Logistic Regression

Ahmad Alqurashi

10/8/2021

Introduction

For week 6, We are going to use logistic regression model. Logistic regression is a predictive analysis that used to describe the effects of independent variables on binary dependent variable. For example, logistic regression can give a probability of getting diagnosed with caner (yes or no). For this assignment, we are going to predict whether a customer churn or not. Churn means that the customer does not continue business with the service provider.

Dataset

Churn dataset is a list of customers that contains information such as monthly payment, gender, contract type, etc. Most of the variables are binary and categorical, and only 3 variables are continuous. This dataset will used to predict whether customer opt out of the service or not. Chrun variable, the dependent variable, has two values, yes and No, meaning that no is the customer is continuing the service and yes is that the customer opt out of the service.

Data Cleaning

Churn dataset has 21 columns. Many of them are useless for this analysis or have aliased coefficients with other variables. Customer Id and total charge are not useful for this analysis because ID is meaningless and would mess up the result, total charge on the other hand, is useless because we have monthly charge and tenure variables which total charge can be calculated with these two variables. Also, Multiple Lines, Online Security, Online Backup, Device Protection, Tech Support, Streaming TV, and Streaming Movies variables has aliased coefficients with phone service variable, and they need to be removed in order to test the model with VIF function. lastly, all of the factor variables are stored as character, so we need to convert them into factors in order to fit them in the model.

Methods and results

- Importing required libraries

```r
# Importing required libraries
library(data.table)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from ’package:data.table’:
##
##     between, first, last

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

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

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: ’car’

## The following object is masked from ’package:dplyr’:
##
##     recode
```

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(caTools)
library(pROC)
```

```
## Type ’citation("pROC")’ for a citation.

##
## Attaching package: ’pROC’

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

```
library(MASS)
```

```
##
## Attaching package: ’MASS’

## The following object is masked from ’package:dplyr’:
##
##     select
```

- setting seed with a constant value in order to get same prediction every time we run the code

```
set.seed(1)
```

- Loading dataset into R environment, convert it to data table, and remove messing values.

```
dt <- read.csv('C:\\Users\\Ahmad\\Desktop\\MSDS\\MSDS660\\week 6\\assignment\\churn.csv', header = TRUE
dt <- as.data.table(dt)
str(dt)
```

```
## Classes ’data.table’ 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(*, ".internal.selfref")=<externalptr>
```

```
summary(dt)
```

```
##   customerID           gender           SeniorCitizen      Partner
##  Length:7043        Length:7043        Min.   :0.0000   Length:7043
##  Class :character   Class :character   1st Qu.:0.0000   Class :character
##  Mode  :character   Mode  :character   Median :0.0000   Mode  :character
##                                        Mean   :0.1621
##                                        3rd Qu.:0.0000
##                                        Max.   :1.0000
##
##   Dependents            tenure      PhoneService       MultipleLines
##  Length:7043        Min.   : 0.00   Length:7043        Length:7043
##  Class :character   1st Qu.: 9.00   Class :character   Class :character
##  Mode  :character   Median :29.00   Mode  :character   Mode  :character
##                     Mean   :32.37
##                     3rd Qu.:55.00
##                     Max.   :72.00
##
##  InternetService    OnlineSecurity     OnlineBackup       DeviceProtection
##  Length:7043        Length:7043        Length:7043        Length:7043
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  TechSupport        StreamingTV        StreamingMovies      Contract
##  Length:7043        Length:7043        Length:7043        Length:7043
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  PaperlessBilling   PaymentMethod      MonthlyCharges    TotalCharges
##  Length:7043        Length:7043        Min.   : 18.25   Min.   :  18.8
##  Class :character   Class :character   1st Qu.: 35.50   1st Qu.: 401.4
##  Mode  :character   Mode  :character   Median : 70.35   Median :1397.5
##                                        Mean   : 64.76   Mean   :2283.3
##                                        3rd Qu.: 89.85   3rd Qu.:3794.7
##                                        Max.   :118.75   Max.   :8684.8
##                                                         NA's   :11
##     Churn
##  Length:7043
##  Class :character
##  Mode  :character
##
##
##
##
```

```
dt <- dt[complete.cases(dt),]
summary(dt)
```

```
##   customerID           gender          SeniorCitizen      Partner
##  Length:7032        Length:7032        Min.   :0.0000   Length:7032
##  Class :character   Class :character   1st Qu.:0.0000   Class :character
##  Mode  :character   Mode  :character   Median :0.0000   Mode  :character
##                                        Mean   :0.1624
##                                        3rd Qu.:0.0000
##                                        Max.   :1.0000
##   Dependents            tenure     PhoneService       MultipleLines
##  Length:7032        Min.   : 1.00   Length:7032        Length:7032
##  Class :character   1st Qu.: 9.00   Class :character   Class :character
##  Mode  :character   Median :29.00   Mode  :character   Mode  :character
##                     Mean   :32.42
##                     3rd Qu.:55.00
##                     Max.   :72.00
##  InternetService    OnlineSecurity     OnlineBackup       DeviceProtection
##  Length:7032        Length:7032        Length:7032        Length:7032
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
```

```
##
##  TechSupport        StreamingTV        StreamingMovies      Contract
##  Length:7032        Length:7032        Length:7032        Length:7032
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  PaperlessBilling   PaymentMethod      MonthlyCharges    TotalCharges
##  Length:7032        Length:7032        Min.   : 18.25   Min.   :  18.8
##  Class :character   Class :character   1st Qu.: 35.59   1st Qu.: 401.4
##  Mode  :character   Mode  :character   Median : 70.35   Median :1397.5
##                                        Mean   : 64.80   Mean   :2283.3
##                                        3rd Qu.: 89.86   3rd Qu.:3794.7
##                                        Max.   :118.75   Max.   :8684.8
##     Churn
##  Length:7032
##  Class :character
##  Mode  :character
##
##
##
```

- Data cleaning: removing unwanted variables, and convert characters to factor

```
dtcln <- dt[, c("customerID", "TotalCharges", "MultipleLines", "OnlineSecurity","OnlineBackup", "Device
dtcln$gender <- as.factor(dtcln$gender)
dtcln$SeniorCitizen <- as.factor(dtcln$SeniorCitizen)
dtcln$Partner <- as.factor(dtcln$Partner)
dtcln$Dependents <- as.factor(dtcln$Dependents)
dtcln$PhoneService <- as.factor(dtcln$PhoneService)
dtcln$InternetService <- as.factor(dtcln$InternetService)
dtcln$Contract <- as.factor(dtcln$Contract)
dtcln$PaperlessBilling <- as.factor(dtcln$PaperlessBilling)
dtcln$PaymentMethod <- as.factor(dtcln$PaymentMethod)
dtcln$Churn <- as.factor(dtcln$Churn)
```

- Splitting data into two subsets: training and testing subsets, because we need to estimate the model performance, and prevent overfitting the model.

```
# Split the data into a train and test set
samp <- sample.split(dt$Churn, SplitRatio = 0.8)
train <- subset(dtcln, samp == TRUE)
test <- subset(dtcln, samp == FALSE)
```

- Create a model with train data.

```
# Create a multilinear binomial logistic regression on survived vs sex
fit <- glm(Churn ~ ., data = train, family = "binomial")
summary(fit)
```

```
##
```

```
## Call:
## glm(formula = Churn ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7668  -0.6860  -0.3032   0.7527   3.1901
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                          -0.617911   0.203095  -3.042 0.002346 **
## genderMale                           -0.028902   0.071860  -0.402 0.687533
## SeniorCitizen1                        0.213886   0.093782   2.281 0.022569 *
## PartnerYes                            0.111064   0.086618   1.282 0.199762
## DependentsYes                        -0.181602   0.098704  -1.840 0.065788 .
## tenure                               -0.035360   0.002537 -13.939  < 2e-16 ***
## PhoneServiceYes                      -0.818550   0.158200  -5.174 2.29e-07 ***
## InternetServiceFiber optic            0.826359   0.147119   5.617 1.94e-08 ***
## InternetServiceNo                    -0.183585   0.202832  -0.905 0.365408
## ContractOne year                     -0.837273   0.118340  -7.075 1.49e-12 ***
## ContractTwo year                     -1.542060   0.191350  -8.059 7.70e-16 ***
## PaperlessBillingYes                   0.357423   0.081936   4.362 1.29e-05 ***
## PaymentMethodCredit card (automatic) -0.123920   0.127112  -0.975 0.329613
## PaymentMethodElectronic check         0.355330   0.104299   3.407 0.000657 ***
## PaymentMethodMailed check            -0.038217   0.125025  -0.306 0.759851
## MonthlyCharges                        0.012058   0.003954   3.050 0.002291 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6513.9  on 5624  degrees of freedom
## Residual deviance: 4748.6  on 5609  degrees of freedom
## AIC: 4780.6
##
## Number of Fisher Scoring iterations: 6
```

```
vif(fit)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## gender           1.002560  1        1.001279
## SeniorCitizen    1.120884  1        1.058718
## Partner          1.393055  1        1.180278
## Dependents       1.292942  1        1.137076
## tenure           2.058215  1        1.434648
## PhoneService     1.898518  1        1.377867
## InternetService  8.617785  2        1.713361
## Contract         1.506527  2        1.107884
## PaperlessBilling 1.117082  1        1.056921
## PaymentMethod    1.334851  3        1.049314
## MonthlyCharges   8.674749  1        2.945293
```

It seems we have 8 variable that are significant. VIF score shows that Internet service and monthly charge have collinearity, both scores are 8.

- performing StepAIC with both directions to find the best model.

```
stepAIC(fit, dirrection = 'both')
```

```
## Start:  AIC=4780.59
## Churn ~ gender + SeniorCitizen + Partner + Dependents + tenure + 
##     PhoneService + InternetService + Contract + PaperlessBilling + 
##     PaymentMethod + MonthlyCharges
## 
##                    Df Deviance    AIC
## - gender            1   4748.8 4778.8
## - Partner           1   4750.2 4780.2
## <none>                  4748.6 4780.6
## - Dependents        1   4752.0 4782.0
## - SeniorCitizen     1   4753.8 4783.8
## - MonthlyCharges    1   4757.9 4787.9
## - PaperlessBilling  1   4767.7 4797.7
## - PaymentMethod     3   4777.7 4803.7
## - PhoneService      1   4775.2 4805.2
## - InternetService   2   4782.3 4810.3
## - Contract          2   4847.4 4875.4
## - tenure            1   4959.3 4989.3
## 
## Step:  AIC=4778.75
## Churn ~ SeniorCitizen + Partner + Dependents + tenure + PhoneService + 
##     InternetService + Contract + PaperlessBilling + PaymentMethod + 
##     MonthlyCharges
## 
##                    Df Deviance    AIC
## - Partner           1   4750.4 4778.4
## <none>                  4748.8 4778.8
## - Dependents        1   4752.1 4780.1
## - SeniorCitizen     1   4753.9 4781.9
## - MonthlyCharges    1   4758.1 4786.1
## - PaperlessBilling  1   4767.9 4795.9
## - PaymentMethod     3   4777.9 4801.9
## - PhoneService      1   4775.5 4803.5
## - InternetService   2   4782.5 4808.5
## - Contract          2   4847.5 4873.5
## - tenure            1   4959.9 4987.9
## 
## Step:  AIC=4778.4
## Churn ~ SeniorCitizen + Dependents + tenure + PhoneService + 
##     InternetService + Contract + PaperlessBilling + PaymentMethod + 
##     MonthlyCharges
## 
##                    Df Deviance    AIC
## <none>                  4750.4 4778.4
## - Dependents        1   4752.4 4778.4
## - SeniorCitizen     1   4756.3 4782.3
## - MonthlyCharges    1   4760.0 4786.0
## - PaperlessBilling  1   4769.5 4795.5
## - PaymentMethod     3   4780.0 4802.0
## - PhoneService      1   4777.4 4803.4
## - InternetService   2   4784.3 4808.3
## - Contract          2   4849.3 4873.3
## - tenure            1   4963.6 4989.6
```

```
##
## Call:  glm(formula = Churn ~ SeniorCitizen + Dependents + tenure + PhoneService +
##     InternetService + Contract + PaperlessBilling + PaymentMethod +
##     MonthlyCharges, family = "binomial", data = train)
##
## Coefficients:
##                          (Intercept)                        SeniorCitizen1
##                             -0.61946                               0.22670
##                        DependentsYes                                tenure
##                             -0.12701                              -0.03469
##                      PhoneServiceYes            InternetServiceFiber optic
##                             -0.82405                               0.82718
##                    InternetServiceNo                      ContractOne year
##                             -0.17776                              -0.83679
##                     ContractTwo year                   PaperlessBillingYes
##                             -1.54390                               0.35670
## PaymentMethodCredit card (automatic)         PaymentMethodElectronic check
##                             -0.13118                               0.35234
##            PaymentMethodMailed check                        MonthlyCharges
##                             -0.04762                               0.01220
##
## Degrees of Freedom: 5624 Total (i.e. Null);  5611 Residual
## Null Deviance:       6514
## Residual Deviance: 4750  AIC: 4778
```

StepAIC shows that the best model contains these variables: +SeniorCitizen +Dependents +tenure +PhoneService +InternetService +Contract +PaperlessBilling +PaymentMethod +MonthlyCharges

- create a model with those variables.

```r
#  AIC=4757.64

fit1 <- glm(Churn ~ SeniorCitizen + Dependents + tenure + PhoneService +
               InternetService + Contract + PaperlessBilling + PaymentMethod +
               MonthlyCharges, data = train, family = "binomial")

summary(fit1)
```

```
##
## Call:
## glm(formula = Churn ~ SeniorCitizen + Dependents + tenure + PhoneService +
##     InternetService + Contract + PaperlessBilling + PaymentMethod +
##     MonthlyCharges, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7880  -0.6860  -0.3013   0.7526   3.1717
##
## Coefficients:
```

```
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                          -0.619462   0.200031  -3.097 0.001956 **
## SeniorCitizen1                        0.226701   0.093231   2.432 0.015032 *
## DependentsYes                        -0.127012   0.089069  -1.426 0.153873
## tenure                               -0.034686   0.002474 -14.021  < 2e-16 ***
## PhoneServiceYes                      -0.824049   0.158064  -5.213 1.85e-07 ***
## InternetServiceFiber optic            0.827177   0.147087   5.624 1.87e-08 ***
## InternetServiceNo                    -0.177757   0.202715  -0.877 0.380552
## ContractOne year                     -0.836795   0.118340  -7.071 1.54e-12 ***
## ContractTwo year                     -1.543900   0.191370  -8.068 7.17e-16 ***
## PaperlessBillingYes                   0.356702   0.081912   4.355 1.33e-05 ***
## PaymentMethodCredit card (automatic) -0.131181   0.126941  -1.033 0.301417
## PaymentMethodElectronic check         0.352339   0.104220   3.381 0.000723 ***
## PaymentMethodMailed check            -0.047616   0.124801  -0.382 0.702808
## MonthlyCharges                        0.012200   0.003951   3.088 0.002015 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6513.9  on 5624  degrees of freedom
## Residual deviance: 4750.4  on 5611  degrees of freedom
## AIC: 4778.4
##
## Number of Fisher Scoring iterations: 6
```

```
vif(fit1)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## SeniorCitizen    1.107920  1        1.052578
## Dependents       1.053878  1        1.026585
## tenure           1.958778  1        1.399564
## PhoneService     1.898267  1        1.377776
## InternetService  8.609852  2        1.712967
## Contract         1.506746  2        1.107924
## PaperlessBilling 1.116912  1        1.056841
## PaymentMethod    1.329013  3        1.048548
## MonthlyCharges   8.667717  1        2.944099
```

- Below shows the results of prediction for train data.

```
# Predict on the train data
trainpreds <- predict(fit1, type = 'response', train)
# Round prediction values at 0.5 cutoff factor and change labels
trainp <- factor(trainpreds >= 0.5, labels = c('No', 'Yes'))
# Build a confusion matrix to see results
trainCM <- confusionMatrix(train$Churn, trainp)
trainCM
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  3701  429
##        Yes  696  799
##
##                Accuracy : 0.8
##                  95% CI : (0.7893, 0.8104)
##     No Information Rate : 0.7817
##     P-Value [Acc > NIR] : 0.0004192
##
##                   Kappa : 0.4566
##
##  Mcnemar's Test P-Value : 2.181e-15
##
##             Sensitivity : 0.8417
##             Specificity : 0.6507
##          Pos Pred Value : 0.8961
##          Neg Pred Value : 0.5344
##              Prevalence : 0.7817
##          Detection Rate : 0.6580
##    Detection Prevalence : 0.7342
##       Balanced Accuracy : 0.7462
##
##        'Positive' Class : No
##
```

Confusion matrix shows a prediction of 3701 false positive, meaning that 3701 customers who will not churn and actually did not churn and 429 false negative meaning that 429 customers who will not churn but actually churn. the prediction accuracy is 80%, sensitivity 84% meaning that 84% is above the curve.

- below is the prediction on testing data

```
# predict on the test data
testpreds <- predict(fit1, type = 'response', test)

# Round prediction values at 0.5 cutoff factor and change labels
testp <- factor(testpreds >= 0.5, labels = c('No', 'Yes'))

# Build a confusion matrix to see results
testCM <- confusionMatrix(test$Churn, testp)
testCM
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  930 103
##        Yes 163 211
##
##                Accuracy : 0.8109
##                  95% CI : (0.7895, 0.8311)
##     No Information Rate : 0.7768
##     P-Value [Acc > NIR] : 0.0009865
##
##                   Kappa : 0.4895
```

```
##
##  Mcnemar's Test P-Value : 0.0002974
##
##             Sensitivity : 0.8509
##             Specificity : 0.6720
##          Pos Pred Value : 0.9003
##          Neg Pred Value : 0.5642
##              Prevalence : 0.7768
##          Detection Rate : 0.6610
##    Detection Prevalence : 0.7342
##       Balanced Accuracy : 0.7614
##
##        'Positive' Class : No
##
```

prediction accuracy is 81%, sensitivity is 85%, which is better than the prediction on the training data.

- below shows ROC curve for train data

```
train_roc_curve <- roc(train$Churn, trainpreds)
```

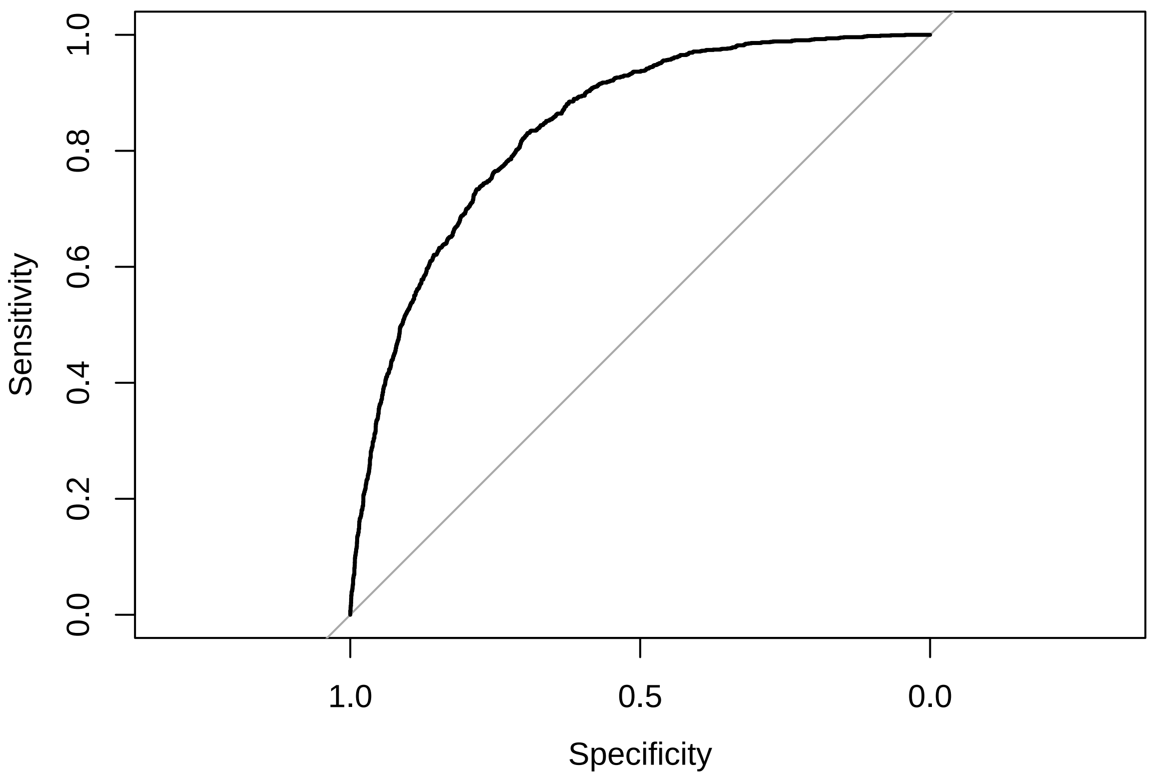
```
## Setting levels: control = No, case = Yes
```

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

```
train_roc_curve
```

```
##
## Call:
## roc.default(response = train$Churn, predictor = trainpreds)
##
## Data: trainpreds in 4130 controls (train$Churn No) < 1495 cases (train$Churn Yes).
## Area under the curve: 0.8403
```

```
plot(train_roc_curve)
```



```
train_rocc <- coords(roc=train_roc_curve, x = 'best', best.method = 'closest.topleft')
train_rocc
```

```
##   threshold specificity sensitivity
## 1  0.294515   0.7523002   0.7632107
```

- below shows ROC curve for test data

```
test_roc_curve <- roc(test$Churn, testpreds)
```

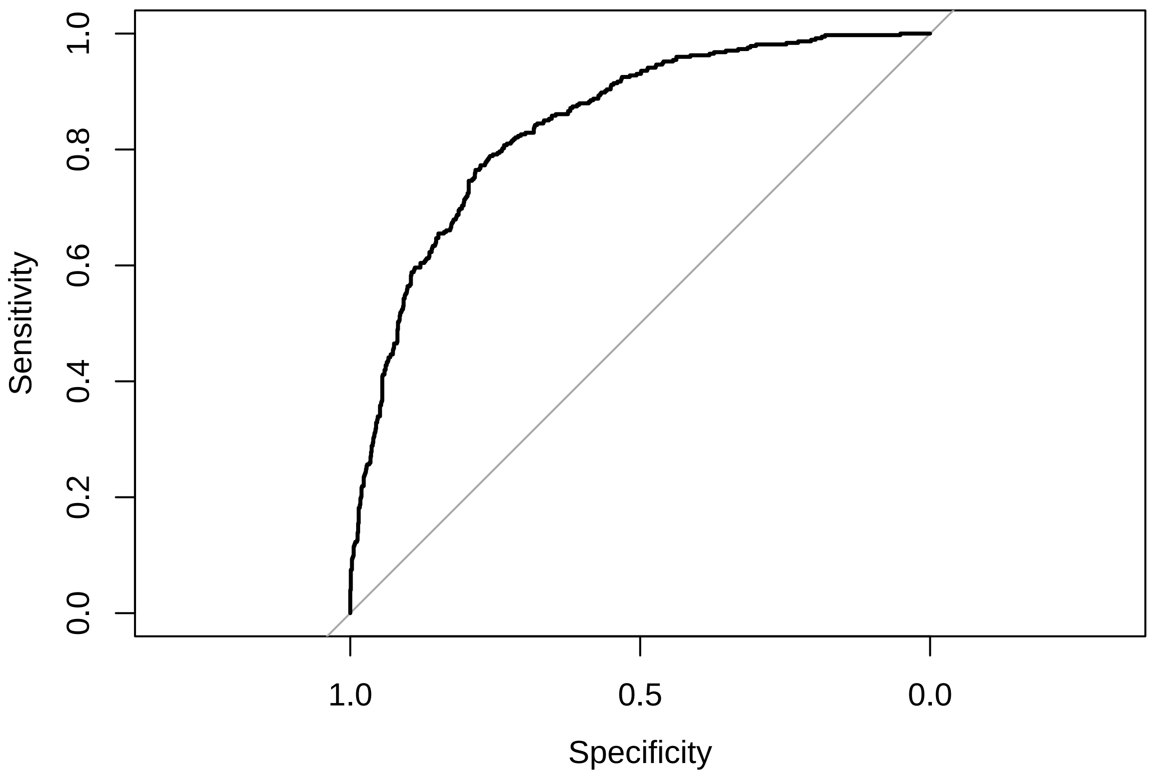
```
## Setting levels: control = No, case = Yes
```

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

```
test_roc_curve
```

```
##
## Call:
## roc.default(response = test$Churn, predictor = testpreds)
##
## Data: testpreds in 1033 controls (test$Churn No) < 374 cases (test$Churn Yes).
## Area under the curve: 0.8431
```

```
plot(test_roc_curve)
```



```
test_rocc <- coords(roc=test_roc_curve, x = 'best', best.method = 'closest.topleft')
test_rocc
```

```
##   threshold specificity sensitivity
## 1 0.3195466   0.7841239   0.7647059
```

Both curves looks similar in terms of sensitivity and specificity.

- below shows prediction on train data using ROC cut-off

```
# Predict on the train data using the ROC cut-off
# Round prediction values at 0.5 cutoff factor and change labels
trainrocp <- factor(trainpreds >= as.numeric(train_rocc[1]), labels = c('No', 'Yes'))

# Build a confusion matrix to see results
trainROCCM <- confusionMatrix(train$Churn, trainrocp)
trainROCCM
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   No  Yes
##        No  3107 1023
##        Yes  354 1141
##
##                Accuracy : 0.7552
##                  95% CI : (0.7437, 0.7664)
##     No Information Rate : 0.6153
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4511
##
##  Mcnemar’s Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8977
##             Specificity : 0.5273
##          Pos Pred Value : 0.7523
##          Neg Pred Value : 0.7632
##              Prevalence : 0.6153
##          Detection Rate : 0.5524
##    Detection Prevalence : 0.7342
##       Balanced Accuracy : 0.7125
##
##        ’Positive’ Class : No
##
```

Accuracy went down from 80% to 75%, while sensitivity rose to 89%.

- below shows prediction on test data using ROC cut-off

```
# Predict on the test data
# Round prediction values at 0.5 cutoff factor and change labels
testp <- factor(testpreds >= as.numeric(test_rocc[1]), labels = c('No', 'Yes'))

# Build a confusion matrix to see results
testROCCM <- confusionMatrix(test$Churn, testp)
testROCCM
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  810 223
##        Yes  88 286
##
##                Accuracy : 0.779
##                  95% CI : (0.7564, 0.8004)
##     No Information Rate : 0.6382
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4922
##
##  Mcnemar’s Test P-Value : 2.997e-14
##
```

```
##             Sensitivity : 0.9020
##             Specificity : 0.5619
##          Pos Pred Value : 0.7841
##          Neg Pred Value : 0.7647
##              Prevalence : 0.6382
##          Detection Rate : 0.5757
##    Detection Prevalence : 0.7342
##       Balanced Accuracy : 0.7319
##
##        ’Positive’ Class : No
##
```

It shows accuracy at 77%, sensitivity at 90%, and specificity at 56%.

- below shows confusion matrix of all predictions.

```
trainCM
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  3701  429
##        Yes  696  799
##
##                Accuracy : 0.8
##                  95% CI : (0.7893, 0.8104)
##     No Information Rate : 0.7817
##     P-Value [Acc > NIR] : 0.0004192
##
##                   Kappa : 0.4566
##
##  Mcnemar’s Test P-Value : 2.181e-15
##
##             Sensitivity : 0.8417
##             Specificity : 0.6507
##          Pos Pred Value : 0.8961
##          Neg Pred Value : 0.5344
##              Prevalence : 0.7817
##          Detection Rate : 0.6580
##    Detection Prevalence : 0.7342
##       Balanced Accuracy : 0.7462
##
##        ’Positive’ Class : No
##
```

```
trainROCCM
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  3107 1023
##        Yes  354 1141
##
##                Accuracy : 0.7552
##                  95% CI : (0.7437, 0.7664)
##     No Information Rate : 0.6153
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4511
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8977
##             Specificity : 0.5273
##          Pos Pred Value : 0.7523
##          Neg Pred Value : 0.7632
##              Prevalence : 0.6153
##          Detection Rate : 0.5524
##    Detection Prevalence : 0.7342
##       Balanced Accuracy : 0.7125
##
##        'Positive' Class : No
##
```

```
testCM
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  930 103
##        Yes 163 211
##
##                Accuracy : 0.8109
##                  95% CI : (0.7895, 0.8311)
##     No Information Rate : 0.7768
##     P-Value [Acc > NIR] : 0.0009865
##
##                   Kappa : 0.4895
##
##  Mcnemar's Test P-Value : 0.0002974
##
##             Sensitivity : 0.8509
##             Specificity : 0.6720
##          Pos Pred Value : 0.9003
##          Neg Pred Value : 0.5642
##              Prevalence : 0.7768
##          Detection Rate : 0.6610
##    Detection Prevalence : 0.7342
##       Balanced Accuracy : 0.7614
##
##        'Positive' Class : No
##
```

```
testROCCM
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  No Yes
##        No  810 223
##        Yes  88 286
##                                           
##                Accuracy : 0.779           
##                  95% CI : (0.7564, 0.8004)
##     No Information Rate : 0.6382          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.4922          
##                                           
##  Mcnemar’s Test P-Value : 2.997e-14       
##                                           
##             Sensitivity : 0.9020          
##             Specificity : 0.5619          
##          Pos Pred Value : 0.7841          
##          Neg Pred Value : 0.7647          
##              Prevalence : 0.6382          
##          Detection Rate : 0.5757          
##    Detection Prevalence : 0.7342          
##       Balanced Accuracy : 0.7319          
##                                           
##        ’Positive’ Class : No              
## 
```

Conclusion

In conclusion, all variables that are used in the model were good predictors of the dependent variable.