Value : 0.8462
##          Neg Pred Value : 0.6794
##              Prevalence : 0.7351
##          Detection Rate : 0.6673
##    Detection Prevalence : 0.7886
##       Balanced Accuracy : 0.7249
##
##        'Positive' Class : 0
##
```

```r
# cbinding actual results with the predicted results
results <- cbind(f.results,testData$Churn)
colnames(results) <- c("predicted","actual")
results <- as.data.frame(results)
```