Modulo 4 - Proyecto

Carlos Galindo

2024-05-19

Como primer paso, vamos a cargar todas las librerías que utilizaremos y posterirmente se realizará la carga de los datos train y test del documento y luego se procederá a realizar un análisis exploratorio de los datos.

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.2.3

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(ggplot2)  
library(rpart)  
library(rpart.plot)  
library(corrplot)

## corrplot 0.92 loaded

library(vcd)

## Loading required package: grid

library(vcdExtra)

## Loading required package: gnm

##   
## Attaching package: 'vcdExtra'

## The following object is masked from 'package:dplyr':  
##   
## summarise

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'lattice'

## The following object is masked from 'package:gnm':  
##   
## barley

library(weights)

## Loading required package: Hmisc

## Loading required package: survival

##   
## Attaching package: 'survival'

## The following object is masked from 'package:caret':  
##   
## cluster

## Loading required package: Formula

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':  
##   
## src, summarize

## The following objects are masked from 'package:base':  
##   
## format.pval, units

library(e1071)

##   
## Attaching package: 'e1071'

## The following object is masked from 'package:Hmisc':  
##   
## impute

library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library(randomForest)

## randomForest 4.7-1.1

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

##   
## Attaching package: 'randomForest'

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

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

train <- read.csv("train.csv", stringsAsFactors = F)  
test <- read.csv("test.csv", stringsAsFactors = F) #

str(train)

## 'data.frame': 381109 obs. of 12 variables:  
## $ id : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Gender : chr "Male" "Male" "Male" "Male" ...  
## $ Age : int 44 76 47 21 29 24 23 56 24 32 ...  
## $ Driving\_License : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Region\_Code : num 28 3 28 11 41 33 11 28 3 6 ...  
## $ Previously\_Insured : int 0 0 0 1 1 0 0 0 1 1 ...  
## $ Vehicle\_Age : chr "> 2 Years" "1-2 Year" "> 2 Years" "< 1 Year" ...  
## $ Vehicle\_Damage : chr "Yes" "No" "Yes" "No" ...  
## $ Annual\_Premium : num 40454 33536 38294 28619 27496 ...  
## $ Policy\_Sales\_Channel: num 26 26 26 152 152 160 152 26 152 152 ...  
## $ Vintage : int 217 183 27 203 39 176 249 72 28 80 ...  
## $ Response : int 1 0 1 0 0 0 0 1 0 0 ...

Se observa que en su mayoría los datos son categóricos como las variables Gender, Driving\_license, Previously\_Insured, Vehicle\_age, Vehicle\_Damage y Policy\_Sales\_channel. Las otras variables como la edad o Vintage (dias asociado) son enteros pero se podrían condensar en rangos.

summary(train)

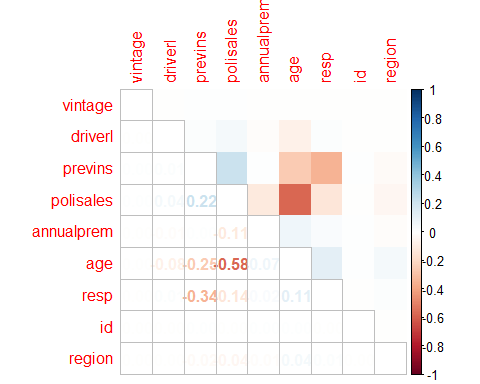
## id Gender Age Driving\_License   
## Min. : 1 Length:381109 Min. :20.00 Min. :0.0000   
## 1st Qu.: 95278 Class :character 1st Qu.:25.00 1st Qu.:1.0000   
## Median :190555 Mode :character Median :36.00 Median :1.0000   
## Mean :190555 Mean :38.82 Mean :0.9979   
## 3rd Qu.:285832 3rd Qu.:49.00 3rd Qu.:1.0000   
## Max. :381109 Max. :85.00 Max. :1.0000   
## Region\_Code Previously\_Insured Vehicle\_Age Vehicle\_Damage   
## Min. : 0.00 Min. :0.0000 Length:381109 Length:381109   
## 1st Qu.:15.00 1st Qu.:0.0000 Class :character Class :character   
## Median :28.00 Median :0.0000 Mode :character Mode :character   
## Mean :26.39 Mean :0.4582   
## 3rd Qu.:35.00 3rd Qu.:1.0000   
## Max. :52.00 Max. :1.0000   
## Annual\_Premium Policy\_Sales\_Channel Vintage Response   
## Min. : 2630 Min. : 1 Min. : 10.0 Min. :0.0000   
## 1st Qu.: 24405 1st Qu.: 29 1st Qu.: 82.0 1st Qu.:0.0000   
## Median : 31669 Median :133 Median :154.0 Median :0.0000   
## Mean : 30564 Mean :112 Mean :154.3 Mean :0.1226   
## 3rd Qu.: 39400 3rd Qu.:152 3rd Qu.:227.0 3rd Qu.:0.0000   
## Max. :540165 Max. :163 Max. :299.0 Max. :1.0000

Podemos ver el resumen de cada una de las variables. Vamos a renombrar algunas columnas para facilitar el script. esto se hará tanto en Train como en test. Ojo porque test no tiene la columna de response debido a que es el dataframe para evaluar las respuestas.

train = train %>%  
 rename(id = id, gender = Gender, age = Age, driverl = Driving\_License,   
 region = Region\_Code, previns = Previously\_Insured,   
 v\_age = Vehicle\_Age, v\_damage = Vehicle\_Damage,  
 annualprem = Annual\_Premium, polisales = Policy\_Sales\_Channel, vintage = Vintage, resp = Response)  
  
test = test %>%  
 rename(id = id, gender = Gender, age = Age, driverl = Driving\_License,   
 region = Region\_Code, previns = Previously\_Insured,   
 v\_age = Vehicle\_Age, v\_damage = Vehicle\_Damage,  
 annualprem = Annual\_Premium, polisales = Policy\_Sales\_Channel, vintage = Vintage)

a continuación analizaremos la relacion que puede haber entre las variables con la variable response. Primero veremos la relacion de las variables cuantitativas.

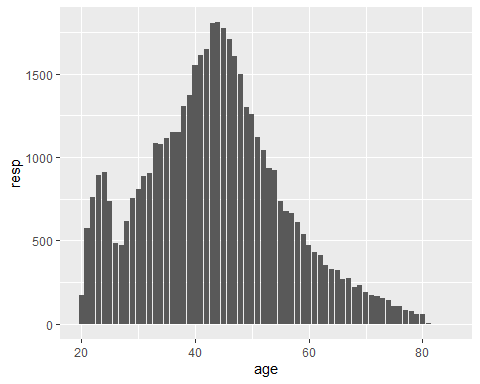
nums <- sapply(train, is.numeric)  
data\_cor = cor(train[,nums])  
View(data\_cor)  
  
corrplot.mixed(cor(train[,nums]),   
 lower="number",   
 upper="color", tl.pos="lt",  
 diag="n", order="hclust")



Podemos observar que las variables que más correlación tienen con la variable response en este caso es - previns (accidentes previos) y - polisales (política de venta). - La edad tiene una correlación pero es muy debil. nos enfocaremos en estas 3 variables

Edad vs Response

ggplot(train, aes(age, resp)) + geom\_col()

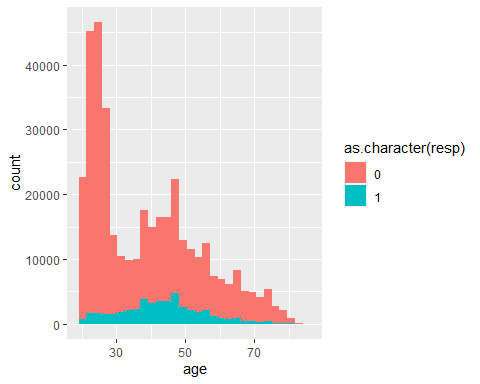


La distribución de la edad respecto a response parece tener una distribución normal. Esto podría ser un indicador de correlación.

Histograma de edad, color-coded por respuesta

ggplot(train,aes(x=age,fill=as.character(resp)))+geom\_histogram()

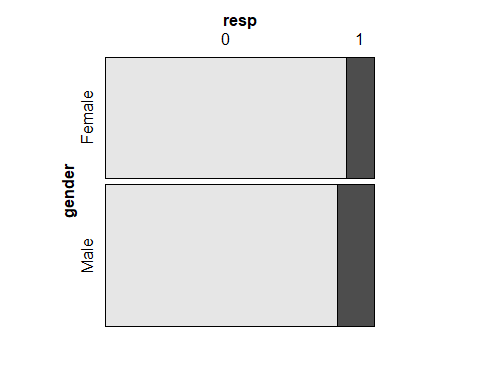
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



Se puede observar que hay diferencia entre la distribución de las edades en función de su respuesta. La diferencia principal es en las personas menores de 30 años, se observa claramente que para este rango de edad la respuesta positiva es muy baja (proporcionalmente a la cantidad de gente en ese rango). Para profundizar en la variable de edad la estaremos dividiendo en rangos de edad.

Genero vs Response

mosaic(resp~ gender, data = train)

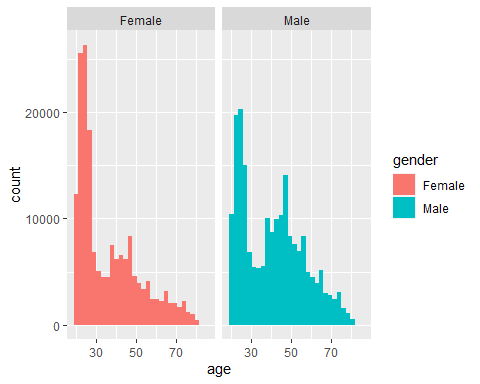
 Muchas personas no estuvieron interesadas pero de las personas que si lo estuvieron, fueron más hombres que mujeres.

prop.table(table(train$gender, train$resp),1)

##   
## 0 1  
## Female 0.8960976 0.1039024  
## Male 0.8615889 0.1384111

ggplot(train,aes(x=age,fill=gender))+geom\_histogram()+facet\_wrap(~gender)

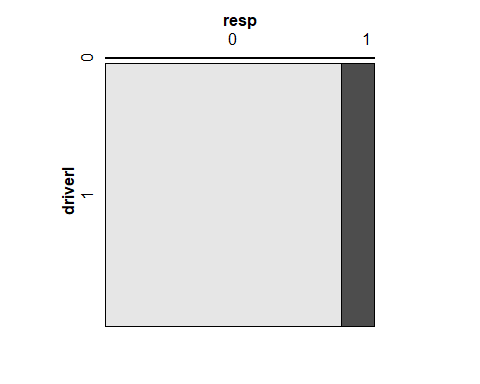
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



La distribución de edad en función del genero es muy similar y ademas las proporciones de la respuesta en función del genero tampoco son muy diferentes. Parce ser que el sexo no es de las variables con mayor relación a la respuesta.

Licencia vs response

mosaic(resp~ driverl, data = train)



table(train$driverl, train$resp)

##   
## 0 1  
## 0 771 41  
## 1 333628 46669

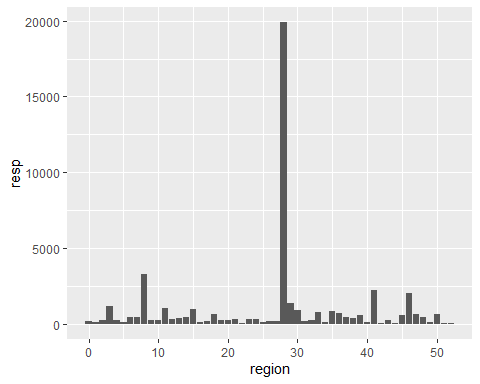
prop.table(table(train$driverl, train$resp),1)

##   
## 0 1  
## 0 0.94950739 0.05049261  
## 1 0.87728276 0.12271724

Por la diferencia en la cantidad de personas para quienes tienen licencia y no la tienen, es más util observar la tabla de proporciones. Podemos observar en la tabla que hay una mayor tendencia a no tener interés para quienes no tienen licencia.

Region vs response

ggplot(train, aes(region, resp)) + geom\_col()



Existen muchas regiones y con la información que tenemos no podemos agruparlas de alguna forma objetiva. Para verificar si puede llegar a ser una variable importante veremos si la proporción de respuesta varía mucho entre las regiones, lo veremos en la siguiente tabla:

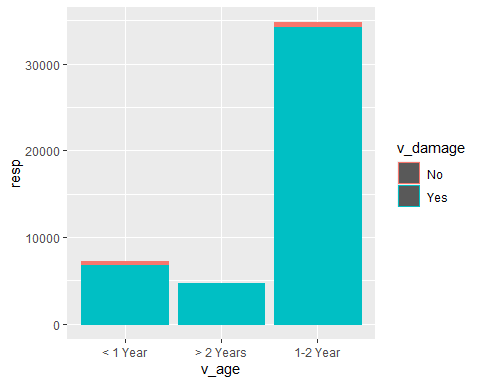
prop.table(table(train$region, train$resp),1)

##   
## 0 1  
## 0 0.91390401 0.08609599  
## 1 0.89186508 0.10813492  
## 2 0.92892521 0.07107479  
## 3 0.87233813 0.12766187  
## 4 0.84175458 0.15824542  
## 5 0.88428460 0.11571540  
## 6 0.93009554 0.06990446  
## 7 0.86825252 0.13174748  
## 8 0.90385807 0.09614193  
## 9 0.91841342 0.08158658  
## 10 0.93987197 0.06012803  
## 11 0.88724003 0.11275997  
## 12 0.90243902 0.09756098  
## 13 0.90981169 0.09018831  
## 14 0.90979051 0.09020949  
## 15 0.92801323 0.07198677  
## 16 0.93323368 0.06676632  
## 17 0.92969049 0.07030951  
## 18 0.86842616 0.13157384  
## 19 0.83713355 0.16286645  
## 20 0.88475452 0.11524548  
## 21 0.92897328 0.07102672  
## 22 0.93659282 0.06340718  
## 23 0.84693878 0.15306122  
## 24 0.85838509 0.14161491  
## 25 0.95725130 0.04274870  
## 26 0.92771550 0.07228450  
## 27 0.92596529 0.07403471  
## 28 0.81283654 0.18716346  
## 29 0.87663805 0.12336195  
## 30 0.92617505 0.07382495  
## 31 0.89948980 0.10051020  
## 32 0.91424471 0.08575529  
## 33 0.90070551 0.09929449  
## 34 0.92427885 0.07572115  
## 35 0.87539614 0.12460386  
## 36 0.92008639 0.07991361  
## 37 0.92074168 0.07925832  
## 38 0.80799605 0.19200395  
## 39 0.87639966 0.12360034  
## 40 0.88185328 0.11814672  
## 41 0.87822373 0.12177627  
## 42 0.92554992 0.07445008  
## 43 0.90754074 0.09245926  
## 44 0.95915842 0.04084158  
## 45 0.89027654 0.10972346  
## 46 0.89710871 0.10289129  
## 47 0.91231845 0.08768155  
## 48 0.89809870 0.10190130  
## 49 0.92521834 0.07478166  
## 50 0.93732305 0.06267695  
## 51 0.84699454 0.15300546  
## 52 0.87640449 0.12359551

Podemos observar que no hay ninguna región que sea muy diferente al resto, en cuanto a proporciones, por lo que podemos determinar que no se necesario separar algunas regiones ni dummificar a cada una.

Response == 1, edad del vehiculo

ggplot(train, aes(v\_age, resp, color = v\_damage)) + geom\_col()



Para la variable de antiguedad del carro, podemos ver que en su mayoría son propietarios de vehiculos de entre 1 y 2 años, le siguen los de menos de 1 año y por ultimo los que tienen un vehículo de más de 2 años. Por su lado, también podemos ver que la variable de daños previos también es significativa para la muestra.

Proporcion de respuesta por edad de vehículo

table(train$v\_age, train$resp)

##   
## 0 1  
## < 1 Year 157584 7202  
## > 2 Years 11305 4702  
## 1-2 Year 165510 34806

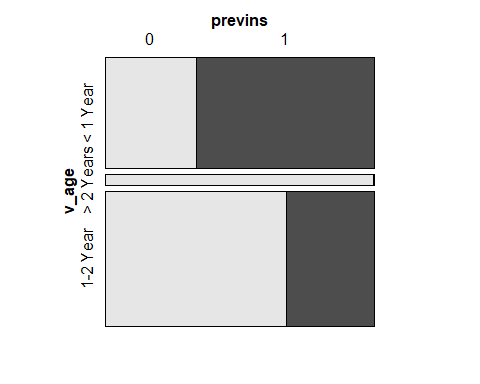
prop.table(table(train$v\_age, train$resp),1)

##   
## 0 1  
## < 1 Year 0.95629483 0.04370517  
## > 2 Years 0.70625351 0.29374649  
## 1-2 Year 0.82624453 0.17375547

Podemos observar que en proporcion, las personas con carros de antiguedad mayor a 2 años están más interesados en un seguro de vehículo. No obstante esto puede deberse a que los vehículos de menor antiguedad ya tienen un seguro de vehículo. Lo estaremos comprobando en el siguiente gráfico:

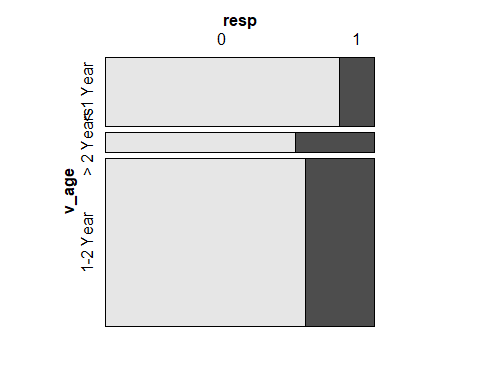
Previamente asegurado, edad de vehículo

mosaic(previns~ v\_age, data = train)



Efectivamente podemos comprobar que la diferencia en proporciones se debe a que los carros de más de 2 años de antiguedad casi no tienen seguro.

Sin\_Seguro = train %>%  
 filter(previns == 0)  
mosaic(resp~ v\_age, data = Sin\_Seguro)



Los clientes cuyo vehículo tiene una antiguedad mayor a 2 años, tienden a tener una respuesta más positiva (separando la muestra en quienes no tienen ya seguro de vehículo).

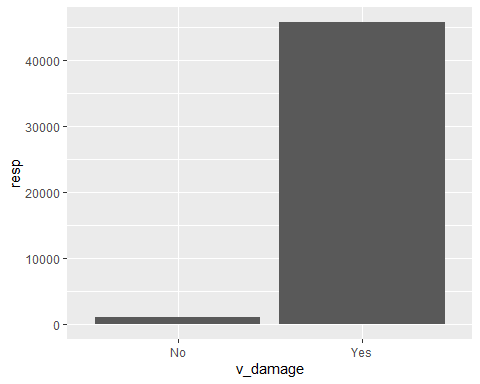
prop.table(table(Sin\_Seguro$v\_age, Sin\_Seguro$resp),1)

##   
## 0 1  
## < 1 Year 0.8714185 0.1285815  
## > 2 Years 0.7054696 0.2945304  
## 1-2 Year 0.7428643 0.2571357

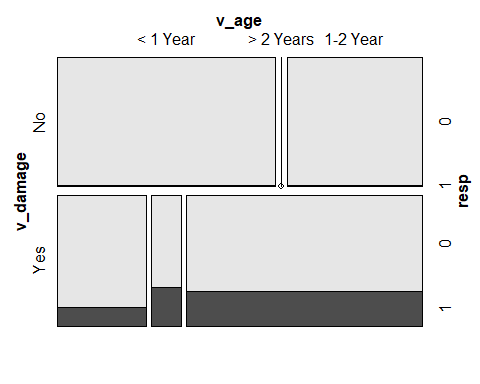
Podemos ver númericamente lo que nos indica el gráfico anterior.

Daños a vehículo

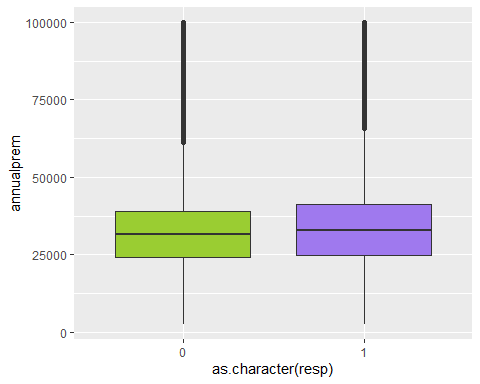
ggplot(train, aes(v\_damage, resp)) + geom\_col()



mosaic(resp~ v\_damage + v\_age, data = train)

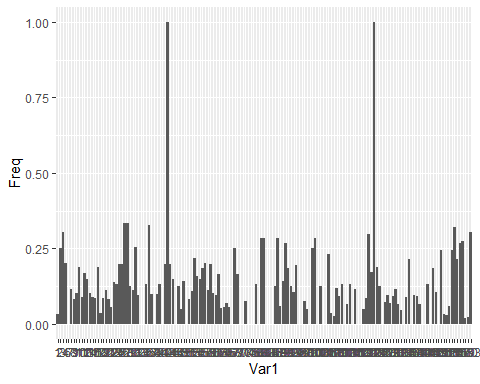
 Parece existir una relación entre las personas que tienen interés y que su vehículo haya sufrido un problema en el pasado.

train %>%  
 filter(annualprem < 1e+05) %>%  
 ggplot(aes(x= as.character(resp),   
 y= annualprem)) +  
 geom\_boxplot(fill=c("olivedrab3","mediumpurple2"))

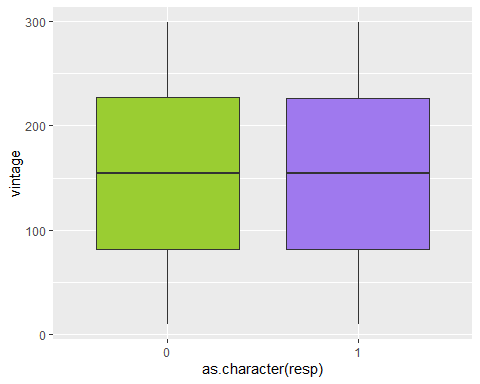


Se puede observar que las personas interesadas en un seguro de vehículo pagan una prima anual parecida a las personas que no están tan interesadas.

Prop\_polisales = as.data.frame(prop.table(table(train$polisales, train$resp),1))  
Prop\_polisales %>%  
 filter(Var2 == 1) %>%  
 arrange(desc(Freq)) %>%  
 ggplot(aes(x= Var1, y=Freq))+geom\_col()

 Podemos observar que si hay hay algunos canales de venta con mayor proporción, la desventaja es que desconocemos a que tipo de canal hace referencia el código y complica que los podamos agrupar en categorías para reducir el número.

ggplot(train, aes(x= as.character(resp),   
 y= vintage)) +  
 geom\_boxplot(fill=c("olivedrab3","mediumpurple2"))



La distribución del Vintage (días de asociación con la empresa) en función del interés del seguro parece ser muy similar, por lo que es muy posible que no sea una variable muy determinante.

#1. Relaciones entre variables que podrían afectar en la decisión del cliente.

Luego de analizar los datos, tomamos las siguientes variables como definitivas para generar el primer modelo: 1. previns 2. polisales 3. age 4. gender 5. v\_age 6. v\_damage

### HIPOTESIS

### Se plantea la hipótesis de que las variables con mayor impacto, según el análisis exploratorio, sobre el interés de una persona en contratar un seguro para su vehículo son las siguientes:

### - Previns, se observa que si la persona ya tiene seguro de vehículo no estará interesada en adquirir otro seguro del mismo tipo.

### - Edad, personas menores de 30 tienen mucho menos interés.

### - Daño al vehículo, para las personas que tienen un vehículo que ha recibido daño en el pasado, están más interesadas en adquirir el seguro.

### Se espera que existan variables que también esten relacionadas pero las 3 anteriores se espera que sean las que tengan un impacto mayor.

### ¿Qué proporción de personas que ya tienen seguro no están interesadas en adquirir otro producto del mismo tipo?

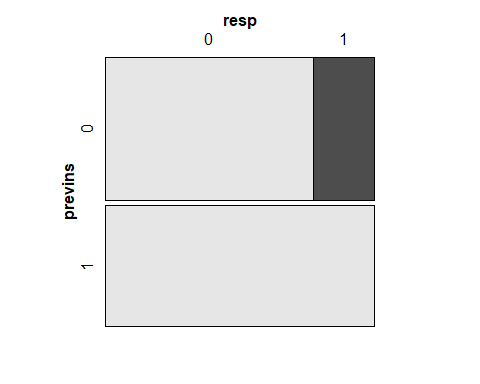
### Para las personas menores a 30 años: ¿Qué proporción no está interesada en adquirir seguro de vehículo y como se compara con otros rangos?

### ¿Qué proporción de la gente que ha dañado su vehículo está interesada en adquirir un seguro de vehículo y como se compara con la gente que no ha dañado su vehículo?

Respondemos las preguntas a continuación:

### ¿Qué proporción de personas que ya tienen seguro no están interesadas en adquirir otro producto del mismo tipo?

mosaic(resp~ previns, data = train)



table(train$previns, train$resp)

##   
## 0 1  
## 0 159929 46552  
## 1 174470 158

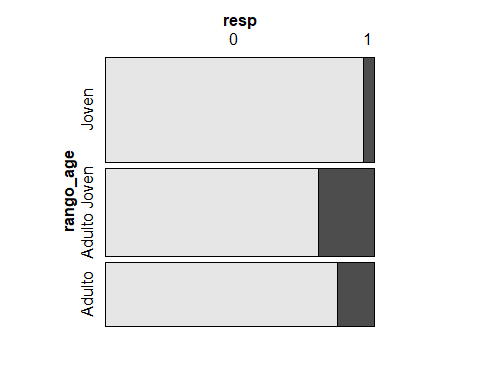
prop.table(table(train$previns, train$resp),1)

##   
## 0 1  
## 0 0.7745458420 0.2254541580  
## 1 0.9990952196 0.0009047804

Se puede observar que si la persona ya tiene seguro de vehículo, la probabilidad de que esté interesado en otro es casi nula. Es decir que esta variable es muy influyente.

###### Para las personas menores a 30 años: ¿Qué proporción no está interesada en adquirir seguro de vehículo y como se compara con otros rangos?

train = train %>%  
 mutate(rango\_age = case\_when(age < 30 ~ "Joven",  
 age < 50 ~ "Adulto Joven",  
 age < 90 ~ "Adulto")) %>%  
 mutate(rango\_age = factor(rango\_age,   
 levels = c("Joven", "Adulto Joven", "Adulto")))  
  
test = test %>%  
 mutate(rango\_age = case\_when(age < 30 ~ "Joven",  
 age < 50 ~ "Adulto Joven",  
 age < 90 ~ "Adulto")) %>%  
 mutate(rango\_age = factor(rango\_age,   
 levels = c("Joven", "Adulto Joven", "Adulto")))  
  
mosaic(resp~ rango\_age, data = train)



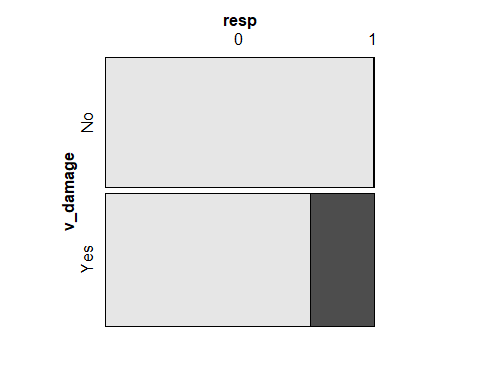
prop.table(table(train$rango\_age, train$resp),1)

##   
## 0 1  
## Joven 0.95893121 0.04106879  
## Adulto Joven 0.79271390 0.20728610  
## Adulto 0.86118114 0.13881886

Con lo anterior se comprueba que la gente mayor 30, tiene una mayor probabilidad a estar interesado en el seguro de automóvil.

### ¿Qué proporción de la gente que ha dañado su vehículo está interesada en adquirir un seguro de vehículo y como se compara con la gente que no ha dañado su vehículo?

mosaic(resp~ v\_damage , data = train)



prop.table(table(train$v\_damage, train$resp),1)

##   
## 0 1  
## No 0.994795862 0.005204138  
## Yes 0.762344540 0.237655460

Parece existir una relación entre las personas que tienen interés y que su vehículo haya sufrido un problema en el pasado.

### MODELOS SUPERVISADOS:

### MODELO DE NAIVE BAYES 1

set.seed(456)  
  
modeloBayes1 = naiveBayes(resp ~ previns + polisales + age + gender + v\_age + v\_damage, data = train)  
predNBtrain1 = predict(modeloBayes1, newdata = train, type="raw")  
  
predictionNBtrain1 = as.data.frame(predNBtrain1)  
predictionNBtrain1$pred1<-ifelse(predictionNBtrain1[,1]>predictionNBtrain1[,2],0,1)  
resultsNBtrain1<-table(train$resp, predictionNBtrain1$pred1)  
resultsNBtrain1

##   
## 0 1  
## 0 231507 102892  
## 1 6152 40558

Podemos hacer el primer modelo utilizando las variables previamente mencionadas y luego, al compararlas con el resultado real obtenemos las siguientes proporciones:

prop.table(resultsNBtrain1)

##   
## 0 1  
## 0 0.60745613 0.26998050  
## 1 0.01614236 0.10642100

Se acertó en un 61% de los casos cuando el usuario NO estaba interesado y en un 11% cuando si estaba interesado. Sin embargo los falsos negativos y falsos positivos fueron en total un 28% restante lo cual puede considerarse todavía bastante. Vamos a calcular ahora los valores de accuracy, recall y precision:

accuracyNBtrain1<-sum(diag(resultsNBtrain1))/sum(resultsNBtrain1)  
accuracyNBtrain1

## [1] 0.7138771

Un accuracy del 71%

recallNBtrain1<-(resultsNBtrain1[2,2]/(resultsNBtrain1[2,1]+resultsNBtrain1[2,2]))  
recallNBtrain1

## [1] 0.8682937

Un recall del 87%

precisionBNtrain1<-(resultsNBtrain1[2,2]/(resultsNBtrain1[1,2]+resultsNBtrain1[2,2]))  
precisionBNtrain1

## [1] 0.2827327

Un precision del 28%

El modelo nos dice entonces que no es muy confiable por la precisión, esto puede ser un problema debido a que queremos saber exactamente quienes si están interesados para tener un mejor mercado objetivo.

Ahora vamos a explorar el modelo pero restringiendo la variable de edad que era la que menor correlación tenía con la variable dependiente.

modeloBayes2 = naiveBayes(resp ~ previns + polisales + gender + v\_age + v\_damage, data = train)  
predNBtrain2 = predict(modeloBayes2, newdata = train, type="raw")  
  
predictionNBtrain2 = as.data.frame(predNBtrain2)  
predictionNBtrain2$pred2<-ifelse(predictionNBtrain2[,1]>predictionNBtrain2[,2],0,1)  
resultsNBtrain2<-table(train$resp, predictionNBtrain2$pred2)  
resultsNBtrain2

##   
## 0 1  
## 0 215630 118769  
## 1 3205 43505

Obtenemos entonces la proporción de los resultados del modelo 2:

prop.table(resultsNBtrain2)

##   
## 0 1  
## 0 0.565796137 0.311640502  
## 1 0.008409668 0.114153694

Ahora obtenemos los datos de accuracy, recall y precision:

accuracyNBtrain2<-sum(diag(resultsNBtrain2))/sum(resultsNBtrain2)  
accuracyNBtrain2

## [1] 0.6799498

Un accuracy del 67%.

recallNBtrain2<-(resultsNBtrain2[2,2]/(resultsNBtrain2[2,1]+resultsNBtrain2[2,2]))  
recallNBtrain2

## [1] 0.9313851

Un recall del 91%

precisionBNtrain2<-(resultsNBtrain2[2,2]/(resultsNBtrain2[1,2]+resultsNBtrain2[2,2]))  
precisionBNtrain2

## [1] 0.2680959

Un precision del 27%

para este caso, el modelo bajó tanto en su nivel de precision como de accuracy, por lo cual no es conveniente restringir la variable de edad ya que solamente el valor del recall aumentó pero no es eso lo que en verdad buscamos.

###—————————–### ### MODELOS DE DECISITION TREE 1 ###—————————–###

set.seed(456)  
inTrain <- sample(nrow(train), 300, replace = FALSE)  
train\_2 <- train[inTrain,]  
test\_2 <- train[-inTrain,]  
prop.table(table(train$resp))

##   
## 0 1   
## 0.8774366 0.1225634

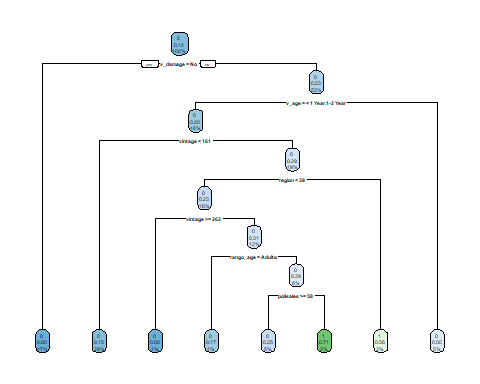
Se creó una muestra con 300 observaciones debido a que se probaron los modelos de árboles de decisión y estos parecen colapsar con muchas observaciones, se tomo la decisión de trabajar con una muestra pequeña y el resto dejarlo para testear el modelo.

modeloDT1<-rpart(resp~.-age-id, data=train\_2,method="class") # recursive partitioning tree  
modeloDT1

## n= 300   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 300 36 0 (0.8800000 0.1200000)   
## 2) v\_damage=No 142 0 0 (1.0000000 0.0000000) \*  
## 3) v\_damage=Yes 158 36 0 (0.7721519 0.2278481)   
## 6) v\_age=< 1 Year,1-2 Year 144 29 0 (0.7986111 0.2013889)   
## 12) vintage< 160.5 88 13 0 (0.8522727 0.1477273) \*  
## 13) vintage>=160.5 56 16 0 (0.7142857 0.2857143)   
## 26) region< 39 47 11 0 (0.7659574 0.2340426)   
## 52) vintage>=263 12 0 0 (1.0000000 0.0000000) \*  
## 53) vintage< 263 35 11 0 (0.6857143 0.3142857)   
## 106) rango\_age=Adulto 12 2 0 (0.8333333 0.1666667) \*  
## 107) rango\_age=Joven,Adulto Joven 23 9 0 (0.6086957 0.3913043)   
## 214) polisales>=58 16 4 0 (0.7500000 0.2500000) \*  
## 215) polisales< 58 7 2 1 (0.2857143 0.7142857) \*  
## 27) region>=39 9 4 1 (0.4444444 0.5555556) \*  
## 7) v\_age=> 2 Years 14 7 0 (0.5000000 0.5000000) \*

Imprimimos el modelo en función de todas las variables menos el id y el age (importante tomar en cuenta que se está tomando la edad pero en ragos anteriormente determinado)

rpart.plot(modeloDT1)

 # El modelo 1 (Decision Tree) de forma visual.

prediDT1<-predict(modeloDT1, newdata=test\_2, type="class")  
prediDT1 = as.character(prediDT1)  
prediDT1 = ifelse(prediDT1 == 1,"SI","NO")  
resultadoDT1<-table(test\_2$resp, prediDT1)  
resultadoDT1

## prediDT1  
## NO SI  
## 0 315359 18776  
## 1 40433 6241

Estos son los resultados del modelo respecto al set de testeo que tiene 380,809 observaciones. #Calculo el acurracy

accuracyDT1<-sum(diag(resultadoDT1))/sum(resultadoDT1)  
accuracyDT1

## [1] 0.8445179

Tenemos un accuracy alto, este nos dice las observaciones correctas que se tuvieron ya sean positivas o negativas.

#TP/(TP+FN)

recallDT1<-(resultadoDT1[2,2]/(resultadoDT1[2,1]+resultadoDT1[2,2]))  
recallDT1

## [1] 0.1337147

Tenemos un recall bajo, mi modelo dice en muchas ocasiones que el cliente no está interesado en el seguro pero realmente si lo está. El costo de equivocarse es alto ya que no estamos podiendo identificar de manera efectiva a la gente que si está interesada para poder eplicar estrategias específicas para ellos.

#TP/(TP+FP)

precisionDT1<-(resultadoDT1[2,2]/(resultadoDT1[1,2]+resultadoDT1[2,2]))  
precisionDT1

## [1] 0.2494704

El precision también es bajo, en este indicador el modelo nos indica que si hay interés cuando realmente no lo hay. el costo de un presicion bajo es invertir recursos en clientes que definitamente no están interesados.

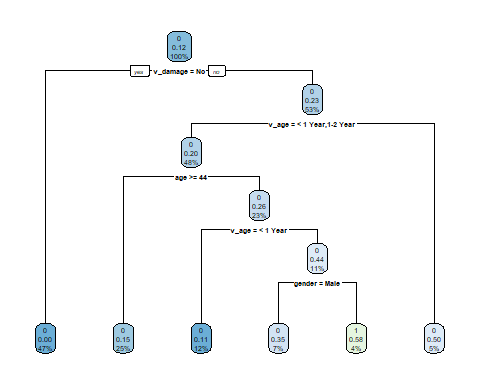
###—————————–### ### MODELOS DE DECISITION TREE 2 ###—————————–###

modeloDT2 <-rpart(formula = resp~previns + age + gender + v\_age + v\_damage+driverl, method="class", data=train\_2)  
modeloDT2

## n= 300   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 300 36 0 (0.8800000 0.1200000)   
## 2) v\_damage=No 142 0 0 (1.0000000 0.0000000) \*  
## 3) v\_damage=Yes 158 36 0 (0.7721519 0.2278481)   
## 6) v\_age=< 1 Year,1-2 Year 144 29 0 (0.7986111 0.2013889)   
## 12) age>=43.5 75 11 0 (0.8533333 0.1466667) \*  
## 13) age< 43.5 69 18 0 (0.7391304 0.2608696)   
## 26) v\_age=< 1 Year 37 4 0 (0.8918919 0.1081081) \*  
## 27) v\_age=1-2 Year 32 14 0 (0.5625000 0.4375000)   
## 54) gender=Male 20 7 0 (0.6500000 0.3500000) \*  
## 55) gender=Female 12 5 1 (0.4166667 0.5833333) \*  
## 7) v\_age=> 2 Years 14 7 0 (0.5000000 0.5000000) \*

Imprimimos el modelo de en función de previns + age + gender + v\_age + v\_damage+driverl, se decidió tomar la edad sin rangos para que modelo sea el que indique las divisiones de forma óptima. La idea también es incluir solo las variables que se mostraron significativas en el análisis exploratorio.

rpart.plot(modeloDT2)

 El modelo 2 (Decision Tree) de forma visual.

prediDT2<-predict(modeloDT2, newdata=test\_2, type="class")  
prediDT2 = as.character(prediDT2)  
prediDT2 = ifelse(prediDT2 == 1,"SI","NO")  
resultadoDT2<-table(test\_2$resp, prediDT2)  
resultadoDT2

## prediDT2  
## NO SI  
## 0 320276 13859  
## 1 40075 6599

Estos son los resultados del modelo respecto al set de testeo que tiene 380,809 observaciones.

#Calculo el acurracy

accuracyDT2<-sum(diag(resultadoDT2))/sum(resultadoDT2)  
accuracyDT2

## [1] 0.8583699

Tenemos un accuracy alto, este nos dice las observaciones correctas que se tuvieron ya sean positivas o negativas. Es ligeramente mayor al modelo anterior. #TP/(TP+FN)

recallDT2<-(resultadoDT2[2,2]/(resultadoDT2[2,1]+resultadoDT2[2,2]))  
recallDT2

## [1] 0.1413849

Tenemos un recall ligeramente más alto que en el modelo anterior, logrando reducir el costo de no indentificar a los clientes que realmente si están interesados. #TP/(TP+FP)

precisionDT2<-(resultadoDT2[2,2]/(resultadoDT2[1,2]+resultadoDT2[2,2]))  
precisionDT2

## [1] 0.3225633

Tenemos un presicion significativamente más alto que en el modelo anterior, logrando reducir el costo de recursos invertidos en clientes que no están interesados en el seguro.

###—————————–### ### MODELOS DE RANDOM FOREST 1 ###—————————–###

modelF1 <- randomForest(resp ~ .-age-id , importance = F, type = "class", data=train\_2)

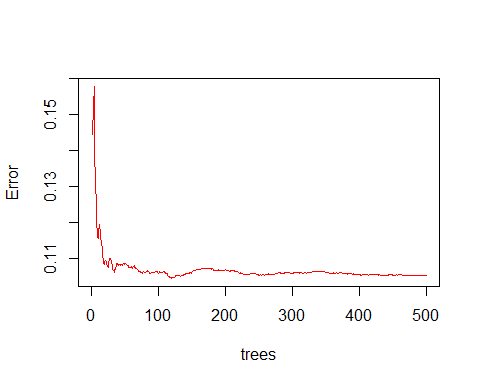
## Warning in randomForest.default(m, y, ...): The response has five or fewer  
## unique values. Are you sure you want to do regression?

modelF1

##   
## Call:  
## randomForest(formula = resp ~ . - age - id, data = train\_2, importance = F, type = "class")   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## Mean of squared residuals: 0.1051149  
## % Var explained: 0.46

Imprimos el modelo 1 de Random Forest en función de todas las variables menos el id y el age (importante tomar en cuenta que se está tomando la edad pero en rangos anteriormente determinados)

plot(modelF1,type="l",col="red",main="")

 Podemos observar que el modelo se estabiliza a partir de los 200 árboles aproximadamente.

predRDF1 <- predict(modelF1, newdata = test\_2, type = "class")  
predRD2F1 = ifelse(predRDF1 > 0.5,"SI","NO")  
resultsDTF1<-table(test\_2$resp, predRD2F1)  
resultsDTF1

## predRD2F1  
## NO SI  
## 0 331057 3078  
## 1 44899 1775

Estos son los resultados del modelo respecto al set de testeo que tiene 380,809 observaciones.

prop.table(resultsDTF1)

## predRD2F1  
## NO SI  
## 0 0.869351827 0.008082792  
## 1 0.117904251 0.004661129

Un vistazo a las proporciones según los resultados obtenidos. #ACCURACY

accuracyDTF1<-sum(diag(resultsDTF1))/sum(resultsDTF1)  
accuracyDTF1

## [1] 0.874013

Tenemos un accuracy alto, este nos dice las observaciones correctas que se tuvieron ya sean positivas o negativas.

#TP/(TP+FN)

recallF1<-(resultsDTF1[2,2]/(resultsDTF1[2,1]+resultsDTF1[2,2]))  
recallF1

## [1] 0.03802974

Tenemos un recall muy bajo, mi modelo dice en muchas ocasiones que el cliente no está interesado en el seguro pero realmente si lo está. El costo de equivocarse es alto ya que no estamos podiendo idenficar de manera efectiva a la gente que si está interesada para poder eplicar estrategias específicas para ellos.

#TP/(TP+FP)

precisionF1<-(resultsDTF1[2,2]/(resultsDTF1[1,2]+resultsDTF1[2,2]))  
precisionF1

## [1] 0.3657531

El precision también es relativamente bajo, en este indicador el modelo nos indica que si hay interés cuando realmente no lo hay. el costo de un presicion bajo es invertir recursos en clientes que definitamente no están interesados.

###—————————–### ### MODELOS DE RANDOM FOREST 2 ###—————————–###

modelF2 <- randomForest(resp ~ previns + age + gender + v\_age + v\_damage+driverl , importance = F, type = "class",data=train\_2)

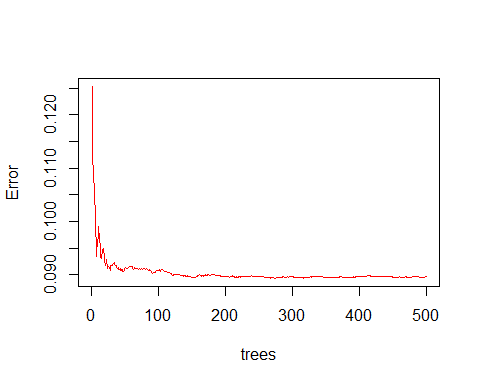
## Warning in randomForest.default(m, y, ...): The response has five or fewer  
## unique values. Are you sure you want to do regression?

modelF2

##   
## Call:  
## randomForest(formula = resp ~ previns + age + gender + v\_age + v\_damage + driverl, data = train\_2, importance = F, type = "class")   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 2  
##   
## Mean of squared residuals: 0.08961063  
## % Var explained: 15.14

Imprimos el modelo 1 de Random Forest en función de todas las variables previns + age + gender + v\_age + v\_damage+driverl, se decidió tomar la edad sin rangos para que modelo sea el que indique las divisiones de forma óptima. La idea también es incluir solo las variables que se mostraron significativas en el análisis exploratorio.

plot(modelF2,type="l",col="red",main="") #Grafica recomendada



#par(new=TRUE)  
#plot(modelF2,type="l",col="green", main="Modelos")

El modelo parece estabilizarse a partir de los 150 árboles.

predRDF2 <- predict(modelF2, newdata = test\_2, type = "class")  
predRD2F2 = ifelse(predRDF2 > 0.5,"SI","NO")  
resultsDTF2<-table(test\_2$resp, predRD2F2)  
resultsDTF2

## predRD2F2  
## NO SI  
## 0 330131 4004  
## 1 44423 2251

Imprimos los resultados del modelo comparados al set de testo.

prop.table(resultsDTF2)

## predRD2F2  
## NO SI  
## 0 0.86692016 0.01051446  
## 1 0.11665428 0.00591110

Observamos las proporciones de los resultados.

accuracyDTF2<-sum(diag(resultsDTF2))/sum(resultsDTF2)  
accuracyDTF2

## [1] 0.8728313

Tenemos un accuracy alto y muy ligeramente por debajo del modelo anterior.

#TP/(TP+FN)  
recallF2<-(resultsDTF2[2,2]/(resultsDTF2[2,1]+resultsDTF2[2,2]))  
recallF2

## [1] 0.04822814

El recall sigue siendo muy bajo pero ligeramente por arriba del modelo anteiror.

#TP/(TP+FP)  
precisionF2<-(resultsDTF2[2,2]/(resultsDTF2[1,2]+resultsDTF2[2,2]))  
precisionF2

## [1] 0.3598721

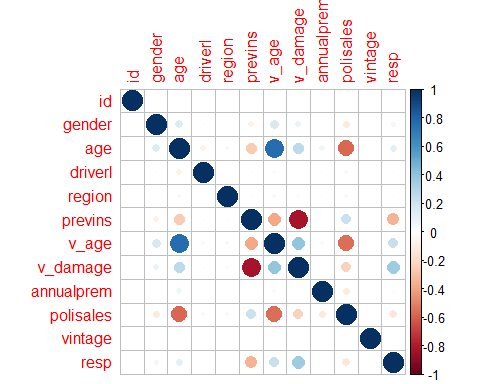
EL precision es ligeramente por debajo del modelo anterior, podriamos decir que son modelos con muy poca diferencia en cuanto a estos indicadores.

### Regresion Logistica

trainRL <- train  
testRL <- test  
  
### cambiamos las variables categóricas a numéricas  
  
trainRL$gender <- ifelse(trainRL$gender == "Male", 1, 0)  
trainRL$v\_damage <- ifelse(trainRL$v\_damage == "Yes", 1, 0)  
trainRL$v\_age <- ifelse(trainRL$v\_age == "> 2 Years", 2, ifelse(trainRL$v\_age == "1-2 Year", 1, 0))  
str(trainRL)

## 'data.frame': 381109 obs. of 13 variables:  
## $ id : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ gender : num 1 1 1 1 0 0 1 0 0 0 ...  
## $ age : int 44 76 47 21 29 24 23 56 24 32 ...  
## $ driverl : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ region : num 28 3 28 11 41 33 11 28 3 6 ...  
## $ previns : int 0 0 0 1 1 0 0 0 1 1 ...  
## $ v\_age : num 2 1 2 0 0 0 0 1 0 0 ...  
## $ v\_damage : num 1 0 1 0 0 1 1 1 0 0 ...  
## $ annualprem: num 40454 33536 38294 28619 27496 ...  
## $ polisales : num 26 26 26 152 152 160 152 26 152 152 ...  
## $ vintage : int 217 183 27 203 39 176 249 72 28 80 ...  
## $ resp : int 1 0 1 0 0 0 0 1 0 0 ...  
## $ rango\_age : Factor w/ 3 levels "Joven","Adulto Joven",..: 2 3 2 1 1 1 1 3 1 2 ...

trainRL$rango\_age = NULL  
corrplot(cor(trainRL))



set.seed(123)  
trainRL\_ind <- sample(nrow(trainRL), 0.7\*nrow(trainRL))  
  
trainRL\_train <- trainRL[trainRL\_ind,]  
trainRL\_test <- trainRL[-trainRL\_ind,]  
  
modeloRL\_01 <- glm(resp ~.,data = trainRL\_train, family=binomial)  
summary(modeloRL\_01) ## podemos ver que las variables ID, vintage, y region no

##   
## Call:  
## glm(formula = resp ~ ., family = binomial, data = trainRL\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3163 -0.6619 -0.0416 -0.0320 3.9206   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.025e+00 2.126e-01 -18.932 < 2e-16 \*\*\*  
## id -6.304e-08 5.860e-08 -1.076 0.282036   
## gender 1.041e-01 1.326e-02 7.850 4.16e-15 \*\*\*  
## age -2.009e-02 6.271e-04 -32.040 < 2e-16 \*\*\*  
## driverl 1.365e+00 2.046e-01 6.673 2.51e-11 \*\*\*  
## region 2.795e-04 5.182e-04 0.539 0.589598   
## previns -4.033e+00 9.998e-02 -40.339 < 2e-16 \*\*\*  
## v\_age 6.916e-01 1.550e-02 44.606 < 2e-16 \*\*\*  
## v\_damage 2.014e+00 4.070e-02 49.489 < 2e-16 \*\*\*  
## annualprem 1.343e-06 3.535e-07 3.799 0.000145 \*\*\*  
## polisales -2.878e-03 1.302e-04 -22.095 < 2e-16 \*\*\*  
## vintage -8.379e-05 7.714e-05 -1.086 0.277362   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 198057 on 266775 degrees of freedom  
## Residual deviance: 147330 on 266764 degrees of freedom  
## AIC: 147354  
##   
## Number of Fisher Scoring iterations: 9

## son estadisticamente significativas

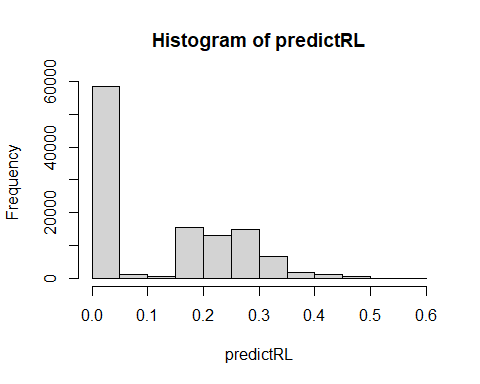
Se estableció un set.seed para hacer reproducibles los resultados obtenidos en el modelo, y se creó un modelo de regresión logística que busca predecir la variable response en función de todas las variables. Al ver el resumen del modelo podemos identificar que las variables ID, vintage y region no son estadísticamente significativas, por lo que las sacaremos y crearemos otro modelo.

modeloRL\_02 <- glm(resp ~ gender+age+driverl+previns+v\_age+v\_damage+annualprem+polisales, data = trainRL\_train, family = binomial)  
  
summary(modeloRL\_02)

##   
## Call:  
## glm(formula = resp ~ gender + age + driverl + previns + v\_age +   
## v\_damage + annualprem + polisales, family = binomial, data = trainRL\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3165 -0.6620 -0.0416 -0.0320 3.9211   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.043e+00 2.115e-01 -19.115 < 2e-16 \*\*\*  
## gender 1.040e-01 1.326e-02 7.845 4.33e-15 \*\*\*  
## age -2.009e-02 6.270e-04 -32.034 < 2e-16 \*\*\*  
## driverl 1.366e+00 2.046e-01 6.677 2.44e-11 \*\*\*  
## previns -4.033e+00 9.998e-02 -40.339 < 2e-16 \*\*\*  
## v\_age 6.916e-01 1.550e-02 44.606 < 2e-16 \*\*\*  
## v\_damage 2.014e+00 4.070e-02 49.493 < 2e-16 \*\*\*  
## annualprem 1.340e-06 3.535e-07 3.791 0.00015 \*\*\*  
## polisales -2.879e-03 1.302e-04 -22.114 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 198057 on 266775 degrees of freedom  
## Residual deviance: 147333 on 266767 degrees of freedom  
## AIC: 147351  
##   
## Number of Fisher Scoring iterations: 9

Vemos ahora que las variables tomadas en cuenta para el modelo son estadísticamente signififcativas, por lo que procederemos con la predicción de este primer modelo de regresión logística.

predictRL <- predict(modeloRL\_02, newdata = trainRL\_test, type = "response")  
hist(predictRL)



### vemos en el histograma que el umbral de probabilidad predecida para saber  
### si alguien optara por un seguro vehicular, no es mayor a 45% por lo que para este  
### caso utilizaremos un umbral del 0.25 para predecir si alguien comprara el seguro

Al correr la función de “predict” y realizar un histograma con el resultado, podemos ver que el umbral de probabilidad predicha para saber si alguien optará por un seguro vehicular no es mayor a 45%, por lo que para este caso de negocio se utilizará un umbral del 0.25 para la predicción de compras de seguros.

predictRL <- as.data.frame(predictRL)  
  
predictRL$respuesta <- ifelse(predictRL > 0.25,"SI","NO")  
  
results\_testRL<-table(trainRL\_test$resp, predictRL$respuesta)  
results\_testRL

##   
## NO SI  
## 0 82824 17388  
## 1 5929 8192

prop.table(results\_testRL)

##   
## NO SI  
## 0 0.72441028 0.15208208  
## 1 0.05185729 0.07165035

Las tablas anteriores representan a los clientes interesados en conseguir un seguro vehicular según nuestro modelo de regresión logísitca, comparados con los casos reales. De igual forma podemos ver una tabla de proporciones de nuestros resultados.

Se procederá ahora con el cálculo del accuracy, recall y precision de nuestro primer modelo de regresión logística.

### Calculamos accuracy  
  
accuracyRL\_01 <-sum(diag(results\_testRL))/sum(results\_testRL)  
print("Accuracy Regresión Logística 01")

## [1] "Accuracy Regresión Logística 01"

accuracyRL\_01 ## 79%

## [1] 0.7960606

### Calculamos recall  
  
recallRL\_01<-(results\_testRL[2,2]/(results\_testRL[2,1]+results\_testRL[2,2]))  
print("Recall Regresión Logística 01")

## [1] "Recall Regresión Logística 01"

recallRL\_01 ## 58%

## [1] 0.5801289

### Calculamos precision  
  
precisionRL\_01<-(results\_testRL[2,2]/(results\_testRL[1,2]+results\_testRL[2,2]))  
print("Precision Regresión Logística 01")

## [1] "Precision Regresión Logística 01"

precisionRL\_01 ## 32%

## [1] 0.3202502

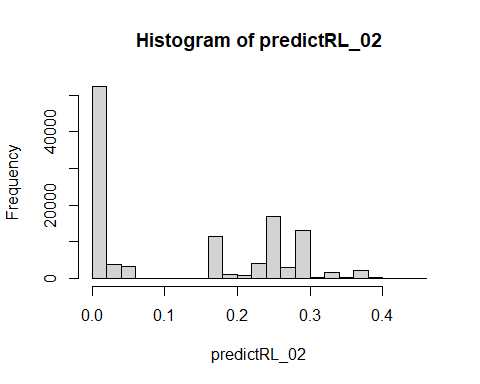
Se puede ver que nuestro modelo tuvo una exactitud del 79% en los resultados predichos, siendo mejor que ambos modelos de Naive Bayes. La precisión de este modelo también fue ligeramente mejor, con un 32%. Sin embargo, el recall fue significativamente menor en contraste al segundo modelo de Naive Bayes.

Se llevará a cabo ahora otro modelo de regresión logística, en el cual únicamente tomaremos en cuenta las siguientes variables: previns, v\_age, v\_damage, annualprem, y polisales.

modeloRL\_03 <- glm(resp ~ previns+v\_age+v\_damage+annualprem+polisales, data = trainRL\_train, family = binomial)  
  
summary(modeloRL\_03)

##   
## Call:  
## glm(formula = resp ~ previns + v\_age + v\_damage + annualprem +   
## polisales, family = binomial, data = trainRL\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0619 -0.7323 -0.0427 -0.0314 3.9094   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.348e+00 4.528e-02 -73.942 <2e-16 \*\*\*  
## previns -4.042e+00 9.994e-02 -40.446 <2e-16 \*\*\*  
## v\_age 4.107e-01 1.283e-02 32.020 <2e-16 \*\*\*  
## v\_damage 2.016e+00 4.065e-02 49.587 <2e-16 \*\*\*  
## annualprem 9.738e-07 3.507e-07 2.776 0.0055 \*\*   
## polisales -1.670e-03 1.240e-04 -13.462 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 198057 on 266775 degrees of freedom  
## Residual deviance: 148550 on 266770 degrees of freedom  
## AIC: 148562  
##   
## Number of Fisher Scoring iterations: 9

predictRL\_02 <- predict(modeloRL\_03, newdata = trainRL\_test, type = "response")  
hist(predictRL\_02)



### vemos en el histograma que el umbral de probabilidad predecida para saber  
### si alguien optara por un seguro vehicular, no es mayor a 45% por lo que para este  
### caso utilizaremos un umbral del 0.25 para predecir si alguien comprara el seguro

Con base en el histograma generado, nuevamente se utilizará un umbral del 0.25 para predecir si alguien comprará el seguro vehicular. Los resultados de este segundo modelo de regresión logística, y sus proporciones, son los siguientes:

predictRL\_02 <- as.data.frame(predictRL\_02)  
  
predictRL\_02$respuesta <- ifelse(predictRL\_02 > 0.25,"SI","NO")  
  
results\_testRL\_02<-table(trainRL\_test$resp, predictRL\_02$respuesta)  
results\_testRL\_02

##   
## NO SI  
## 0 79040 21172  
## 1 5769 8352

prop.table(results\_testRL\_02)

##   
## NO SI  
## 0 0.69131397 0.18517838  
## 1 0.05045787 0.07304978

Procedemos ahora a calcular el accuracy, recall y precision del segundo modelo de regresión.

### Calculamos accuracy  
  
accuracyRL\_02 <-sum(diag(results\_testRL\_02))/sum(results\_testRL\_02)  
print("Accuracy Regresión Logística 02")

## [1] "Accuracy Regresión Logística 02"

accuracyRL\_02 ## 76%

## [1] 0.7643637

### Calculamos recall  
  
recallRL\_02<-(results\_testRL\_02[2,2]/(results\_testRL\_02[2,1]+results\_testRL\_02[2,2]))  
print("Recall Regresión Logística 02")

## [1] "Recall Regresión Logística 02"

recallRL\_02 ## 59%

## [1] 0.5914595

### Calculamos precision  
  
precisionRL\_02<-(results\_testRL\_02[2,2]/(results\_testRL\_02[1,2]+results\_testRL\_02[2,2]))  
print("Precision Regresión Logística 02")

## [1] "Precision Regresión Logística 02"

precisionRL\_02 ## 28%

## [1] 0.2828885

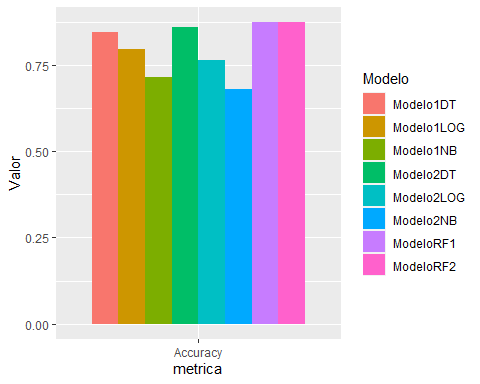
Podemos observar que nuestro accuracy y precision fueron menores, en 3 y 4 puntos porcentuales respectivamente. Únicamente resultó siendo mejor el recall, aunque por solamente un punto porcentual en comparación al modelo de regresión logística anterior.

### Comparacion de modelos

#Comparacion final  
  
titulos\_modelos = c("Modelo1DT", "Modelo2DT", "Modelo1NB", "Modelo2NB", "Modelo1LOG", "Modelo2LOG","ModeloRF1","ModeloRF2")  
Accuracy = c(round(accuracyDT1,4),   
 round(accuracyDT2,4),  
 round(accuracyNBtrain1,4),  
 round(accuracyNBtrain2,4),  
 round(accuracyRL\_01,4),  
 round(accuracyRL\_02,4),  
 round(accuracyDTF1,4),  
 round(accuracyDTF2,4))  
Recall = c(round(recallDT1,4),   
 round(recallDT2,4),  
 round(recallNBtrain1,4),  
 round(recallNBtrain2,4),  
 round(recallRL\_01,4),  
 round(recallRL\_02,4),  
 round(recallF1,4),  
 round(recallF2,4))  
Precision = c(round(precisionDT1,4),   
 round(precisionDT2,4),  
 round(precisionBNtrain1,4),  
 round(precisionBNtrain2,4),  
 round(precisionRL\_01,4),  
 round(precisionRL\_02,4),  
 round(precisionF1,4),  
 round(precisionF2,4))  
  
Resultado = as.data.frame(rbind(Accuracy,   
 Recall,   
 Precision))  
colnames(Resultado) = titulos\_modelos  
  
Resultado

## Modelo1DT Modelo2DT Modelo1NB Modelo2NB Modelo1LOG Modelo2LOG  
## Accuracy 0.8445 0.8584 0.7139 0.6799 0.7961 0.7644  
## Recall 0.1337 0.1414 0.8683 0.9314 0.5801 0.5915  
## Precision 0.2495 0.3226 0.2827 0.2681 0.3203 0.2829  
## ModeloRF1 ModeloRF2  
## Accuracy 0.8740 0.8728  
## Recall 0.0380 0.0482  
## Precision 0.3658 0.3599

library(tidyr)  
  
ResultadoDF = as.data.frame(Resultado) %>%  
 mutate(metrica = row.names(Resultado)) %>%  
 gather("Modelo", "Valor", -metrica) %>%  
 filter(metrica == "Accuracy")   
  
ResultadoDF %>%  
 ggplot(aes(x=metrica, y = Valor, fill = Modelo)) +   
 geom\_col(position = "dodge")



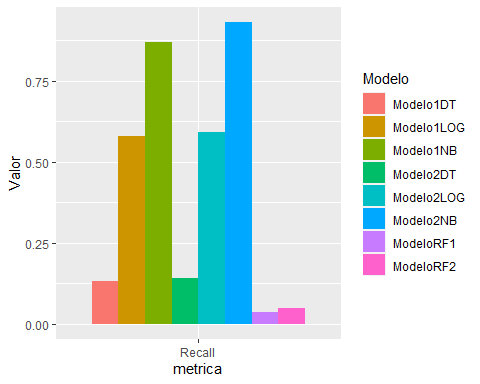
ResultadoDF

## metrica Modelo Valor  
## 1 Accuracy Modelo1DT 0.8445  
## 2 Accuracy Modelo2DT 0.8584  
## 3 Accuracy Modelo1NB 0.7139  
## 4 Accuracy Modelo2NB 0.6799  
## 5 Accuracy Modelo1LOG 0.7961  
## 6 Accuracy Modelo2LOG 0.7644  
## 7 Accuracy ModeloRF1 0.8740  
## 8 Accuracy ModeloRF2 0.8728

ResultadoDF = as.data.frame(Resultado) %>%  
 mutate(metrica = row.names(Resultado)) %>%  
 gather("Modelo", "Valor", -metrica) %>%  
 filter(metrica == "Recall")   
ResultadoDF

## metrica Modelo Valor  
## 1 Recall Modelo1DT 0.1337  
## 2 Recall Modelo2DT 0.1414  
## 3 Recall Modelo1NB 0.8683  
## 4 Recall Modelo2NB 0.9314  
## 5 Recall Modelo1LOG 0.5801  
## 6 Recall Modelo2LOG 0.5915  
## 7 Recall ModeloRF1 0.0380  
## 8 Recall ModeloRF2 0.0482

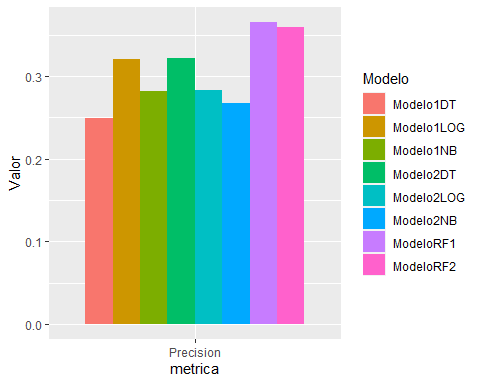
ResultadoDF %>%  
 ggplot(aes(x=metrica, y = Valor, fill = Modelo)) +   
 geom\_col(position = "dodge")



ResultadoDF = as.data.frame(Resultado) %>%  
 mutate(metrica = row.names(Resultado)) %>%  
 gather("Modelo", "Valor", -metrica) %>%  
 filter(metrica == "Precision")   
ResultadoDF

## metrica Modelo Valor  
## 1 Precision Modelo1DT 0.2495  
## 2 Precision Modelo2DT 0.3226  
## 3 Precision Modelo1NB 0.2827  
## 4 Precision Modelo2NB 0.2681  
## 5 Precision Modelo1LOG 0.3203  
## 6 Precision Modelo2LOG 0.2829  
## 7 Precision ModeloRF1 0.3658  
## 8 Precision ModeloRF2 0.3599

ResultadoDF %>%  
 ggplot(aes(x=metrica, y = Valor, fill = Modelo)) +   
 geom\_col(position = "dodge")



### Conclusiones y recomendaciones

### Conclusion 1

Por el caso de negocio presentado se prioriza el RECALL ya que el objetivo con el que se construye el modelo es identificar a la gente interesada en el seguro del vehiculo. Es más costoso en este caso no identificar a gente sí interesada que el modelo identifica a personas no interesada como sí interesadas.

Los modelos de NAIVE BAIYES son muy superiores en cuanto a Recall, en ACCURACY Y PRECISION no son el mejor modelo pero la diferencia con el resto de modelos no es muy grande.

Por lo anterior se escoge el modelo #2 del NAIVE BAYES.

### Conclusion 2

Se encontró que las variables planteadas a la hipótesis si tienen impacto en si las persona está o no interesada en el seguro de vehículo. No obstante ninguna de esas variables, ni ninguna de las que tiene el set de datos en general, tienen un impacto que sea muy determinante en el respuesta del cliente.

### Recomendaciones

Buscar más variables que puedan explicar el comportamiento de los clientes.

Recomendaría repetir los modelos una vez se tenga una muestra más grande con personas que sí están interesadas, ya que la proporción excesiva de personas no interesadas hace difícil que los modelos identifiquen a personas si interesadas.