Homework 2

Sam Kuhn

11/28/22

Table of contents

Objective

In this homework assignment, you will explore, analyze and model a data set containing approximately 8000 records representing a customer at an auto insurance company. Each record has two response variables. The first response variable, TARGET_FLAG, is a 1 or a 0. A "1" means that the person was in a car crash. A zero means that the person was not in a car crash. The second response variable is TARGET_AMT. This value is zero if the person did not crash their car. But if they did crash their car, this number will be a value greater than zero.

Your objective is to build multiple linear regression and binary logistic regression models on the training data to predict the probability that a person will crash their car and also the amount of money it will cost if the person does crash their car. You can only use the variables given to you (or variables that you derive from the variables provided).

Data Exploration

Load data

```
# Install pacman package and load libraries
# install.packages("pacman")
pacman::p_load(tidyverse, here, tidymodels, corrplot, MASS, gt, stargazer)

# Makes sure dplyr::filter and dplyr::select will be used
conflicted::conflict_prefer("select", "dplyr")
conflicted::conflict_prefer("filter", "dplyr")

# Load training set from data folder and clean variable names
training_set <- readr::read_csv(
  here::here("data", "insurance_training_data.csv")
) |>
  janitor::clean_names()
```

First, lets rename the variables to a more readable format. Then, print a summary of the data to get a sense of what datatypes there are. From the table below, we can see we have a few variables where the datatype is wrong. In particular, `Income`, `Bluebook Value`, and `Total Claims` are all character columns rather than numeric. We will convert them. Furthermore, let's convert any binary variables to factors for modelling purposes.

```
Rows: 8,161
Columns: 26
$ index                 <dbl> 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, ~
$ `Crash Dummy`         <dbl> 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, ~
$ `Crash Damage`        <dbl> 0.000, 0.000, 0.000, 0.000, 0.000, 2946.000, ~
$ `Teen Drivers`        <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, ~
$ `Age of Driver`       <dbl> 60, 43, 35, 51, 50, 34, 54, 37, 34, 50, 53, 4~
$ `Kids at Home`        <dbl> 0, 0, 1, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, ~
$ `Years on Job`        <dbl> 11, 11, 10, 14, NA, 12, NA, NA, 10, 7, 14, 5,~
$ Income                <chr> "$67,349", "$91,449", "$16,039", NA, "$114,98~
$ `Single Parent`       <chr> "No", "No", "No", "No", "No", "Yes", "No", "N~
$ `Home Value`          <chr> "$0", "$257,252", "$124,191", "$306,251", "$2~
$ `Marital Status`      <chr> "z_No", "z_No", "Yes", "Yes", "Yes", "z_No", ~
$ Sex                   <chr> "M", "M", "z_F", "M", "z_F", "z_F", "z_F", "M~
$ Education             <chr> "PhD", "z_High School", "z_High School", "<Hi~
```

```
$ Job                      <chr> "Professional", "z_Blue Collar", "Clerical", ~
$ `Distance to Work`       <dbl> 14, 22, 5, 32, 36, 46, 33, 44, 34, 48, 15, 36~
$ car_use                  <chr> "Private", "Commercial", "Private", "Private"~
$ `Bluebook Value`         <chr> "$14,230", "$14,940", "$4,010", "$15,440", "$~
$ `Time as Customer`       <dbl> 11, 1, 4, 7, 1, 1, 1, 1, 1, 7, 1, 7, 7, 6, 1,~
$ `Car Type`               <chr> "Minivan", "Minivan", "z_SUV", "Minivan", "z_~
$ `Is Red Car`             <chr> "yes", "yes", "no", "yes", "no", "no", "no", ~
$ `Total Claims`           <chr> "$4,461", "$0", "$38,690", "$0", "$19,217", "~
$ `Claim Frequency`        <dbl> 2, 0, 2, 0, 2, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, ~
$ `License Revoked`        <chr> "No", "No", "No", "No", "Yes", "No", "No", "Y~
$ `Licences Record Points` <dbl> 3, 0, 3, 0, 3, 0, 0, 10, 0, 1, 0, 0, 3, 3, 3,~
$ `Car Age`                <dbl> 18, 1, 10, 6, 17, 7, 1, 7, 1, 17, 11, 1, 9, 1~
$ Geography                <chr> "Highly Urban/ Urban", "Highly Urban/ Urban",~
```

```r
training_set <- training_set |>
  mutate(
    across(
      .cols = c(`Income`, `Home Value`, `Bluebook Value`, `Total Claims`),
      .fns = parse_number
    )
  )

training_set <- training_set |>
  mutate(
    across(
      .cols = c(`Is Red Car`, `Crash Dummy`, `Marital Status`, "Sex"),
      .fns = as.factor
    )
  )
```

Check for missing values

To check for NA values, we are going to take the sum of every value matching `NA` across the entire dataframe and print the results. Then, replace all the `NA` values with the median value of the corresponding variable. The variables with the most `NA` observations are: `team_batting_hbp`, `team_baserun_cs`, and `team_fielding_dp`. Based on the percentages from `Table 1`, it does not appear that only columns have more than 10% of their observations missing.

```
# Sum NAs across columns, divide by length of column and get percent missing NAs per column
missing_vals <- training_set |>
  summarise(across(everything(), ~ sum(is.na(.) / length(.)))) |>
  pivot_longer(cols = where(is.numeric), names_to = "variable")

missing_vals |>
  filter(value > 0) |>
  gt::gt() |>
  tab_header(
    title = "Table 1: Percent of observations with missing values"
  ) |>
  cols_label(
    variable = "Variable",
    value = "Percent"
  ) |>
  fmt_percent(
    columns = value,
    decimals = 2
  )
```

Table 1: Percent of observations with missing values

Variable	Percent
Age of Driver	0.07%
Years on Job	5.56%
Income	5.45%
Home Value	5.69%
Job	6.45%
Car Age	6.25%

Appendix