Flujo de análisis en clasificación supervisada

Métodos supervisados

Laura Rodríguez Navas

Septiembre 2020

Contents

Comenzamos cargando los paquetes necesarios.

```
library(caret)
library(dplyr)
library(doParallel)
```

0.1 Descripción

Para la realización del ejercicio propuesto se ha elegido la competición en Kaggle: **Real or Not? NLP with Disaster Tweets**. El dataset de la competición se puede encontrar en el siguiente enlace: https://www.kaggle.com/c/nlp-getting-started/data. Este dataset, con 10.876 instancias, contiene 4 variables explicativas: **id**, **keyword**, **location** y **text**, y dos valores en la variable clase **target** (0 y 1). La variable clase es binaria, así que, vamos a aprender un modelo de clasificación binaria. El objetivo de este modelo será predecir si dado un tweet, este tweet trata sobre un desastre real o no. Si un tweet trata sobre un desastre real, se predice un 1. Si no, se predice un 0.

La métrica de evaluación esperada por la competición es F1. Y se calcula de la siguiente manera:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

donde:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

La partición inicial train-test, no se tiene que realizar, ya que las instancias de train y test ya vienen definidas en el dataset de la competición (ficheros **train.csv** y **test.csv**).

A continuación, cargaremos el conjunto de datos de train y test, nombrando los valores perdidos como **NA** para que los podamos tratar más adelante, y mostraremos sus dimensiones.

```
train <- read.csv("train.csv", na.strings=c("", "NA"))
test <- read.csv("test.csv", na.strings=c("", "NA"))
dim(train)
```

```
## [1] 7613    5
```

```
dim(test)
```

```
## [1] 3263    4
```

El conjunto de datos de train contiene 7613 instancias y el conjunto de datos de test contiene 3263 instancias. Cada instancia de estos conjuntos contiene la siguiente información:

- **id**: un identificador único para cada tweet.
- **keyword**: una palabra clave del tweet.
- **location**: la ubicación desde la que se envió el tweet.
- **text**: el texto del tweet.
- **target**: solo en el conjunto de datos de train porqué es la variable clase a predecir. Indica si un tweet es sobre un desastre real (1) o no (0).

```
str(train, width = 85, strict.width = "cut")
```

```
## 'data.frame':    7613 obs. of  5 variables:
##  $ id      : int  1 4 5 6 7 8 10 13 14 15 ...
##  $ keyword : chr  NA NA NA NA ...
##  $ location: chr  NA NA NA NA ...
##  $ text    : chr  "Our Deeds are the Reason of this #earthquake May ALLAH Forgive "..
##  $ target  : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
str(test, width = 85, strict.width = "cut")
```

```
## 'data.frame':    3263 obs. of  4 variables:
##  $ id      : int  0 2 3 9 11 12 21 22 27 29 ...
##  $ keyword : chr  NA NA NA NA ...
##  $ location: chr  NA NA NA NA ...
##  $ text    : chr  "Just happened a terrible car crash" "Heard about #earthquake is"..
```

Unimos los conjuntos de train y test (*7613 + 3263 observaciones*) para poder analizar y extraer los sentimientos más adelante.

```
complete_df <- bind_rows(train, test)
dim(complete_df)
```

```
## [1] 10876     5
```

El análisis de sentimientos es una técnica de *Machine Learning*, basada en el procesado del lenguaje natural, que pretende obtener información subjetiva de una serie de textos. Su aplicación es este caso, consiste en resolver si un tweet es real o no real en relación a un desastre.

Echamos un vistazo más de cerca a las variables del nuevo conjunto de datos **complete_df**.

```
summary(complete_df)
```

```
##        id          keyword            location            text
##  Min.   :    0   Length:10876       Length:10876       Length:10876
##  1st Qu.: 2719   Class :character   Class :character   Class :character
##  Median : 5438   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 5438
##  3rd Qu.: 8156
##  Max.   :10875
##
##      target
##  Min.   :0.00
##  1st Qu.:0.00
##  Median :0.00
##  Mean   :0.43
##  3rd Qu.:1.00
##  Max.   :1.00
##  NA's   :3263
```

Vemos que la variable **id** es solo un identificador único y la eliminaremos.

```
complete_df$id <- NULL
dim(complete_df)
```

```
## [1] 10876     4
```

- Confirmamos que la variable **target** es la variable dependiente, la que nos interesa es predecir. Al observarla podemos decir que el 32% de los tweets en realidad se refieren a un desastre y el 43% no.

0.2 Gráficas de visualización

Relacionado con la descripción anterior, mostraremos algunas gráficas de visualización de algunas variables de interés. Pero primero tenemos que categorizar la variable a predecir, ya que inicialmente es de tipo entero.

```
complete_df$target <- as.factor(ifelse(complete_df$target == 0, "No", "Yes"))
str(complete_df, width = 85, strict.width = "cut")
```

```
## 'data.frame':    10876 obs. of  4 variables:
##  $ keyword : chr  NA NA NA NA ...
##  $ location: chr  NA NA NA NA ...
##  $ text    : chr  "Our Deeds are the Reason of this #earthquake May ALLAH Forgive "..
##  $ target  : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
```

La primera visualización representa la distribución de la variable a predecir.

```
ggplot(train, aes(x=target)) + geom_bar(aes(fill=target))
```