ModelosNoSupervisado-ActividadFacebook

Javier Eduardo Jaimes Velasquez[[1]](#footnote-20)

Brandon Valencia Murillo[[2]](#footnote-22)

2024-07-11

Table of Contents

# 1 Modelos No Supervisados - Actividad en Facebook

Hay situacion en donde el interes es encontrar patrones a nivel grupal mas alla de lo individual, y es el caso que se estudiara a lo largo del libro.

Es importante recordar que:

* las tareas de agrupamiento, difieren un poco de las tareas de clasificacion.
* el algoritmo de agrupamiento objeto de estudio es el llamado *kmeans*.

## 1.1 Entiendo el agrupamiento no supervisado.

Este de algoritmo es de caracter no supervisado, en donde automaticamente se definen grupos (*clusters*) de elementos similares. Es claro que estos grupos no se han definido previamente por tal razon estos algoritmos son escenciales a la hora de descubrir patrones, mas alla de hacer predicciones.

El principio principal de agrupamiento se basa en que los elementos de un grupo deben ser tan similares como se pueda de los demas pero muy diferente del resto, es decir que en terminos de definicion de similiradidad la distancia puede ser un elemento clave.

### 1.1.1 Algunos ejemplos de criterios de similaridad

* Agrupamiento por caracteristicas socio-demograficas.
* Patrones de Comportamiento de uso de un sistema, comportamiento de compra en una plataforma de ventas.
* Tan simple como por columnas de un dataset con valores similares.

## 1.2 El modelo de agrupamiento

A diferencia de los modelos de prediccion, los modelos de agrupamiento buscan crear nuevos datos a partir de la logica definidad de agrupamiento, pero en resumen el proceso no difiere de a mucho, excepto de que al super no supervisado implica que los datos se agrupan siguiendo por ejemplo heuristicas definidas para ello.

library(magrittr)  
library(factoextra)

## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(corrplot)

## corrplot 0.92 loaded

library(flextable)  
library(readr)  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ stringr 1.5.1  
## ✔ forcats 1.0.0 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ purrr::compose() masks flextable::compose()  
## ✖ tidyr::extract() masks magrittr::extract()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ purrr::set\_names() masks magrittr::set\_names()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

### 1.2.1 Descripcion del conjunto de datos

fb\_activity <- read.csv2("./datasets/Live\_20210128.csv", sep = ",", header = T)

Ahora se realiza un poco de limpieza sobre el conjunto de los datos:

fb\_activity <- fb\_activity %>% select(-c("status\_type", "status\_id", "status\_published", "Column1", "Column2", "Column3", "Column4"))

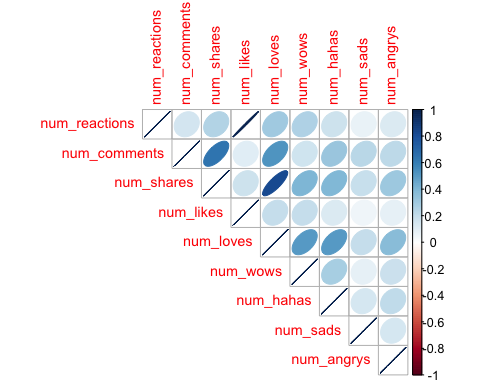
fb\_activity %>% summarizor %>% as\_flextable(spread\_first\_col = F)

|  |  |  | Statistic (N=7,050) |
| --- | --- | --- | --- |
| num\_reactions | Mean (SD) |  | 230.1 (462.6) |
| Median (IQR) |  | 59.5 (202.0) |
| Range |  | 0.0 - 4,710.0 |
| num\_comments | Mean (SD) |  | 224.4 (889.6) |
| Median (IQR) |  | 4.0 (23.0) |
| Range |  | 0.0 - 20,990.0 |
| num\_shares | Mean (SD) |  | 40.0 (131.6) |
| Median (IQR) |  | 0.0 (4.0) |
| Range |  | 0.0 - 3,424.0 |
| num\_likes | Mean (SD) |  | 215.0 (449.5) |
| Median (IQR) |  | 58.0 (167.8) |
| Range |  | 0.0 - 4,710.0 |
| num\_loves | Mean (SD) |  | 12.7 (40.0) |
| Median (IQR) |  | 0.0 (3.0) |
| Range |  | 0.0 - 657.0 |
| num\_wows | Mean (SD) |  | 1.3 (8.7) |
| Median (IQR) |  | 0.0 (0.0) |
| Range |  | 0.0 - 278.0 |
| num\_hahas | Mean (SD) |  | 0.7 (4.0) |
| Median (IQR) |  | 0.0 (0.0) |
| Range |  | 0.0 - 157.0 |
| num\_sads | Mean (SD) |  | 0.2 (1.6) |
| Median (IQR) |  | 0.0 (0.0) |
| Range |  | 0.0 - 51.0 |
| num\_angrys | Mean (SD) |  | 0.1 (0.7) |
| Median (IQR) |  | 0.0 (0.0) |
| Range |  | 0.0 - 31.0 |

#### 1.2.1.1 Variables duplicadas

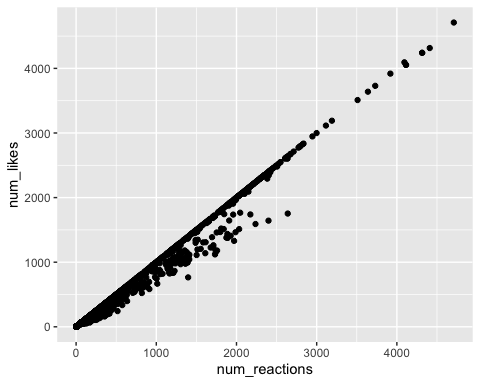
Es importante determinar si algunas variables estan corelacionadas entre si, esto para dejarlas por fuera de nuestro modelo de agrupamiento:

corrplot(cor(fb\_activity), type = "upper", method = "ellipse", tl.cex = 0.9)



Entiendo que, la variable *num\_reactions* y *num\_likes*, tienen un coeficiente de corelacion de 1, haria sentido que al menos una de ella no fuera parte de nuestro modelo. Antes de eliminarlas del todo, se procede a verificar su nivel de dispersion por medio de un grafico de variacion:

fb\_activity %>% ggplot() + aes(x = num\_reactions, y =num\_likes) + geom\_point()



Al comprobarse la corelacion de forma grafica procedemos a limpiar la variable de nuestro conjunto datos:

fb\_activity <- fb\_activity[, -1]

En el siguiente paso, estandarizemos los valores siguiendo el modelo *z-score*. En este proceso se buscan que todas las caracteristicas sean llevadas en una escala siguiendo el llamandom del metodo *scale* de R, en dicho caso la media sera de y la desviacion estandar de :

fb\_activity\_zs <- as.data.frame(scale(fb\_activity))

## 1.3 K-means, como funciona.

Por lo general lo algoritmos de agrupamiento se pueden dividir en dos grupos, los basados en una metrica de distancia o los basado en un funcion de agrupamiento. Y a su vez podemos tambien clasificarlos dependiendo del metodo, los basadois en metodos jerarquicos, los basados en particiones y lo basados densidad. K-means esta basado en particiones, entre sus ventajas estan>

* Principios basicos que son de facil entendimiento.
* Altamente configurabble, adaptable para acomodarse a diversas situaciones.
* Su desempeno es bueno bajo la mayoria de situaciones de la vida real.

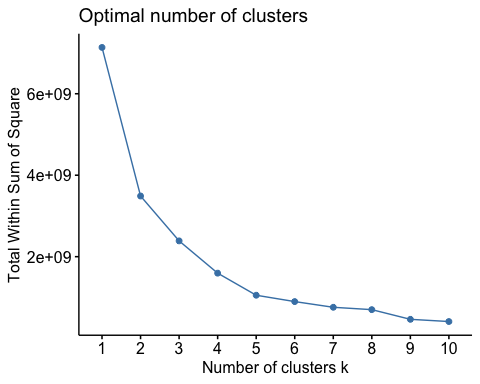
Entre sus desventajas: - No tan sofisticado como algunos de los otros metodos. - No tiene garantia de seleccion en situacion optima de agrupamiento. - Requiere esfuerzos en la eleccion del numero , de clusters. - Y no es ideal para ciertos cosas como aquellos donde la alta variacion en la densidad del modelo es clave.

En simples terminos: k-means assigna un numero determinado de observaciones a un numero de clusters, donde fue determinado con antelacion. El algoritmo usa un proceso de heuristicas para encontrar una solucion local optima, basicamente inicia con su mejor supocision y compara diferentes agrupamientos hasta llegar al supuesto “mejor”.

## 1.4 Determinar k, numero de clusters.

Empezaremos determinando el numero a partir del metodo del codo (*elbow*): aqui la idea es en como la homogeneidad y la heterogeneidad de los clusters cambia para diferentes valores de :

fviz\_nbclust(x = fb\_activity, FUNcluster = kmeans, method = "wss")



Dado lo anteorior, el numero debe ser igual a ., Con definido procedemos a crear el modelo utilizando el metodo *k-means* parte de la libreria *stats*:

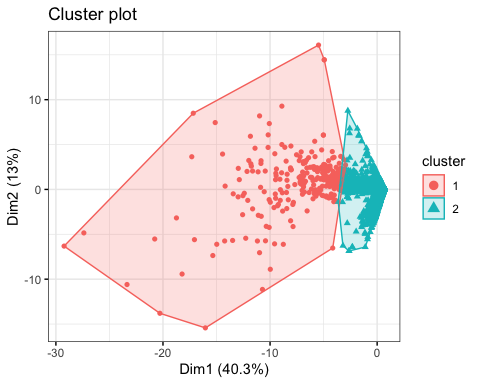
km\_2 <- kmeans(x = fb\_activity\_zs, centers = 2)

Y asignamos al dataframe *fb\_activity\_zs* el cluster asignado a cada observacion:

fb\_activity\_zs\_agg <- fb\_activity\_zs %>% mutate(cluster\_medias = km\_2$cluster)

Por ultimo visualizacion la agrupacion creada por el metodo k-means:

fviz\_cluster(km\_2, data = fb\_activity, geom = "point", ellipse.type = "convex", ggtheme = theme\_bw())



1. Politécnico Grancolombiano, [jajaimes4@poligran.edu.co](mailto:jajaimes4@poligran.edu.co) [↑](#footnote-ref-20)
2. Politécnico Grancolombiano, [brvalencia6@poligran.edu.co](mailto:brvalencia6@poligran.edu.co) [↑](#footnote-ref-22)