Insurance Rate Column

Elizabeth Jones

5/11/2022

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

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

```
library(rpart)
library(rpart.plot)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

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

```
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:purrr':
##
##     cross
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
library(e1071)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(randomForest)
```

```
## randomForest 4.7-1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(smotefamily)
library(readr)
library(dplyr)
```

#CREATE MODEL

```
#IMPORT DATA
car_insurance_data <- read.csv("C:/Users/eljon/OneDrive/Desktop/IST 707/Final Project/car insura
nce data.csv")

#remove rows with NAs
insured<-na.omit(car_insurance_data)

#ADD Column with customer rank
r.insured<-insured %>% mutate (RANKED = case_when(PAST_ACCIDENTS == 0 ~ "good", PAST_ACCIDENTS =
= 1 ~"ok", PAST_ACCIDENTS > 1 ~ "bad"))

#REMOVE uneeded columns
d.insured<-r.insured[,-c(1,3,4,8)]

#Discrentize columns
d.insured<-d.insured %>% mutate (DUIS = case_when(DUIS == 0 ~ "0", DUIS >= 1 ~ "1"),
                                 SPEEDING_VIOLATIONS = case_when(SPEEDING_VIOLATIONS == 0 ~ "0",
SPEEDING_VIOLATIONS >= 1 ~ "1"))

#Factor Columns
f.insured = d.insured |> mutate_if(is.character, as.factor)
f.insured = f.insured |> mutate_if(is.numeric, as.factor)

#Create train & test dataset
set.seed(111)

trainList <-
  createDataPartition(y=f.insured$OUTCOME,p=0.4,list=FALSE)

insured.trainSet <-f.insured[trainList,]
insured.testSet <- f.insured[-trainList,]

#balance training set
s.insured<-DMwR::SMOTE(OUTCOME ~., insured.trainSet, perc.over = 100, perc.under = 200)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
#create loss matrix
loss2<-matrix(c(0,2,1,0),ncol=2)

#make model
third.model<-rpart(OUTCOME ~.,data = s.insured, method = 'class', parms = list(loss=loss2))
```

#ADD COLUMN TO DETERMINE PREDICTION

```
#create new dataframe
final.insured<-data.frame(f.insured)

#add columns
final.insured$PREDICTION<-predict(third.model,newdata=f.insured, type = "class")
```

#ADD COLUMN TO DETERMINE RATE

```
#Make column
final.insured$RATE <- ifelse(final.insured$PREDICTION == 0 & final.insured$RANKED == "good", "BE
ST RATE",ifelse(final.insured$PREDICTION == 0 & final.insured$RANKED == "ok", "GOOD RATE",
            ifelse(final.insured$PREDICTION == 0 & final.insured$RANKED == "bad", "OK RATE",
            ifelse(final.insured$PREDICTION == 1 & final.insured$RANKED == "good", "GOOD RATE",
            ifelse(final.insured$PREDICTION == 1 & final.insured$RANKED == "ok", "OK RATE",
            ifelse(final.insured$PREDICTION == 1 & final.insured$RANKED == "bad", "BAD RATE", "B
AD RATE"))))))
```

#View Dataset

```
glimpse(final.insured)
```

```
## Rows: 8,149
## Columns: 18
## $ AGE                 <fct> 65+, 16-25, 16-25, 16-25, 26-39, 40-64, 65+, 26-39~
## $ DRIVING_EXPERIENCE  <fct> 0-9y, 0-9y, 0-9y, 0-9y, 10-19y, 20-29y, 30y+, 0-9y~
## $ EDUCATION           <fct> high school, none, high school, university, none, ~
## $ INCOME              <fct> upper class, poverty, working class, working class~
## $ VEHICLE_OWNERSHIP   <fct> 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1,~
## $ VEHICLE_YEAR        <fct> after 2015, before 2015, before 2015, before 2015,~
## $ MARRIED             <fct> 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1,~
## $ CHILDREN            <fct> 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1,~
## $ POSTAL_CODE         <fct> 10238, 10238, 10238, 32765, 32765, 10238, 10238, 1~
## $ ANNUAL_MILEAGE      <fct> 12000, 16000, 11000, 11000, 12000, 13000, 13000, 1~
## $ VEHICLE_TYPE        <fct> sedan, sedan, sedan, sedan, sedan, sedan, sedan, s~
## $ SPEEDING_VIOLATIONS <fct> 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0,~
## $ DUIS                <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0,~
## $ PAST_ACCIDENTS      <fct> 0, 0, 0, 0, 1, 3, 3, 0, 0, 0, 7, 0, 2, 1, 0, 0, 0,~
## $ OUTCOME             <fct> 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1,~
## $ RANKED              <fct> good, good, good, good, ok, bad, bad, good, good, ~
## $ PREDICTION          <fct> 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0,~
## $ RATE                <chr> "BEST RATE", "GOOD RATE", "GOOD RATE", "GOOD RATE"~
```