Credit-Data_script.R

shank

2019-10-20

```
#=====================================================================================================
# R version 3.6.1 and Rattle version 5.2.7

# We begin most scripts by loading the required packages.
# Here are some initial packages to load and others will be
# identified as we proceed through the script. When writing
# our own scripts we often collect together the library
# commands at the beginning of the script here.

library(rattle)   # Access to the Rattle package for data mining and data analysis.
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.7 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(magrittr) # Utilise %>% and %<>% pipeline operators.

# This script generally records the process of building a model.
# It is also used to evaluate and score the model based on the training, validation, and
# test dataset.The logical variable 'building'
# is used to toggle between generating transformations,
# when building a model and using the transformations,
# when scoring a dataset.

building <- TRUE
scoring  <- ! building

# A pre-defined value is used to reset the random seed
# so that results are repeatable.

crv$seed <- 42

#=====================================================================================================

# Load a dataset from file.
#Note that file location will be different based on user.

library(readxl, quietly=TRUE)

crs$dataset <- read_excel("C:/Users/shank/Documents/DAT650/Credit Data.xls", guess_max=1e4)

crs$dataset
```

```
## # A tibble: 1,000 x 32
##    `OBS#` CHK_ACCT DURATION HISTORY NEW_CAR USED_CAR FURNITURE RADIO_TV
##     <dbl>    <dbl>    <dbl>   <dbl>   <dbl>    <dbl>     <dbl>    <dbl>
```

```
##  1      1        0        6       4       0        0        0        1
##  2      2        1       48       2       0        0        0        1
##  3      3        3       12       4       0        0        0        0
##  4      4        0       42       2       0        0        1        0
##  5      5        0       24       3       1        0        0        0
##  6      6        3       36       2       0        0        0        0
##  7      7        3       24       2       0        0        1        0
##  8      8        1       36       2       0        1        0        0
##  9      9        3       12       2       0        0        0        1
## 10     10        1       30       4       1        0        0        0
## # ... with 990 more rows, and 24 more variables: EDUCATION <dbl>,
## #   RETRAINING <dbl>, AMOUNT <dbl>, SAV_ACCT <dbl>, EMPLOYMENT <dbl>,
## #   INSTALL_RATE <dbl>, MALE_DIV <dbl>, MALE_SINGLE <dbl>,
## #   MALE_MAR_or_WID <dbl>, CO_APPLICANT <dbl>, GUARANTOR <dbl>,
## #   PRESENT_RESIDENT <dbl>, REAL_ESTATE <dbl>, PROP_UNKN_NONE <dbl>,
## #   AGE <dbl>, OTHER_INSTALL <dbl>, RENT <dbl>, OWN_RES <dbl>,
## #   NUM_CREDITS <dbl>, JOB <dbl>, NUM_DEPENDENTS <dbl>, TELEPHONE <dbl>,
## #   FOREIGN <dbl>, DEFAULT <dbl>
```

```
#============================================================================================================

# Build the train/validate/test datasets.

# nobs=1000 train=700 validate=150 test=150

set.seed(crv$seed)

crs$nobs <- nrow(crs$dataset)

crs$train <- sample(crs$nobs, 0.7*crs$nobs)

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
  crs$validate

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
  crs$test

# The following variable selections have been noted.

crs$input     <- c("OBS#", "CHK_ACCT", "DURATION", "HISTORY",
                   "NEW_CAR", "USED_CAR", "FURNITURE", "RADIO_TV",
                   "EDUCATION", "RETRAINING", "AMOUNT", "SAV_ACCT",
                   "EMPLOYMENT", "INSTALL_RATE", "MALE_DIV",
                   "MALE_SINGLE", "MALE_MAR_or_WID", "CO_APPLICANT",
                   "GUARANTOR", "PRESENT_RESIDENT", "REAL_ESTATE",
                   "PROP_UNKN_NONE", "AGE", "OTHER_INSTALL", "RENT",
                   "OWN_RES", "NUM_CREDITS", "JOB", "NUM_DEPENDENTS",
                   "TELEPHONE", "FOREIGN")
```

```
crs$numeric   <- c("OBS#", "CHK_ACCT", "DURATION", "HISTORY",
                   "NEW_CAR", "USED_CAR", "FURNITURE", "RADIO_TV",
                   "EDUCATION", "RETRAINING", "AMOUNT", "SAV_ACCT",
                   "EMPLOYMENT", "INSTALL_RATE", "MALE_DIV",
                   "MALE_SINGLE", "MALE_MAR_or_WID", "CO_APPLICANT",
                   "GUARANTOR", "PRESENT_RESIDENT", "REAL_ESTATE",
                   "PROP_UNKN_NONE", "AGE", "OTHER_INSTALL", "RENT",
                   "OWN_RES", "NUM_CREDITS", "JOB", "NUM_DEPENDENTS",
                   "TELEPHONE", "FOREIGN")

crs$categoric <- NULL

crs$target    <- "DEFAULT"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
crs$weights   <- NULL

#============================================================================================

# Remap variables.

# Transform into a factor.

crs$dataset[["TFC_CHK_ACCT"]] <- as.factor(crs$dataset[["CHK_ACCT"]])
crs$dataset[["TFC_HISTORY"]] <- as.factor(crs$dataset[["HISTORY"]])
crs$dataset[["TFC_NEW_CAR"]] <- as.factor(crs$dataset[["NEW_CAR"]])
crs$dataset[["TFC_USED_CAR"]] <- as.factor(crs$dataset[["USED_CAR"]])
crs$dataset[["TFC_FURNITURE"]] <- as.factor(crs$dataset[["FURNITURE"]])
crs$dataset[["TFC_RADIO_TV"]] <- as.factor(crs$dataset[["RADIO_TV"]])
crs$dataset[["TFC_EDUCATION"]] <- as.factor(crs$dataset[["EDUCATION"]])
crs$dataset[["TFC_RETRAINING"]] <- as.factor(crs$dataset[["RETRAINING"]])
crs$dataset[["TFC_SAV_ACCT"]] <- as.factor(crs$dataset[["SAV_ACCT"]])
crs$dataset[["TFC_EMPLOYMENT"]] <- as.factor(crs$dataset[["EMPLOYMENT"]])
crs$dataset[["TFC_MALE_DIV"]] <- as.factor(crs$dataset[["MALE_DIV"]])
crs$dataset[["TFC_MALE_SINGLE"]] <- as.factor(crs$dataset[["MALE_SINGLE"]])
crs$dataset[["TFC_MALE_MAR_or_WID"]] <- as.factor(crs$dataset[["MALE_MAR_or_WID"]])
crs$dataset[["TFC_CO_APPLICANT"]] <- as.factor(crs$dataset[["CO_APPLICANT"]])
crs$dataset[["TFC_GUARANTOR"]] <- as.factor(crs$dataset[["GUARANTOR"]])
crs$dataset[["TFC_PRESENT_RESIDENT"]] <- as.factor(crs$dataset[["PRESENT_RESIDENT"]])
crs$dataset[["TFC_REAL_ESTATE"]] <- as.factor(crs$dataset[["REAL_ESTATE"]])
crs$dataset[["TFC_PROP_UNKN_NONE"]] <- as.factor(crs$dataset[["PROP_UNKN_NONE"]])
crs$dataset[["TFC_OTHER_INSTALL"]] <- as.factor(crs$dataset[["OTHER_INSTALL"]])
crs$dataset[["TFC_RENT"]] <- as.factor(crs$dataset[["RENT"]])
crs$dataset[["TFC_OWN_RES"]] <- as.factor(crs$dataset[["OWN_RES"]])
crs$dataset[["TFC_JOB"]] <- as.factor(crs$dataset[["JOB"]])
crs$dataset[["TFC_TELEPHONE"]] <- as.factor(crs$dataset[["TELEPHONE"]])
crs$dataset[["TFC_FOREIGN"]] <- as.factor(crs$dataset[["FOREIGN"]])
crs$dataset[["TFC_DEFAULT"]] <- as.factor(crs$dataset[["DEFAULT"]])

ol <- levels(crs$dataset[["TFC_CHK_ACCT"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
```

```
levels(crs$dataset[["TFC_CHK_ACCT"]]) <- nl

ol <- levels(crs$dataset[["TFC_HISTORY"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_HISTORY"]]) <- nl

ol <- levels(crs$dataset[["TFC_NEW_CAR"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_NEW_CAR"]]) <- nl

ol <- levels(crs$dataset[["TFC_USED_CAR"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_USED_CAR"]]) <- nl

ol <- levels(crs$dataset[["TFC_FURNITURE"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_FURNITURE"]]) <- nl

ol <- levels(crs$dataset[["TFC_RADIO_TV"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_RADIO_TV"]]) <- nl

ol <- levels(crs$dataset[["TFC_EDUCATION"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_EDUCATION"]]) <- nl

ol <- levels(crs$dataset[["TFC_RETRAINING"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_RETRAINING"]]) <- nl

ol <- levels(crs$dataset[["TFC_SAV_ACCT"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_SAV_ACCT"]]) <- nl

ol <- levels(crs$dataset[["TFC_EMPLOYMENT"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_EMPLOYMENT"]]) <- nl

ol <- levels(crs$dataset[["TFC_MALE_DIV"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_MALE_DIV"]]) <- nl

ol <- levels(crs$dataset[["TFC_MALE_SINGLE"]])
```

```
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_MALE_SINGLE"]]) <- nl

ol <- levels(crs$dataset[["TFC_MALE_MAR_or_WID"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_MALE_MAR_or_WID"]]) <- nl

ol <- levels(crs$dataset[["TFC_CO_APPLICANT"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_CO_APPLICANT"]]) <- nl

ol <- levels(crs$dataset[["TFC_GUARANTOR"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_GUARANTOR"]]) <- nl

ol <- levels(crs$dataset[["TFC_PRESENT_RESIDENT"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_PRESENT_RESIDENT"]]) <- nl

ol <- levels(crs$dataset[["TFC_REAL_ESTATE"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_REAL_ESTATE"]]) <- nl

ol <- levels(crs$dataset[["TFC_PROP_UNKN_NONE"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_PROP_UNKN_NONE"]]) <- nl

ol <- levels(crs$dataset[["TFC_OTHER_INSTALL"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_OTHER_INSTALL"]]) <- nl

ol <- levels(crs$dataset[["TFC_RENT"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_RENT"]]) <- nl

ol <- levels(crs$dataset[["TFC_OWN_RES"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_OWN_RES"]]) <- nl

ol <- levels(crs$dataset[["TFC_JOB"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_JOB"]]) <- nl
```

```
ol <- levels(crs$dataset[["TFC_TELEPHONE"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_TELEPHONE"]]) <- nl

ol <- levels(crs$dataset[["TFC_FOREIGN"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_FOREIGN"]]) <- nl

ol <- levels(crs$dataset[["TFC_DEFAULT"]])
lol <- length(ol)
nl <- c(sprintf("[%s,%s]", ol[1], ol[1]), sprintf("(%s,%s]", ol[-lol], ol[-1]))
levels(crs$dataset[["TFC_DEFAULT"]]) <- nl

#============================================================================

# Build the train/validate/test datasets again after transformation of variables.

# nobs=1000 train=700 validate=150 test=150

set.seed(crv$seed)

crs$nobs <- nrow(crs$dataset)

crs$train <- sample(crs$nobs, 0.7*crs$nobs)

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
  crs$validate

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
  crs$test

# The following variable selections have been noted.

crs$input     <- c("DURATION", "AMOUNT", "INSTALL_RATE", "AGE",
                   "NUM_CREDITS", "NUM_DEPENDENTS", "TFC_CHK_ACCT",
                   "TFC_HISTORY", "TFC_NEW_CAR", "TFC_USED_CAR",
                   "TFC_FURNITURE", "TFC_RADIO_TV", "TFC_EDUCATION",
                   "TFC_RETRAINING", "TFC_SAV_ACCT",
                   "TFC_EMPLOYMENT", "TFC_MALE_DIV",
                   "TFC_MALE_SINGLE", "TFC_MALE_MAR_or_WID",
                   "TFC_CO_APPLICANT", "TFC_GUARANTOR",
                   "TFC_PRESENT_RESIDENT", "TFC_REAL_ESTATE",
                   "TFC_PROP_UNKN_NONE", "TFC_OTHER_INSTALL",
                   "TFC_RENT", "TFC_OWN_RES", "TFC_JOB",
                   "TFC_TELEPHONE", "TFC_FOREIGN")
```

```
crs$numeric   <- c("DURATION", "AMOUNT", "INSTALL_RATE", "AGE",
                   "NUM_CREDITS", "NUM_DEPENDENTS")

crs$categoric <- c("TFC_CHK_ACCT", "TFC_HISTORY", "TFC_NEW_CAR",
                   "TFC_USED_CAR", "TFC_FURNITURE", "TFC_RADIO_TV",
                   "TFC_EDUCATION", "TFC_RETRAINING", "TFC_SAV_ACCT",
                   "TFC_EMPLOYMENT", "TFC_MALE_DIV",
                   "TFC_MALE_SINGLE", "TFC_MALE_MAR_or_WID",
                   "TFC_CO_APPLICANT", "TFC_GUARANTOR",
                   "TFC_PRESENT_RESIDENT", "TFC_REAL_ESTATE",
                   "TFC_PROP_UNKN_NONE", "TFC_OTHER_INSTALL",
                   "TFC_RENT", "TFC_OWN_RES", "TFC_JOB",
                   "TFC_TELEPHONE", "TFC_FOREIGN")

crs$target    <- "TFC_DEFAULT"
crs$risk      <- NULL
crs$ident     <- "OBS#"
crs$ignore    <- c("CHK_ACCT", "HISTORY", "NEW_CAR", "USED_CAR", "FURNITURE", "RADIO_TV", "EDUCATION",
crs$weights   <- NULL

#============================================================================================

# The 'Hmisc' package provides the 'describe' and 'contents' function.

library(Hmisc, quietly=TRUE)
```

```
##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# Obtain a summary of the dataset.

contents(crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)])
```

```
##
## Data frame:crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)]    700 observations and 31 vari
##
##
##                       Levels Storage
## DURATION                      double
## AMOUNT                        double
## INSTALL_RATE                  double
## AGE                           double
## NUM_CREDITS                   double
## NUM_DEPENDENTS                double
## TFC_CHK_ACCT                 4 integer
## TFC_HISTORY                  5 integer
## TFC_NEW_CAR                  2 integer
## TFC_USED_CAR                 2 integer
```

```
## TFC_FURNITURE             2 integer
## TFC_RADIO_TV              2 integer
## TFC_EDUCATION             2 integer
## TFC_RETRAINING            2 integer
## TFC_SAV_ACCT              5 integer
## TFC_EMPLOYMENT            5 integer
## TFC_MALE_DIV              2 integer
## TFC_MALE_SINGLE           2 integer
## TFC_MALE_MAR_or_WID       2 integer
## TFC_CO_APPLICANT          2 integer
## TFC_GUARANTOR             2 integer
## TFC_PRESENT_RESIDENT      4 integer
## TFC_REAL_ESTATE           2 integer
## TFC_PROP_UNKN_NONE        2 integer
## TFC_OTHER_INSTALL         2 integer
## TFC_RENT                  2 integer
## TFC_OWN_RES               2 integer
## TFC_JOB                   4 integer
## TFC_TELEPHONE             2 integer
## TFC_FOREIGN               2 integer
## TFC_DEFAULT               2 integer
##
## +--------------------+----------------------------+
## |Variable            |Levels                      |
## +--------------------+----------------------------+
## |TFC_CHK_ACCT        |[0,0],(0,1],(1,2],(2,3]     |
## |TFC_JOB             |                            |
## +--------------------+----------------------------+
## |TFC_HISTORY         |[0,0],(0,1],(1,2],(2,3],(3,4]|
## |TFC_SAV_ACCT        |                            |
## |TFC_EMPLOYMENT      |                            |
## +--------------------+----------------------------+
## |TFC_NEW_CAR         |[0,0],(0,1]                 |
## |TFC_USED_CAR        |                            |
## |TFC_FURNITURE       |                            |
## |TFC_RADIO_TV        |                            |
## |TFC_EDUCATION       |                            |
## |TFC_RETRAINING      |                            |
## |TFC_MALE_DIV        |                            |
## |TFC_MALE_SINGLE     |                            |
## |TFC_MALE_MAR_or_WID |                            |
## |TFC_CO_APPLICANT    |                            |
## |TFC_GUARANTOR       |                            |
## |TFC_REAL_ESTATE     |                            |
## |TFC_PROP_UNKN_NONE  |                            |
## |TFC_OTHER_INSTALL   |                            |
## |TFC_RENT            |                            |
## |TFC_OWN_RES         |                            |
## |TFC_TELEPHONE       |                            |
## |TFC_FOREIGN         |                            |
## |TFC_DEFAULT         |                            |
## +--------------------+----------------------------+
## |TFC_PRESENT_RESIDENT|[1,1],(1,2],(2,3],(3,4]     |
## +--------------------+----------------------------+
```

```
summary(crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)])
```

```
##     DURATION         AMOUNT       INSTALL_RATE        AGE
##  Min.   : 4.00   Min.   :  250   Min.   :1.000   Min.   :19.00
##  1st Qu.:12.00   1st Qu.: 1342   1st Qu.:2.000   1st Qu.:27.00
##  Median :18.00   Median : 2317   Median :3.000   Median :33.00
##  Mean   :20.66   Mean   : 3181   Mean   :2.984   Mean   :35.44
##  3rd Qu.:24.00   3rd Qu.: 3952   3rd Qu.:4.000   3rd Qu.:41.00
##  Max.   :72.00   Max.   :15672   Max.   :4.000   Max.   :75.00
##   NUM_CREDITS   NUM_DEPENDENTS  TFC_CHK_ACCT TFC_HISTORY TFC_NEW_CAR
##  Min.   :1.00   Min.   :1.000   [0,0]:185    [0,0]: 25   [0,0]:533
##  1st Qu.:1.00   1st Qu.:1.000   (0,1]:199    (0,1]: 36   (0,1]:167
##  Median :1.00   Median :1.000   (1,2]: 49    (1,2]:361
##  Mean   :1.41   Mean   :1.147   (2,3]:267    (2,3]: 59
##  3rd Qu.:2.00   3rd Qu.:1.000                (3,4]:219
##  Max.   :4.00   Max.   :2.000
##  TFC_USED_CAR TFC_FURNITURE TFC_RADIO_TV TFC_EDUCATION TFC_RETRAINING
##  [0,0]:631    [0,0]:576     [0,0]:498    [0,0]:664     [0,0]:627
##  (0,1]: 69    (0,1]:124     (0,1]:202    (0,1]: 36     (0,1]: 73
##
##
##
##
##  TFC_SAV_ACCT TFC_EMPLOYMENT TFC_MALE_DIV TFC_MALE_SINGLE
##  [0,0]:433    [0,0]: 45      [0,0]:662    [0,0]:318
##  (0,1]: 70    (0,1]:122      (0,1]: 38    (0,1]:382
##  (1,2]: 40    (1,2]:236
##  (2,3]: 32    (2,3]:121
##  (3,4]:125    (3,4]:176
##
##  TFC_MALE_MAR_or_WID TFC_CO_APPLICANT TFC_GUARANTOR TFC_PRESENT_RESIDENT
##  [0,0]:638           [0,0]:674        [0,0]:664     [1,1]: 93
##  (0,1]: 62           (0,1]: 26        (0,1]: 36     (1,2]:216
##                                                     (2,3]:102
##                                                     (3,4]:289
##
##
##  TFC_REAL_ESTATE TFC_PROP_UNKN_NONE TFC_OTHER_INSTALL  TFC_RENT
##  [0,0]:498       [0,0]:598          [0,0]:577         [0,0]:574
##  (0,1]:202       (0,1]:102          (0,1]:123         (0,1]:126
##
##
##
##
##  TFC_OWN_RES  TFC_JOB    TFC_TELEPHONE TFC_FOREIGN TFC_DEFAULT
##  [0,0]:196   [0,0]: 15   [0,0]:417     [0,0]:668   [0,0]:489
##  (0,1]:504   (0,1]:138   (0,1]:283     (0,1]: 32   (0,1]:211
##              (1,2]:450
##              (2,3]: 97
##
##
```

```
# Generate a description of the dataset.

describe(crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)])
```

```
## crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)]
##
##  31  Variables      700  Observations
## ---------------------------------------------------------------------------
## DURATION
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      700        0       29    0.985    20.66    12.86        6        9
##      .25      .50      .75      .90      .95
##       12       18       24       36       48
##
## lowest :  4  6  7  8  9, highest: 42 45 48 60 72
## ---------------------------------------------------------------------------
## AMOUNT
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      700        0      662        1     3181     2667    706.7    914.4
##      .25      .50      .75      .90      .95
##   1341.8   2317.0   3951.5   6970.2   8471.8
##
## lowest :   250   338   339   362   368, highest: 14318 14555 14782 15653 15672
## ---------------------------------------------------------------------------
## INSTALL_RATE
##        n  missing distinct     Info     Mean      Gmd
##      700        0        4    0.869    2.984      1.2
##
## Value          1     2     3     4
## Frequency     96   158   107   339
## Proportion 0.137 0.226 0.153 0.484
## ---------------------------------------------------------------------------
## AGE
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      700        0       52    0.999    35.44     12.4       22       23
##      .25      .50      .75      .90      .95
##       27       33       41       52       60
##
## lowest : 19 20 21 22 23, highest: 66 67 68 74 75
## ---------------------------------------------------------------------------
## NUM_CREDITS
##        n  missing distinct     Info     Mean      Gmd
##      700        0        4    0.717     1.41   0.5345
##
## Value          1     2     3     4
## Frequency    436   243    19     2
## Proportion 0.623 0.347 0.027 0.003
## ---------------------------------------------------------------------------
## NUM_DEPENDENTS
##        n  missing distinct     Info     Mean      Gmd
##      700        0        2    0.376    1.147   0.2513
##
## Value          1     2
```

```
## Frequency    597   103
## Proportion 0.853 0.147
## ---------------------------------------------------------------------------
## TFC_CHK_ACCT
##        n  missing distinct
##      700        0        4
##
## Value      [0,0] (0,1] (1,2] (2,3]
## Frequency    185   199    49   267
## Proportion 0.264 0.284 0.070 0.381
## ---------------------------------------------------------------------------
## TFC_HISTORY
##        n  missing distinct
##      700        0        5
##
## Value      [0,0] (0,1] (1,2] (2,3] (3,4]
## Frequency     25    36   361    59   219
## Proportion 0.036 0.051 0.516 0.084 0.313
## ---------------------------------------------------------------------------
## TFC_NEW_CAR
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    533   167
## Proportion 0.761 0.239
## ---------------------------------------------------------------------------
## TFC_USED_CAR
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    631    69
## Proportion 0.901 0.099
## ---------------------------------------------------------------------------
## TFC_FURNITURE
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    576   124
## Proportion 0.823 0.177
## ---------------------------------------------------------------------------
## TFC_RADIO_TV
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    498   202
## Proportion 0.711 0.289
## ---------------------------------------------------------------------------
## TFC_EDUCATION
##        n  missing distinct
##      700        0        2
```

```
##
## Value      [0,0] (0,1]
## Frequency    664    36
## Proportion 0.949 0.051
## ---------------------------------------------------------------------------
## TFC_RETRAINING
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    627    73
## Proportion 0.896 0.104
## ---------------------------------------------------------------------------
## TFC_SAV_ACCT
##        n  missing distinct
##      700        0        5
##
## Value      [0,0] (0,1] (1,2] (2,3] (3,4]
## Frequency    433    70    40    32   125
## Proportion 0.619 0.100 0.057 0.046 0.179
## ---------------------------------------------------------------------------
## TFC_EMPLOYMENT
##        n  missing distinct
##      700        0        5
##
## Value      [0,0] (0,1] (1,2] (2,3] (3,4]
## Frequency     45   122   236   121   176
## Proportion 0.064 0.174 0.337 0.173 0.251
## ---------------------------------------------------------------------------
## TFC_MALE_DIV
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    662    38
## Proportion 0.946 0.054
## ---------------------------------------------------------------------------
## TFC_MALE_SINGLE
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    318   382
## Proportion 0.454 0.546
## ---------------------------------------------------------------------------
## TFC_MALE_MAR_or_WID
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    638    62
## Proportion 0.911 0.089
## ---------------------------------------------------------------------------
## TFC_CO_APPLICANT
```

```
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    674    26
## Proportion 0.963 0.037
## -------------------------------------------------------------------------
## TFC_GUARANTOR
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    664    36
## Proportion 0.949 0.051
## -------------------------------------------------------------------------
## TFC_PRESENT_RESIDENT
##        n  missing distinct
##      700        0        4
##
## Value      [1,1] (1,2] (2,3] (3,4]
## Frequency     93   216   102   289
## Proportion 0.133 0.309 0.146 0.413
## -------------------------------------------------------------------------
## TFC_REAL_ESTATE
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    498   202
## Proportion 0.711 0.289
## -------------------------------------------------------------------------
## TFC_PROP_UNKN_NONE
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    598   102
## Proportion 0.854 0.146
## -------------------------------------------------------------------------
## TFC_OTHER_INSTALL
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    577   123
## Proportion 0.824 0.176
## -------------------------------------------------------------------------
## TFC_RENT
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    574   126
## Proportion  0.82  0.18
```

```
## ------------------------------------------------------------------------
## TFC_OWN_RES
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    196   504
## Proportion  0.28  0.72
## ------------------------------------------------------------------------
## TFC_JOB
##        n  missing distinct
##      700        0        4
##
## Value      [0,0] (0,1] (1,2] (2,3]
## Frequency     15   138   450    97
## Proportion 0.021 0.197 0.643 0.139
## ------------------------------------------------------------------------
## TFC_TELEPHONE
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    417   283
## Proportion 0.596 0.404
## ------------------------------------------------------------------------
## TFC_FOREIGN
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    668    32
## Proportion 0.954 0.046
## ------------------------------------------------------------------------
## TFC_DEFAULT
##        n  missing distinct
##      700        0        2
##
## Value      [0,0] (0,1]
## Frequency    489   211
## Proportion 0.699 0.301
## ------------------------------------------------------------------------
```

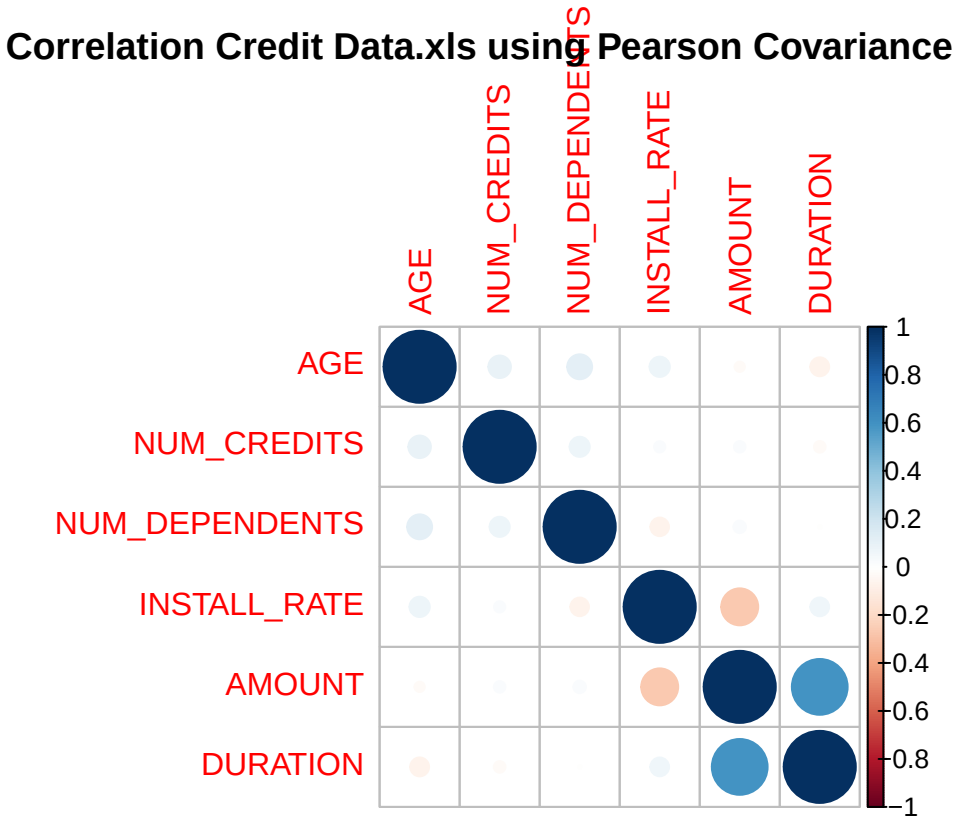
```
#=============================================================================================

# Generate a correlation plot for the numeric data type variables.

# The 'corrplot' package provides the 'corrplot' function.

library(corrplot, quietly=TRUE)
```

```
## corrplot 0.84 loaded
```

```
# Correlations work for numeric variables only.

crs$cor <- cor(crs$dataset[crs$train, crs$numeric], use="pairwise", method="pearson")

# Order the correlations by their strength.

crs$ord <- order(crs$cor[1,])
crs$cor <- crs$cor[crs$ord, crs$ord]

# Display the actual correlations.

print(crs$cor)
```

```
##                       AGE NUM_CREDITS NUM_DEPENDENTS INSTALL_RATE
## AGE             1.00000000  0.09733562    0.118936115   0.07885786
## NUM_CREDITS     0.09733562  1.00000000    0.077590746   0.02391158
## NUM_DEPENDENTS  0.11893611  0.07759075    1.000000000  -0.06608108
## INSTALL_RATE    0.07885786  0.02391158   -0.066081084   1.00000000
## AMOUNT         -0.02015010  0.02522412    0.029585776  -0.26276401
## DURATION       -0.06843380 -0.02572992   -0.002018986   0.06751893
##                     AMOUNT     DURATION
## AGE            -0.02015010 -0.068433798
## NUM_CREDITS     0.02522412 -0.025729920
## NUM_DEPENDENTS  0.02958578 -0.002018986
## INSTALL_RATE   -0.26276401  0.067518928
## AMOUNT          1.00000000  0.598987755
## DURATION        0.59898775  1.000000000
```

```
# Graphically display the correlations.

corrplot(crs$cor, mar=c(0,0,1,0))
title(main="Correlation Credit Data.xls using Pearson Covariance")
```




```
#============================================================================================

# Build a Random Forest model using the traditional approach.

set.seed(crv$seed)

crs$rf <- randomForest::randomForest(TFC_DEFAULT ~ .,
                                     data=crs$dataset[crs$train, c(crs$input, crs$target)],
                                     ntree=500,
                                     mtry=5,
                                     importance=TRUE,
                                     na.action=randomForest::na.roughfix,
                                     replace=FALSE)


#============================================================================================
# Build a Random Forest model using the traditional approach.

set.seed(crv$seed)

crs$rf <- randomForest::randomForest(TFC_DEFAULT ~ .,
                                     data=crs$dataset[crs$train, c(crs$input, crs$target)],
                                     ntree=500,
                                     mtry=5,
                                     importance=TRUE,
                                     na.action=randomForest::na.roughfix,
```

```
                                      replace=FALSE)

# Generate textual output of the 'Random Forest' model.

crs$rf
```

```
##
## Call:
##  randomForest(formula = TFC_DEFAULT ~ ., data = crs$dataset[crs$train,      c(crs$input, crs$target)]
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 25.43%
## Confusion matrix:
##       [0,0] (0,1] class.error
## [0,0]   448    41  0.08384458
## (0,1]   137    74  0.64928910
```

```
# The `pROC' package implements various AUC functions.

# Calculate the Area Under the Curve (AUC).

pROC::roc(crs$rf$y, as.numeric(crs$rf$predicted))
```

```
## Setting levels: control = [0,0], case = (0,1]
```

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

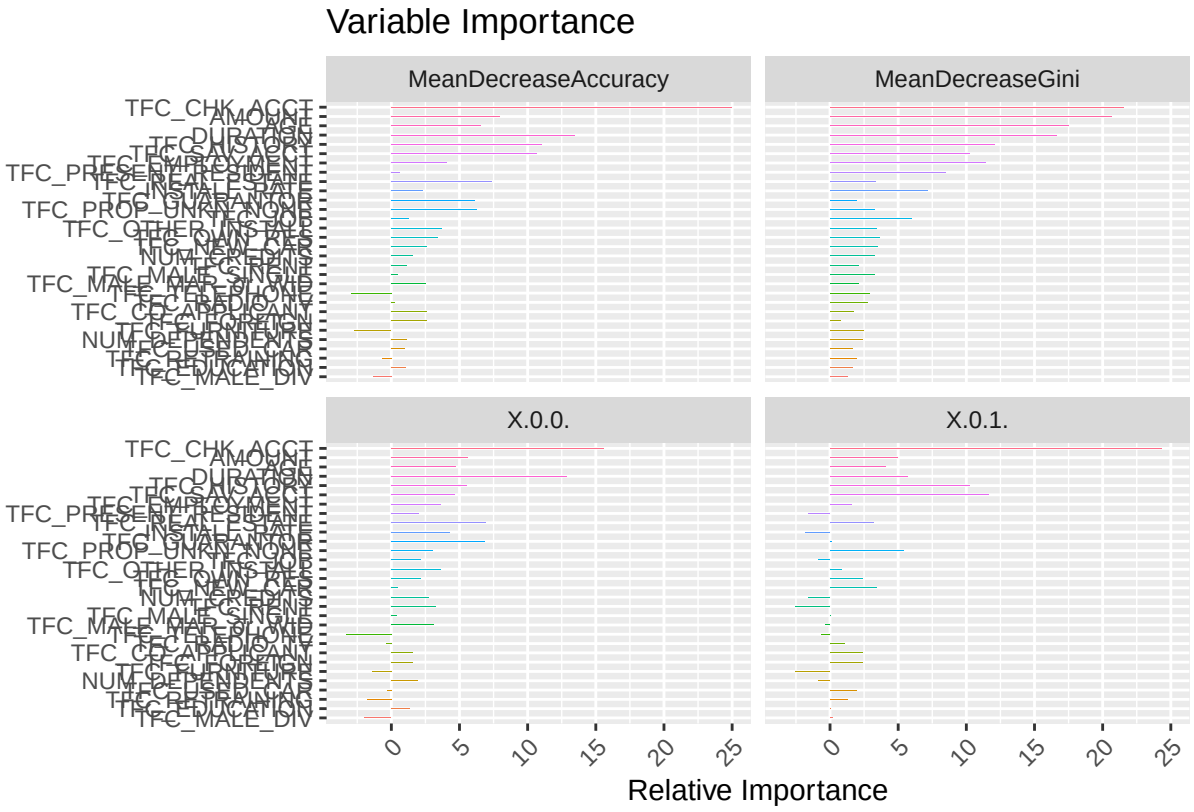
```
##
## Call:
## roc.default(response = crs$rf$y, predictor = as.numeric(crs$rf$predicted))
##
## Data: as.numeric(crs$rf$predicted) in 489 controls (crs$rf$y [0,0]) < 211 cases (crs$rf$y (0,1]).
## Area under the curve: 0.6334
```

```
# Calculate the AUC Confidence Interval.

#pROC::ci.auc(crs$rf$y, as.numeric(crs$rf$predicted))FALSE

# List the importance of the variables.

rn <- round(randomForest::importance(crs$rf), 2)
rn[order(rn[,3], decreasing=TRUE),]
```

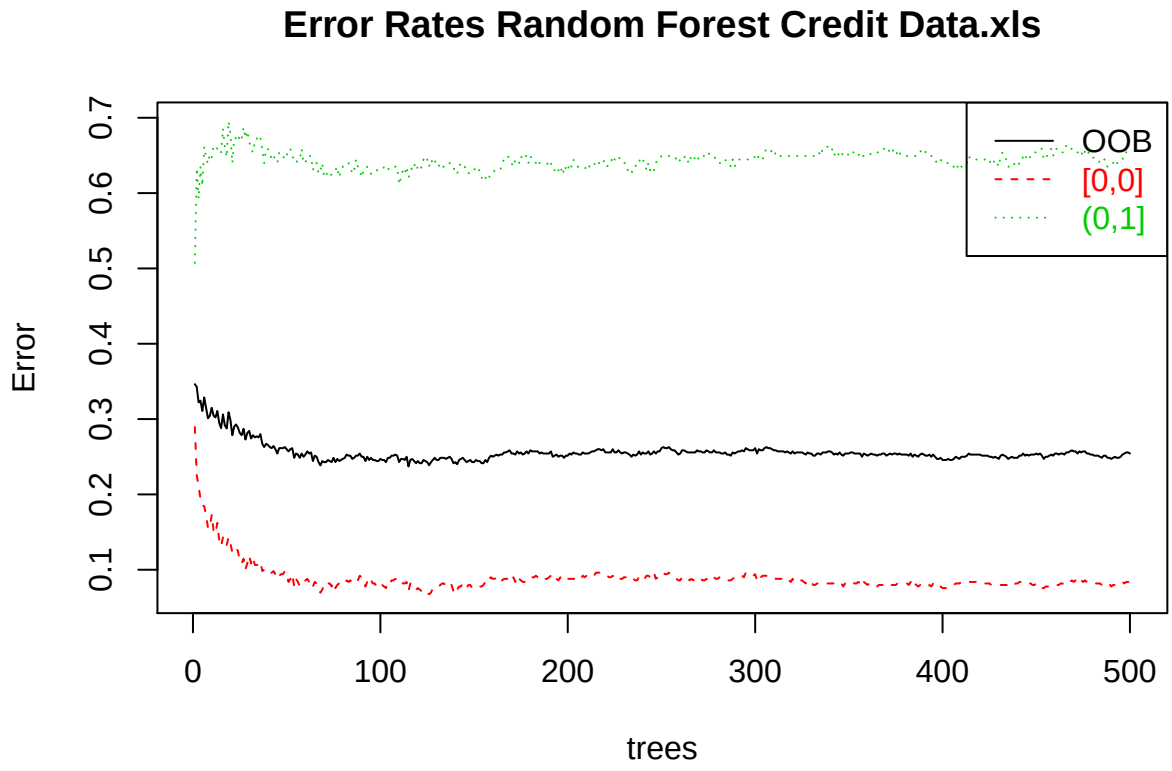
```
##                     [0,0] (0,1] MeanDecreaseAccuracy MeanDecreaseGini
## TFC_CHK_ACCT        15.58 24.35                24.95            21.56
## DURATION            12.84  5.71                13.49            16.65
## TFC_HISTORY          5.55 10.23                11.04            12.07
## TFC_SAV_ACCT         4.69 11.64                10.64            10.22
## AMOUNT               5.58  4.96                 7.94            20.65
```

```
## TFC_REAL_ESTATE       6.91  3.23                7.38             3.35
## AGE                   4.71  4.10                6.54            17.48
## TFC_PROP_UNKN_NONE    3.05  5.38                6.25             3.27
## TFC_GUARANTOR         6.85  0.11                6.09             1.94
## TFC_EMPLOYMENT        3.60  1.61                4.10            11.42
## TFC_OTHER_INSTALL     3.66  0.87                3.72             3.42
## TFC_OWN_RES           2.17  2.42                3.41             3.66
## TFC_NEW_CAR           0.45  3.38                2.59             3.52
## TFC_CO_APPLICANT      1.61  2.35                2.59             1.71
## TFC_FOREIGN           1.59  2.42                2.58             0.78
## TFC_MALE_MAR_or_WID   3.12 -0.31                2.54             2.07
## INSTALL_RATE          4.27 -1.78                2.32             7.17
## NUM_CREDITS           2.76 -1.61                1.61             3.26
## TFC_JOB               2.19 -0.87                1.29             5.99
## TFC_RENT              3.25 -2.51                1.17             2.10
## NUM_DEPENDENTS        1.97 -0.84                1.16             2.37
## TFC_EDUCATION         1.39  0.01                1.08             1.66
## TFC_USED_CAR         -0.34  1.97                0.99             1.64
## TFC_PRESENT_RESIDENT  2.02 -1.56                0.60             8.47
## TFC_MALE_SINGLE       0.40  0.03                0.45             3.23
## TFC_RADIO_TV         -0.39  1.08                0.23             2.73
## TFC_RETRAINING       -1.80  1.29               -0.70             1.96
## TFC_MALE_DIV         -1.98  0.16               -1.34             1.28
## TFC_FURNITURE        -1.41 -2.55               -2.71             2.43
## TFC_TELEPHONE        -3.33 -0.60               -2.96             2.93
```

```
#=====================================================================================================

# Plot the relative importance of the variables.

p <- ggVarImp(crs$rf,
              title="Variable Importance Random Forest Credit Data.xls")
p
```




```
#============================================================================================================

# Plot the error rate against the number of trees.
#This can determine the optimal number of trees to build

plot(crs$rf, main="")
legend("topright", c("OOB", "[0,0]", "(0,1]"), text.col=1:6, lty=1:3, col=1:3)
title(main="Error Rates Random Forest Credit Data.xls")
```




```
#=======================================================================================================


#=======================================================================================================
# Rattle timestamp: 2019-10-12 20:09:24 x86_64-w64-mingw32

# Re-Build a Random Forest model using the traditional approach with the optimal number of trees as
#indicated by the error rate plot.

set.seed(crv$seed)

crs$rf <- randomForest::randomForest(TFC_DEFAULT ~ .,
                                     data=crs$dataset[crs$train, c(crs$input, crs$target)],
                                     ntree=150,
                                     mtry=5,
                                     importance=TRUE,
                                     na.action=randomForest::na.roughfix,
                                     replace=FALSE)

# Generate textual output of the 'Random Forest' model.

crs$rf
```

```
##
## Call:
##  randomForest(formula = TFC_DEFAULT ~ ., data = crs$dataset[crs$train,      c(crs$input, crs$target)]
```

```
##                Type of random forest: classification
##                      Number of trees: 150
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 24.29%
## Confusion matrix:
##       [0,0] (0,1] class.error
## [0,0]   452    37  0.07566462
## (0,1]   133    78  0.63033175
```

```
# The `pROC' package implements various AUC functions.

# Calculate the Area Under the Curve (AUC).

pROC::roc(crs$rf$y, as.numeric(crs$rf$predicted))
```

```
## Setting levels: control = [0,0], case = (0,1]
## Setting direction: controls < cases
```

```
##
## Call:
## roc.default(response = crs$rf$y, predictor = as.numeric(crs$rf$predicted))
##
## Data: as.numeric(crs$rf$predicted) in 489 controls (crs$rf$y [0,0]) < 211 cases (crs$rf$y (0,1]).
## Area under the curve: 0.647
```

```
# Calculate the AUC Confidence Interval.

#pROC::ci.auc(crs$rf$y, as.numeric(crs$rf$predicted))FALSE

# List the importance of the variables.

rn <- round(randomForest::importance(crs$rf), 2)
rn[order(rn[,3], decreasing=TRUE),]
```

```
##                     [0,0] (0,1] MeanDecreaseAccuracy MeanDecreaseGini
## TFC_CHK_ACCT         8.40 13.06                13.33            21.15
## DURATION             6.65  3.03                 7.36            17.19
## TFC_SAV_ACCT         2.08  7.33                 6.11            10.55
## TFC_HISTORY          2.45  5.48                 5.82            12.47
## AGE                  3.66  1.85                 4.38            17.28
## TFC_REAL_ESTATE      3.35  2.25                 3.95             3.48
## TFC_PROP_UNKN_NONE   0.54  4.35                 3.37             3.04
## AMOUNT               1.41  3.34                 3.31            20.61
## TFC_GUARANTOR        3.48 -0.99                 3.13             1.98
## TFC_EMPLOYMENT       2.16  1.71                 2.98            11.27
## TFC_OTHER_INSTALL    2.65  0.76                 2.95             3.29
## TFC_NEW_CAR          0.83  2.82                 2.52             3.43
## TFC_OWN_RES          1.65  0.65                 1.85             3.97
## INSTALL_RATE         3.21 -1.30                 1.83             7.25
## TFC_MALE_MAR_or_WID  1.61  0.17                 1.55             1.89
## NUM_DEPENDENTS       1.63 -0.21                 1.36             2.32
```

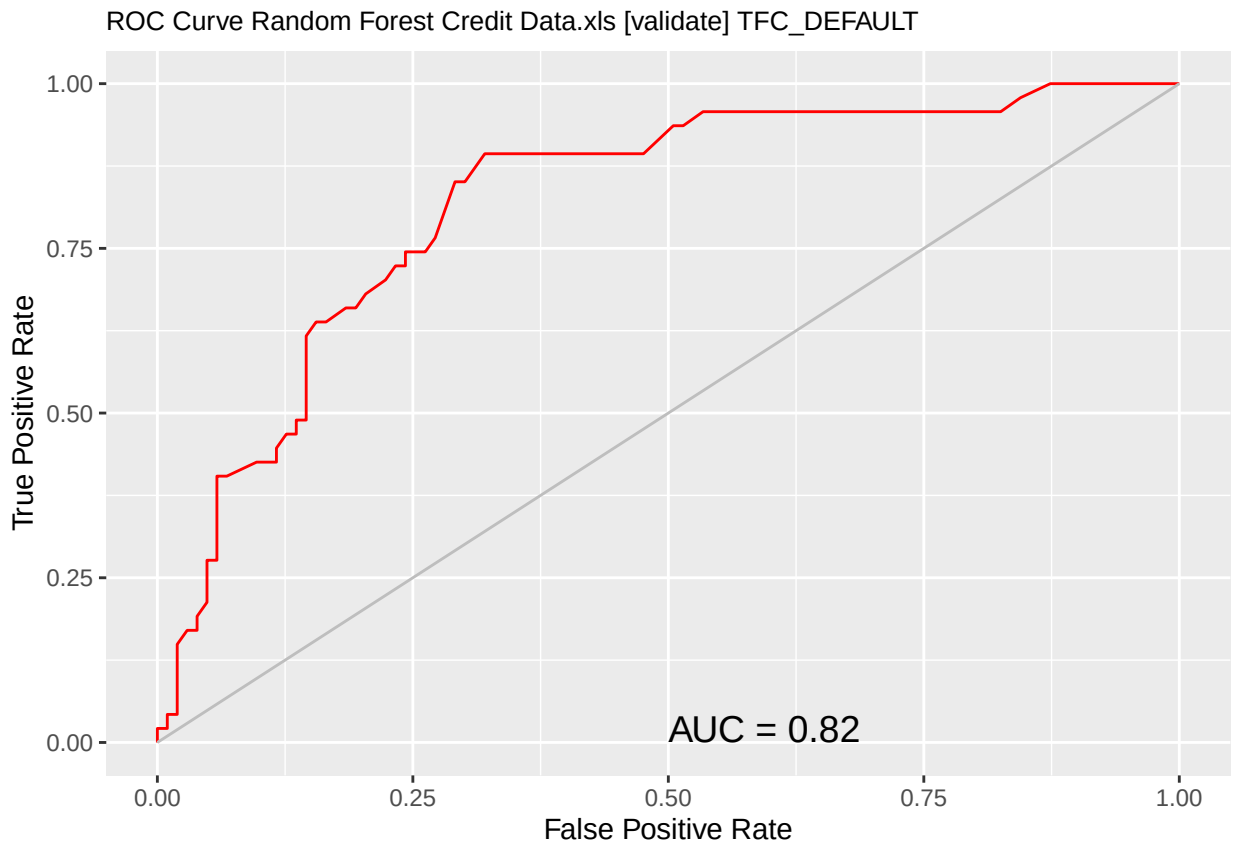
```
## TFC_CO_APPLICANT      0.87  0.70                1.01            1.87
## TFC_FOREIGN           1.00  0.28                1.00            0.70
## TFC_JOB               1.48 -1.00                0.77            5.91
## NUM_CREDITS           0.24 -0.11                0.24            3.26
## TFC_EDUCATION         0.59 -0.57                0.14            1.70
## TFC_RETRAINING       -0.25  0.31               -0.05            1.96
## TFC_MALE_SINGLE      -0.57  0.48               -0.10            3.29
## TFC_RADIO_TV         -0.66  0.64               -0.22            2.83
## TFC_PRESENT_RESIDENT -0.21 -0.07               -0.27            8.09
## TFC_USED_CAR         -0.74  0.40               -0.38            1.69
## TFC_RENT              1.33 -2.36               -0.38            1.89
## TFC_TELEPHONE        -2.35  0.57               -1.57            2.82
## TFC_FURNITURE        -1.74 -0.35               -1.73            2.39
## TFC_MALE_DIV         -1.72 -1.72               -2.18            1.36
```

```
#============================================================================================

# Evaluate model performance on the validation dataset.

# ROC Curve: requires the ROCR package.

library(ROCR)
```

```
## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```
# ROC Curve: requires the ggplot2 package.

library(ggplot2, quietly=TRUE)

# Generate an ROC Curve for the rf model on Credit Data.xls [validate].

crs$pr <- predict(crs$rf, newdata=na.omit(crs$dataset[crs$validate, c(crs$input, crs$target)]),
                  type    = "prob")[,2]

# Remove observations with missing target.

no.miss   <- na.omit(na.omit(crs$dataset[crs$validate, c(crs$input, crs$target)])$TFC_DEFAULT)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
```

```
}

pe <- performance(pred, "tpr", "fpr")
au <- performance(pred, "auc")@y.values[[1]]
pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve Random Forest Credit Data.xls [validate] TFC_DEFAULT")
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)), colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5,
                  label=paste("AUC =", round(au, 2)))
print(p)
```



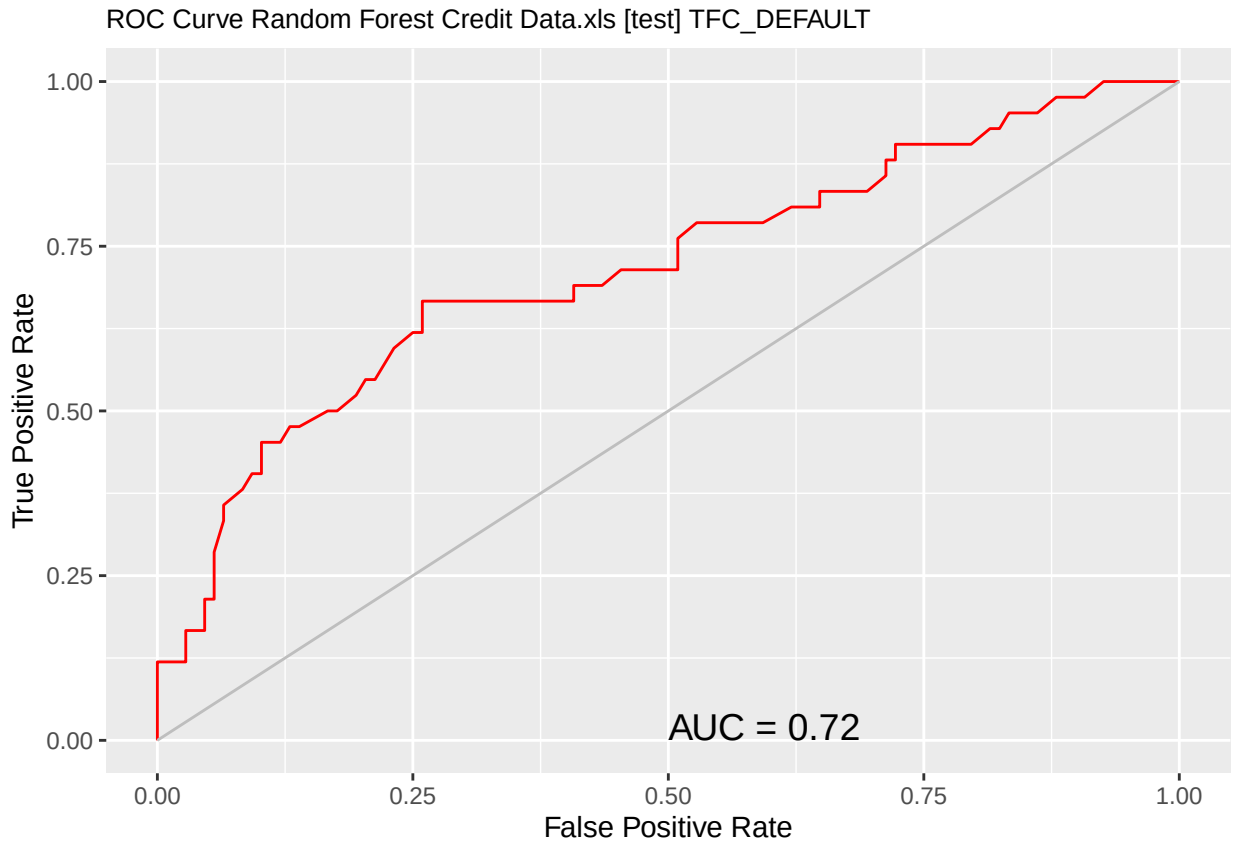

```
# Calculate the area under the curve for the plot.
# Remove observations with missing target.

no.miss   <- na.omit(na.omit(crs$dataset[crs$validate, c(crs$input, crs$target)])$TFC_DEFAULT)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
```

```
} else
{
  pred <- prediction(crs$pr, no.miss)
}
performance(pred, "auc")
```

```
## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.8199752
##
##
## Slot "alpha.values":
## list()
```

```
#=============================================================================================

# Evaluate model performance on the testing dataset.

# ROC Curve: requires the ROCR package.

library(ROCR)

# ROC Curve: requires the ggplot2 package.

library(ggplot2, quietly=TRUE)

# Generate an ROC Curve for the rf model on Credit Data.xls [test].

crs$pr <- predict(crs$rf, newdata=na.omit(crs$dataset[crs$test, c(crs$input, crs$target)]),
                  type    = "prob")[,2]

# Remove observations with missing target.

no.miss   <- na.omit(na.omit(crs$dataset[crs$test, c(crs$input, crs$target)])$TFC_DEFAULT)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
```

```
} else
{
  pred <- prediction(crs$pr, no.miss)
}

pe <- performance(pred, "tpr", "fpr")
au <- performance(pred, "auc")@y.values[[1]]
pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve Random Forest Credit Data.xls [test] TFC_DEFAULT")
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)), colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5,
                  label=paste("AUC =", round(au, 2)))
print(p)
```




```
# Calculate the area under the curve for the plot.


# Remove observations with missing target.

no.miss   <- na.omit(na.omit(crs$dataset[crs$test, c(crs$input, crs$target)])$TFC_DEFAULT)
miss.list <- attr(no.miss, "na.action")
```

```r
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
performance(pred, "auc")
```

```
## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.7158289
##
##
## Slot "alpha.values":
## list()
```