Module-4-R-Script-Template.R

Maniteja Kurukunda

2022-09-17

```
### Project: Modeling with Classification Trees ##############################


### Introduction #################################################################

# Data analysis should be reproducible, meaning: every step taken to manipulate,
# clean, transform, summarize, visualize or model data should be documented
# exactly so that results can be replicated.  An R Script is a tool---or,
# specifically, a document type---for doing reproducible data science. You
# should use comments like this to make notes (for your future self or
# colleagues) about the purpose and meaning of your code, as well as to add
# interpretation of your results.

# You can can easily compile an .R script file to HTML by selecting "Compile Report"
# under the top-level RStudio File menu. (File -> Compile Report...)

# You do not need to submit this script (or the compiled HTML).  It is provided for
# you to practice coding and writing using a script file.

### Preparation ##################################################################

# Load packages

library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6     ✓ purrr   0.3.4
## ✓ tibble  3.1.8     ✓ dplyr   1.0.9
## ✓ tidyr   1.2.0     ✓ stringr 1.4.1
## ✓ readr   2.1.2     ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
# Additionally, here are the packages needed to fit and visualize classification
# trees.  You will need to install them beforehand with this code (uncomment the
# lines first):

# install.packages("rpart")
# install.packages("rpart.plot")

# Then load the packages:
library(rpart)
library(rpart.plot)

# Load Data

# Below is code to load the dataset into memory. Before running that code,
# follow these preparatory steps:
#
# 1. Download this template and the dataset for the assignment from Canvas.
#
# 2. Copy or move these files from your downloads folder to a folder dedicated
# to this class--say, MKTG-6487.
#
# 3. You need to define this dedicated folder as your "working directory."  To
# do so, navigate to that folder using the files tab in the lower right quadrant
# in RStudio.  (You should see your files you moved into this folder in the
# previous step.) Click the "More" button in the menu under the Files tab and
# select "Set As Working Directory."
#
# Once the files are in the right location on your computer then you are ready
# to begin working run this code to clean and format the data:

advise_invest <- read_csv("adviseinvest.csv")
```

```
## Rows: 29504 Columns: 14
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (14): answered, income, female, age, job, num_dependents, rent, own_res,...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Clean and format data, using the following code chunk:

advise_invest <- read_csv("adviseinvest.csv")  %>%     # Download data and save it (via assignme
nt operator)
  select(-product) %>%                                 # Remove the product column
  na.omit %>%                                          # Remove rows with NAs
  filter(income > 0,                                   # Filter out mistaken data
         num_accts < 5) %>%
  mutate(answered = ifelse(answered==0, "no","yes"),   # Turn answered into yes/no
         answered = factor(answered,                   # Turn answered into factor
                           levels  = c("no", "yes")),  # Specify factor levels
         female = factor(female),                      # Make other binary and categorical
# variables into factors
         job = factor(job),
         rent = factor(rent),
         own_res = factor(own_res),
         new_car = factor(new_car),
         mobile = factor(mobile),
         chk_acct = factor(chk_acct),
         sav_acct = factor(sav_acct))
```

```
## Rows: 29504 Columns: 14
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## dbl (14): answered, income, female, age, job, num_dependents, rent, own_res,...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
### Assignment ########################################################################

# Write the code below that will enable you to answer the questions in the project
# quiz. Some of the questions do not require writing code and have been omitted
# from this template.

##Q2
round(mean(advise_invest$answered=="yes"),3)
```

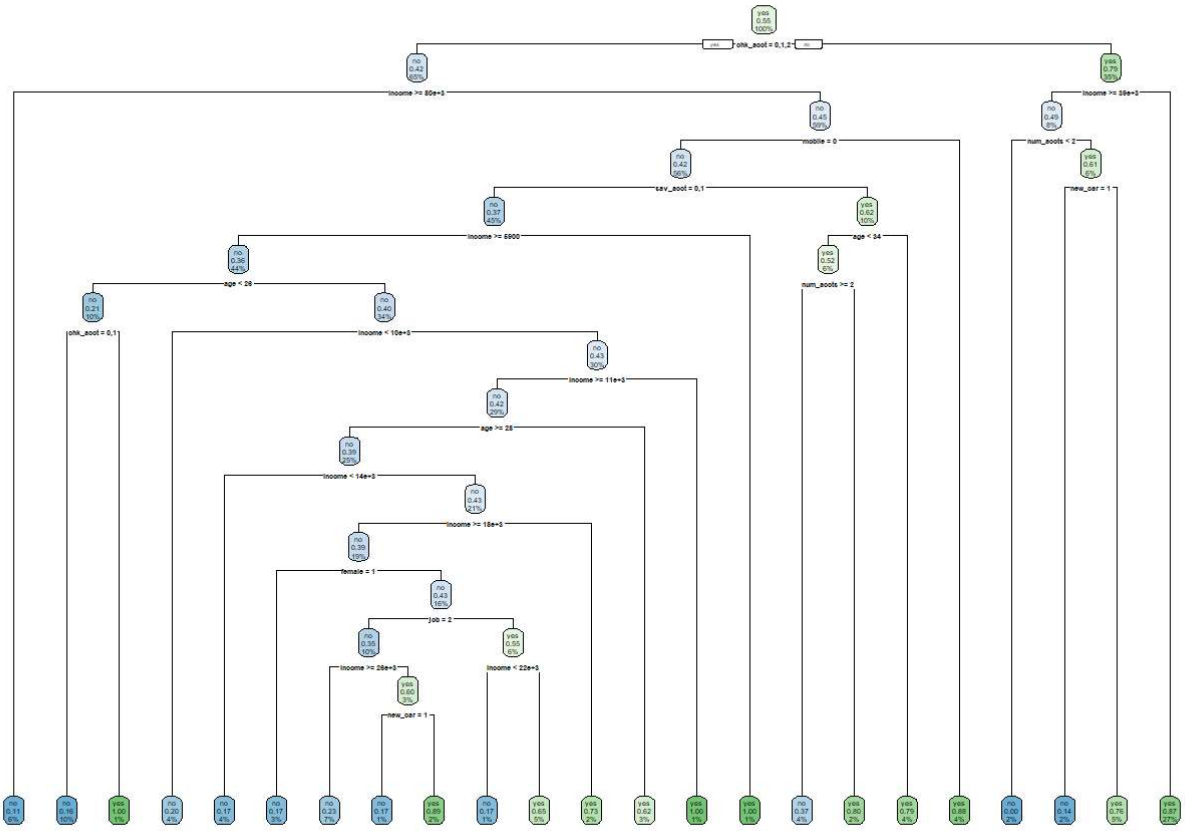
```
## [1] 0.547
```

```
# answer=0.547
##Q3
income_model<- rpart(formula = answered~income,data =advise_invest)
round((predict(object = income_model,type = "class")==advise_invest$answered)%>%mean,3)
```

```
## [1] 0.642
```

```
#0.642

#Q5
tree_model<- rpart(formula = answered~.,data =advise_invest)
rpart.plot(x = tree_model,tweak = 0.95, roundint=T)
```



```
#Q6
round((predict(object = tree_model,type = "class")==advise_invest$answered)%>%mean,3)
```

```
## [1] 0.82
```

```
#0.82
```