Report\_classificationTree.R

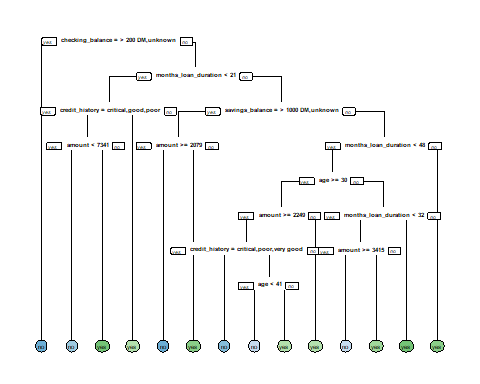
#setwd("")  
  
########################################################################  
#Build a classification tree  
########################################################################  
credit <- read.csv("credit.csv", stringsAsFactors = TRUE)  
########################################################################  
#Split data in 80% 20%  
########################################################################  
# Total number of rows in the credit data frame  
n <- nrow(credit)  
# Number of rows for the training set (80% of the dataset)  
n\_train <- round(0.8 \* n)   
# Create a vector of indices which is an 80% random sample  
set.seed(123)  
train\_indices <- sample(1:n, n\_train)  
# Subset the credit data frame to training indices only  
credit\_train <- credit[train\_indices, ]   
# Exclude the training indices to create the test set  
credit\_test <- credit[-train\_indices, ]  
########################################################################  
#Train a classification tree model  
########################################################################  
library(rpart)  
library(rpart.plot)  
# Train the model (to predict 'default')  
credit\_model <- rpart(formula = default ~.,   
 data = credit\_train,   
 method = "class")  
  
# Look at the model output   
print(credit\_model)

## n= 800   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 800 238 no (0.70250000 0.29750000)   
## 2) checking\_balance=> 200 DM,unknown 369 45 no (0.87804878 0.12195122) \*  
## 3) checking\_balance=< 0 DM,1 - 200 DM 431 193 no (0.55220418 0.44779582)   
## 6) months\_loan\_duration< 20.5 231 84 no (0.63636364 0.36363636)   
## 12) credit\_history=critical,good,poor 207 66 no (0.68115942 0.31884058)   
## 24) amount< 7341 200 60 no (0.70000000 0.30000000) \*  
## 25) amount>=7341 7 1 yes (0.14285714 0.85714286) \*  
## 13) credit\_history=perfect,very good 24 6 yes (0.25000000 0.75000000) \*  
## 7) months\_loan\_duration>=20.5 200 91 yes (0.45500000 0.54500000)   
## 14) savings\_balance=> 1000 DM,unknown 35 9 no (0.74285714 0.25714286)   
## 28) amount>=2079 26 2 no (0.92307692 0.07692308) \*  
## 29) amount< 2079 9 2 yes (0.22222222 0.77777778) \*  
## 15) savings\_balance=< 100 DM,100 - 500 DM,500 - 1000 DM 165 65 yes (0.39393939 0.60606061)   
## 30) months\_loan\_duration< 47.5 132 60 yes (0.45454545 0.54545455)   
## 60) age>=29.5 77 35 no (0.54545455 0.45454545)   
## 120) amount>=2249 62 24 no (0.61290323 0.38709677)   
## 240) credit\_history=critical,poor,very good 25 5 no (0.80000000 0.20000000) \*  
## 241) credit\_history=good,perfect 37 18 yes (0.48648649 0.51351351)   
## 482) age< 41 21 7 no (0.66666667 0.33333333) \*  
## 483) age>=41 16 4 yes (0.25000000 0.75000000) \*  
## 121) amount< 2249 15 4 yes (0.26666667 0.73333333) \*  
## 61) age< 29.5 55 18 yes (0.32727273 0.67272727)   
## 122) months\_loan\_duration< 31.5 38 16 yes (0.42105263 0.57894737)   
## 244) amount>=3415 17 6 no (0.64705882 0.35294118) \*  
## 245) amount< 3415 21 5 yes (0.23809524 0.76190476) \*  
## 123) months\_loan\_duration>=31.5 17 2 yes (0.11764706 0.88235294) \*  
## 31) months\_loan\_duration>=47.5 33 5 yes (0.15151515 0.84848485) \*

# Display the results  
rpart.plot(x = credit\_model, yesno = 2, type = 0, extra = 0)  
########################################################################  
#Compute confusion matrix  
########################################################################  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2



# Generate predicted classes using the model object  
class\_prediction <- predict(object = credit\_model,   
 newdata = credit\_test,   
 type = "class")   
  
# Calculate the confusion matrix for the test set  
confusionMatrix(data = class\_prediction,   
 reference = credit\_test$default)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 125 46  
## yes 13 16  
##   
## Accuracy : 0.705   
## 95% CI : (0.6366, 0.7672)  
## No Information Rate : 0.69   
## P-Value [Acc > NIR] : 0.3543   
##   
## Kappa : 0.192   
## Mcnemar's Test P-Value : 3.099e-05   
##   
## Sensitivity : 0.9058   
## Specificity : 0.2581   
## Pos Pred Value : 0.7310   
## Neg Pred Value : 0.5517   
## Prevalence : 0.6900   
## Detection Rate : 0.6250   
## Detection Prevalence : 0.8550   
## Balanced Accuracy : 0.5819   
##   
## 'Positive' Class : no   
##

#Accuracy 0.75 VS No info 0.69  
########################################################################  
#Compare models with a different splitting criterion  
########################################################################  
#Train two models that use a different splitting criterion  
#use the validation set to choose a "best" model from this group  
# Train a gini-based model  
credit\_model1 <- rpart(formula = default ~ .,   
 data = credit\_train,   
 method = "class",  
 parms = list(split = "gini"))  
# Train an information-based model  
credit\_model2 <- rpart(formula = default ~ .,   
 data = credit\_train,   
 method = "class",  
 parms = list(split = "information"))  
# Generate predictions on the validation set using the gini model  
pred1 <- predict(object = credit\_model1,   
 newdata = credit\_test,  
 type = "class")   
# Generate predictions on the validation set using the information model  
pred2 <- predict(object = credit\_model2,   
 newdata = credit\_test,  
 type = "class")  
library(Metrics)

##   
## Attaching package: 'Metrics'

## The following objects are masked from 'package:caret':  
##   
## precision, recall

# Compare classification error  
ce(actual = credit\_test$default,   
 predicted = pred1)

## [1] 0.295

ce(actual = credit\_test$default,   
 predicted = pred2)

## [1] 0.275

#Information parameter is better from -0.20 points  
  
###################################################################################  
#Prepare Prediction parameter "info" for model comparison  
###################################################################################  
# Generate predictions on the validation set using the information model  
pred3 <- predict(object = credit\_model2,   
 newdata = credit\_test,  
 type = "prob")  
#saveRDS(pred3[,"yes"],file="dt\_preds")