**UNIVERSIDAD CENTRAL DEL ECUADOR**

**FACULTAD DE INGENIERÍA, CIENCIAS FÍSICAS Y MATEMÁTICA**

**INGENIERÍA INFORMÁTICA**

**ANÁLISIS DE DATOS**



**PROYECTO ANALISIS DE DATOS**

**RADIOS DEL ECUADOR**



**INTEGRANTES DEL GRUPO:**

Alexis Bautista

Bryan Catucuamba

David Jácome

Alejandro Naranjo

Richard Quimbiulco

**Contenido**

[TEMA 2](#_Toc489479595)

[OBJETIVOS 3](#_Toc489479596)

[Objetivo general 3](#_Toc489479597)

[Objetivos específicos 3](#_Toc489479598)

[DATOS 3](#_Toc489479599)

[Criterios de selección 3](#_Toc489479600)

[Extracción 3](#_Toc489479601)

[Depuración 4](#_Toc489479602)

[BASE TEÓRICA 5](#_Toc489479603)

[**Aprendizaje no supervisado** 5](#_Toc489479604)

[**Aprendizaje supervisado** 6](#_Toc489479605)

[**Máquinas de Soporte Vectorial** 6](#_Toc489479606)

[**Índices de relevancia** 7](#_Toc489479607)

[**Factor de ponderación** 7](#_Toc489479608)

[RESULTADOS 7](#_Toc489479609)

[APRENDIZAJE SUPERVISADO 7](#_Toc489479610)

[APRENDIZAJE NO SUPERVISADO 9](#_Toc489479611)

[FACTORES DE PONDERACIÓN 11](#_Toc489479612)

[ANALISIS DE LA RELEVANCIA DE LOS ACTORES 13](#_Toc489479613)

[Análisis con los textos clasificados con el aprendizaje supervisado 13](#_Toc489479614)

[Análisis con los textos clasificados con el aprendizaje no supervisado 17](#_Toc489479615)

[CONCLUSIONES Y RECOMENDACIONES 20](#_Toc489479616)

[REFERENCIAS 20](#_Toc489479617)

[ANEXOS 21](#_Toc489479618)

[SCRIPTS UTILIZADOS 21](#_Toc489479619)

# TEMA

El análisis de audiencias se define como el conjunto de métodos de investigación utilizados para obtener información sobre la audiencia, y que permite estimar el número de personas que han estado expuestos a un evento medial en un momento del tiempo determinado. Este puede referirse por ejemplo al consumo de un producto audiovisual, ya sea textual, sonoro, fotográfico o multimedia.

# OBJETIVOS

## Objetivo general

Identificar cual o cuales son los artistas musicales que la gente más solicita a través de Twitter para realizar un análisis de los datos obtenidos de las principales cadenas radiales musicales y concluir cual es la demanda de los usuarios.

## Objetivos específicos

* Determinar las fuentes de los datos y depurar los datos para su análisis.
* Identificar la mejor métrica de polaridad para los datos, ya sea usando diccionario de palabras o máquinas de soporte vectoriales.
* Identificar cual es el mejor kernel para máquinas de soporte vectorial que se ajusta a la clasificación de textos.
* Establecer un índice de relevancia que mejor nos permita valorar la importancia de los artistas obtenidos.
* Estimar un factor de ponderación a cada provincia del Ecuador según los datos recogidos y los datos censales.

# DATOS

## Criterios de selección

Se han descargado los datos desde el 10 de Junio hasta el Lunes 24 de Julio de las diferentes emisoras de radio elegidas de acuerdo al criterio de: mayor cantidad de seguidores.

Elegimos las radios musicales con más seguidores en el Ecuador:

|  |  |
| --- | --- |
| EMISORA | NÚMERO DE SEGUIDORES |
| La Bruja | 86700 |
| Canela | 59900 |
| Sucre | 54100 |
| Exa | 39000 |
| Alfa | 26700 |
| WQ Radio | 17800 |
| Metro | 11800 |
| La Otra | 11600 |

## Extracción

Para la descarga de tweets usamos el siguiente código:

#usamos las libreria necesaria

library(twitteR)

#credenciales de acceso para descarga de tweets

consumer\_key <-"RFHeP55qj0ejWv7YiSceWTphX"

consumer\_secret<-"YOOUNzPvlJCAIHo23AWGU6hRm9VYxP6AY60H0n3u3dBM44aHZS"

access\_token<-"366852754-QE8L7hZs1J0WbtDTImo4P3qxzabRZ6OEY0E9ckOs"

access\_secret<-"zUZ3rnyPHCLQxso43A8PvXY1a3jWp81yS8Z2Rr0w96YzB"

#iniciamos las conexión para la descarga de tweets

setup\_twitter\_oauth(consumer\_key, consumer\_secret, access\_token, access\_secret)

#empezamos la descarga de tweets

listaTweets <- searchTwitter("metrostereo", n=1000)

#guardamos los tweets obtenidos en un archivo .csv

tweets <- twListToDF(listaTweets)

write.csv(tweets, 'tweets-metro-1.csv')

Para el posterior análisis necesitamos el número de seguidores, la localidad del tweet, el número de veces que ha recibido un "me gusta" un tweet, obtenemos esta información con el siguiente código:

#Aumentamos las columnas necesarias en la base de datos para el posterior #análisis:

baseMetro$statusCount <- c(rep(NA, NROW(baseMetro)))

baseMetro$followers <- c(rep(NA, NROW(baseMetro)))

baseMetro$favorites <- c(rep(NA, NROW(baseMetro)))

baseMetro$friends <- c(rep(NA, NROW(baseMetro)))

baseMetro$location <- c(rep(NA, NROW(baseMetro)))

for (i in 1:NROW(base)) {

usuario <- as.character(base[i, 12]) tryCatch({

informacionUsuario <- getUser(usuario) #obtenemos el usuario que envio el tweet

base[i, 20] <- statusesCount(informacionUsuario)

base[i, 21] <- followersCount(informacionUsuario) #seguidores

base[i, 22] <- favoritesCount(informacionUsuario)

base[i, 23] <- friendsCount(informacionUsuario)

base[i, 24] <- location(informacionUsuario) #localidad

})

}

## Depuración

Para la depuración de datos hemos utilizado el siguiente código, aplicado a cada uno de los archivos de datos descargados de las diferentes emisoras seleccionadas:

Debemos tener en cuenta que la "codificación" que usamos para ver los tweets en este caso es UTF-8.

#DEPURACION DE DATOS

#RADIO METRO

#depuracion de los textos de los tweets

for (i in 1:NROW(baseMetro)) {

texto <- baseMetro[i, 2]

#eliminar simbolos de retweets

sinRT <- gsub("(RT|via)((?:\\b\\W\*@\\w+)+)", "", texto)

#eliminar simbolo @ y #

sinCuentas <- gsub("@|#", "", sinRT)

#eliminar signos de puntuacion

sinSimbolos <- gsub("[[:punct:]].", "", sinCuentas)

#eliminar numeros

sinNumeros <- gsub("[[:digit:]]", "", sinSimbolos)

#eliminar enlaces

sinEnlaces <- gsub("http\\w+", "", sinNumeros)

baseMetro$depurado[i] <- sinEnlaces

baseMetro$sinEspacios[i] <- gsub("[[:space:]]|", "", baseMetro$depurado[i])

}

Una vez depurados los datos ordenamos la base de datos de acuerdo al texto depurado sin espacios:

#ordenamos al dataframe por el texto depurado sin espacios

baseMetro <- baseMetro[order(baseMetro$sinEspacios), ]

Eliminamos los tweets que están vacíos:

#eliminamos tweets que quedaron vacios

baseMetro <- baseMetro[!(baseMetro$sinEspacios == ""), ]

Eliminamos los tweets que están repetidos:

#eliminamos repetidos

baseMetro <- baseMetro[!duplicated(baseMetro$sinEspacios), ]

Obtenemos la siguiente tabla de resultados:

|  |  |  |
| --- | --- | --- |
| Emisora | Cantidad de tweets  Antes de la depuración | Cantidad de tweets  Después de la depuración |
| La Bruja | 3446 | 1511 |
| Canela | 13300 | 1722 |
| Sucre | 2050 | 700 |
| Exa | 1383 | 514 |
| Alfa | 1328 | 700 |
| WQ Radio | 12692 | 1240 |
| Metro | 4707 | 1106 |
| La Otra | 6679 | 572 |
| TOTAL | 45585 | 8065 |

**Base de datos final depurada:**

Número de observaciones: **8065**

# BASE TEÓRICA

## **Aprendizaje no supervisado**

Todo el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan sólo por entradas al sistema. No se tiene información sobre las categorías de esos ejemplos. Por lo tanto, en este caso, el sistema tiene que ser capaz de reconocer patrones para poder etiquetar las nuevas entradas.

Dentro del campo de investigación de la minería de datos, haciendo uso de la lingüística computacional, el procesamiento del lenguaje natural (PLN) y el análisis de textos, se halla una disciplina llamada análisis de sentimientos (*sentiment analysis),* también citada en la bibliografía como minería de sentimientos (*opinion mining)* o análisis de subjetividad (*subjetiviy analysis).*

El análisis sentimental, según Jorge Carrillo de Albornoz (2011), “hace referencia a la tarea de análisis, identificación y clasificación de todo tipo de contenido emocional, subjetivo u opinado”.

En términos generales, el análisis sentimental tiene como objetivo determinar la actitud de un orador o un escritor con respecto a algún tema, o la polaridad global contextual de un documento.

## **Aprendizaje supervisado**

### **Máquinas de Soporte Vectorial**

**Definición:**

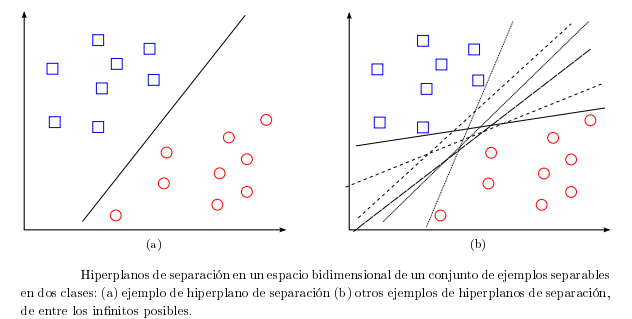
Dado un conjunto de puntos, subconjunto de un conjunto mayor (espacio), en el que cada uno de ellos pertenece a una de dos posibles categorías, un algoritmo basado en SVM construye un modelo capaz de predecir si un punto nuevo (cuya categoría desconocemos) pertenece a una categoría o a la otra.

Como en la mayoría de los métodos de clasificación supervisada, los datos de entrada (los puntos) son vistos como un vector *p*-dimensional (una lista ordenada de *p* números).

La SVM busca un hiperplano que separe de forma óptima a los puntos de una clase de la de otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior.

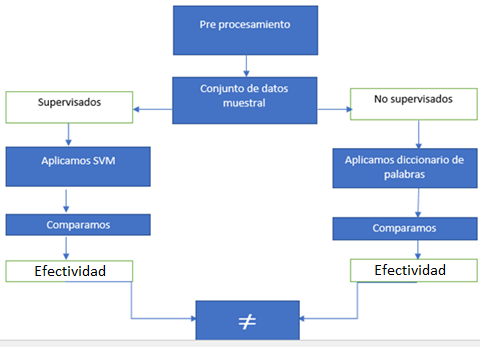
En ese concepto de "separación óptima" es donde reside la característica fundamental de las SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado. (Support-vector-machines.org, 2017)

Los algoritmos SVM pertenecen a la familia de los clasificadores lineales.



**Máquina de Soporte Vectorial para clasificación**

Entre las aplicaciones más relevantes de las SVM se encuentra la Clasificación, el problema de la clasificación puede reducirse a examinar dos clases sin pérdida de generalidad. La tarea es encontrar un clasificador que funcione bien en datos futuros, es decir que generalice bien la clasificación.



## **Índices de relevancia**

Consiste en medir la importancia de una cuenta de twitter o el contenido de un tuit. En muchas ocasiones, para medir la importancia de una cuenta de Twitter, tendemos a fijarnos en una sola métrica: Los seguidores, sin embargo existen algunas otras de donde se puede obtener información valiosa que se puede interpretar y monitorizar.

## **Factor de ponderación**

Consiste en convertir los resultados de los valores caracterizados a una unidad común y sumable, multiplicándolos por su factor de ponderación. Posteriormente se suman todos ellos para obtener una puntuación única del impacto que tienen.

La fórmula empleada para encontrar el factor de ponderación es el siguiente:

# RESULTADOS

## APRENDIZAJE SUPERVISADO

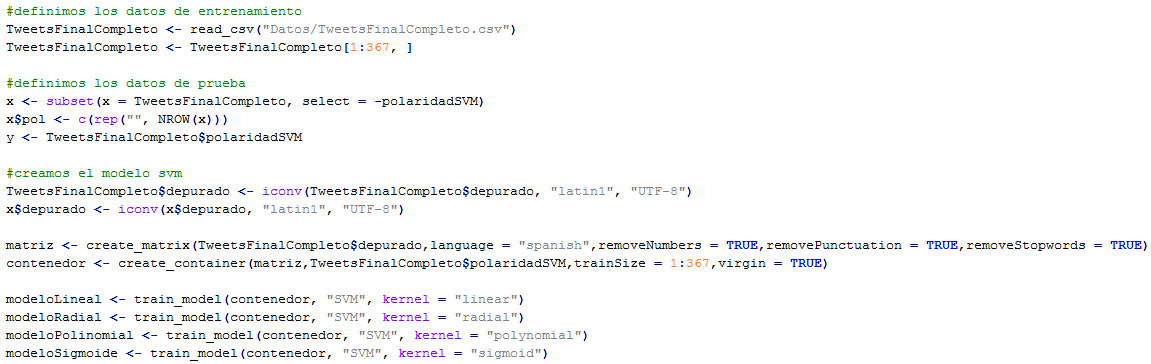
Para el análisis supervisado el algoritmo usado es máquinas de soporte para lo cual del total de 8065 se extrajeron 367 datos del total de los datos para ser de entrenamiento y 7698 datos serán para prueba. La cantidad de datos de entrenamiento lo determinamos usando la fórmula para calcular el tamaño de la muestra:



Luego de obtener aleatoriamente los 367 datos de la muestra se les asigna un valor de polaridad según el contenido del tuit.

A partir de los datos de entrenamiento se le aplica el algoritmo de máquinas de soporte al resto de datos.

Antes de aplicar el modelo de máquinas de soporte vectorial, debemos verificar cuál es el kernel de SVM que mejor se aplica a la clasificación de textos, para ello hemos encontrado el modelo SVM usando los datos de entrenamiento y los diferentes kernels que existen. Cabe recalcar que los datos de prueba para verificar cual modelo se ajustan más son los datos de entrenamiento, pero su clasificación con SVM se lo puso en una nueva columna. Una vez clasificados los textos, encontramos las siguientes tablas de confusión con los diferentes núcleos.



KERNEL LINEAL

|  |  |  |  |
| --- | --- | --- | --- |
| **Resultados** | | **Predichos** | |
| **0** | **1** |
| **Reales** | **0** | 293 | 0 |
| **1** | 0 | 74 |

Una vez clasificados los tweets con SVM con el núcleo lineal, el modelo nos dice que de los 293 tweets negativos son todos negativos y de los 74 tweets positivos son todos positivos, por lo que obtuvimos que el porcentaje de aciertos del modelo es del 100%.

KERNEL RADIAL

|  |  |  |  |
| --- | --- | --- | --- |
| **Resultados** | | **Predichos** | |
| **0** | **1** |
| **Reales** | **0** | 293 | 0 |
| **1** | 10 | 64 |

Una vez clasificados los tweets con SVM con el núcleo radial, el modelo nos dice que de los 293 tweets negativos todos pasaron a ser negativos y de los 74 tweets positivos 10 pasaron a ser negativos, por lo que obtuvimos que el porcentaje de aciertos del modelo es del 97,28%.

KERNEL POLINOMIAL

|  |  |  |  |
| --- | --- | --- | --- |
| **Resultados** | | **Predichos** | |
| **0** | **1** |
| **Reales** | **0** | 293 | 0 |
| **1** | 74 | 0 |

Una vez clasificados los tweets con SVM con el núcleo polinomial, el modelo nos dice que de los 293 tweets negativos todos pasaron a ser negativos y de los 74 tweets positivos todos pasaron a ser negativos, por lo que obtuvimos que el porcentaje de aciertos del modelo es del 79,84%.

KERNEL SIGMOIDE

|  |  |  |  |
| --- | --- | --- | --- |
| **Resultados** | | **Predichos** | |
| **0** | **1** |
| **Reales** | **0** | 293 | 0 |
| **1** | 18 | 56 |

Una vez clasificados los tweets con SVM con el núcleo sigmoide, el modelo nos dice que de los 293 tweets negativos todos pasaron a ser negativos y de los 74 tweets positivos 18 pasaron a ser negativos, por lo que obtuvimos que el porcentaje de aciertos del modelo es del 95,1%.

Una vez que hemos determinado cual es el kernel que mejor se ajusta para la clasificación de textos, que en este caso es el lineal, procedemos a clasificar los datos de prueba.

# Creamos una matriz de los tweets

matriz <- create\_matrix(TweetsFinal$depurado,

language = "spanish",

removeNumbers = TRUE,

removePunctuation = TRUE,

removeStopwords = TRUE)

# Creamos un contenedor para separar de la matriz anterior, los tweets

# de entrenamiento y los tweets de prueba y escojemos la variable dependiente

# que en este caso es la polaridad

contenedor <- create\_container(matriz, TweetsFinal$polaridadSVM,

trainSize = 1:367,

testSize = 368:8065,

virgin = FALSE)

# Con el comando train\_model vamos a crear el modelo SVM

model <- train\_model(contenedor, "SVM", kernel="linear")

# Con el comando classify\_model clasificaremos los tweets de prueba

clasificacion <- classify\_model(contenedor, model)

# Añadimos la clasificación de los tweets de prueba en el dataframe

TweetsFinal$polaridadSVM[368:8065] <- as.numeric(clasificacion$SVM\_LABEL)-1

Una vez realizada la clasificación verificamos que, de los 7698 tweets de prueba, 6808 fueron tweets negativos y 890 tweets fueron positivos, es decir, son tweets con peticiones de artistas.

## APRENDIZAJE NO SUPERVISADO

Para el caso de aprendizaje no supervisado se ha usado un diccionario con 44 palabras positivas que verifiquen cuales son los tweets que corresponden a una petición de un artista.

Cuando el tuit ya se encuentra depurado se contabiliza las palabras que coinciden con las del diccionario de palabras, una vez finalizado el conteo se compara entre el número de palabras positivas con un valor de 2 porque usualmente las peticiones se las realiza con un mínimo de 2 palabras, luego con la comparación se determina la polaridad respectiva.



#Agregamos el diccionario con las palabras positivas, en total tenemos 44 palabras

diccionarioPositivo <- c("positiva","solicito","quisiera","necesito","mejor","complacer","divina","divino","hermosa",

"hermoso","ponga","pongan","pongamen","complazcamen","uno","mucho","top","deseo","exito",

"éxito","alegría","entusiasmo","buenos","momentos","confianza","excelente","escuchar",

"sonar","poner","tocar","ponen","sonar","canción","cancion","complacen","favorito",

"quisiera","quiero","puede","pueden","programen","podrian","podrían","queremos")

#Creamos la columna de polaridad con diccionarios

TweetsFinal$polaridadDiccionarios <- c(rep(0, NROW(TweetsFinal)))

#Recorremos todos los tweets, por cada tweet separamos en palabras al texto y las comparamos con las palabras del diccionario.

for (i in 1:NROW(TweetsFinal)) {

texto <- TweetsFinal[i,17]

v <- strsplit(texto," ")

palabras <- data.frame(v)

numPositivas <- 0

numNegativas <- 0

#Contamos el número de palabras postivias en cada texto del tweet

for (j in 1:NROW(palabras)) {

for (k in 1:NROW(diccionarioPositivo)) {

if (palabras[j,1]==diccionarioPositivo[k]) {

numPositivas<-numPositivas+1

}

}

}

#Realizamos la métrica, que seria que si encuentra mas de dos palabras positivas en el tweet,ese tweet es postivo

if (numPositivas >= 2) {

TweetsFinal[i,29]<- 1

}

}

Una vez realizada la clasificación de los tweets con los diccionarios, vemos que 7564 tweets son negativos y 501 son positivos. Además, realizamos la tabla de confusión entre los tweets leídos y los tweets predichos, obteniendo los siguientes resultados.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resultados** | | **Predichos** | |
| **0** | **1** |
| **Reales** | **0** | 287 | 6 |
| **1** | 43 | 31 |

De donde podemos determinar que: de los 293 tweets negativos que nosotros determinamos, 287 son negativos y 6 pasaron a ser positivos; y de los 74 tweets positivos leídos, 31 son positivos y 43 son negativos, lo que da un porcentaje de efectividad del 86,65%

## FACTORES DE PONDERACIÓN

Para calcular los factores de ponderación realizamos los siguientes pasos:

1. Convertimos la columna localidad a mayúsculas
2. Verificamos a donde pertenece cada tweet mediante el uso de expresiones regulares, para ello usamos el comando grepl, y le asignamos a cada tweet un código de localidad.

Los códigos de las localidades son los siguientes:

|  |  |
| --- | --- |
| Código | Localidad |
| 1 | Azuay |
| 2 | Bolívar |
| 3 | Cañar |
| 4 | Carchi |
| 5 | Cotopaxi |
| 6 | Chimborazo |
| 7 | El Oro |
| 8 | Esmeraldas |
| 9 | Guayas |
| 10 | Imbabura |
| 11 | Loja |
| 12 | Los Ríos |
| 13 | Manabí |
| 14 | Morona Santiago |
| 15 | Napo |
| 16 | Pastaza |
| 17 | Pichincha |
| 18 | Tungurahua |
| 19 | Zamora Chinchipe |
| 20 | Galápagos |
| 21 | Sucumbíos |
| 22 | Orellana |
| 23 | Santo Domingo |
| 24 | Santa Elena |
| 25 | Otros |

1. Calculamos el número total de tuits por localidad
2. Ingresamos los datos del censo realizado en el 2010 para conocer el número de personas mayores a 11 años en el Ecuador a la fecha y su respectivo porcentaje.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| codprov | provincias | numero | menores4 | menores9 | menores14 | mayores11 | porcentaje |
| 1 | Azuay | 712127 | 68737 | 72122 | 75507 | 643390 | 4,95200 |
| 2 | Bolívar | 183641 | 18722 | 21226 | 21560 | 164919 | 1,26934 |
| 3 | Cañar | 225184 | 22921 | 23984 | 26851 | 202263 | 1,55676 |
| 4 | Carchi | 164524 | 15362 | 16888 | 17584 | 149162 | 1,14806 |
| 5 | Cotopaxi | 409205 | 42491 | 47367 | 46489 | 366714 | 2,82250 |
| 6 | Chimborazo | 458581 | 45264 | 49074 | 50710 | 413317 | 3,18119 |
| 7 | El Oro | 600659 | 55780 | 61450 | 63128 | 544879 | 4,19379 |
| 8 | Esmeraldas | 534092 | 64498 | 67581 | 64963 | 469594 | 3,61434 |
| 9 | Guayas | 3645483 | 359678 | 362896 | 373511 | 3285805 | 25,28995 |
| 10 | Imbabura | 398244 | 38996 | 43646 | 44326 | 359248 | 2,76503 |
| 11 | Loja | 448966 | 44203 | 47751 | 49974 | 404763 | 3,11535 |
| 12 | Los Ríos | 778115 | 83631 | 86598 | 88134 | 694484 | 5,34525 |
| 13 | Manabí | 1369780 | 139481 | 150953 | 152584 | 1230299 | 9,46928 |
| 14 | Morona Santiago | 147940 | 22011 | 21198 | 19412 | 125929 | 0,96924 |
| 15 | Napo | 103697 | 13624 | 14028 | 13023 | 90073 | 0,69327 |
| 16 | Pastaza | 83933 | 10730 | 11013 | 10208 | 73203 | 0,56342 |
| 17 | Pichincha | 2576287 | 236893 | 244844 | 241334 | 2339394 | 18,00568 |
| 18 | Tungurahua | 504583 | 45198 | 48391 | 49194 | 459385 | 3,53576 |
| 19 | Zamora Chinchipe | 91376 | 11129 | 11976 | 11976 | 80247 | 0,61764 |
| 20 | Galápagos | 25124 | 2135 | 2472 | 2358 | 22989 | 0,17694 |
| 21 | Sucumbíos | 176472 | 21294 | 22292 | 20926 | 155178 | 1,19436 |
| 22 | Orellana | 136396 | 18635 | 18189 | 16546 | 117761 | 0,90637 |
| 23 | Santo Domingo | 368013 | 40407 | 42413 | 41439 | 327606 | 2,52149 |
| 24 | Santa Elena | 308693 | 36762 | 34252 | 33446 | 271931 | 2,09298 |
| 25 | Otros | 0 | 0 | 0 | 0 | 0 | 0,00000 |

1. Calculamos el porcentaje de tweets por localidad.

porcentajeT[1, 4] <- (AZUAY \* 100) / NROW(TweetsFinal)

porcentajeT[2, 4] <- (BOLIVAR \* 100) / NROW(TweetsFinal)

porcentajeT[3, 4] <- (CAÑAR \* 100) / NROW(TweetsFinal)

porcentajeT[4, 4] <- (CARCHI \* 100) / NROW(TweetsFinal)

porcentajeT[5, 4] <- (CHIMBORAZO \* 100) / NROW(TweetsFinal)

porcentajeT[6, 4] <- (COTOPAXI \* 100) / NROW(TweetsFinal)

porcentajeT[7, 4] <- (EL\_ORO \* 100) / NROW(TweetsFinal)

porcentajeT[8, 4] <- (ESMERALDAS \* 100) / NROW(TweetsFinal)

porcentajeT[9, 4] <- (GUAYAS \* 100) / NROW(TweetsFinal)

porcentajeT[10, 4] <- (IMBABURA \* 100) / NROW(TweetsFinal)

porcentajeT[11, 4] <- (LOJA \* 100) / NROW(TweetsFinal)

porcentajeT[12, 4] <- (LOS\_RIOS \* 100) / NROW(TweetsFinal)

porcentajeT[13, 4] <- (MANABI \* 100) / NROW(TweetsFinal)

porcentajeT[14, 4] <- (MORONA\_SANTIAGO \* 100) / NROW(TweetsFinal)

porcentajeT[15, 4] <- (NAPO \* 100) / NROW(TweetsFinal)

porcentajeT[16, 4] <- (PASTAZA \* 100) / NROW(TweetsFinal)

porcentajeT[17, 4] <- (PICHINCHA \* 100) / NROW(TweetsFinal)

porcentajeT[18, 4] <- (TUNGURAHUA \* 100) / NROW(TweetsFinal)

porcentajeT[19, 4] <- (ZAMORA\_CHINCHIPE \* 100) / NROW(TweetsFinal)

porcentajeT[20, 4] <- (GALAPAGOS \* 100) / NROW(TweetsFinal)

porcentajeT[21, 4] <- (SUCUMBIOS \* 100) / NROW(TweetsFinal)

porcentajeT[22, 4] <- (ORELLANA \* 100) / NROW(TweetsFinal)

porcentajeT[23, 4] <- (SANTO\_DOMINGO\_TSACHILAS \* 100) / NROW(TweetsFinal)

porcentajeT[24, 4] <- (SANTA\_ELENA \* 100) / NROW(TweetsFinal)

porcentajeT[25, 4] <- (OTROS \* 100) / NROW(TweetsFinal)

1. Calculamos el porcentaje de tweets por localidad excluyendo los otros y calculamos el factor de ponderación correspondiente de cada localidad aplicando la fórmula detalla anteriormente.

for (i in 1:NROW(porcentajeT)) {

porcentajeT[i, 5] <-

(porcentajeT[i, 4]) / (1 - (porcentajeT[25, 4] / 100))

if (porcentajeT[i, 2] == "OTROS") {

porcentajeT[i, 6] <- 1

} else {

fp <- porcentajeT[i, 3] / porcentajeT[i, 5]

if (is.infinite(fp) == FALSE) {

porcentajeT[i, 6] <- fp

}

}

}

1. Finalmente incluimos el factor de ponderación al dataframe de tweets.

(Código completo en Anexos)

## ANALISIS DE LA RELEVANCIA DE LOS ACTORES

Para determinar cuáles son los actores a analizar, es decir, cuales son los artistas más nombrados en los tweets, hemos realizado una nube de palabras y así determinar los actores.

### Análisis con los textos clasificados con el aprendizaje supervisado

Este procedimiento lo hemos realizado, primeramente, filtrando solamente los tweets que hayan sido clasificados como positivos usando las máquinas de soporte vectorial.

TweetsPositivos <-subset(TweetsFinal, TweetsFinal$polaridadSVM == 1)

corpus = Corpus(VectorSource(TweetsPositivos$depuraSinRadios))

corpus = tm\_map(

corpus,

removeWords,

c(stopwords("spanish"))

)

frequencies = DocumentTermMatrix(corpus)

length(corpus)

frequencies

tweetsDePeticiones = as.data.frame(as.matrix(frequencies))

colnames(tweetsDePeticiones) = make.names(colnames(tweetsDePeticiones))

library(wordcloud)

positivas = as.data.frame(colSums(tweetsDePeticiones))

positivas$word = row.names(positivas)

colnames(positivas) = c("frecuencia", "palabra")

positivas = positivas[order(positivas$frecuencia, decreasing = T),]

x11()

wordcloud(

positivas$palabra,

positivas$frecuencia,

colors = brewer.pal(8, "Dark2"),

max.words = 85,

min.freq = 17

)

Obteniendo una nube de palabras como la siguiente:



Encontramos los artistas más nombrados en los tweets y le asignamos un código a cada uno de ellos:

|  |  |  |
| --- | --- | --- |
| Código de artista | Artista | Número de veces nombrado |
| 3 | Maluma | 79 |
| 1 | Joel y Brian | 73 |
| 5 | RKM & Ken-Y | 63 |
| 6 | JBalvin | 40 |
| 4 | Jesus Miranda - Chyno | 33 |
| 8 | Jostin Ramirez | 32 |
| 2 | Pablo Jara | 30 |
| 10 | Wilson Franco | 22 |
| 7 | Johann Vera | 18 |
| 11 | Marqués | 17 |
| 12 | Melibea | 15 |
| 9 | CNCO | 14 |
| 16 | Reik | 10 |
| 15 | Sebastián Yatra | 9 |
| 13 | Daniel Paez | 9 |
| 14 | Carlos Rivera | 6 |

Luego filtramos solo los tweets en donde se han nombrado a los artistas que describimos anteriormente, y procedemos a encontrar la relevancia de cada tweet usando la fórmula:

tweetsArtistas$RelevanciaArtista <-

((tweetsArtistas$favoriteCount + tweetsArtistas$retweetCount) / NROW(tweetsArtistas))

Luego, a esta relevancia lo multiplicamos por el factor de ponderación de la localidad del tweet.

tweetsArtistas$RelevanciaArtistaFP <-

tweetsArtistas$RelevanciaArtista \* tweetsArtistas$factorPonderacion

Luego debemos encontrar a qué percentil pertenece cada tweet, para ello primero ordenamos los tweets por su relevancia ponderada.

tweetsArtistas <- tweetsArtistas[order(tweetsArtistas$RelevanciaArtistaFP), ]

Luego calculamos el percentil al que pertenece cada tweet usando la siguiente fórmula:

Donde, i es la posición del tweet y N es el número de tweets.

for (i in 1:NROW(tweetsArtistas)) {

tweetsArtistas[i,35] <- ((i-(0.5))/NROW(tweetsArtistas))\*100

}

Una vez que encontramos el percentil al que pertenece cada tweet, procedemos a calcular los votos para cada tweet, siguiendo el siguiente esquema:

PROCESO DE VALORACION DE LA RELEVANCIA

for (i in 1:NROW(tweetsArtistas)) {

if (tweetsArtistas[i,35] <= 50) {

tweetsArtistas[i,36] <- 1

} else if (tweetsArtistas[i,35] > 50 & tweetsArtistas[i,35] <= 75) {

tweetsArtistas[i,36] <- 2

} else if (tweetsArtistas[i,35] > 75 & tweetsArtistas[i,35] <= 90) {

tweetsArtistas[i,36] <- 3

} else if (tweetsArtistas[i,35] > 90 & tweetsArtistas[i,35] <= 95) {

tweetsArtistas[i,36] <- 4

} else {

tweetsArtistas[i,36] <- 5

}

}

Luego procedemos a contar el número de votos para cada artista, obteniendo el siguiente resultado:

|  |  |
| --- | --- |
| Artista | Número de Votos |
| Joel Y Brian | 138 |
| Pablo Jara | 49 |
| Maluma | 97 |
| Jesus Miranda - Chyno | 65 |
| RKM & Ken-Y | 75 |
| JBalvin | 56 |
| Johann Vera | 43 |
| Jostin Ramirez | 84 |
| CNCO | 34 |
| Wilson Franco | 32 |
| Marques | 43 |
| Melibea | 15 |
| Daniel Paez | 25 |
| Carlos Rivera | 15 |
| Sebastián Yatra | 16 |
| Reik | 24 |

Además, realizamos un diagrama de pastel en donde consta el porcentaje de votos de cada artista.

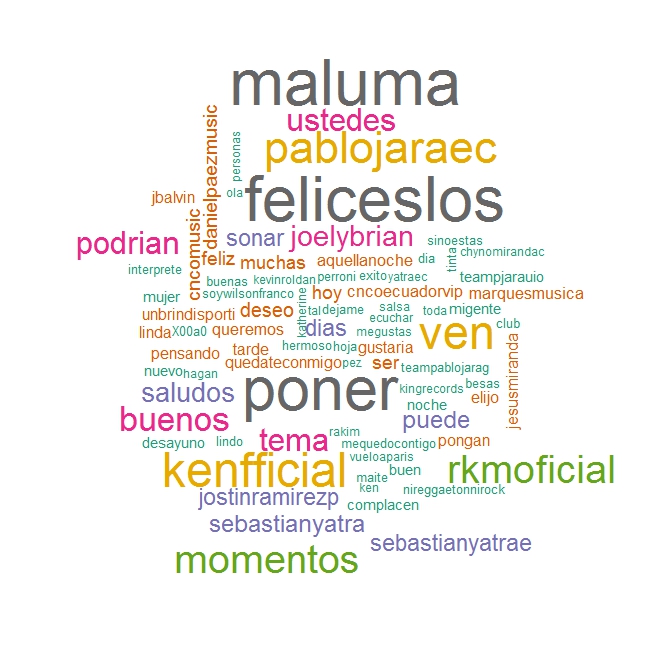
Y para concluir un diagrama de barras, con el número de votos de cada artista.

### Análisis con los textos clasificados con el aprendizaje no supervisado

Para realizar el análisis de los actores que fueron encontrados con el aprendizaje no supervisado, seguimos los mismos pasos que en el aprendizaje supervisado, pero los resultados son un poco diferentes:

La nube de palabras es la siguiente y su respectiva tabla de frecuencias:

|  |  |  |
| --- | --- | --- |
| Código de artista | Artista | Número de veces nombrado |
| 3 | Maluma | 75 |
| 1 | Joel y Brian | 29 |
| 5 | RKM & Ken-Y | 53 |
| 6 | JBalvin | 10 |
| 4 | Jesus Miranda - Chyno | 11 |
| 8 | Jostin Ramirez | 20 |
| 2 | Pablo Jara | 50 |
| 10 | Wilson Franco | 8 |
| 7 | Johann Vera | 5 |
| 11 | Marqués | 12 |
| 12 | Melibea | 0 |
| 9 | CNCO | 14 |
| 16 | Reik | 1 |
| 15 | Sebastián Yatra | 22 |
| 13 | Daniel Paez | 15 |
| 14 | Carlos Rivera | 3 |



Número de votos por cada artista es:

|  |  |
| --- | --- |
| Artista | Número de Votos |
| Joel Y Brian | 56 |
| Pablo Jara | 105 |
| Maluma | 98 |
| Jesus Miranda - Chyno | 24 |
| RKM & Ken-Y | 62 |
| JBalvin | 28 |
| Johann Vera | 12 |
| Jostin Ramirez | 51 |
| CNCO | 36 |
| Wilson Franco | 5 |
| Marques | 31 |
| Melibea | 0 |
| Daniel Paez | 44 |
| Carlos Rivera | 11 |
| Sebastián Yatra | 46 |
| Reik | 2 |

Su respectivo diagrama de pastel con el porcentaje de votos:

Y su respectivo diagrama de barras con el número de votos:

De los dos análisis, podemos determinar el top 5 de los artistas más solicitados en las radios ecuatorianas:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Análisis Con Aprendizaje Supervisado** | | | **Análisis Con Aprendizaje No Supervisado** | | |
| **Posición** | **Artista** | **Votos** | **Posición** | **Artista** | **Votos** |
| 1 | Joel y Brian | 138 | 1 | Pablo Jara | 105 |
| 2 | Maluma | 97 | 2 | Maluma | 98 |
| 3 | Jostin Ramirez | 84 | 3 | RKM & Ken-y | 62 |
| 4 | RKM & Ken-y | 75 | 4 | Joel y Brian | 56 |
| 5 | Jesús Miranda - Chyno | 65 | 5 | Jostin Ramirez | 51 |

De donde podemos decir que 4 de los 5 artistas están en el podio sin importar el tipo de clasificación que se hizo, ya sea con aprendizaje supervisado o aprendizaje no supervisado, pero para cumplir con el objetivo hemos optado por escoger al top 5 de los artistas realizado con Aprendizaje Supervisado, mejor dicho, con el Análisis de máquinas de soporte, ya que el error que este arroja es muy bajo a comparación del análisis no supervisado.

# CONCLUSIONES Y RECOMENDACIONES

1. El artista más solicitado por los ecuatorianos es Joel y Brian, seguido de Maluma y en tercer puesto se encuentra Jostin Ramirez.
2. Al momento de clasificar texto, el aprendizaje supervisado es más efectivo que el análisis no supervisado.
3. Cuando se clasifica texto, entre los kernels que usa las Máquinas de Soporte Vectorial, el más efectivo al momento de la categorización es el kernel lineal.

# REFERENCIAS

* RESENDIZ, J. A. Las Máquinas de Soporte Vectorial para identificación en Linea. Maestria, Control Automático. Instituto Politecnico Nacional, 2006.
* BOURDIEU, Pierre. Sobre la televisión. Editorial Anagrama. Barcelona, 1994.
* CALLEJO, Javier; Investigar las audiencias; editorial : Paidós, Barcelona , 2001.
* Dorian, P. (2003), " Business Modeling and Data Mining ", Morgan Kaufmann
* Support-vector-machines.org. (2017). *SVM - Support Vector Machines*. [online] Disponible en: <http://www.support-vector-machines.org/>
* Collingwood, L., Jurka, T., Boydstun, A. E., Grossman, E., & van Atteveldt, W. H. (2013). RTextTools: A supervised learning package for text classification.
* Curiosoando. (2017). ¿Qué son los percentiles?. [online] Available at: <https://curiosoando.com/que-son-los-percentiles>.
* KOWALCZYK, A. (2017). Linear Kernel: Why is it recommended for text classification ?. [online] SVM Tutorial. Available at: https://www.svm-tutorial.com/2014/10/svm-linear-kernel-good-text-classification/.

# ANEXOS

## SCRIPTS UTILIZADOS

Lista de Scripts usados para el proyecto (Adjuntos en el CD):

SCRIPT PARA DESCARGA DE TWEETS

SCRIPT PARA DEPURACION DE TWEETS

SCRIPT PARA HALLAR LA LOCALIDAD DEL TWEET

SCRIPT PARA EL CALCULO DEL FACTOR DE RELEVANCIA

SCRIPT PARA LA CLASIFICACIÓN DE TEXTOS USANDO SVM

SCRIPT PARA LA CLASIFICACIÓN DE TEXTOS USANDO DICCIONARIOS

SCRIPT PARA EL ANÁLISIS DE LA RELEVANCIA CON LOS TWEETS POSITIVOS CLASIFICADOS CON SVM.

SCRIPT PARA EL ANÁLISIS DE LA RELEVANCIA CON LOS TWEETS POSITIVOS CLASIFICADOS CON DICCIONARIOS.