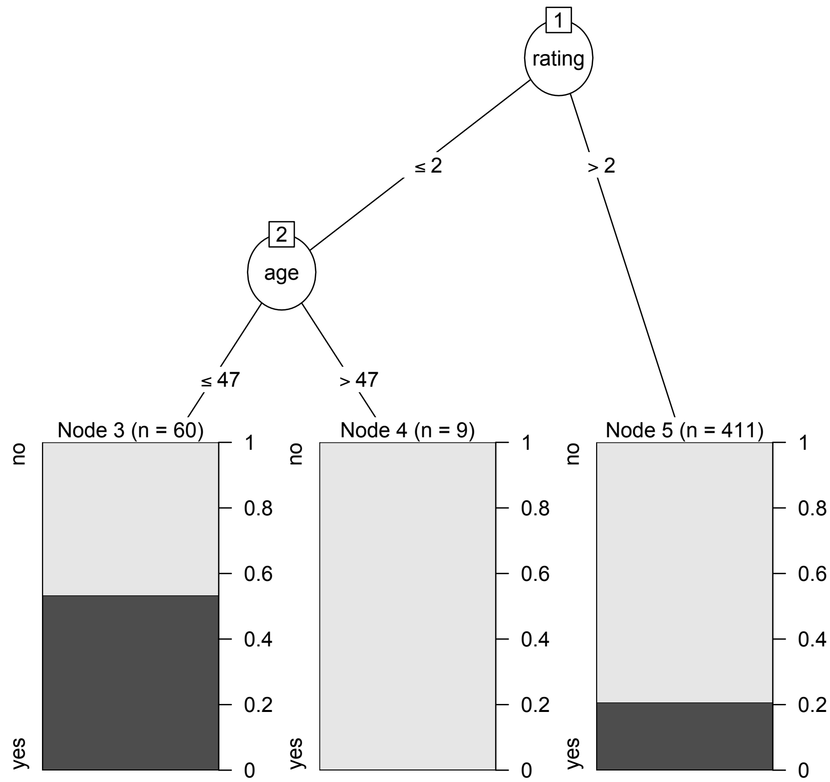
MIS 545 - Kirsten Fure, 8/20/2019

1. Decision Tree Fitting:

dt <- C5.0(x = train[, var_names], y = train$if_affair)



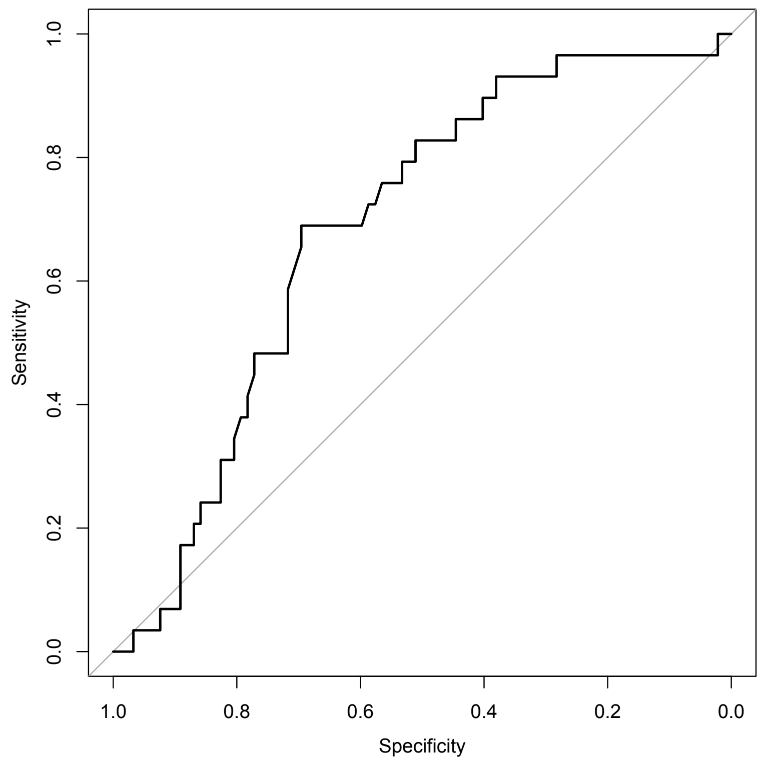
Regression Model Fitting:

reg <- glm(if_affair ~ . , data = train2, family = binomial())

```
Call:
glm(formula = if_affair ~ ., family = binomial(), data = train2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6726  -0.7540  -0.5560   0.7719   2.4087

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.79728    0.67329   2.669 0.007599 **
age           -0.03788    0.01936  -1.956 0.050449 .
yearsmarried   0.11421    0.03298   3.463 0.000534 ***
religiousness -0.24954    0.10147  -2.459 0.013924 *
rating        -0.49466    0.09822  -5.036 4.75e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 542.03  on 479  degrees of freedom
Residual deviance: 488.35  on 475  degrees of freedom
AIC: 498.35
```

2. a. The most useful attribute for the decision tree was “rating” at 100%.
   b. For the Decision Tree, the Precision was 75%. And the Recall was only 27%.

3. a. For the regression, the Baseline of a positive result for an affair was 25%.
   b. The area under the ROC curve was 69%.



4. Our model was good at predicting no affair, but not very good at predicting yes for an affair. The Baseline for affairs was 25% and the Recall was similar at 27%, due to the fact that the model did not predict very many affairs overall. The False Negative rate was high (73%) and the True Positive Rate was low (27%), showing that our model was having difficulty predicting affairs. The Precision Rate was decently high at 75% because there was a low number of predicted affairs overall. Often the F-score can give a better result balancing the Recall and Precision. In this particular case, I would think the main goal would be to have a very low false positive rate. We would not want to predict an affair if there wasn’t one and cause unnecessary problems in the marriage. So, in this case, using the precision rate as a metric of performance at 75% would be adequate, making sure to keep the false positive rate low.

5. R Code:

```
###     decision tree
if (!require(C50)) {
      install.packages("C50")
      }
###     ROC curve\n
  if (!require(pROC)) {
      install.packages("pROC")
      }
library(C50)
library(pROC)

setwd("/Users/kirsten 1/Documents/Masters Programs/MIS 545 Data Mining - UofA/Lab
 6 Model Evalution")
getwd()
affair <- read.csv("affairs.csv")
str(affair)
head(affair)
nrow(affair)
nrow(!complete.cases(affair))
###     partition dataset for training (80%) and testing (20%)
size <- floor(0.8 * nrow(affair))
size
###     randomly decide which ones for training
training_index <- sample(nrow(affair), size = size, replace = FALSE)
train <- affair[training_index,]
test <- affair[-training_index,]
###     names of variables that used for prediction
var_names <- names(affair)[-5]    #this gives column titles of first 4 variables (so
 everything except #5)
var_names
# fit the model
dt <- C5.0(x = train[, var_names], y = train$if_affair)
# see the summary of model
summary(dt)
plot(dt)
###     now, validate test
##      predict() method returns a vector of result
dt_pred <- predict(dt, newdata = test)
###     merger dt_prediction value to test dataset
dt_evaluation <- cbind(test, dt_pred)
```

```
head(test)
head(dt_evaluation)
### compare dt_prediction result to actual value
dt_evaluation$correct <- ifelse(dt_evaluation$if_affair == dt_evaluation$dt_pred, 1, 0)
head(dt_evaluation)
###      accuracy rate
sum(dt_evaluation$correct) / nrow(dt_evaluation)
###     confusion matrix
table(dt_evaluation$if_affair, dt_evaluation$dt_pred)
###     True Positive Rate (Sensitivity)    TPR = TP / P
###     = count of true positive dt_prediction divided by total positive truth
TPR <- sum(dt_evaluation$dt_pred == 'yes' & dt_evaluation$if_affair == 'yes') /
 sum(dt_evaluation$if_affair == 'yes')
TPR
###     True Negative Rate (Specificity)    TNR = TN / N
###     = count of true negative dt_prediction divided by total negative truth
TNR <- sum(dt_evaluation$dt_pred == 'no' & dt_evaluation$if_affair == 'no') /
 sum(dt_evaluation$if_affair == 'no')
TNR
###     False Positive Rate    (1 - Specificity)    FPR = FP / N
###     = count of false positive dt_prediction divided by total negative truth
###     = sum(dt_evaluation$dt_pred == 'yes'& dt_evaluation$if_affair == 'no') /
 sum(dt_evaluation$if_affair_50K == 'no')
FPR <- 1 - TNR
FPR
###     False Negative Rate FNR (1 - Sensitivity)    FNR = FN / P
###     = count of false negative dt_prediction divided by total positive truth
###     = sum(dt_evaluation$dt_pred == 'no'& dt_evaluation$if_affair == 'yes') /
 sum(dt_evaluation$if_affair == 'yes')
FNR <- 1 - TPR
FNR
###     dt_precision equals
###     = number of true positive dt_prediction    / total positive dt_prediction
dt_precision <- sum(dt_evaluation$if_affair == 'yes' & dt_evaluation$dt_pred == 'yes') /
 sum(dt_evaluation$dt_pred == 'yes')
dt_precision
###     dt_recall equals = TPR
###     = true positive dt_prediction / total true positive
dt_recall <- sum(dt_evaluation$if_affair == 'yes' & dt_evaluation$dt_pred == 'yes') /
 sum(dt_evaluation$if_affair == 'yes')
dt_recall
```

```
 ###   F score
F <- 2 * dt_precision * dt_recall / (dt_precision + dt_recall)
F
################################################################################
 #
###            ROC Curve: Receiver Operating Characteristic Curve                ###
################################################################################
 #
###     randomly decide which ones for training
training_index2 <- sample(nrow(affair), size = size, replace = FALSE)
train2 <- affair[training_index2,]
test2 <- affair[-training_index2,]
#######################################
###             fitting model            ###
#######################################
###     fitting regression model
reg <- glm(if_affair ~ . , data = train2, family = binomial() )
###    model detail
summary(reg)
###     validate test dataset
evaluation <- test2
###     return risk instead of classification
evaluation$prob <- predict(reg, newdata = evaluation, type = "response")
#     Calculate baseline
count_affair <- nrow(subset(affair, affair$if_affair == "yes") )
baseline <- count_affair / nrow(affair)
baseline     #this is the affair rate which you want to beat with prediction
#######################################
 ###        create roc graph              ###
###S####################################
#   sensitive     Specificity
g <- roc(evaluation$if_affair ~ evaluation$prob, data = evaluation)
#     ROC curve
plot(g)
summary(g)
#     Show Area Under the Curve (AUC)
g
```

Screen Output:

```
> library(C50)
> library(pROC)
> setwd("/Users/kirsten 1/Documents/Masters Programs/MIS 545 Data Mining - UofA/Lab 6 Model Evalution")
> getwd()
[1] "/Users/kirsten 1/Documents/Masters Programs/MIS 545 Data Mining - UofA/Lab 6 Model Evalution"
```

```
 affair <- read.csv("affairs.csv")
> str(affair)
'data.frame':	601 obs. of  5 variables:
 $ age           : num  37 27 32 57 22 32 22 57 32 22 ...
 $ yearsmarried  : num  10 4 15 15 0.75 1.5 0.75 15 15 1.5 ...
 $ religiousness: int  3 4 1 5 2 2 2 2 4 4 ...
 $ rating        : int  4 4 4 5 3 5 3 4 2 5 ...
 $ if_affair     : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
> head(affair)
  age yearsmarried religiousness rating if_affair
1  37        10.00             3      4        no
2  27         4.00             4      4        no
3  32        15.00             1      4        no
4  57        15.00             5      5        no
5  22         0.75             2      3        no
6  32         1.50             2      5        no
> nrow(affair)
[1] 601
> nrow(!complete.cases(affair))
NULL
>
> size
[1] 480
```

```
>  ###   randomly decide which ones for training
> training_index <- sample(nrow(affair), size = size, replace = FALSE)
>
> train <- affair[training_index,]
> test <- affair[-training_index,]
> var_names <- names(affair)[-5]  #this gives column titles of first 4 va
#5)
> var_names
[1] "age"           "yearsmarried"  "religiousness" "rating"
> dt <- C5.0(x = train[, var_names], y = train$if_affair)
> summary(dt)

Call:
C5.0.default(x = train[, var_names], y = train$if_affair)


C5.0 [Release 2.07 GPL Edition]   Tue Aug 20 17:41:23 2019
-------------------------------

Class specified by attribute `outcome'

Read 480 cases (5 attributes) from undefined.data

Decision tree:

rating > 2: no (411/85)
rating <= 2:
:...age <= 47: yes (60/28)
    age > 47: no (9)


Evaluation on training data (480 cases):

	    Decision Tree
	  ----------------
	  Size      Errors

	     3  113(23.5%)   <<


	   (a)   (b)    <-classified as
	  ----  ----
	   335    28    (a): class no
	    85    32    (b): class yes


	Attribute usage:

	100.00%  rating
	 14.38%  age


Time: 0.0 secs
```

```
> dt_pred <- predict(dt, newdata = test)
> dt_evaluation <- cbind(test, dt_pred)
> head(test)
   age yearsmarried religiousness rating if_affair
8   57         15.0             2      4        no
10  22          1.5             4      5        no
17  37         15.0             2      3        no
26  22          1.5             2      5        no
27  27          7.0             4      5        no
32  22          1.5             3      5        no
> head(dt_evaluation)
   age yearsmarried religiousness rating if_affair dt_pred
8   57         15.0             2      4        no      no
10  22          1.5             4      5        no      no
17  37         15.0             2      3        no      no
26  22          1.5             2      5        no      no
27  27          7.0             4      5        no      no
32  22          1.5             3      5        no      no
> dt_evaluation$correct <- ifelse(dt_evaluation$if_affair == dt_evaluation$dt_pred, 1, 0)
> head(dt_evaluation)
   age yearsmarried religiousness rating if_affair dt_pred correct
8   57         15.0             2      4        no      no       1
10  22          1.5             4      5        no      no       1
17  37         15.0             2      3        no      no       1
26  22          1.5             2      5        no      no       1
27  27          7.0             4      5        no      no       1
32  22          1.5             3      5        no      no       1
```

```
> sum(dt_evaluation$correct) / nrow(dt_evaluation)
[1] 0.7768595
> table(dt_evaluation$if_affair, dt_evaluation$dt_pred)

      no yes
  no  85   3
  yes 24   9
> TPR <- sum(dt_evaluation$dt_pred == 'yes' & dt_evaluation$if_affair == 'yes') /
sum(dt_evaluation$if_affair == 'yes')
> TPR
[1] 0.2727273
> TNR <- sum(dt_evaluation$dt_pred == 'no' & dt_evaluation$if_affair == 'no') /
sum(dt_evaluation$if_affair == 'no')
> TNR
[1] 0.9659091
> FPR <- 1 - TNR
> FPR
[1] 0.03409091
> FNR <- 1 - TPR
> FNR
[1] 0.7272727
>  dt_precision <- sum(dt_evaluation$if_affair == 'yes' & dt_evaluation$dt_pred == 'yes') /
sum(dt_evaluation$dt_pred == 'yes')
> dt_precision
[1] 0.75
> dt_recall <- sum(dt_evaluation$if_affair == 'yes' & dt_evaluation$dt_pred == 'yes') /
sum(dt_evaluation$if_affair == 'yes')
> dt_recall
[1] 0.2727273
```

```
>         training_index2 <- sample(nrow(titanic), size = size, replace = FALSE)
>         training_index2 <- sample(nrow(affair), size = size, replace = FALSE)
> training_index2 <- sample(nrow(affair), size = size, replace = FALSE)
> train2 <- affair[training_index2,]
> test2 <- affair[-training_index2,]
> reg <- glm(if_affair ~ . , data = train2, family = binomial() )
> summary(reg)

Call:
glm(formula = if_affair ~ ., family = binomial(), data = train2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6726  -0.7540  -0.5560   0.7719   2.4087

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.79728    0.67329   2.669 0.007599 **
age           -0.03788    0.01936  -1.956 0.050449 .
yearsmarried   0.11421    0.03298   3.463 0.000534 ***
religiousness -0.24954    0.10147  -2.459 0.013924 *
rating        -0.49466    0.09822  -5.036 4.75e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 542.03  on 479  degrees of freedom
Residual deviance: 488.35  on 475  degrees of freedom
AIC: 498.35

Number of Fisher Scoring iterations: 4
> evaluation <- test2
> evaluation$prob <- predict(reg, newdata = evaluation, type = "response")
> count_affair <- nrow(subset(affair, affair$if_affair == "yes") )
> baseline <- count_affair / nrow(affair)
> baseline   #this is the affair rate which you want to beat with prediction
[1] 0.249584
> g <- roc(evaluation$if_affair ~ evaluation$prob, data = evaluation)
Setting levels: control = no, case = yes
Setting direction: controls < cases
> plot(g)
> summary(g)
                   Length Class  Mode
percent              1    -none- logical
sensitivities       95    -none- numeric
specificities       95    -none- numeric
thresholds          95    -none- numeric
direction            1    -none- character
cases               29    -none- numeric
controls            92    -none- numeric
fun.sesp             1    -none- function
auc                  1    auc    numeric
call                 3    -none- call
original.predictor 121    -none- numeric
original.response  121    factor numeric
predictor          121    -none- numeric
response           121    factor numeric
levels               2    -none- character
> g

Call:
roc.formula(formula = evaluation$if_affair ~ evaluation$prob,     data = evaluation)

Data: evaluation$prob in 92 controls (evaluation$if_affair no) < 29 cases (evaluation$if_affair yes).
Area under the curve: 0.6855
```