Mani.R

Maniteja Kurukunda

2022-09-25

```
### Project: Model Evaluation and Deployment #################################



### 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()
```

```
library(rpart)

# 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.
```

```
# And here is code to load the dataset of prospective customers from your
# working directory. Note that in order to use this dataset for prediction, the
# variables need to be formatted exactly the same as in the data used to fit the
# model. It does not include a target variable because the event of answering
# or not answering has not happened yet for scheduled customers.

prospective <- read_csv("customer_data.csv") %>%
  mutate(female = factor(female),
         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: 1000 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): customer_id
## dbl (12): income, female, age, job, num_dependents, rent, own_res, new_car, ...
##
## 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 #################################################################

## Questions

# One of the simplifying assumptions we will make in this project is that all
# the customers who answer the phone will purchase a product. (This assumption
# is actually verified by the data.) To model "answered" in this case is
# therefore equivalent to modeling "purchased."

# There are costs and benefits in this case. We will assume that customers
# purchase a product for $100 dollars. This was the average cost of
# AdviseInvest products, according to the Director of Sales.  Also, as we
# learned in the interview, the agent time to make the sale is worth $25.
# Profit would therefore be $75 dollars for an answered call and a purchase. In
# sum:

# Benefit: True positive. The customer is predicted to answer, does answer,
#  and purchases a product for $100 for a profit of 100 - 25 = $75.

# Cost: False positive. The customer is predicted to answer, but does not
# answer, so there is a loss of $25. (We assume the agent cannot schedule
# another call at the last minute, or spends the entire time slot trying to make
# the call.)

# For this exercise, we propose that customers who are not predicted to answer
# will not be called, so there would be no benefits and no costs for them.

# However, this proposal is for illustration only.  Below you will be asked to
# come up with a final recommendation for the Director of Sales, and you should
# feel free to craft a solution--whatever that might be--that fits the details
# of the case.

# One thing to keep in mind for this final phase of the project is that a
# predictive model is always developed using historical data.  The end goal,
# however, is to predict the future occurrence of the event that has been
# modeled. In this exercise, you will practice using data on new
# customers---that is, customers who have not yet been called---to predict
# whether they will answer. How you use these predictions in solving the
# business problem is up to you.

### Q1.
tree_model<- rpart(formula = answered~.,data =advise_invest)
table(predicted = predict(tree_model, type = "class"),
      observed = advise_invest$answered)
```

```
##          observed
## predicted    no   yes
##       no  10367  2304
##       yes  3008 13820
```

```
### Q2
75*13820-25*3008
```

```
## [1] 961300
```

```
### Q3
table(advise_invest$answered)
```

```
##
##    no   yes
## 13375 16124
```

```
16124*75-13375*25
```

```
## [1] 874925
```

```
### Q4
table(predicted = ifelse(predict(tree_model, type = "prob")[,1] >= .3, "yes", "no"),
      observed = advise_invest$answered)
```

```
##          observed
## predicted    no   yes
##       no   2112 12220
##       yes 11263  3904
```

```
3904*75-25*11263
```

```
## [1] 11225
```

```
### Q5
predictions<-data.frame(predict(tree_model, newdata = prospective,type = "prob"))
predictions$yes<-ifelse(predictions$yes>=0.3 ,"yes", "no")
predictions <- predictions%>%filter(predictions$yes =="yes")
contact_list<-predictions$yes

###Q6
```