Model Fitting III

Carolina Alves de Lima Salge

3/23/2021

Slide Code

```
library(tidyverse)
```

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

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     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()
```

```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: ’caret’

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

```
churn <- read_csv("churn.csv")
```

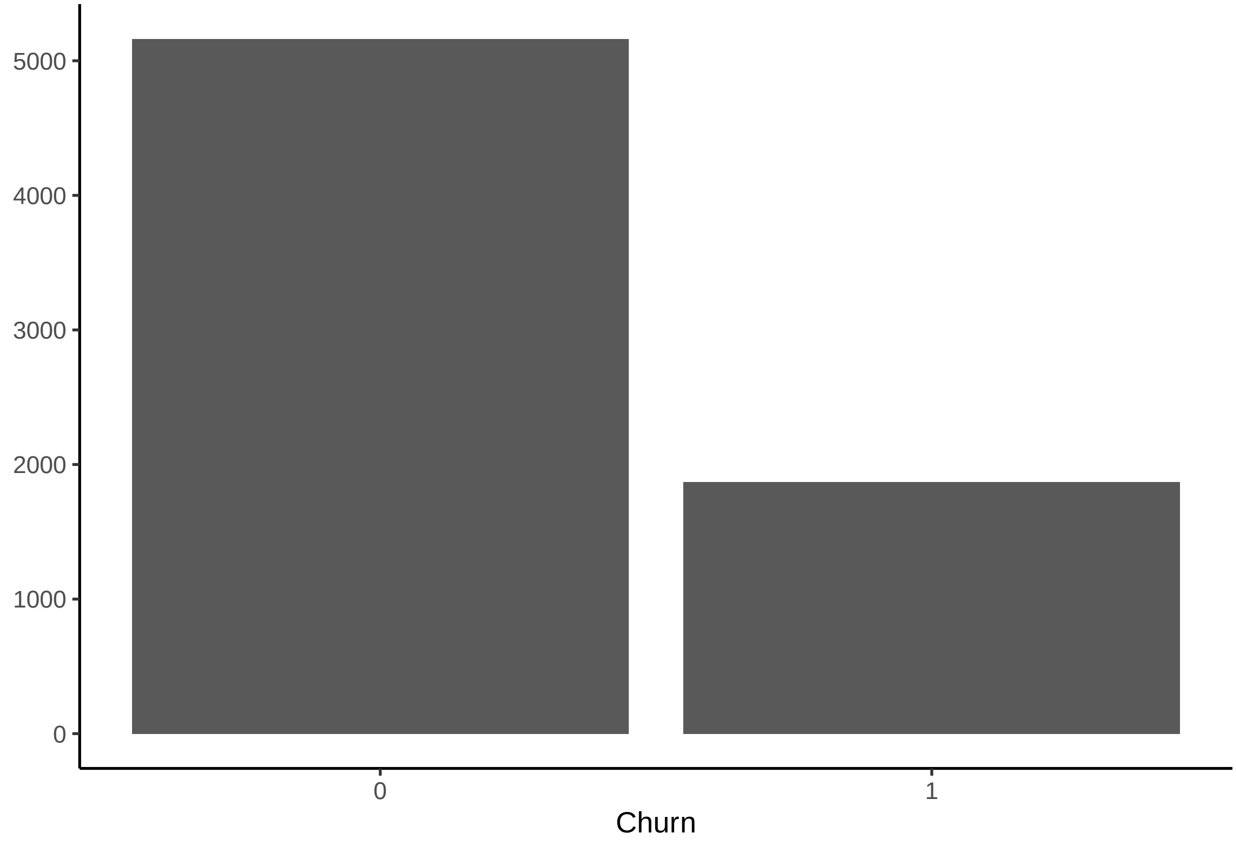
```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_character(),
##   SeniorCitizen = col_double(),
##   tenure = col_double(),
##   MonthlyCharges = col_double(),
##   TotalCharges = col_double()
## )
## i Use ‘spec()‘ for the full column specifications.
```

```r
# transform categories to numbers
churn <- churn %>%
  mutate(genderN = case_when(
    gender == "Male" ~ 1,
    gender == "Female" ~ 0
    )) %>%
  mutate(PartnerN = case_when(
    Partner == "Yes" ~ 1,
    Partner == "No" ~ 0
    )) %>%
  mutate(DependentsN = case_when(
    Dependents == "Yes" ~ 1,
    Dependents == "No" ~ 0
    )) %>%
  mutate(PhoneServiceN = case_when(
    PhoneService == "Yes" ~ 1,
    PhoneService == "No" ~ 0
    )) %>%
  mutate(MultipleLinesN = case_when(
    MultipleLines == "Yes" ~ 1,
    MultipleLines == "No" ~ 0,
    MultipleLines == "No phone service" ~ 0
    )) %>%
  mutate(InternetServiceN = case_when(
    InternetService == "Fiber optic" ~ 2,
    InternetService == "DSL" ~ 1,
    InternetService == "No" ~ 0
    )) %>%
  mutate(OnlineSecurityN = case_when(
    OnlineSecurity == "Yes" ~ 1,
    OnlineSecurity == "No" ~ 0,
    OnlineSecurity == "No internet service" ~ 0
    )) %>%
  mutate(OnlineBackupN = case_when(
    OnlineBackup == "Yes" ~ 1,
    OnlineBackup == "No" ~ 0,
    OnlineBackup == "No internet service" ~ 0
    )) %>%
  mutate(DeviceProtectionN = case_when(
    DeviceProtection == "Yes" ~ 1,
    DeviceProtection == "No" ~ 0,
    DeviceProtection == "No internet service" ~ 0
    )) %>%
  mutate(TechSupportN = case_when(
    TechSupport == "Yes" ~ 1,
    TechSupport == "No" ~ 0,
    TechSupport == "No internet service" ~ 0
    )) %>%
  mutate(StreamingTVN = case_when(
    StreamingTV == "Yes" ~ 1,
    StreamingTV == "No" ~ 0,
    StreamingTV == "No internet service" ~ 0
    )) %>%
  mutate(StreamingMoviesN = case_when(
    StreamingMovies == "Yes" ~ 1,
    StreamingMovies == "No" ~ 0,
    StreamingMovies == "No internet service" ~ 0
    )) %>%
  mutate(ContractN = case_when(
    Contract == "Month-to-month" ~ 0,
    Contract == "One year" ~ 1,
    Contract == "Two year" ~ 1
    )) %>%
  mutate(PaperlessN = case_when(
    PaperlessBilling == "Yes" ~ 1,
    PaperlessBilling == "No" ~ 0
    )) %>%
  mutate(PaymentN = case_when(
    PaymentMethod == "Electronic check" ~ 0,
    PaymentMethod == "Mailed check" ~ 0,
    PaymentMethod == "Bank transfer (automatic)" ~ 1,
    PaymentMethod == "Credit card (automatic)" ~ 1
    )) %>%
  mutate(ChurnN = case_when(
    Churn == "Yes" ~ 1,
    Churn == "No" ~ 0
    ))

# only select numeric variables
df <- churn %>% dplyr::select(Churn, ChurnN, SeniorCitizen, tenure,
                              MonthlyCharges, TotalCharges, genderN:PaymentN)

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

# is the target skewed?
ggplot(df1, aes(ChurnN)) +
  geom_bar() +
  theme_classic() +
  labs(x = "Churn", y = NULL) +
  scale_x_continuous(breaks = c(0,1))
```



```r
# yes - use precision and recall not only accuracy

# transform target into a factor
df1$Churn <- as.factor(df1$Churn)

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

# partition data
trainIndex <- createDataPartition(df1$Churn, # 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

churn_train <- df1[trainIndex, ] # data frame for training
```

```
## Warning: The ‘i‘ argument of ‘‘[‘()‘ can’t be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_warnings()‘ to see where this warning was generated.
```

```r
churn_test <- df1[-trainIndex, ] # data frame for testing

# compute correlation between predictors
predCor <- cor(churn_train[,3:21])
```

```
# which variables to remove to avoid multicollinearity?
findCorrelation(predCor, cutoff =  .7, names = TRUE)
```

```
## [1] "TotalCharges"   "MonthlyCharges"
```

```
churn_train <- churn_train %>%
  dplyr::select(Churn, ChurnN, SeniorCitizen, tenure, genderN:PaymentN)

# compute correlation between predictors and the target
predTargetCor <- cor(churn_train[,2:19])

model <- train(Churn ~ InternetServiceN + PaperlessN + SeniorCitizen +
                 PartnerN + TechSupportN + DependentsN + OnlineSecurityN +
                 PaymentN + tenure + ContractN,
               data = churn_train, # use training set
               method = "glm") # simple additive logistic regression

# now predict outcomes in test set
p <- predict(model, churn_test, type = 'raw')

# add predictions to initial dataset
churn_test$pred_churn <- p

# how did we do? confusion matrix
confusionMatrix(data = churn_test$pred_churn,
                reference = churn_test$Churn,
                mode = "prec_recall",
                positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  925 176
##        Yes 107 197
##
##                Accuracy : 0.7986
##                  95% CI : (0.7766, 0.8193)
##     No Information Rate : 0.7345
##     P-Value [Acc > NIR] : 1.342e-08
##
##                   Kappa : 0.4511
##
##  Mcnemar's Test P-Value : 5.296e-05
##
##               Precision : 0.6480
##                  Recall : 0.5282
##                      F1 : 0.5820
##              Prevalence : 0.2655
##          Detection Rate : 0.1402
##    Detection Prevalence : 0.2164
##       Balanced Accuracy : 0.7122
```

```
##
##        'Positive' Class : Yes
##
```