Model Fitting II

Carolina Alves de Lima Salge

3/16/2021

Slide Code

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
insurance <- read_csv("insurance.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   age = col_double(),
##   sex = col_character(),
##   bmi = col_double(),
##   children = col_double(),
##   smoker = col_character(),
##   region = col_character(),
##   charges = col_double()
## )
```

```
insurance
```

```
## # A tibble: 1,338 x 7
##      age sex      bmi children smoker region    charges
##    <dbl> <chr>  <dbl>    <dbl> <chr>  <chr>       <dbl>
##  1    19 female  27.9        0 yes    southwest  16885.
##  2    18 male    33.8        1 no     southeast   1726.
##  3    28 male    33          3 no     southeast   4449.
##  4    33 male    22.7        0 no     northwest  21984.
##  5    32 male    28.9        0 no     northwest   3867.
##  6    31 female  25.7        0 no     southeast   3757.
##  7    46 female  33.4        1 no     southeast   8241.
##  8    37 female  27.7        3 no     northwest   7282.
##  9    37 male    29.8        2 no     northeast   6406.
## 10    60 female  25.8        0 no     northwest  28923.
## # ... with 1,328 more rows
```

```
# transform categories to numbers
insurance <- insurance %>%
  mutate(sexN = case_when(
    sex == "male" ~ 1,
    sex == "female" ~ 0
    )) %>%
  mutate(smokerN = case_when(
    smoker == "yes" ~ 1,
    smoker == "no" ~ 0
    )) %>%
  mutate(regionN = case_when(
    region == "southwest" ~ 1,
    region == "southeast" ~ 2,
    region == "northwest" ~ 3,
    region == "northeast" ~ 4
    ))

# only select numeric variables
df <- insurance %>%
  dplyr::select(charges, age, sexN, bmi, children, smokerN, regionN)

# drop missing values NAs
df1 <- drop_na(df)

set.seed(12L) # set a starting seed to be able to get reproducible results

# partition data
trainIndex <- createDataPartition(df1$charges, # target variable
                                  p = 0.8, # percentage that goes to training
                                  list = FALSE, # results will not be in a list
                                  times = 1) # number of partitions to create

charges_train <- df1[trainIndex, ] # data frame for training
charges_test <- df1[-trainIndex, ] # data frame for testing

# compute correlation between predictos
```

```r
cor(charges_train[,2:7])
```

```
##                   age         sexN           bmi      children      smokerN
## age       1.000000000 -0.008477524  0.1013065882  0.0569025927 -0.033065845
## sexN     -0.008477524  1.000000000  0.0504461319  0.0046657032  0.058039957
## bmi       0.101306588  0.050446132  1.0000000000 -0.0007692116 -0.014603519
## children  0.056902593  0.004665703 -0.0007692116  1.0000000000  0.007271808
## smokerN  -0.033065845  0.058039957 -0.0146035194  0.0072718079  1.000000000
## regionN   0.002014072  0.018623882 -0.1744241089 -0.0371746216  0.001804169
##               regionN
## age       0.002014072
## sexN      0.018623882
## bmi      -0.174424109
## children -0.037174622
## smokerN   0.001804169
## regionN   1.000000000
```

```r
# compute correlation between predictos and the target
cor(charges_train[,1:7])
```

```
##             charges          age         sexN           bmi      children
## charges  1.00000000  0.293517648  0.044956612  0.1924010282  0.0573252106
## age      0.29351765  1.000000000 -0.008477524  0.1013065882  0.0569025927
## sexN     0.04495661 -0.008477524  1.000000000  0.0504461319  0.0046657032
## bmi      0.19240103  0.101306588  0.050446132  1.0000000000 -0.0007692116
## children 0.05732521  0.056902593  0.004665703 -0.0007692116  1.0000000000
## smokerN  0.77555146 -0.033065845  0.058039957 -0.0146035194  0.0072718079
## regionN  0.01172501  0.002014072  0.018623882 -0.1744241089 -0.0371746216
##               smokerN      regionN
## charges   0.775551461  0.011725009
## age      -0.033065845  0.002014072
## sexN      0.058039957  0.018623882
## bmi      -0.014603519 -0.174424109
## children  0.007271808 -0.037174622
## smokerN   1.000000000  0.001804169
## regionN   0.001804169  1.000000000
```

```r
# age, bmi, and smoking are highly correlated with health costs

# use training set to build model
model <- train(charges ~ age + bmi + smokerN,
               data = charges_train, # use training set
               method = "lm") # linear regression

# now predict outcomes in test set
p <- predict(model, charges_test)

# how did we do? calculate performance across resamples
# RMSE and R-squared
postResample(pred = p, obs = charges_test$charges)
```

```
##         RMSE     Rsquared          MAE
## 5808.0045894    0.7989742 4184.9721150
```

```
# on average, our prediction is off by $5,808.00

# how can we improve performance? Try a different method!
model2 <- train(charges ~ age + bmi + smokerN,
               data = charges_train, # use training set
               method = "ranger") # random forest
```

```
## note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .
```

```
# now predict outcomes in test set
p1 <- predict(model2, charges_test)

# how did we do? calculate performance across resamples
# RMSE and R-squared
postResample(pred = p1, obs = charges_test$charges)
```

```
##         RMSE     Rsquared          MAE
## 4205.4282707    0.8933816 2436.1432025
```

```
# on average, our prediction is off by $4,632.99

# first collect the resampling results of each model
resamps <- resamples(list(LM = model,
                          RF = model2))
resamps
```

```
##
## Call:
## resamples.default(x = list(LM = model, RF = model2))
##
## Models: LM, RF
## Number of resamples: 25
## Performance metrics: MAE, RMSE, Rsquared
## Time estimates for: everything, final model fit
```

```
# then use a simple t-test to evaluate the null hypothesis that there is no difference
summary(diff(resamps))
```

```
##
## Call:
## summary.diff.resamples(object = diff(resamps))
##
## p-value adjustment: bonferroni
## Upper diagonal: estimates of the difference
## Lower diagonal: p-value for H0: difference = 0
##
## MAE
##    LM        RF
## LM           1403
## RF < 2.2e-16
##
```

```
## RMSE
##    LM         RF
## LM            1156
## RF 2.139e-15
##
## Rsquared
##    LM       RF
## LM          -0.09466
## RF 1.17e-14
```