Assignment: Esoph and Young People Survey

BOUN - ETM 58D

Due Date April 24, 2018

This is an individual assignment. You are given 3 data sets to explore and build models. Add the assignment outcomes to your individual Progress Journals in 3 different html pages built with RMarkdown with all your code and outputs. **Please** do not forget to add detailed comments and explanations in proper content length to show that you understand these concepts and you can express your findings in a coherent way. Even though there is verbal guidance about the models you can use in the questions, you can use all the models you learned in the class (including random forests).

Collaboration is allowed but your work should be your own. Since copy pasting is so easy these days, high similarity between two submissions will result in penalized or nulled grades for this assignment. Those data sets are popular on internet. If you find an inspiration, please state it in a references section with links.

Assignment 1: Esoph and Young People Survey Data

Questions below ask for your insight about the data. There is no single and constructed way to the right answer. Objective of this assignment is to help you help yourself to get data from outside resources, analyze the data, validate your model and convey your conclusions in a clear and reproducible way. You are expected to use R Markdown outputs on your Progress Journals to show your work.

1. Use `esoph` data set to see if (o)esophageal cancer is related to alcohol consumption, age or tobacco consumption. (Just write `esoph` to your R console to get the data.)

2. Use the Young People Survey's Hobbies & Interests category answers to infer "meaning" from the data. You are expected to use methods described in the lecture notes. But also you are welcome to use different methods as well. You can get the data from Kaggle.

    1. History: Not interested 1-2-3-4-5 Very interested (integer)
    2. Psychology: Not interested 1-2-3-4-5 Very interested (integer)
    3. Politics: Not interested 1-2-3-4-5 Very interested (integer)
    4. Mathematics: Not interested 1-2-3-4-5 Very interested (integer)
    5. Physics: Not interested 1-2-3-4-5 Very interested (integer)
    6. Internet: Not interested 1-2-3-4-5 Very interested (integer)
    7. PC Software, Hardware: Not interested 1-2-3-4-5 Very interested (integer)
    8. Economy, Management: Not interested 1-2-3-4-5 Very interested (integer)
    9. Biology: Not interested 1-2-3-4-5 Very interested (integer)
    10. Chemistry: Not interested 1-2-3-4-5 Very interested (integer)
    11. Poetry reading: Not interested 1-2-3-4-5 Very interested (integer)
    12. Geography: Not interested 1-2-3-4-5 Very interested (integer)
    13. Foreign languages: Not interested 1-2-3-4-5 Very interested (integer)
    14. Medicine: Not interested 1-2-3-4-5 Very interested (integer)
    15. Law: Not interested 1-2-3-4-5 Very interested (integer)
    16. Cars: Not interested 1-2-3-4-5 Very interested (integer)
    17. Art: Not interested 1-2-3-4-5 Very interested (integer)
    18. Religion: Not interested 1-2-3-4-5 Very interested (integer)
    19. Outdoor activities: Not interested 1-2-3-4-5 Very interested (integer)
    20. Dancing: Not interested 1-2-3-4-5 Very interested (integer)
    21. Playing musical instruments: Not interested 1-2-3-4-5 Very interested (integer)

22. Poetry writing: Not interested 1-2-3-4-5 Very interested (integer)
23. Sport and leisure activities: Not interested 1-2-3-4-5 Very interested (integer)
24. Sport at competitive level: Not interested 1-2-3-4-5 Very interested (integer)
25. Gardening: Not interested 1-2-3-4-5 Very interested (integer)
26. Celebrity lifestyle: Not interested 1-2-3-4-5 Very interested (integer)
27. Shopping: Not interested 1-2-3-4-5 Very interested (integer)
28. Science and technology: Not interested 1-2-3-4-5 Very interested (integer)
29. Theatre: Not interested 1-2-3-4-5 Very interested (integer)
30. Socializing: Not interested 1-2-3-4-5 Very interested (integer)
31. Adrenaline sports: Not interested 1-2-3-4-5 Very interested (integer)
32. Pets: Not interested 1-2-3-4-5 Very interested (integer)

Assignment 2: Diamonds Data

Your assignment consists of finding the price of a diamond given its properties. You will use the `diamonds` data set in ggplot2 package (which is inside tidyverse). You need to do your exploratory analysis well and come up with a predictive model. Your performance depends on the difference between the actual price of the diamond and the predicted price by the model. Use the `price` column as the response variable and other columns (except `diamond_id`) as predictors.

You are recommended to use CART but welcome to use any advanced method you like. Add your exploratory analysis to form a basis of your model and include references (with links) if you are inspired from similar analysis. Use the following code (and random seed) to form your train and test data. Remember, you should train your model on the train data and your real performance depends on the test data.

```
set.seed(503)
library(tidyverse)
diamonds_test <- diamonds %>% mutate(diamond_id = row_number()) %>%
    group_by(cut, color, clarity) %>% sample_frac(0.2) %>% ungroup()

diamonds_train <- anti_join(diamonds %>% mutate(diamond_id = row_number()),
    diamonds_test, by = "diamond_id")

diamonds_train
```

```
## # A tibble: 43,143 x 11
##    carat cut       color clarity depth table price     x     y     z
##    <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
##  1 0.230 Ideal     E     SI2      61.5  55.0   326  3.95  3.98  2.43
##  2 0.210 Premium   E     SI1      59.8  61.0   326  3.89  3.84  2.31
##  3 0.230 Good      E     VS1      56.9  65.0   327  4.05  4.07  2.31
##  4 0.290 Premium   I     VS2      62.4  58.0   334  4.20  4.23  2.63
##  5 0.240 Very Good J     VVS2     62.8  57.0   336  3.94  3.96  2.48
##  6 0.240 Very Good I     VVS1     62.3  57.0   336  3.95  3.98  2.47
##  7 0.260 Very Good H     SI1      61.9  55.0   337  4.07  4.11  2.53
##  8 0.220 Fair      E     VS2      65.1  61.0   337  3.87  3.78  2.49
##  9 0.230 Very Good H     VS1      59.4  61.0   338  4.00  4.05  2.39
## 10 0.300 Good      J     SI1      64.0  55.0   339  4.25  4.28  2.73
## # ... with 43,133 more rows, and 1 more variable: diamond_id <int>
```

```
diamonds_test
```

```
## # A tibble: 10,797 x 11
##    carat cut   color clarity depth table price     x     y     z
##    <dbl> <ord> <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
##  1 3.40  Fair  D     I1       66.8  52.0 15964  9.42  9.34  6.27
##  2 0.900 Fair  D     SI2      64.7  59.0  3205  6.09  5.99  3.91
##  3 0.950 Fair  D     SI2      64.4  60.0  3384  6.06  6.02  3.89
##  4 1.00  Fair  D     SI2      65.2  56.0  3634  6.27  6.21  4.07
##  5 0.700 Fair  D     SI2      58.1  60.0  2358  5.79  5.82  3.37
##  6 1.04  Fair  D     SI2      64.9  56.0  4398  6.39  6.34  4.13
##  7 0.700 Fair  D     SI2      65.6  55.0  2167  5.59  5.50  3.64
##  8 1.03  Fair  D     SI2      66.4  56.0  3743  6.31  6.19  4.15
##  9 1.10  Fair  D     SI2      64.6  54.0  4725  6.56  6.49  4.22
## 10 2.01  Fair  D     SI2      59.4  66.0 15627  8.20  8.17  4.86
## # ... with 10,787 more rows, and 1 more variable: diamond_id <int>
```

Assignment 3: Spam Data

Original library is in UCI Database. See documentation on the website for further detail.

Your assignment consists of buiding a CART model to detect spam mail using UCI's Spambase data and analyze it. You performance depends on correct specification of spam/non-spam mails in the test subset. You are going to use the RData file given on course webpage. Report your way of thinking, methodology, code and results.

You can load the data by using `load` command from your working directory or anywhere if you specify the path. For some installations, you can also double click the on the RData file to load. Name of the data frame is `spam_data` (same as the file name).

```
load("spam_data.RData")
head(spam_data)
```

Column names and short explanations are given below. For further details see the UCI documentation given in the above link.

train_or_test - 0 train, 1 test

spam_or_not - 0 not spam, 1 spam

V1 - word_freq_make

V2 - word_freq_address

V3 - word_freq_all

V4 - word_freq_3d

V5 - word_freq_our

V6 - word_freq_over

V7 - word_freq_remove

V8 - word_freq_internet

V9 - word_freq_order

V10 - word_freq_mail

V11 - word_freq_receive

V12 - word_freq_will

V13 - word_freq_people

V14 - word_freq_report

V15 - word_freq_addresses

V16 - word_freq_free

V17 - word_freq_business

V18 - word_freq_email

V19 - word_freq_you

V20 - word_freq_credit

V21 - word_freq_your

V22 - word_freq_font

V23 - word_freq_000

V24 - word_freq_money

V25 - word_freq_hp

V26 - word_freq_hpl

V27 - word_freq_george

V28 - word_freq_650

V29 - word_freq_lab

V30 - word_freq_labs

V31 - word_freq_telnet

V32 - word_freq_857

V33 - word_freq_data

V34 - word_freq_415

V35 - word_freq_85

V36 - word_freq_technology

V37 - word_freq_1999

V38 - word_freq_parts

V39 - word_freq_pm

V40 - word_freq_direct

V41 - word_freq_cs

V42 - word_freq_meeting

V43 - word_freq_original

V44 - word_freq_project

V45 - word_freq_re

V46 - word_freq_edu

V47 - word_freq_table

V48 - word_freq_conference

V49 - char_freq_;

V50 - char_freq_(

V51 - char_freq_[

V52 - char_freq_!

V53 - char_freq_$

V54 - char_freq_#

V55 - capital_run_length_average

V56 - capital_run_length_longest

V57 - capital_run_length_total