2251 Midterm

BQOM 2578 | Data Mining

Theresa Wohlever

Thursday, October 9, 2025

Table of contents

Libraries

```
rm(list = ls())
#Loading some neccesary libraries
#Add any missing ones as you need them!
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
```

```
v purrr     1.1.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

2. Exploration

get a sense of the data.

```r
df<-read.csv("driverschurn.csv")
str(df)
```

```
'data.frame':   1046 obs. of  7 variables:
 $ CHURN   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ AGE     : int  28 24 24 27 24 28 31 27 25 28 ...
 $ STATUS  : chr  "MARRIED" "MARRIED" "MARRIED" "MARRIED" ...
 $ GENDER  : chr  "Female" "Female" "Male" "Female" ...
 $ CHILDREN: chr  "Yes" "Yes" "Yes" "Yes" ...
 $ EXPSAL  : num  22000 20000 22000 18000 19000 16000 24000 17000 23000 20000 ...
 $ ACTSAL  : num  15986 21554 19090 21779 16432 ...
```

```r
summary(df)
```

```
     CHURN             AGE           STATUS             GENDER         
 Min.   :0.0000   Min.   :24.00   Length:1046        Length:1046       
 1st Qu.:0.0000   1st Qu.:27.00   Class :character   Class :character  
 Median :0.0000   Median :30.00   Mode  :character   Mode  :character  
 Mean   :0.2409   Mean   :30.68                                        
 3rd Qu.:0.0000   3rd Qu.:33.00                                        
 Max.   :1.0000   Max.   :46.00                                        
   CHILDREN             EXPSAL          ACTSAL     
 Length:1046        Min.   :15000   Min.   :11831  
 Class :character   1st Qu.:17000   1st Qu.:15752  
 Mode  :character   Median :20000   Median :17928  
                    Mean   :19902   Mean   :17967  
                    3rd Qu.:22000   3rd Qu.:20234  
                    Max.   :28000   Max.   :23000  
```

```r
unique(df$STATUS)
```

```
[1] "MARRIED" "SINGLE"
```

```r
table(df$STATUS)
```

```
MARRIED  SINGLE 
    510     536 
```

```r
unique(df$GENDER)
```

```
[1] "Female" "Male"
```

```r
table(df$GENDER)
```

```
Female   Male 
   486    560 
```

```r
unique(df$CHILDREN)
```

```
[1] "Yes" "No"
```

```r
table(df$CHILDREN)
```

```
 No Yes 
432 614 
```

Next, to prepare his dataset, split it between training and testing dataset:

```r
set.seed(1013, sample.kind = "Rejection")

#split the dataset leaving 80% of observations in the training dataset and 20% in the test dataset:
spl = sample(nrow(df),0.8*nrow(df))
head(spl)
```

```
[1]  990  610  654  518  802 1026
```

```
# Now lets split our dataset into train and test:

train.df = df[spl,]
test.df = df[-spl,]
```

3. Modeling

After that, John Knowsall decided to run a linear regression to test if expected and actual salary played a role. He decided a clever way to do so would be to start with actual salary, which surely would play a role. Then, he added expected salary. Finally, he added the demographic control variables.

```
m1<-lm(CHURN~ACTSAL,data=train.df)
summary(m1)
```

```
Call:
lm(formula = CHURN ~ ACTSAL, data = train.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.3950 -0.2783 -0.1966 -0.1098  0.8861 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  7.280e-01  9.666e-02   7.531 1.31e-13 ***
ACTSAL      -2.701e-05  5.320e-06  -5.078 4.71e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4228 on 834 degrees of freedom
Multiple R-squared:  0.02999,	Adjusted R-squared:  0.02883 
F-statistic: 25.78 on 1 and 834 DF,  p-value: 4.713e-07
```

```
m2<-lm(CHURN~ACTSAL+EXPSAL,data=train.df)
summary(m2)
```

```
Call:
lm(formula = CHURN ~ ACTSAL + EXPSAL, data = train.df)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.60029 -0.27396 -0.15880  0.01094  1.02429

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.579e-02  1.358e-01  -0.263    0.792
ACTSAL      -2.157e-05  5.189e-06  -4.157 3.56e-05 ***
EXPSAL       3.341e-05  4.313e-06   7.745 2.77e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4086 on 833 degrees of freedom
Multiple R-squared:  0.09515,   Adjusted R-squared:  0.09298
F-statistic:  43.8 on 2 and 833 DF,  p-value: < 2.2e-16
```

```r
m3<-lm(CHURN~.,data=train.df)
summary(m3)
```

```
Call:
lm(formula = CHURN ~ ., data = train.df)

Residuals:
    Min      1Q  Median      3Q     Max
-0.7908 -0.2474 -0.1085  0.1217  1.0881

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -5.410e-01  1.516e-01  -3.569  0.00038 ***
AGE           1.712e-02  2.539e-03   6.742 2.92e-11 ***
STATUSSINGLE  1.396e-01  2.685e-02   5.198 2.53e-07 ***
GENDERMale    1.125e-02  2.670e-02   0.421  0.67359
CHILDRENYes  -1.346e-01  2.730e-02  -4.931 9.86e-07 ***
EXPSAL        2.661e-05  4.105e-06   6.481 1.56e-10 ***
ACTSAL       -1.504e-05  4.921e-06  -3.056  0.00232 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.384 on 829 degrees of freedom
Multiple R-squared:  0.2048,    Adjusted R-squared:  0.1991
F-statistic: 35.59 on 6 and 829 DF,  p-value: < 2.2e-16
```

4. Presenting

Although John liked his model's results, he knew that this is not the proper way to present them. So, he made a nice looking table:

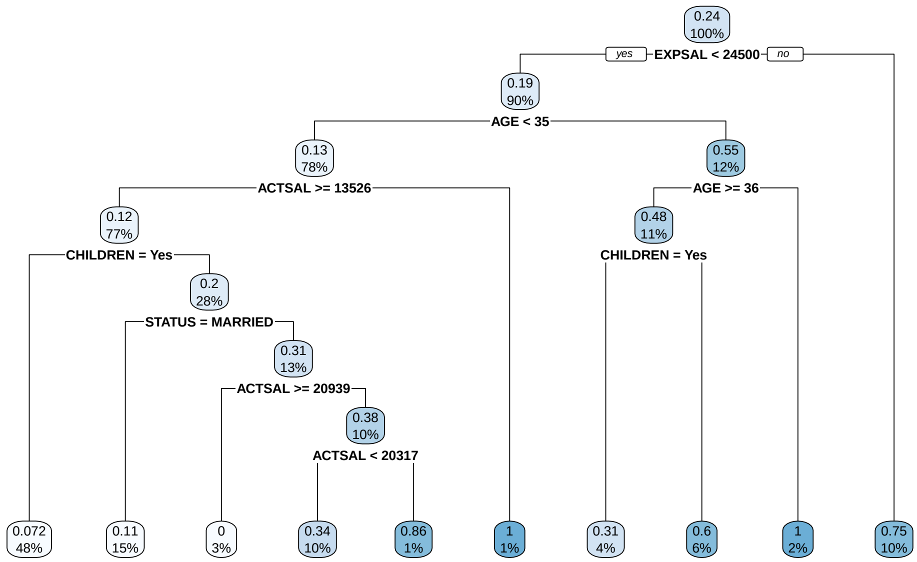
```r
library(jtools)
library(sjPlot)



# Generate the table using tab_model()
#transform
model_table <- tab_model(m1, m2, m3,dv.labels = c("(1):", "(2)","(3)"),show.ci=FALSE,collapse.se=TRUE, p.threshold = c(0.1, 0.05, 0.01
  string.se = "std. Error",
  string.p = "p-value",transform=NULL,digits=6)



# Print the table
model_table
```

Predictors	(1): Estimates	(2) Estimates	(3) Estimates
(Intercept)	0.728009 *** (0.096664)	-0.035791 (0.135835)	-0.541023 *** (0.151607)
ACTSAL	-0.000027 *** (0.000005)	-0.000022 *** (0.000005)	-0.000015 *** (0.000005)
EXPSAL		0.000033 *** (0.000004)	0.000027 *** (0.000004)
AGE			0.017117 *** (0.002539)
STATUS [SINGLE]			0.139560 *** (0.026847)
GENDER [Male]			0.011251 (0.026700)
CHILDREN [Yes]			-0.134619 *** (0.027298)
Observations	836	836	836
R^2 / R^2 adjusted	0.030 / 0.029	0.095 / 0.093	0.205 / 0.199

* p<0.1 ** p<0.05 *** p<0.01

5. 2nd modeling attempt

```
library(rpart)
library(rpart.plot)
tree<-rpart(CHURN ~ ., data=train.df, method="anova")
rpart.plot(tree,digits=-2)
```



CANVAS EXAM

Question 5: Classification Tree

```
library(rpart)
library(rpart.plot)
library(tidyverse)
library(caTools)
library(ROCR)
library(caret)
```

```
Loading required package: lattice
```

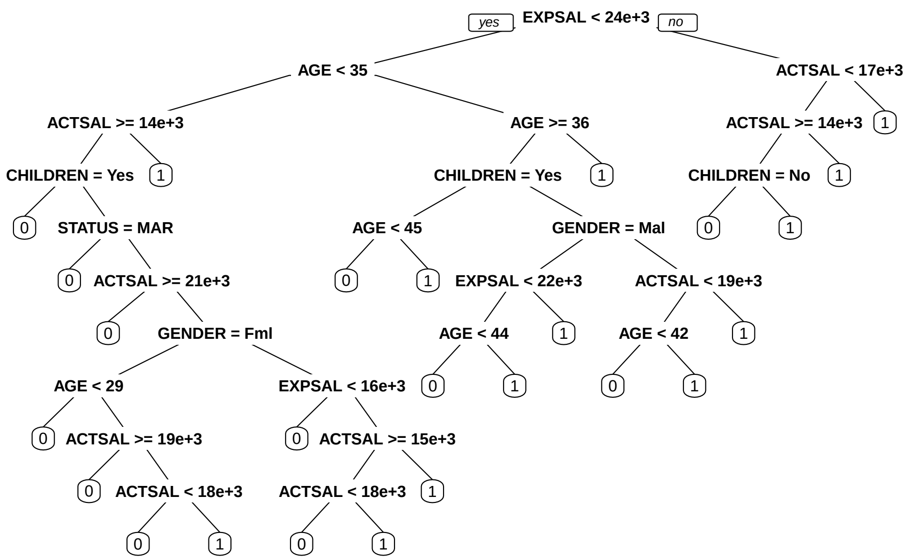
```
Attaching package: 'caret'
```

```
The following object is masked from 'package:purrr':

    lift
```

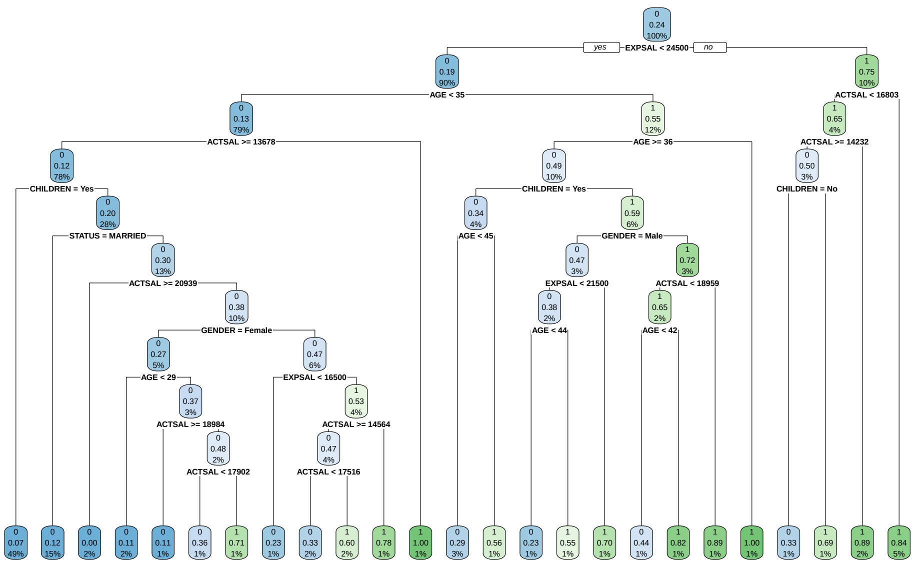
```r
library(corrplot)
```

```
corrplot 0.95 loaded
```

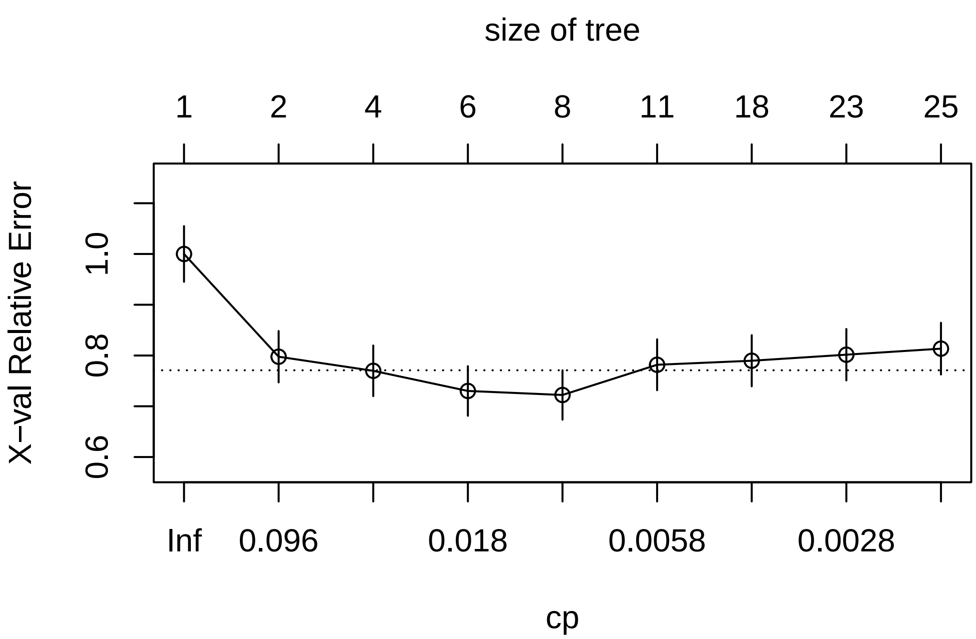
```r
library("lm.beta")



tree<-rpart(CHURN ~ ., data=df, method="class",cp=0.0005)
prp(tree)
```



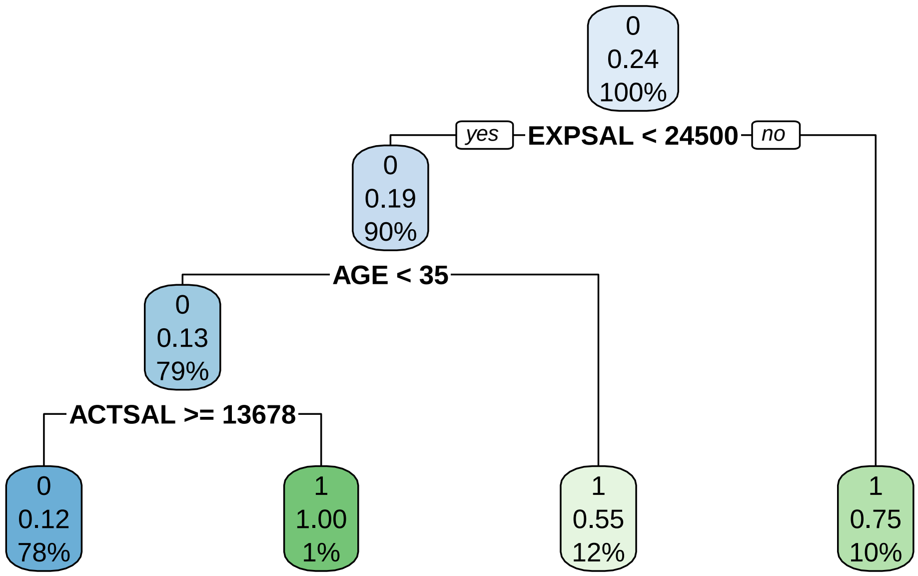
```r
rpart.plot(tree,digits=-2)
```



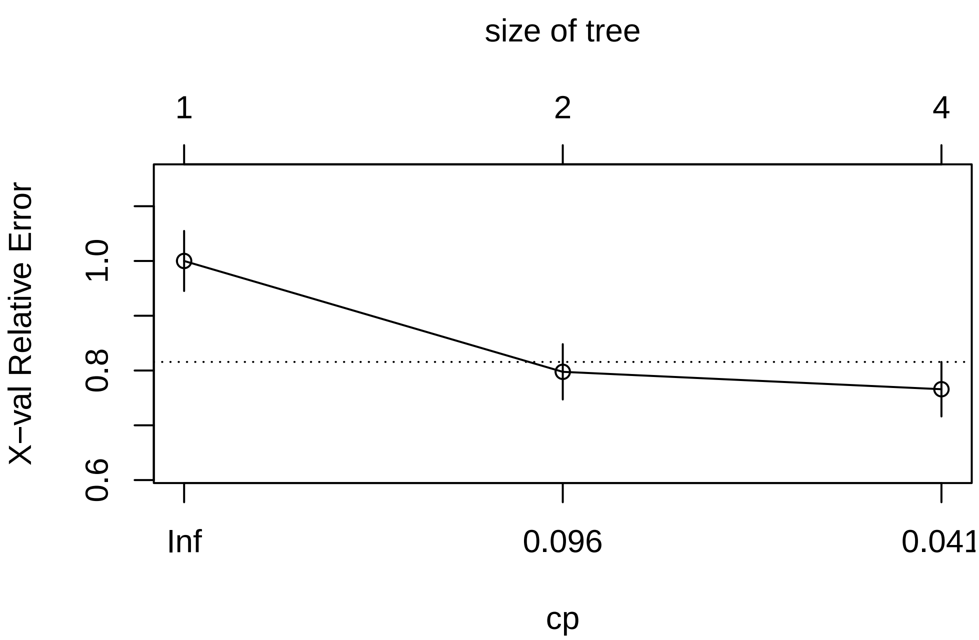

```r
plotcp(tree)
```




```r
tree_cp036 = rpart(CHURN ~ .,df, method="class",cp= 0.036 )
rpart.plot(tree_cp036 ,digits=-2)
```




```
plotcp(tree_cp036)
```



Question 6: Confusion Matrix

```r
df_pred <- df
df_pred$pred = predict(tree_cp036, newdata = df, type="class")
confusionMatrix(df_pred$pred,as.factor(df_pred$CHURN), positive="1")
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 714  98
         1  80 154

               Accuracy : 0.8298
                 95% CI : (0.8057, 0.8521)
    No Information Rate : 0.7591
    P-Value [Acc > NIR] : 1.741e-08

                  Kappa : 0.5231

 Mcnemar's Test P-Value : 0.2026

            Sensitivity : 0.6111
            Specificity : 0.8992
         Pos Pred Value : 0.6581
         Neg Pred Value : 0.8793
             Prevalence : 0.2409
         Detection Rate : 0.1472
   Detection Prevalence : 0.2237
      Balanced Accuracy : 0.7552

       'Positive' Class : 1
```

Question 7: Logistic Regression

```r
logreg <- glm(CHURN ~ ., data=df, family="binomial")
summary(logreg)
```

```
Call:
```

```
glm(formula = CHURN ~ ., family = "binomial", data = df)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)  -5.806e+00  9.259e-01  -6.271 3.59e-10 ***
AGE           9.422e-02  1.468e-02   6.420 1.36e-10 ***
STATUSSINGLE  8.698e-01  1.691e-01   5.145 2.68e-07 ***
GENDERMale    5.215e-02  1.646e-01   0.317 0.751408
CHILDRENYes  -9.095e-01  1.642e-01  -5.539 3.05e-08 ***
EXPSAL        1.771e-04  2.517e-05   7.038 1.95e-12 ***
ACTSAL       -1.111e-04  3.029e-05  -3.668 0.000244 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1155.07  on 1045  degrees of freedom
Residual deviance:  931.76  on 1039  degrees of freedom
AIC: 945.76

Number of Fisher Scoring iterations: 5
```

```r
df_logreg <- df
df_logreg$pred <- predict(logreg, newdata = df, type="response")
summary(df_logreg$pred)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01405 0.08579 0.17390 0.24092 0.33420 0.91024
```

```r
# Calculate the exp(coefficient)
coeftable <- data.frame(col1=coef(logreg),col2=exp(coef(logreg)))
colnames(coeftable)<-c('Coefficient (log-odds)','e^coefficient (odds)')
coeftable
```

```
             Coefficient (log-odds) e^coefficient (odds)
(Intercept)           -5.8060837666          0.003009192
AGE                    0.0942152192          1.098796203
STATUSSINGLE           0.8698395647          2.386527939
GENDERMale             0.0521532636          1.053537199
CHILDRENYes           -0.9094738754          0.402736058
EXPSAL                 0.0001771061          1.000177122
ACTSAL                -0.0001110970          0.999888909
```

Question 8: Logistic Regression Cutoff 0.5

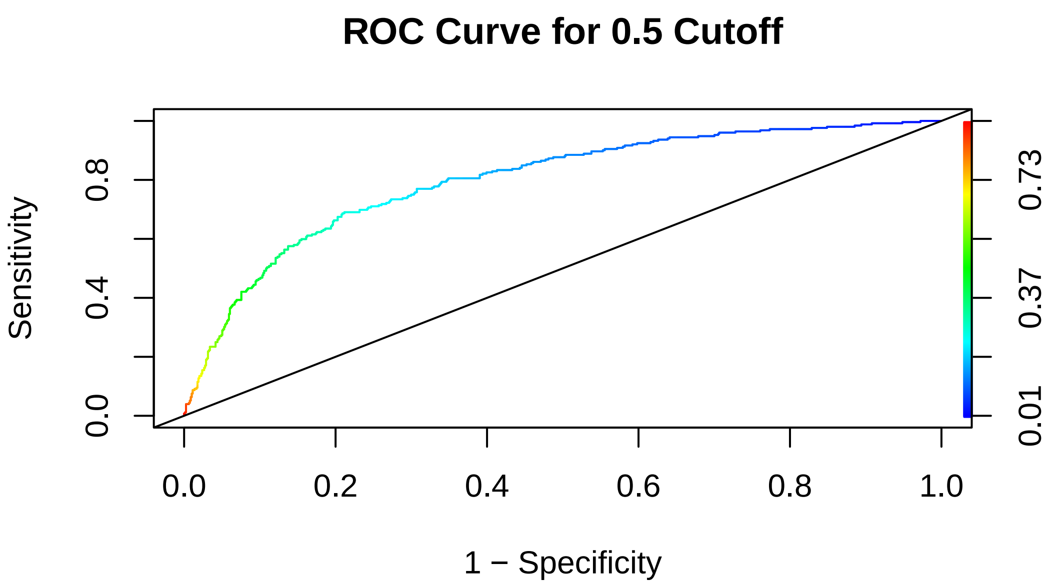
```r
cutoff <- 0.50
levels <- c("0", "1")
## Using Cut Off value for prediction

# Convert predicted probabilities to predicted classes using a threshold (e.g., 0.5)
df_logreg$pred_class <- factor(ifelse(df_logreg$pred > cutoff, "1", "0"), levels = levels )

# Ensure the reference vector is a factor with the same levels
df_logreg$CHURN <- factor(df_logreg$CHURN, levels = levels )

# create the confusion matrix
conmat <- caret::confusionMatrix(df_logreg$pred_class, df_logreg$CHURN, positive = "1")
conmat
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 747 167
         1  47  85

               Accuracy : 0.7954
                 95% CI : (0.7697, 0.8195)
    No Information Rate : 0.7591
    P-Value [Acc > NIR] : 0.002939

                  Kappa : 0.3321

 Mcnemar's Test P-Value : 4.131e-16

            Sensitivity : 0.33730
            Specificity : 0.94081
         Pos Pred Value : 0.64394
         Neg Pred Value : 0.81729
             Prevalence : 0.24092
         Detection Rate : 0.08126
   Detection Prevalence : 0.12620
      Balanced Accuracy : 0.63905
```

```
'Positive' Class : 1
```

Question 9: Logistic Regression Cutoff Update

```
roc.pred = prediction(df_logreg$pred, df_logreg$CHURN)
perf = performance(roc.pred, "tpr", "fpr")

plot(perf,                      # the data
     main = "ROC Curve for 0.5 Cutoff",        # the chart's title
     xlab = "1 - Specificity",  # the name of the x-axis
     ylab = "Sensitivity",      # the name of the y-axis
     colorize=TRUE)             # add color to curve depending on threshold prob.

abline(0,1) # adds line at intercept 0, with slope 1
```



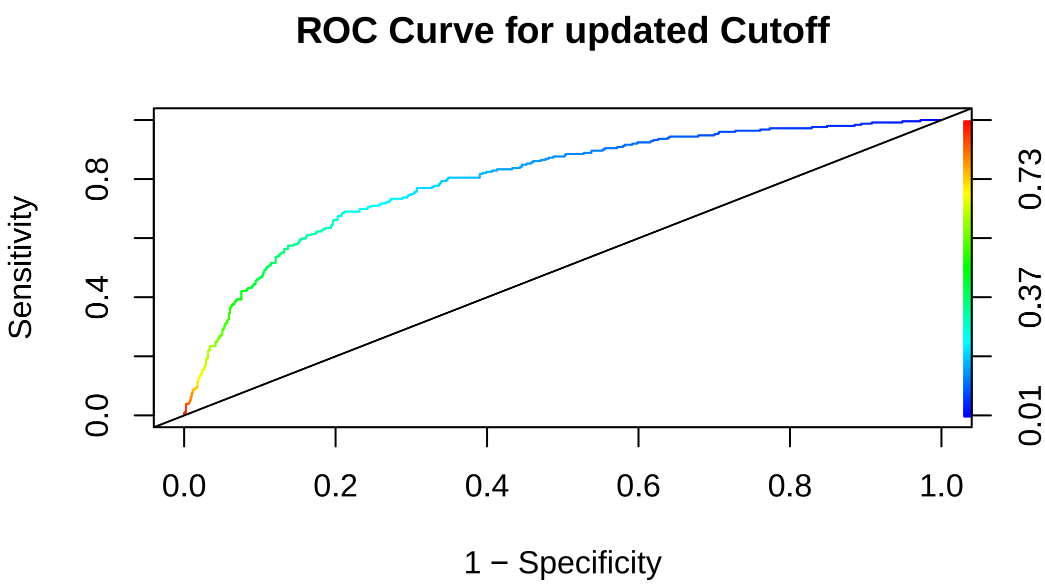

```
perf_auc = performance(roc.pred, "auc")
as.numeric(perf_auc@y.values)
```

```
[1] 0.7980189
```

```r
##
## CONFUSION MATRIX
##
cutoff <- 0.35
levels <- c("0", "1")
## Using Cut Off value for prediction

# Convert predicted probabilities to predicted classes using a threshold (e.g., 0.5)
df_logreg$pred_class <- factor(ifelse(df_logreg$pred > cutoff, "1", "0"), levels = levels )
# Ensure the reference vector is a factor with the same levels
df_logreg$CHURN <- factor(df_logreg$CHURN, levels = levels )
# create the confusion matrix
conmat <- caret::confusionMatrix(df_logreg$pred_class, df_logreg$CHURN, positive = "1")
conmat
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 686 110
         1 108 142

               Accuracy : 0.7916
                 95% CI : (0.7657, 0.8158)
    No Information Rate : 0.7591
    P-Value [Acc > NIR] : 0.007045

                  Kappa : 0.4286

 Mcnemar's Test P-Value : 0.946002

            Sensitivity : 0.5635
            Specificity : 0.8640
         Pos Pred Value : 0.5680
         Neg Pred Value : 0.8618
             Prevalence : 0.2409
         Detection Rate : 0.1358
   Detection Prevalence : 0.2390
      Balanced Accuracy : 0.7137

       'Positive' Class : 1
```

```r
roc.pred = prediction(df_logreg$pred, df_logreg$CHURN)
perf = performance(roc.pred, "tpr", "fpr")

plot(perf,                          # the data
     main = "ROC Curve for updated Cutoff",        # the chart's title
     xlab = "1 - Specificity",  # the name of the x-axis
     ylab = "Sensitivity",      # the name of the y-axis
     colorize=TRUE)             # add color to curve depending on threshold prob.

abline(0,1) # adds line at intercept 0, with slope 1
```




```r
perf_auc = performance(roc.pred, "auc")
as.numeric(perf_auc@y.values)
```

```
[1] 0.7980189
```