# PROYECTO FINAL: PREDICCION GÉNERO USUARIO TWITTER

NURIA REDO

24 de febrero de 2018

## 0.-INTRODUCCION

*No hace falta ser una persona demasiado observadora para mirar a nuestro alrededor y darnos cuenta de que hombres y mujeres nos comunicamos con nuestros iguales de una manera diferente.*

*Las chicas tienden a ser más habladoras, más correctas y discretas en sus discursos y a expresar sus ideas de forma más emotiva. Son más expresivas en sus gestos faciales y gesticulan más.*

*Hacen preguntas de cortesía o crean breves charlas previas para crear un ambiente cercano. Además, dan más rodeos para expresar sus ideas y con frecuencia utilizan expresiones para modular sus afirmaciones ("algo así como", "una especie de", "a mí me parece.*

*Por el contrario, los chicos se expresan de manera más firme y segura, articulando discursos más racionales que emotivos y usando términos menos expresivos. Los hombres usan una entonación más plana, con menos sensación de implicación emocional. Sonríen y gesticulan menos, y escuchan a su interlocutor sin hacer referencias explícitas a que están receptivos y atentos a la comunicación.*

*Las expresiones coloquiales o vulgares, los tacos y las formas despectivas o insultos leves sirven para reforzar lazos de solidaridad masculina, frente a la afinidad femenina que se potencia a través de cumplidos mutuos y la ausencia de crítica abierta.*

*Existen numerosos estudios sobre la lengua escrita, pero ¿es posible localizar esta forma de hablar en la nueva forma de comunicarse como son las redes sociales?.*

*Las redes sociales son el medio de comunicación más extendido. Mezcla patrones de comunicación escrita con patrones propios de comunicación oral. Pero, ¿puede distinguirse el género de un usuario de twitter con solo 140 caracteres para su estudio? ¿Podemos ver si realmente se cumplen los patrones descritos anteriormente? ¿Realmente, las chicas optan por el rosa y los chicos por el negro? Este proyecto pretende clasificar en base un dataset de tweets a los usuarios por su género y crear un modelo de predicción en base a diversas variables.*

# 1.- ANÁLISIS EXPLORATORIO

*En este apartamos configuramos el escritorio y exploramos los datos del dataset. La información que cargaremos será unicamente aquella que nos podria ser útil para la predicción de género. Cargamos aquellos campos resaltados.*

- **_unit_id**: a unique id for user
- **_golden**: whether the user was included in the gold standard for the model; TRUE or FALSE
- **_unit_state**: state of the observation; one of *finalized* (for contributor-judged) or *golden* (for gold standard observations)
- **_trusted_judgments**: number of trusted judgments (int); always 3 for non-golden, and what may be a unique id for gold standard observations
- **_last_judgment_at**: date and time of last contributor judgment; blank for gold standard observations
- **gender***: one of *male*, *female*, or *brand* (for non-human profiles)
- **gender:confidence**: a float representing confidence in the provided gender
- **profile_yn**: "no" here seems to mean that the profile was meant to be part of the dataset but was not available when contributors went to judge it
- **profile_yn:confidence**: confidence in the existence/non-existence of the profile
- **created**: date and time when the profile was created
- **description***: the user's profile description
- **fav_number**: number of tweets the user has favorited
- **gender_gold**: if the profile is golden, what is the gender?
- **link_color**: the link color on the profile, as a hex value
- **name**: the user's name
- **profile_yn_gold**: whether the profile y/n value is golden
- **profileimage**: a link to the profile image
- **retweet_count**: number of times the user has retweeted (or possibly, been retweeted)
- **sidebar_color**: color of the profile sidebar, as a hex value
- **text**: text of a random one of the user's tweets
- **tweet_coord**: if the user has location turned on, the coordinates as a string with the format "[*latitude*, *longitude*]"
- **tweet_count**: number of tweets that the user has posted
- **tweet_created**: when the random tweet (in the **text** column) was created
- **tweet_id**: the tweet id of the random tweet
- **tweet_location**: location of the tweet; seems to not be particularly normalized
- **user_timezone**: the timezone of the user

```
##################CONFIGURATION#################################################
rm(list=ls())
ls()

## character(0)

memory.limit(70000)

## [1] 70000

if (Sys.getenv("JAVA_HOME")!="")
  Sys.setenv(JAVA_HOME="")
Sys.setenv(JAVA_HOME='C:/Program Files/Java/jre1.8.0_161')
options(stringsAsFactors = FALSE)
.libPaths("F:/Program Files/R/R-3.4.3/library")
library(rJava)

############IMPORTACION ARCHIVO Y EXPLORACION#############################
file <- read.csv("F:/NitroPC/Google Drive/BDATA/proyecto/gender-
classifier-DFE-791531.csv/gender-classifier-DFE-791531.csv")
#nos quedamos solo con las columnas que queremos
file <- file[,c(1,6,7,11,12,14,15,17,18,19,20,22)]
#exploramos los datos
summary(file)

##    X_unit_id            gender          gender.confidence
##  Min.   :815719226   Length:20050       Min.   :0.0000
##  1st Qu.:815724318   Class :character   1st Qu.:0.6778
##  Median :815729384   Mode  :character   Median :1.0000
##  Mean   :815729449                      Mean   :0.8828
##  3rd Qu.:815734514                      3rd Qu.:1.0000
##  Max.   :815757985                      Max.   :1.0000
##                                         NA's   :26
##  description          fav_number      link_color            name
##  Length:20050       Min.   :     0   Length:20050       Length:20050
##  Class :character   1st Qu.:    11   Class :character   Class
:character
##  Mode  :character   Median :   456   Mode  :character   Mode
:character
##                     Mean   :  4382
##                     3rd Qu.:  3316
##                     Max.   :341621
##
##  profileimage       retweet_count      sidebar_color
##  Length:20050       Min.   :  0.0000   Length:20050
##  Class :character   1st Qu.:  0.0000   Class :character
##  Mode  :character   Median :  0.0000   Mode  :character
##                     Mean   :  0.0794
##                     3rd Qu.:  0.0000
##                     Max.   :330.0000
##
##      text            tweet_count
##  Length:20050       Min.   :      1
```

```
##  Class :character   1st Qu.:   2398
##  Mode  :character   Median :  11442
##                     Mean   :  38925
##                     3rd Qu.:  40028
##                     Max.   :2680199
##
```

```
head(file[ ,1:5])
```

```
##   X_unit_id gender gender.confidence
## 1 815719226   male            1.0000
## 2 815719227   male            1.0000
## 3 815719228   male            0.6625
## 4 815719229   male            1.0000
## 5 815719230 female            1.0000
## 6 815719231 female            1.0000
##
description
## 1
i sing my own rhythm.
## 2
I'm the author of novels filled with family drama and romance.
## 3
louis whining and squealing and all
## 4               Mobile guy.  49ers, Shazam, Google, Kleiner Perkins,
Yahoo!, Sprint PCS, AirTouch, Air Force.  Stanford GSB, UVa.  Dad,
Husband, Brother.  Golfer.
## 5 Ricky Wilson The Best FRONTMAN/Kaiser Chiefs The Best BAND Xxxx
Thank you Kaiser Chiefs for an incredible year of gigs and memories to
cherish always :) Xxxxxxx
## 6
you don't know me.
##   fav_number
## 1          0
## 2         68
## 3       7696
## 4        202
## 5      37318
## 6       3901
```

```
dim(file)  #numero observaciones/columnas
```

```
## [1] 20050    12
```

```
str(file) #cor(data)
```

```
## 'data.frame':    20050 obs. of  12 variables:
##  $ X_unit_id        : int  815719226 815719227 815719228 815719229
815719230 815719231 815719232 815719233 815719234 815719235 ...
##  $ gender           : chr  "male" "male" "male" "male" ...
##  $ gender.confidence: num  1 1 0.662 1 1 ...
##  $ description      : chr  "i sing my own rhythm." "I'm the author of
novels filled with family drama and romance." "louis whining and
```

```
squealing and all" "Mobile guy.  49ers, Shazam, Google, Kleiner Perkins,
Yahoo!, Sprint PCS, AirTouch, Air Force.  Stanford GSB, UV"|
__truncated__ ...
##  $ fav_number       : int  0 68 7696 202 37318 3901 4122 80 1825 3115
...
##  $ link_color       : chr  "08C2C2" "0084B4" "ABB8C2" "0084B4" ...
##  $ name             : chr  "sheezy0" "DavdBurnett" "lwtprettylaugh"
"douggarland" ...
##  $ profileimage     : chr
"https://pbs.twimg.com/profile_images/414342229096808449/fYvzqXN7_normal.
png"
"https://pbs.twimg.com/profile_images/539604221532700673/WW16tBbU_normal.
jpeg"
"https://pbs.twimg.com/profile_images/657330418249658368/SBLCXdF7_normal.
png" "https://pbs.twimg.com/profile_images/259703936/IMG_8444_normal.JPG"
...
##  $ retweet_count    : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ sidebar_color    : chr  "FFFFFF" "C0DEED" "C0DEED" "C0DEED" ...
##  $ text             : chr  "Robbie E Responds To Critics After Win
Against Eddie Edwards In The #WorldTitleSeries https://t.co/NSybBmVjKZ"
"%ÛÏIt felt like they were my friends and I was living the story with
them%Û https://t.co/arngE0YHNO #retired #"| __truncated__ "i absolutely
adore when louis starts the songs it hits me hard but it feels good" "Hi
@JordanSpieth - Looking at the url - do you use @IFTTT?!  Don't typically
see an advanced user on the @PGATO"| __truncated__ ...
##  $ tweet_count      : int  110964 7471 5617 1693 31462 20036 13354
112117 482 26085 ...
```

```
#vamos a ver si hay un tweet por usuario
length(unique(file$x_unit_id))   #hay un tweet por usuario.
```

```
## [1] 0
```

```
#Encoding
library(stringi)
stri_enc_mark(file$text)  #la mayoria son ASCII y alguna native
```

```
##     [1] "ASCII"  "native" "ASCII"  "ASCII"  "native" "ASCII"  "ASCII"
##     [8] "native" "ASCII"  "ASCII"  "ASCII"  "ASCII"  "ASCII"  "ASCII"
##    [15] "ASCII"  "ASCII"  "native" "ASCII"  "ASCII"  "ASCII"  "ASCII"
```

```
#Vamos a a ver el tema de los valores na.De momento no los descartaremos.
sapply(file, function(x) sum(is.na(x)))
```

```
##         X_unit_id            gender gender.confidence
description
##                 0                 0                26
0
##        fav_number        link_color              name
profileimage
##                 0                 0                 0
0
##     retweet_count     sidebar_color              text
```

```
tweet_count
##                 0                 0                 0
0

#miramos valores unicos
sapply(file,function(x) length(unique(x)))

##         X_unit_id            gender gender.confidence
description
##             20050                 5               924
15141
##        fav_number        link_color              name
profileimage
##              6784              3001             18795
17164
##     retweet_count     sidebar_color              text
tweet_count
##                22               561             18412
14280

#convertir a minusculas los nombres de las columnas
names(file) <-tolower(names(file))

#agrupamos por genero las observ vemos que tenemos: male, female, brand,
en blanco
library(plyr)
library(dplyr)

file%>%
  group_by(gender) %>%
  summarise(n=n())

## # A tibble: 5 x 2
##   gender      n
##   <chr>   <int>
## 1 ""         97
## 2 brand    5942
## 3 female   6700
## 4 male     6194
## 5 unknown  1117

#Vamos a ver el idioma de los tweets. Tendremos que realizer limpieza y
tratamiento del texto por lo que es un paso necesario.
#Es una primera aproximación, para más efectividad deberiamos limpiar el
texto.
library(textcat)
my.profiles <- TC_byte_profiles[names(TC_byte_profiles)]
file$language <- textcat(file$text,my.profiles )
file%>%
  group_by(language) %>%
  summarise(n=n())
```

```
## # A tibble: 42 x 2
##    language            n
##    <chr>           <int>
##  1 afrikaans           7
##  2 albanian            1
##  3 basque              1
##  4 breton             31
##  5 catalan            84
##  6 chinese-big5      118
##  7 czech-iso8859_2     2
##  8 danish             57
##  9 dutch               3
## 10 english         13961
## # ... with 32 more rows
```

```
#vemos si realmente las detección el idioma chino es real o es errónea
filechinese <- subset(file, language =='chinese-big5')
head(filechinese$text)
```

```
## [1] "I HATE THE WALKING DEAD NO I DON'T BELIEVE HE IS DEAD"
## [2] "@LukeIsNotSexy I WISH YOU WOULD NOTICE ME BC I KNOW YOU AND WE
WENT TO THE SAME PRIMARY SCHOOL AND ITS SO EXCITING IM CRYING"
## [3] "mADELEINE THIS AINT THE TIME\n\nHOPE IS GOIN ON A JOURNEY"
## [4] "@milkcartn DO YOU WANT THE HALLOWEEN OR CHRISTMAS ONE"
## [5] "WHY THE FUCK IS MY TAP WATER YELLOW BRO"
## [6] "TIDAL RAPED THE SHIT OUT OF AMERICAN OXYGEN \n\nAPPLE MUSIC TOOK
THE #1 AWAY FROM DRAKE\n\n#BOYCOTTSTREAMINGSERVICESEXCLUSIVES"
```

```
.Es errónea. Todo es Inglés.Así que trataremos todo de manera masiva. No
hará falta separar por idiomas.


#convertimos el id numérico a caracter ya que no nos interesa que sea
nuéerico
file$x_unit_id <-as.character(file$x_unit_id)
```

# 2.-LIMPIEZA DE TEXTO

```
#El dataset contiene varios campos de texto, tales como
descripción del usuario y el tweet propiamente, vamos a realizar
limpieza de texto ya que posteriormente trabajaremos con él.

#Primero creamos la función que usaremos para limpiar los
diferentes campos.
```

```
##################LIMPIEZA DE TEXTO############
library(stringr)

cleaning <- function(s){
  s = tolower(s)
  s = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", s) #remove retweets
  s = gsub("@\\w+", "", s) #remove other screen names
  s = gsub("[[:punct:]]", "", s) #remove punctuation
  s = gsub("[[:digit:]]", "", s) #remove numbers
  s = gsub("http\\w+", "", s) #remove links
  s = gsub("https\\w+", "", s) #remove links
  s = gsub("[ \t]{2,}", "", s)
  s = gsub("^\\s+|\\s+$", "", s)
  s = str_replace_all(s," "," ") #get rid of unnecessary spaces
  s = trimws(s)
  return(s)
}
#pasamos la función por los diferentes strings.

file$description <- cleaning(file$description)
file$clean_tweet <- cleaning(file$text)
file$clean_tweet <- iconv(file$clean_tweet, from = "latin1", to =
"ascii", sub = "byte")
file$description <- iconv(file$description, from = "latin1", to =
"ascii", sub = "byte")
file$name <-  cleaning(file$name)

head(file$clean_tweet)
```

```
## [1] "robbie e responds to critics after win against eddie edwards in
the worldtitleseries"
## [2] "<e2><80><b0><c3><bb><c3><af>it felt like they were my friends and
i was living the story with them<e2><80><b0><c3><bb><c2><9d>retired ian"
## [3] "i absolutely adore when louis starts the songs it hits me hard
but it feels good"
## [4] "hilooking at the urldo you usedont typically see an advanced user
on the"
## [5] "watching neighbours on sky catching up with the neighbs xxx
<c3><b9><c3><a4><c3><b9><c3><a4><c3><b9><c3><a4><c3><b9><c3><b4><c3><ae><
c3><b9><c2><8f><c3><a8><c3><b9><c3><b4><c2><8d><c3><b9><c2><8f><c3><a8>
xxx"
## [6] "ive seen people on the train with lamps chairs tvs etc"
```

# 3.-DETECCION DE TWEETS CON MULTIPLES VOCALES

```
#Vamos a ver si la detección de patrones de repetición de vocales nos
ayuda a la predicción del género. En general las mujeres tendemos más a
escribir looooooool, hellooooo que los hombres.
```

```
######## EXTRACT AAAA-UUUUU PATTERNS ####################################
pattern <- c("aaa", "eee", "iii", "ooo", "uuu")
file$rep_vowels<-str_detect(file$clean_tweet,pattern)
file%>%
  group_by(rep_vowels) %>%
  summarise(n=n())

## # A tibble: 2 x 2
##   rep_vowels     n
##   <lgl>      <int>
## 1 F          20020
## 2 T             30

#no es una variable a tener en cuenta en esta muestra, solo ha encontrado
30 tuits con vocales repetidas.
file$rep_vowels <-NULL
```

# 4.-EMOTICONOS Y EMOJIS

```
#Los hombres y las mujeres nos expresamos diferente. Vamos a ver si el
hecho de contener emoticonos o emojis pueden ayudarnos a la predicción
del género.

##################EMOTICONOS Y EMOJIS##########

#Extraemos los emoticonos del texto


library(qdapRegex)
file$emoticon <-ex_emoticon(file$clean_tweet)
file$clean_tweet<-rm_emoticon(file$clean_tweet)

#sustituimos por si hay o no emoticon. 0-no hay  1-hay
file %>%
  plyr::mutate(emoticon =
                 dplyr::case_when(file$emoticon  == " " ~ "0",
                                  file$emoticon  != " " ~ "1",
                                  TRUE                  ~ " 0"
                 )) -> file
file$emoticon <-as.factor(file$emoticon)
```

```
#########emojis#################
#La idea inicial era extraer los emojis con un diccionario de emojis y
contarlos, incluso usarlos para analisis de sentimiento pero finalmente
he optado por extraer los caracteres que son no ascii para detector los
emojis.


#extraemos los caracteres que son no ascii para detectar los emojis.

#0-sin emojis 1-con emojis.
#extrae algun texto al no haber espacios pero dado que el campo recoge
unicamente si hay o no, no es una
#incidencia que nos afecte.
file$emoji <-ex_non_ascii(file$clean_tweet, extract=T)

file %>%
  plyr::mutate(emoji =
                 dplyr::case_when(file$emoji  != " " ~ "1",
                                  TRUE               ~ "0"
                 )) -> file

file$emoji <-as.factor(file$emoji)
```

# 4.-TRATAMIENTO DEL COLOR

```
#¿somos las mujeres más de rosa o violeta? Prefieren los hombres más el
negro o gris? Esta variable nos ayudara a ver si el color nos ayuda en la
construcción de nuestro modelo.

#Para ello creo un diccionario de colores en base a los colores
existentes en el dataset.

##########COLOR VALUES####################

#vamos a convertir en numérico los campos sidebar_color y link_color
#creamos el diccionario de colores propio.
library(data.table)
bar <-data.table(unique(file$sidebar_color))
str(bar)

## Classes 'data.table' and 'data.frame':   561 obs. of  1 variable:
##  $ V1: chr  "FFFFFF" "C0DEED" "0" "181A1E" ...
##  - attr(*, ".internal.selfref")=<externalptr>

link <-data.table(unique(file$link_color))


#unimos los colores y quitamos los duplicados con la funcion unique y
creamos el id numerico
color <-unique(rbind(bar,link))
```

```
color$id_color <-seq.int(nrow(color))

#substitute colors by id number
library(dplyr)
library(gdata)
file <- merge(file, color, by.x='sidebar_color', by.y='V1', all.x=TRUE)
file$sidebar_color <-file$id_color
file <- merge(file, color, by.x='link_color', by.y='V1', all.x=TRUE)
file$link_color <-file$id_color.y

file <- file[ ,!names(file) %in%c("id_color.y","id_color.x")]

################COVERT GENDER TO VALUES##########

file %>%
  plyr::mutate(gender =
                 dplyr::case_when(file$gender  == "male"   ~ "0",
                                  file$gender  == "female" ~ "1",
                                  file$gender  == "brand"  ~ "2",
                                  )) -> file
levels(as.factor(file$gender))

## [1] "0" "1" "2"
```

# 5.-TRATAMIENTO USERNAME

```
#El nombre de usuario puede ayudarnos a predecir el género. Vamos a
verlo.

 ################PREDICT GENDER WITH NAMES##############################

#genderizeR package funcionaria perfecto pero tenemos la limitación de la
API por lo que buscaremos alguna otra alternativa. Utilizaremos la tabla
names existente en qdapDictionaries que recoge nombres con la predicción
de genero,

library(qdapDictionaries)
library(fuzzyjoin)
#name. A first name.
names

names <- data.table(NAMES_SEX)
names$name <-tolower(names$name)
names$lenght <-nchar(names$name)
names <-names[names$lenght>3]

 #filtramos para limitar un poco el matching,p perderemos nombres como
Eva o Ed pero evitaremos múltiples matchings erroneos.
```

```r
names$gender_by_name <- ifelse(names$pred.sex == "F", 1, 0)
str(names)
```

```
## Classes 'data.table' and 'data.frame':   4937 obs. of  5 variables:
##  $ name          : chr  "mary" "patricia" "linda" "barbara" ...
##  $ gender2       : Factor w/ 3 levels "B","F","M": 1 1 2 2 2 2 1 2 2 2
...
##  $ pred.sex      : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
##  $ lenght        : int  4 8 5 7 9 8 5 5 8 7 ...
##  $ gender_by_name: num  1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
names <-names[ ,c(1,5)]

#incorporamos algunas palabras que se identifican con el genero y que
nos:

dictionary <-
data.frame('name'=c("dad","daddy","mum","mummy","mr","mrs","pretty","man"
, "woman","miss","mister"),
             'gender_by_name'=c(0,0,1,1,0,1,1,0,1,1,0))

names <- rbind(names,dictionary)

file<-regex_left_join(file,names, by='name', ignore_case=FALSE)

file%>%
  group_by(gender_by_name) %>%
  summarise(n=n())
```

```
## # A tibble: 3 x 2
##   gender_by_name     n
##            <dbl> <int>
## 1           0     5846
## 2           1.00  9108
## 3          NA    11380
```

```r
#nos ha casado  un 56% de los datos con uno de los nombres de la tabla.
#vemos tambien como se han incrementado los registros ya que ha casado
mas de un nombre con el username.
#ya veremos posteriormente si el hecho de tener un nombre de usuario u
otro ayuda a la prediccion del genero.
```

# 6.-PART-OF-SPEECH TAGGING

```
#Dicen que las mujeres hablamos más, el estudio POS del texto del tweet
puede ayudarnos a la construccion de nuestro modelo.Utilizamos más
conjunciones las mujeres? Ahora lo veremos.

Veremos la composición de los tweets: número de determinantes, adjetivos,
adverbios, verbos, conjunciones..nos medirá la complejidad del tweet.

###############POS TAGGING#################################################
#existe la libraria Korpus, spacy. Finalmente he elegido udpipe ya que es
un proceso lento y que consume mucha memoria.
# Undepipe no tiene el problema de memoria que he encontrado en Korpus y
no depende de tener instalado spaCy en Python.


library(data.table)
library(koRpus)
library(openNLP)
library(NLP)
library(openNLPdata)
library(reticulate)
library(udpipe)
library(tibble)
#gc()    #limpiamos ya que el proceso pesa un poco.
t <-file$clean_tweet
model <- udpipe_download_model(language = "english")

## Downloading udpipe model from
https://raw.githubusercontent.com/jwijffels/udpipe.models.ud.2.0/master/i
nst/udpipe-ud-2.0-170801/english-ud-2.0-170801.udpipe to f:/Program
Files/RStudio/english-ud-2.0-170801.udpipe



model

##   language                                                file_model
## 1  english f:/Program Files/RStudio/english-ud-2.0-170801.udpipe
##
url
## 1
https://raw.githubusercontent.com/jwijffels/udpipe.models.ud.2.0/master/i
nst/udpipe-ud-2.0-170801/english-ud-2.0-170801.udpipe

udmodel_english <- udpipe_load_model(file = "english-ud-2.0-
170801.udpipe")
c<- udpipe_annotate(udmodel_english,t)
d <- as.data.frame(c)
str(d)
```

```
## 'data.frame':    432746 obs. of  14 variables:
##  $ doc_id       : chr  "doc1" "doc1" "doc1" "doc1" ...
##  $ paragraph_id : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ sentence_id  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ sentence     : chr  "like disco and hammer pants quoteoftheday you
guys quote of the day" "like disco and hammer pants quoteoftheday you
guys quote of the day" "like disco and hammer pants quoteoftheday you
guys quote of the day" "like disco and hammer pants quoteoftheday you
guys quote of the day" ...
##  $ token_id     : chr  "1" "2" "3" "4" ...
##  $ token        : chr  "like" "disco" "and" "hammer" ...
##  $ lemma        : chr  "like" "disco" "and" "hammer" ...
##  $ upos         : chr  "ADP" "NOUN" "CCONJ" "NOUN" ...
##  $ xpos         : chr  "IN" "NN" "CC" "NN" ...
##  $ feats        : chr  NA "Number=Sing" NA "Number=Sing" ...
##  $ head_token_id: chr  "2" "9" "5" "5" ...
##  $ dep_rel      : chr  "case" "nmod" "cc" "compound" ...
##  $ deps         : chr  NA NA NA NA ...
##  $ misc         : chr  NA NA NA NA ...

e <- document_term_frequencies(d[, c("doc_id", "upos")])
f <- document_term_matrix(e)
g <-as.data.frame(as.matrix(f))
rownames(g) <- as.numeric(sub("doc","", rownames(g)))
file <-cbind(file, g)
```

# 7.-SENTIMENTAL ANALYSIS

```
#Siempre se ha dicho que las mujeres somos más sentimentales. ¿Es posible
detectarlo en este dataset?. Para ello vamos a realizar el estudio de
sentimiento. Segun esta regla, el sentimiento "neutral" deberia ser
mayoritario para los hombres…¿se cumplirá?


############ESTUDIO SENTIMIENTO EN EL TEXTO#################################


library(NLP)
library(tm)
library(SentimentAnalysis)

documents <-file$clean_tweet


# Analyze sentiment
file$sentiment <- analyzeSentiment(documents)
corps<- VCorpus(VectorSource(file$clean_tweet))
corps <- tm_map(corps, removeWords,stopwords('english'))
```

```
sent <- analyzeSentiment(corps)
sent$SentimentQDAP
```

```
##     [1]  0.12500000  0.14285714  0.00000000  0.00000000  0.00000000
##     [6] -0.25000000  0.00000000  0.00000000  0.07692308  0.15384615
##    [11]  0.20000000  0.20000000  0.00000000  0.00000000  0.00000000
##    [16]  0.00000000  0.00000000  0.00000000  0.08333333  0.25000000
##    [21] -0.20000000  0.10000000  0.10000000 -0.12500000 -0.11111111
##    [26]  0.22222222  0.09090909  0.09090909  0.00000000 -0.05882353
##    [31]  0.25000000  0.00000000 -0.33333333  0.00000000  0.00000000
##    [36] -0.07142857 -0.07142857 -0.07692308  0.00000000  0.12500000
```

….

```
# View sentiment direction (i.e. negative=1, neutral=2, positive=3)
file$sent2 <-as.factor(convertToDirection(sent$SentimentGI))
file[, levels(file$sent2) <- c(1, 2, 3)]
```

# 8.-ANALISIS DE LAS PALABRAS MÁS FRECUENTES

```
# En este apartado vamos a realizar el tratamiento de las palabras encontradas en el tweet. ¿Sirven para detectar el género? Son los hombres más de Lucha, Guerra y tacos? ¿Vamos repartiendo las mujeres amor a todo el mundo? Vamos a verlo.


##############FREQUENT WORDS BY GENDER#######################################


#tomamos una muestra que recoja usuarios con confianza >0.6 y gender=(0,1)
library(RColorBrewer)
library(NLP)
library(tm)

file$gender <- as.numeric(file$gender)
filegender <-file[file$gender%in%c(0,1) & file$gender.confidence > 0.6 ,]
str(file)
```

```
## 'data.frame':    26334 obs. of  37 variables:
##  $ link_color       : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ sidebar_color    : int  3 1 1 1 3 1 3 1 3 3 ...
##  $ x_unit_id        : chr  "815729481" "815726166" "815739448"
"815733021" ...
##  $ gender           : num  1 2 1 1 1 NA 2 1 2 1 ...
```

```
##  $ gender.confidence: num  1 1 1 1 1 ...
##  $ description      : chr  "lover of words imagesany combination of
the two nyc writer director filmmakerphoto hobbyist aspiring world
traveler" "this is the live account forcheck here to see what song is
playing right now on our soon comingweb stream" "im not gonna say that im
sorry gonna see the end of this storyis my princess
<e2><80><b0><c2><9d><e2><80><a2><c"| __truncated__ "how can you stan sos
im not racist but luke looks like a potato" ...
##  $ fav_number       : int  9201 0 23619 10847 41548 26256 21512 15916
1598 4475 ...
##  $ name.x           : chr  "writedirect" "wmullive" "hejkabeaux"
"newbrokencena" ...
##  $ profileimage     : chr
"https://pbs.twimg.com/profile_images/536243659466104832/DF8E7Ddd_normal.
jpeg"
"https://pbs.twimg.com/profile_images/439436014231109632/erV9G6qX_normal.
png"
"https://pbs.twimg.com/profile_images/657972761348960256/mDTLIZ5f_normal.
png"
"https://pbs.twimg.com/profile_images/658249901458006016/FE1lQl9__normal.
jpg" ...
##  $ retweet_count    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ text             : chr  "\"Like disco and hammer pants\"
\n\n#quoteoftheday, you guys. Quote. Of. The. Day." "Just played: The
Misfit (Got To Keep Movin' - Earl Hooker - unknown(unknown)" "5sos hate
don't stop and its hilarious loool https://t.co/u1K0qGsHnD" "should demi
run for president and call herself POOTUS" ...
##  $ tweet_count      : int  525 154569 88691 24816 16468 58538 1731
56063 27900 14209 ...
##  $ language         : chr  "english" "scots" "middle_frisian"
"english" ...
##  $ clean_tweet      : chr  "like disco and hammer pants quoteoftheday
you guys quote of the day" "just played the misfit got to keep movinearl
hookerunknownunknown" "sos hate dont stop and its hilarious loool"
"should demi run for president and call herself pootus" ...
##  $ emoticon         : Factor w/ 2 levels " 0","1": 1 1 1 1 1 1 1 1 1 1
...
##  $ emoji            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2
...
##  $ name.y           : chr  NA NA "beau" NA ...
##  $ gender_by_name   : num  NA NA 0 NA NA NA NA 1 NA NA ...
##  $ ADJ              : num  0 1 1 0 0 0 0 1 0 0 ...
##  $ ADP              : num  2 0 0 1 0 1 5 0 1 1 ...
##  $ ADV              : num  0 1 1 0 1 0 1 0 0 0 ...
##  $ AUX              : num  1 3 3 0 3 0 0 0 0 0 ...
##  $ CCONJ            : num  1 1 1 0 1 0 2 1 0 0 ...
##  $ DET              : num  1 0 0 1 1 1 2 0 1 1 ...
##  $ INTJ             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NOUN             : num  6 2 2 4 3 4 4 2 4 4 ...
##  $ NUM              : num  0 2 2 0 0 0 0 0 0 0 ...
##  $ PART             : num  0 1 1 0 0 0 1 0 0 0 ...
##  $ PRON             : num  1 5 5 0 3 0 4 0 0 0 ...
```

```
##  $ PROPN            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PUNCT            : num  0 6 6 0 0 0 0 1 0 0 ...
##  $ SCONJ            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SYM              : num  0 9 9 0 0 0 0 3 0 0 ...
##  $ VERB             : num  0 6 6 1 5 1 4 0 1 1 ...
##  $ X                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ sentiment        :'data.frame':   26334 obs. of  14 variables:
##   ..$ WordCount         : num  8 7 6 5 10 4 5 5 13 13 ...
##   ..$ SentimentGI       : num  0.125 0.286 0 0 -0.1 ...
##   ..$ NegativityGI      : num  0 0 0.167 0.2 0.4 ...
##   ..$ PositivityGI      : num  0.125 0.286 0.167 0.2 0.3 ...
##   ..$ SentimentHE       : num  0 0 0 0 0.1 0 0 0 0 0 ...
##   ..$ NegativityHE      : num  0 0 0 0 0 0 0 0 0 0 ...
##   ..$ PositivityHE      : num  0 0 0 0 0.1 0 0 0 0 0 ...
##   ..$ SentimentLM       : num  0 0 -0.167 0 0.1 ...
##   ..$ NegativityLM      : num  0 0 0.167 0 0 ...
##   ..$ PositivityLM      : num  0 0 0 0 0.1 ...
##   ..$ RatioUncertaintyLM: num  0 0 0 0 0 0 0 0 0 0 ...
##   ..$ SentimentQDAP     : num  0.125 0.143 0 0 0 ...
##   ..$ NegativityQDAP    : num  0 0.143 0.167 0 0.1 ...
##   ..$ PositivityQDAP    : num  0.125 0.286 0.167 0 0.1 ...
##  $ sent2            : Factor w/ 3 levels "1","2","3": 3 3 2 2 1 1 2 2
3 3 ...
```

```
#creamos dos datasets para cada usuario y trataremos las palabras más
usadas según cada género.

male_file <- filegender[filegender$gender == 0,]
female_file <- filegender[filegender$gender == 1,]

WordFreq <- function(d){
  d = Corpus(VectorSource(d))
  d <- tm_map(d,removeWords,stopwords('english'))
  tdm <- TermDocumentMatrix(d)
  m <- as.matrix(tdm)
  v <- sort(rowSums(m),decreasing=TRUE)
  d <- data.frame(word = names(v),freq=v)
  return(d)
}
male_words = WordFreq(male_file$clean_tweet)
female_words = WordFreq(female_file$clean_tweet)

#We will calculate the probablity of gender (male in this case) given a
words i.e. P(Gender | word)

all_words = merge(x = male_words, y = female_words, by = "word", all =
TRUE)

colnames(all_words) <- c("word", "freq_m", "freq_f")

all_words[is.na(all_words)] <- 0
```

```r
all_words$sum = all_words$freq_m + all_words$freq_f

all_words$male_prob_words = all_words$freq_m/all_words$sum
all_words$female_prob_words = all_words$freq_f/all_words$sum

d = Corpus(VectorSource(file$clean_tweet))
d <- tm_map(d, removeWords,stopwords('english'))
tdm <- TermDocumentMatrix(d)
library(tidytext)
DF <- tidy(tdm)
max(as.numeric(DF$document))
```

```
## [1] 26334
```

```r
DF <- DF[DF$term %in% all_words$word,]
merged_set <- merge(x = DF, y = all_words, by.x = "term", by.y = "word")
sapply(merged_set, class)
```

```
##              term          document             count
freq_m
##       "character"       "character"         "numeric"
"numeric"
##            freq_f               sum   male_prob_words
female_prob_words
##         "numeric"         "numeric"         "numeric"
"numeric"
```

```r
merged_set$document <- as.numeric(merged_set$document)
merged_set = merged_set[order(order(merged_set$document)),]
max(merged_set$document)
```

```
## [1] 26334
```

```r
aggr = aggregate(cbind(freq_m, sum) ~ document, data = merged_set, sum)
aggr$male_prob_words = aggr$freq_m / aggr$sum
max(aggr$document)
```

```
## [1] 26334
```

```r
file$x_unit_id = 1:nrow(file)
filegender$x_unit_id = 1:nrow(filegender)
file <- merge(file, aggr, by.x = "x_unit_id", by.y = "document", all.x =
T)
filegender <- merge(filegender, aggr, by.x = "x_unit_id", by.y =
"document", all.x = T)

# en función de la probabilidad de ser hombre creada anteriormente vamos
a crear el campo gender_by_words.

file$gender_bywords = ifelse(file$male_prob_words >0.5, 1, 0)
filegender$gender_bywords = ifelse(filegender$male_prob_words >= 0.5, 1,
0)
```

```
str(filegender)
```

```
## 'data.frame':    17914 obs. of  41 variables:
##  $ x_unit_id        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ link_color       : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ sidebar_color    : int  3 1 1 3 1 1 1 1 3 3 ...
##  $ PRON             : num  1 5 0 3 0 2 2 0 1 2 ...
##  $ PROPN            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PUNCT            : num  0 6 0 0 1 0 0 0 0 0 ...

…
##  $ SCONJ            : num  0 0 0 0 0 1 1 0 0 0 ...
##  $ SYM              : num  0 9 0 0 3 0 0 0 0 0 ...
##  $ VERB             : num  0 6 1 5 0 3 3 1 1 4 ...
##  $ X                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ sentiment        :'data.frame':   17914 obs. of  14 variables:
##   ..$ WordCount         : num  8 6 5 10 5 10 10 13 2 2 ...
##   ..$ SentimentGI       : num  0.125 0 0 -0.1 0 0.1 0.1 0 0 0 ...
##   ..$ NegativityGI      : num  0 0.167 0.2 0.4 0 ...
##   ..$ PositivityGI      : num  0.125 0.167 0.2 0.3 0 ...
..

##   ..$ PositivityQDAP    : num  0.125 0.167 0 0.1 0 ...
##  $ sent2            : Factor w/ 3 levels "1","2","3": 3 2 2 1 2 3 3 2
2 2 ...
##  $ freq_m           : num  688 853 440 96 114 112 260 19 622 917 ...
##  $ sum              : num  1851 2103 1042 203 235 ...
##  $ male_prob_words  : num  0.372 0.406 0.422 0.473 0.485 ...
##  $ gender_bywords   : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
file <-data.frame(subset(file,select=-
c(sentiment)),unclass(file$sentiment))
filegender <-data.frame(subset(filegender,select=-
c(sentiment)),unclass(filegender$sentiment))
```

# 9.-GUARDAMOS EN TABLEAU

```
#guardamos los archivos para poder usarlos posteriormente en Tableau.

#write.csv2(file,"F:/NitroPC/Google Drive/BDATA/proyecto/gender-
classifier-DFE-791531.csv/file.txt", na="")
#write.csv2(filegender,"F:/NitroPC/Google Drive/BDATA/proyecto/gender-
classifier-DFE-791531.csv/filegender.txt")
#write.csv2(male_words,"F:/NitroPC/Google Drive/BDATA/proyecto/gender-
classifier-DFE-791531.csv/male_words.txt")
#write.csv2(female_words,"F:/NitroPC/Google Drive/BDATA/proyecto/gender-
```

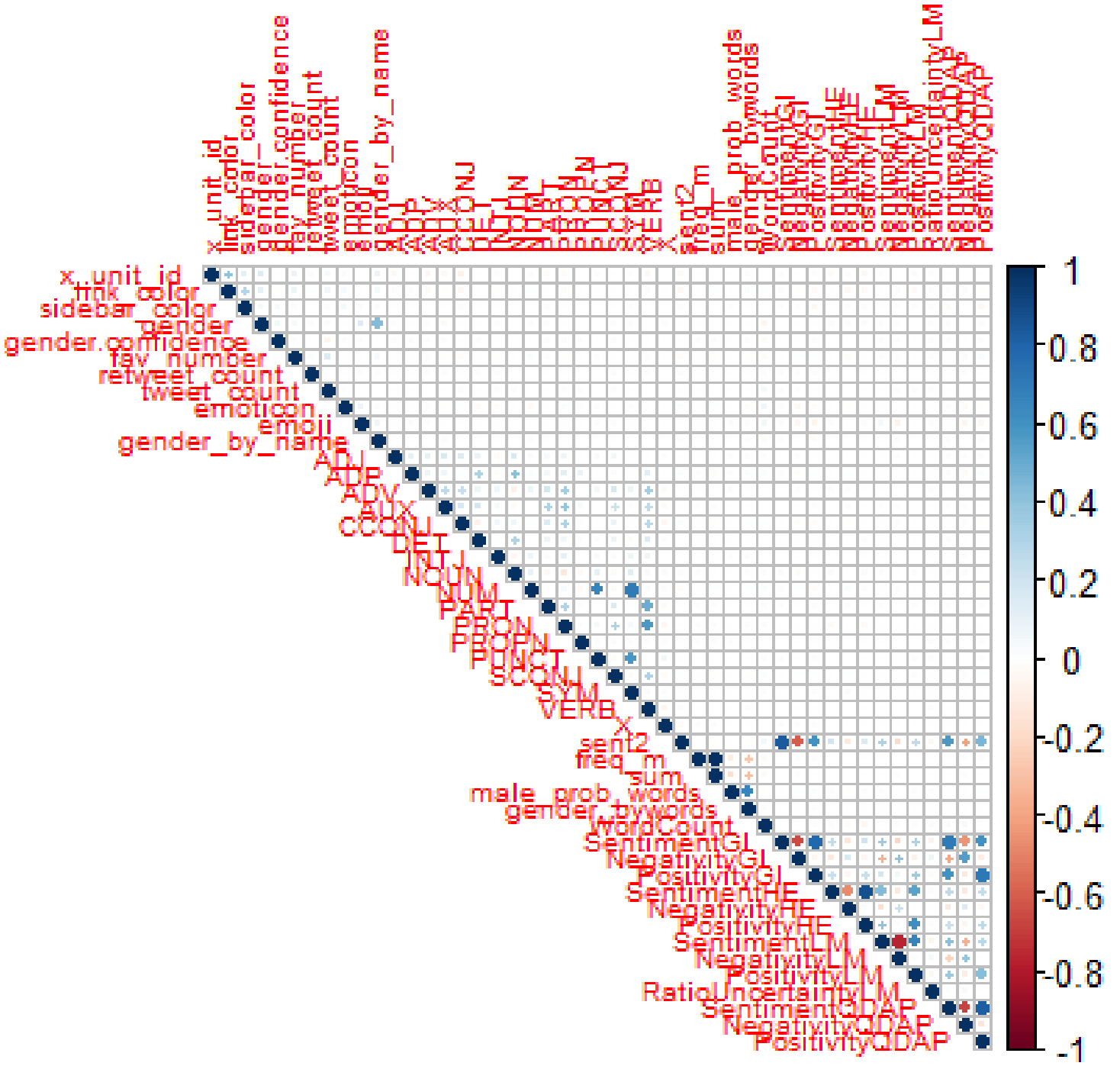
```
classifier-DFE-791531.csv/female_words.txt")
#write.csv2(color,"F:/NitroPC/Google Drive/BDATA/proyecto/gender-
classifier-DFE-791531.csv/color.txt")
```

# 10.-MATRIZ DE CORRELACION

```
#Una vez creadas las diferentes variables vamos a ver cuanto
correlacionadas se encuentran con "gender".

################################CORRELATIN MATRIX##########{}
library(corrplot)

## corrplot 0.84 loaded

library(RColorBrewer)
#vamos a ver cuales son numericas y cuales no. Hay algunas que nos
interesa estudiarlas, asi que las cambiaremos a numéricas.

sapply(file, class)

##          x_unit_id         link_color      sidebar_color
##          "integer"          "integer"          "integer"
##             gender  gender.confidence        description
##          "numeric"          "numeric"        "character"
##         fav_number             name.x       profileimage
##          "integer"        "character"        "character"
##      retweet_count               text        tweet_count
##          "integer"        "character"          "integer"
##           language        clean_tweet           emoticon
##        "character"        "character"           "factor"
##              emoji             name.y     gender_by_name
##           "factor"        "character"          "numeric"
##                ADJ                ADP                ADV
##          "numeric"          "numeric"          "numeric"
##                AUX              CCONJ                DET
##          "numeric"          "numeric"          "numeric"
##               INTJ               NOUN                NUM
##          "numeric"          "numeric"          "numeric"
##               PART               PRON              PROPN
##          "numeric"          "numeric"          "numeric"
##              PUNCT              SCONJ                SYM
##          "numeric"          "numeric"          "numeric"
##               VERB                  X              sent2
##          "numeric"          "numeric"           "factor"
##             freq_m                sum    male_prob_words
##          "numeric"          "numeric"          "numeric"
##     gender_bywords          WordCount        SentimentGI
##          "numeric"          "numeric"          "numeric"
```

```
##       NegativityGI       PositivityGI        SentimentHE
##          "numeric"          "numeric"          "numeric"
##       NegativityHE       PositivityHE        SentimentLM
##          "numeric"          "numeric"          "numeric"
##       NegativityLM       PositivityLM RatioUncertaintyLM
##          "numeric"          "numeric"          "numeric"
##      SentimentQDAP     NegativityQDAP     PositivityQDAP
##          "numeric"          "numeric"          "numeric"
```

```
filegender$sent2 <-as.numeric(filegender$sent2)
filegender$emoticon <-as.numeric(filegender$emoticon)
filegender$emoji <-as.numeric(filegender$emoji)
filegendernumeric <-select_if(filegender, is.numeric)
res <-cor(filegendernumeric, method = "pearson", use = "complete.obs")
res <-corrplot(res,type = "upper", method="circle", number.font=2,
tl.cex=0.6)
```



```r
#vemos la correlación con la variable target del resto de variables

gender <- cor(filegendernumeric, method = 'pearson', use =
'pairwise.complete.obs')[4,-4]
gender
```

```
##          x_unit_id         link_color      sidebar_color
##        0.143542975        0.106022866        0.042728597
##  gender.confidence         fav_number      retweet_count
##        0.059371530        0.037894874       -0.017699933
##       tweet_count            emoticon              emoji
##       -0.034832610        0.001411312        0.137848266
##     gender_by_name                 ADJ                ADP
##        0.435083397       -0.012346899        0.001372972
##                ADV                AUX               CCONJ
##       -0.010611954       -0.019429197       -0.014183237
##                DET               INTJ               NOUN
##       -0.008039218        0.007863439       -0.002406457
##                NUM               PART               PRON
##        0.010828344       -0.006870645       -0.002724647
##              PROPN              PUNCT               SCONJ
##        0.002593037       -0.004709101       -0.008741471
##                SYM               VERB                  X
##        0.004621625       -0.006230027       -0.005298758
##              sent2             freq_m                sum
##        0.004746050       -0.004967096       -0.005872638
##    male_prob_words     gender_bywords          WordCount
##       -0.001309761        0.006248525       -0.044806833
##        SentimentGI       NegativityGI       PositivityGI
##        0.001916965        0.003204282        0.005237660
##        SentimentHE       NegativityHE       PositivityHE
##       -0.024807304        0.012943330       -0.020924300
##        SentimentLM       NegativityLM       PositivityLM
##        0.014871942       -0.022889115       -0.003138123
## RatioUncertaintyLM      SentimentQDAP     NegativityQDAP
##       -0.004734596        0.008809999        0.003415795
##     PositivityQDAP
##        0.014772460
```

```r
#write.csv2(res,"F:/NitroPC/Google Drive/BDATA/proyecto/gender-
classifier-DFE-791531.csv/correlationmatrix.txt")

#parece que el color del link (0,10), el hecho que haya emojis(0.137) y
el nombre de usuario(gender_by_name) (0.43) son las variables más
correlacionadas con el género de usuario.
#También la predicción de genero por palabras usadas (0.19276),pero
tampoco lo alta que hubieramos pensado al inicio del Proyecto.
```

```
Veremos la composición de los tweets: número de determinantes, adjetivos, adverbios, verbos, conjunciones..


#no vemos ninguna más significativa de las anteriores encontradas.


#gender_by_name <-cor(filegendernumeric, method = 'pearson', use = 'pairwise.complete.obs')[11,-11]
#gender_by_name

#gender_by_words <-cor(filegendernumeric, method = 'pearson', use = 'pairwise.complete.obs')[33,-33]
#gender_by_words
```

# 11.-MODELO NO SUPERVISADO

```
# Vamos a construir un modelo de clusterización con dos clusters, a ver si conseguimos clasificar correctamente los grupos en funcion de sus caracteristicas iguales y si el modelo los agrupa según género.

##################MODELO  NO SUPERVISADO#####################################
library(knitr)
#clusterizacion
filegender_m <-filegender%>%select(-1,-3,-(5:14),-17,-(37:39), -(42:54))
filegender_m <-na.omit(filegender_m )
filegender_m <- filegender_m %>% mutate_if(is.numeric,as.factor)
set.seed(1234)
library(dplyr)
gender_train<-sample_frac(filegender_m, 0.65)
sid<-as.numeric(rownames(gender_train)) # because rownames() returns character
gender_test<-gender_train[-sid,]
str(gender_train)

#check de split
prop.table(table(gender_train$gender))

##
##         0         1
## 0.3801978 0.6198022

prop.table(table(gender_test$gender))

##
##         0         1
## 0.3893295 0.6106705
```

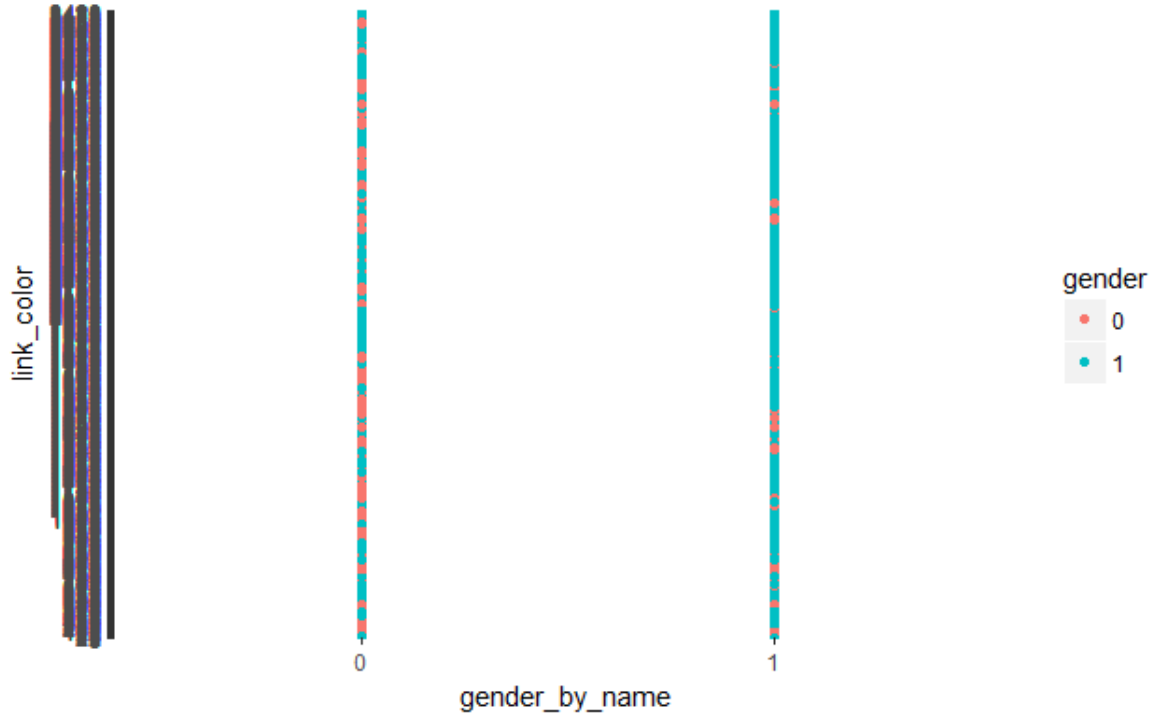
```r
set.seed(1234)
kmeans_clust <- kmeans(filegender_m, 2)
kmeans_clust
```

```
## K-means clustering with 2 clusters of sizes 10678, 1458
##
## Cluster means:
##   link_color    gender emoticon    emoji gender_by_name      ADJ
ADP
## 1   399.8952 0.6010489 1.038022 1.222514      0.6016108 1.086533
1.292190
## 2  2482.8745 0.7379973 1.057613 1.233882      0.6543210 1.069273
1.248285
##         ADV       AUX     CCONJ      DET       INTJ     NOUN       NUM
## 1 0.9031654 0.7065930 0.7125866 1.231129 0.08812512 3.799588 0.3835924
## 2 0.8237311 0.7064472 0.6776406 1.261317 0.08847737 3.816187 0.4032922
##        PART     PRON      PROPN     PUNCT     SCONJ      SYM     VERB
## 1 0.3658925 1.564712 0.04729350 0.3920210 0.1952613 1.575295 2.036898
## 2 0.3806584 1.491770 0.04252401 0.4190672 0.2071331 1.665981 2.021262
##            X    sent2 gender_bywords WordCount
## 1 0.05047762 2.241806      0.1165949  7.962165
## 2 0.03840878 2.192044      0.1262003  7.897119
##
## Clustering vector:
##     [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 696052105 506087247
##  (between_SS / total_SS =  82.2 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
table(filegender_m$gender, kmeans_clust$cluster)
```

```
##
##        1    2
##   0 4260  382
##   1 6418 1076
```

```r
#  No queda muy claro a que cluster corresponde cada género. Vamos a ver
si lo vemos mejor gráficamente.
#PLOT
library(ggplot2)

ggplot(filegender_m, aes(gender_by_name, sidebar_color, color
=gender))+geom_point()
kmeans_clust$centers
```




```
#Parece que el cluster 1 corresponde mas a mujeres y el cluster 2 a
hombres pero este segundo no está del todo claro.



##   link_color    gender emoticon    emoji gender_by_name      ADJ
ADP
## 1   399.8952 0.6010489 1.038022 1.222514      0.6016108 1.086533
1.292190
## 2  2482.8745 0.7379973 1.057613 1.233882      0.6543210 1.069273
1.248285
##         ADV       AUX     CCONJ       DET       INTJ     NOUN       NUM
## 1 0.9031654 0.7065930 0.7125866 1.231129 0.08812512 3.799588 0.3835924
## 2 0.8237311 0.7064472 0.6776406 1.261317 0.08847737 3.816187 0.4032922
##        PART     PRON      PROPN     PUNCT     SCONJ      SYM     VERB
## 1 0.3658925 1.564712 0.04729350 0.3920210 0.1952613 1.575295 2.036898
## 2 0.3806584 1.491770 0.04252401 0.4190672 0.2071331 1.665981 2.021262
##            X    sent2 gender_bywords WordCount
## 1 0.05047762 2.241806      0.1165949  7.962165
## 2 0.03840878 2.192044      0.1262003  7.897119

#unimos en la tabla final
filegender_m_clust <-filegender_m%>% mutate(cluster_id =
kmeans_clust$cluster)
kable(head(filegender_m_clust))
```

# 12.-MODELO SUPERVISADO

```
#Vamos a construir un modelo predictivo en base a clasificación binaria
segun el modelo Naives Bayes, por su sencillez y porque es el ideal para
este caso.


#######MODELO DE APRENDIZAJE NO SUPERVISADO##############################
##NAIVES BAYES
library(e1071)
library(gmodels)

#Creamos el modelo y usamos los dataset de traininig y testing que ya
teniamos creado.
gender_model <- naiveBayes(gender ~., gender_train)

gender_pred <-predict(gender_model,gender_train)
CrossTable(gender_pred, gender_train[,3], prop.chisq =FALSE, proc.c=
FALSE, prop.r=FALSE, dnn =c('actual_gender', 'predicted_gender'))


##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  7888
##
##
##                | predicted_gender
## actual_gender |         1 |         2 | Row Total |
## --------------|-----------|-----------|-----------|
##             0 |      1771 |      1126 |      2897 |
##               |     0.591 |     0.230 |           |
##               |     0.348 |     0.143 |           |
## --------------|-----------|-----------|-----------|
##             1 |      1228 |       221 |      4991 |
##               |     0.409 |     0.697 |           |
##               |     0.156 |     0.477 |           |
## --------------|-----------|-----------|-----------|
##  Column Total |      2999 |       317 |      7888 |
##               |     0.380 |     0.620 |           |
## --------------|-----------|-----------|-----------|
##
##

#Realzamos Tambien la matriz de Confusion.
```

```
##
##  Confusion Matrix and Statistics
##
##            Reference
##
##  Prediction    0    1
##
##           0 1771 1126
##           1 1228 3763
##
##                Accuracy : 0.7016
##                  95% CI : (0.6913, 0.7117)
##     No Information Rate : 0.6198
##     P-Value [Acc > NIR] : < 2e-16
##                   Kappa : 0.3626
##  Mcnemar's Test P-Value : 0.03737
##             Sensitivity : 0.5905
##             Specificity : 0.7697
##          Pos Pred Value : 0.6113
##          Neg Pred Value : 0.7540
##              Prevalence : 0.3802
##          Detection Rate : 0.2245
##    Detection Prevalence : 0.3673
##       Balanced Accuracy : 0.6801
##
##        'Positive' Class : 0
```

```
#vemos que tiene un acierto del 70% pero clasifica major el genero mujer
(69%) que el hombre en el que solo tiene un acierto del 59%.

# vamos a ver como se comporta con el dataset de test.

##EVALUATION MODEL PERFORMANCE
library(gmodels)
gender_pred_test <-predict(gender_model, gender_test)

CrossTable(gender_pred_test, gender_test[,3], prop.chisq =FALSE, proc.c=
FALSE, prop.r=FALSE, dnn =c('actual_gender', 'predicted_gender'))
```

```
## ##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##  ## Total Observations in Table:  2774
##
##                | predicted_gender
## actual_gender |         1 |         2 | Row Total |
## --------------|-----------|-----------|-----------|
```

```
##             0 |       622 |       394 |      1016 |
##               |     0.576 |     0.233 |           |
##               |     0.224 |     0.142 |           |
## --------------|-----------|-----------|-----------|
##             1 |       458 |        77 |      1758 |
##               |     0.424 |     0.767 |           |
##               |     0.165 |     0.469 |           |
## --------------|-----------|-----------|-----------|
##  Column Total |      1080 |       111 |      2774 |
##               |     0.389 |     0.611 |           |
## --------------|-----------|-----------|-----------|
```

```
# vemos que el resultado prácticamente es el mismo, con ligeras
variaciones solamente.

###PREDICCION DE GENDER EN TODO EL DATASET.
gender_pred_file <- predict(gender_model, filegender_m)

CrossTable(gender_pred_file, filegender_m[,3], prop.chisq =FALSE, proc.c=
FALSE, prop.r=FALSE, dnn =c('actual_gender', 'predicted_gender'))
```

```
## ##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
## ##
## Total Observations in Table:  12136
## ##
##                | predicted_gender
## actual_gender |         1 |         2 | Row Total |
## --------------|-----------|-----------|-----------|
##             0 |      2734 |      1699 |      4362 |
##               |     0.589 |     0.227 |           |
##               |     0.225 |     0.140 |           |
## --------------|-----------|-----------|-----------|
##             1 |      1908 |      5795 |      7703 |
##               |     0.411 |     0.773 |           |
##               |     0.157 |     0.478 |           |
## --------------|-----------|-----------|-----------|
##  Column Total |      4642 |      7494 |     12136 |
##               |     0.382 |     0.618 |           |
## --------------|-----------|-----------|-----------|
```

```
table(gender_pred_file, filegender_m$gender)
```

```
##
## gender_pred_file    0    1
##                0 2723 1639
##                1 1919 5855
```

# 13.-CONCLUSION

*#Tanto el modelo de clusterización como el modelo de clasificación tienen un acierto de predicción de las mujeres a cosiderar, mientras que el % de acierto el genero másculino es mucho más bajo.*

*#Parece que los hombres y las mujeres nos expresamos de manera similar en las redes sociales y dificulta la predicción.*

*La predicción en función del nombre de usuario has sido más acertada de lo que esperaba. Casi un acierto del 50%. Un error del 17%  y la unica pega son los usuarios con nombre "raros".*

*Mi intención en un inicio era poder tratar tambien los archivos de las fotos de los perfiles para ver si nos ayudaban en la predicción, pero hubiera "pesado" demasiado y no lo he podido ni probar por falta de tiempo. Queda pendiente para un estudio posterior.*

# 14.-ANEXO

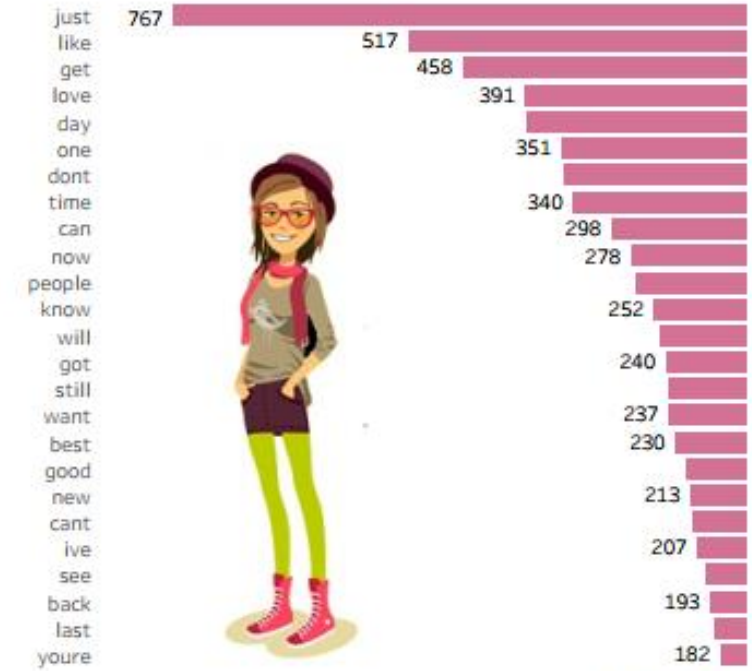
```
#Dejo en este anexo el estudio de los emojis, comentado anteriormente. La
idea era contarlos y tenerlos en cuenta en el sentiment analysis.


#################ESTUDIO SENTIMIENTO EN BASE A EMOJIS###############

#load dictionary function

load_dict_emoji <- function(emoji_file = "emojis.csv"){
  # Download emoji_file if it does not exist
  if (!file.exists(emoji_file)) {
    emoji_file <- "emojis.csv"
    emoji_file_url <-
      paste0(
        "https://raw.githubusercontent.com/today-is-a-good-
day/emojis/master/",
        emoji_file)

    download.file(emoji_file_url, destfile = emoji_file)
  }

  readr::read_delim(
    emoji_file,
    delim = ";",
```

```
    col_names = c("number","unicode","EN", "ES", "tag", "utf8","ftu8"),
    skip = 1,
    progress = FALSE
  ) %>%
    mutate(EN = tolower(EN)) %>%
    rename(description = EN, descripcion_espanyol=ES, r.encoding = ftu8)
}

#cargamos el diccionari de emojis con la descripcion, r.encoding,
descripcion
EmDict_raw<-load_dict_emoji()

## Parsed with column specification:
## cols(
##   number = col_integer(),
##   unicode = col_character(),
##   EN = col_character(),
##   ES = col_character(),
##   tag = col_character(),
##   utf8 = col_character(),
##   ftu8 = col_character()
## )

#limpiamos descripciones
# plain skin tones
skin_tones <- c("light skin tone",
                "medium-light skin tone",
                "medium skin tone",
                "medium-dark skin tone",
                "dark skin tone")

# remove plain skin tones and remove skin tone info in description
library(Unicode)
emDict <- EmDict_raw %>%
  # remove plain skin tones emojis
  filter(!description %in% skin_tones) %>%
  # remove emojis with skin tones info, e.g. remove woman: light skin
tone and only
  # keep woman
  filter(!grepl(":", description)) %>%
  mutate(description = tolower(description))%>%
  mutate(unicode = as.u_char(unicode))

#nos quedamos solo con descripcion, r.encoding, unicode, utf8
#emDict <- emDict%>% select(description, r.encoding, unicode, utf8)
emDict$description <- cleaning(emDict$description)
matchto <- emDict$r.encoding
description <- emDict$description


# this function outputs the emojis found in a string as well as their
occurences
```

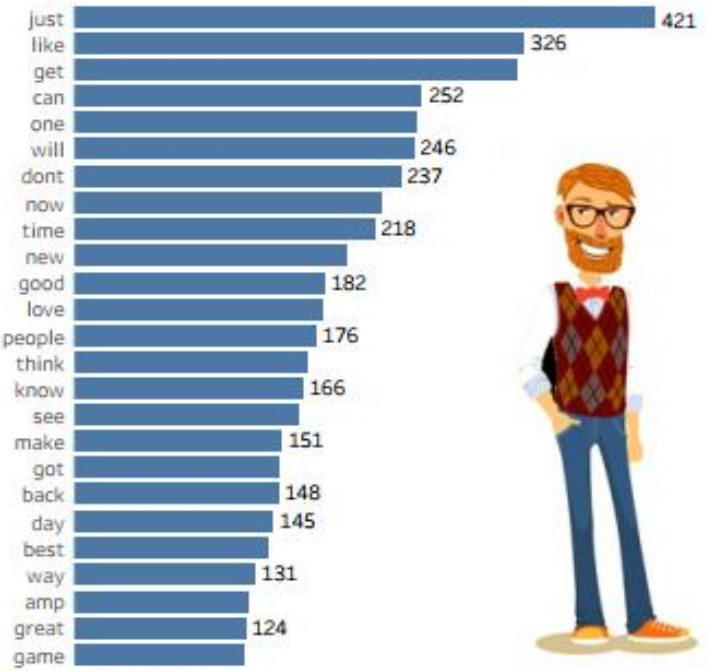
```
count_matches <- function(string, matchto, description, sentiment = NA) {

  vec <- str_count(string, matchto)
  matches <- which(vec != 0)

  descr <- NA
  cnt <- NA

  if (length(matches) != 0) {

    descr <- description[matches]
    cnt <- vec[matches]

  }
df <- data.frame(text = string, description = descr, count = cnt)

  return(df)

}

# this function applies count_matches on a vector o texts and outputs a
data.frame
emojis_matching <- function(texts, matchto, description, sentiment = NA)
{

  texts %>%
    lapply(count_matches, matchto = matchto, description = description,
sentiment = sentiment) %>%
    bind_rows
  }

tweets <- emojis_matching(file$clean_tweet, matchto, description) %>%
  group_by(text) %>%
  summarise(n = sum(count)) %>%
  #merge(file, by.y = "clean_tweet", by.x="text") %>%
  merge(file, by="text") %>%
  select(text,n) %>%
  arrange(-n)
```

# 15.-TABLEAU

# PALABRAS MÁS USUALES





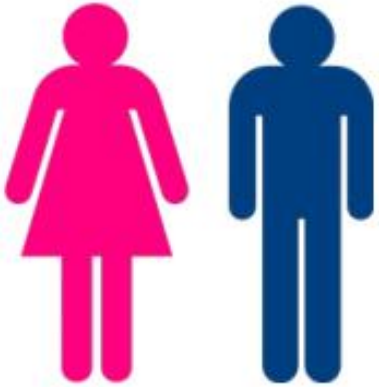






## MUESTRA ESTUDIADA

8.536 users

6.046 users

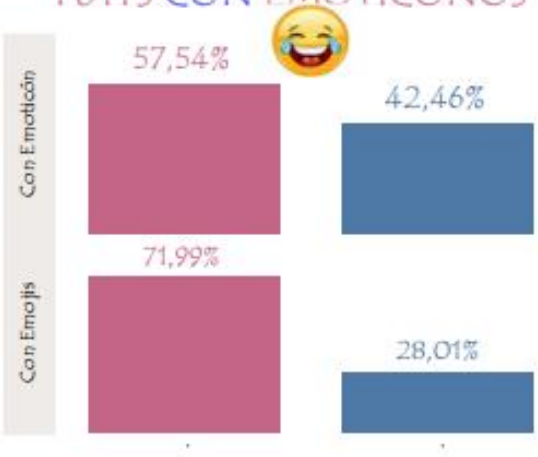


## PREDICCIÓN GÉNERO USUARIO SEGÚN SUS TUITS

por Núria Redó

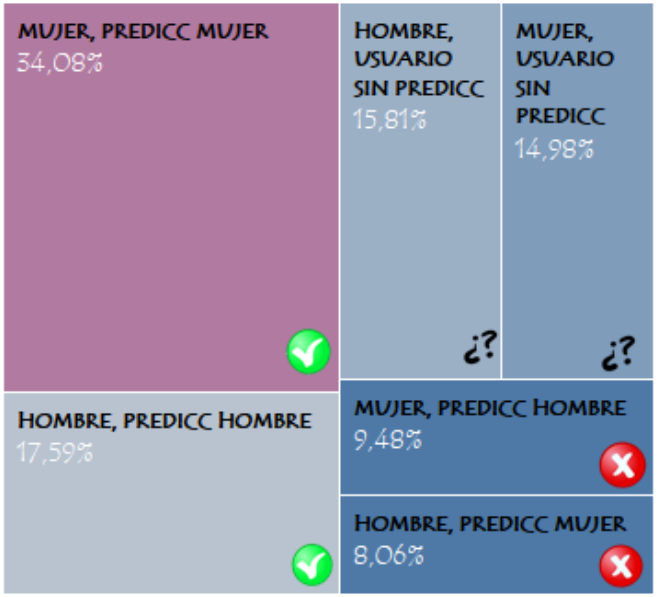
| | | |
|---|---|---|
| SENT. NEGATIVO | 57,95% | 42,05% |
| SENT. NEUTRAL | 59,40% | 40,60% |
| SENT. POSITIVO | 58,28% | 41,72% |

## TUITS CON EMOTICONOS



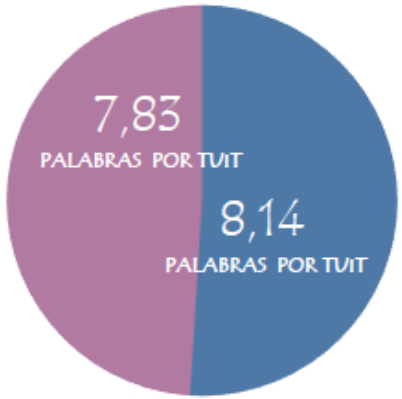
| | | |
|---|---|---|
| aghsmckenzie | PREDICC MUJER | MUJER |
| baileyyq | PREDICC MUJER | MUJER |
| haiingrid | PREDICC MUJER | MUJER |
| kirstenjjp | PREDICC MUJER | MUJER |
| laineygillon | PREDICC MUJER | MUJER |
| lousieaghes | PREDICC MUJER | MUJER |
| molalathommy | PREDICC MUJER | HOMBRE |
| monicalym | PREDICC MUJER | MUJER |
| pjongup | PREDICC MUJER | MUJER |
| swaggysational | PREDICC MUJER | MUJER |
| dinahstyreigns | PREDICC MUJER | MUJER |
| vantillasqueen | PREDICC MUJER | MUJER |

## RESULTADO DE LA PREDICCIÓN SEGÚN NOMBRE USUARIO




## PROMEDIOS PART-OF-SPEECH TAGGING

| Prom. Intj | Prom. DET | Prom. Part | Prom. Noun | Prom. ADV | Prom. Verb | |
|---|---|---|---|---|---|---|
| 0,08 | 1,24 | 0,38 | 3,79 | 0,92 | 2,06 |  |
| 0,09 | 1,24 | 0,37 | 3,80 | 0,89 | 2,02 |  |


## COLOR LINK

| MUJER | HOMBRE |
|---|---|
| 28,77% tuits | 26,34% tuits |
| 3,18% tuits | 0,57% tuits |
| 2,90% tuits | 0,19% tuits |
| 2,18% tuits | 0,68% tuits |
| 1,78% tuits | 1,39% tuits |
| 1,76% tuits | 1,14% tuits |
| 1,64% tuits | 2,24% tuits |
| 1,35% tuits | 1,18% tuits |
| 1,34% tuits | 2,59% tuits |
| 1,23% tuits | 0,31% tuits |
| 1,18% tuits | 0,69% tuits |
| 1,14% tuits | 0,04% tuits |
| 1,03% tuits | 0,35% tuits |
| 0,94% tuits | 1,02% tuits |
| 0,84% tuits | 0,29% tuits |
| 0,80% tuits | 1,77% tuits |
| 0,73% tuits | 0,34% tuits |
| 0,71% tuits | 0,69% tuits |
| 0,60% tuits | 0,10% tuits |
| 0,57% tuits | 0,37% tuits |
| 0,54% tuits | 0,08% tuits |
| 0,50% tuits | 0,15% tuits |
| 0,37% tuits | 0,20% tuits |
| 0,29% tuits | 0,41% tuits |
| 0,22% tuits | 0,27% tuits |