Agrupando Segmentos de Adolescentes

La interacción por medio de redes sociales como facebook o tuenti es una de relación habitual entre personas a escala mundial. Estas interacciones abren la puerta a anuncios y publicidad específica. Una forma de generar anuncios específicos es agrupar a los usuarios por gustos similares. Por ejemplo anuncios de deportes concretos se pueden mostrar a grupos de usuarios con el mismo interés en esos deportes.

El clustering puede ayudar a identificar grupos de usuarios similares. Sin embargo siempre es necesario el criterio del desarrollador para analizar si estos grupos tienen sentido o no.

Conjunto de datos

Vamos a utilizar un conjunto de datos que consiste en una muestra de 30.000 estudiantes Norte-Américanos de High School con perfiles en una importante red social. El conjunto de datos se proporciona junto al enunciado de la práctica.

Los datos corresponden a cuatro institutos durante los años 2006 al 2009 y representan cuatro tipos de clases: **senior**, **junior**, **sophomore** y **freshman**.

Se ha utilizado una herramienta de minería de textos para obtener 500 palabras, de las cuales 36 palabras han sido utilizadas para identificar cinco categorías de intereses: **(1) actividades extracurriculares, (2) fashion, (3) religión, (4) romance y (5) comportamiento anitsocial**. El dataset proporciona cuantas veces aparecen cada una de las 36 palabras en los perfiles de las personas.

Preparación y exploración de los datos

En primer lugar comenzamos leyendo el conjunto de datos del disco:

```
social_data <- read.csv("social_network.csv")
```

Ahora podemos echar un primer vistazo a los datos:

```
str(social_data)
```

Como esperábamos los datos incluyen 30.000 instancias de adolescentes con cuatro variables que indican situaciones personales y 36 palabras que indican intereses. Sin embargo algunas variables contienen datos NA, podemos verlos con el siguiente comando:

```
summary(social_data)
```

También son sospechosos los rangos de edad (3 y 106 años). Para asegurarnos que estos valores extremos no generan un error en el análisis es necesario filtrarlos. Un valor razonable de edades es entre 13 y 20, así que ponemos NA en aquellos valores fuera de ese rango:

```
social_data$age <- ifelse(social_data$age >= 13 & social_data$age < 20,
social_data$age, NA )
```

Ahora podemos volver a comprobar con summary() los rangos de edades. Sin embargo tenemos que solucionar el problema de los valores que faltan.
La solución más sencilla es eliminar todos los registros con valores que faltan. Pero esto implica perder gran parte de información valiosa. Una buena alternativa para datos que faltan que son catégoricos como *gender* es asignarles una nueva clase. En nuestro caso vamos a crear dos nuevas variables: *female* y *no_gender*.

```
social_data$female <- ifelse(social_data$gender == "F" &
!is.na(social_data$gender), 1, 0)

social_data$no_gender <- ifelse(is.na(social_data$gender), 1, 0)
```

Ahora vamos a solucionar el problema de los 5.253 valores faltantes en age. Al ser una variable numérica vamos a rellenar los valores con los números apropiados. Un buen valor es la media de las edades:

```
mean(social_data$age, na.rm = TRUE)
[1] 17.25243
```

Esto nos indica que la edad media de todos los estudiantes es de 17 años. Para ser precisos sería conveniente obtener la edad media de los estudiantes por año de graduación. Para ello utilizamos:

```
aggregate(data = social_data, age ~ gradyear, mean, na.rm = TRUE)
  gradyear      age
1     2006 18.65586
2     2007 17.70617
3     2008 16.76770
4     2009 15.81957
```

Observamos que ahora la media de edad difiere significativamente en función del año de graduación. Para poder utilizar estos números vamos a utilizar la función ave() pues la salida de la función aggregate() es un data.frame y es un poco más tedioso de utilizar.

```
> ave_age <- ave(social_data$age, social_data$gradyear, FUN = function(x) mean(x,
na.rm = TRUE))
> social_data$age <- ifelse(is.na(social_data$age), ave_age, social_data$age)
> summary(social_data$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  13.03   16.28   17.24   17.24   18.21   20.00
```

Entrenando un modelo de clustering sobre los datos

Ahora que ya tenemos los datos en un formato que puede ser utilizado para un modelo de clustering se solicita al alumno/a que genere un modelo de clustering utilizando la función kmeans() del paquete stats y/o la función hclust() de cluster jerárquico. Se aconseja lo siguiente:

1. Empezar el análisis utilizando solo las 36 features que representan los intereses.
2. Normalizar los valores de estas 36 variables de forma que todas tengan la misma escala (media 0 y desviación estándar 1) utilizando para ello la función scale().
3. Utilizar valores de cluster entre (K) entre 3 y 10.

Se pide analizar los resultados obtenidos con el clustering modificando para ellos los parámetros que el alumno/a estime oportuno.

En particular un análisis interesante es ver si el algoritmo de clustering es capaz de agrupar de forma satisfactoria clusters dependiento de las edades, gender y amigos de los adolescentes. Para ello es necesario que estás variables no estén dentro de variables del clustering. Posteriormente una vez realizado el clustering habría que obtener los valores de estás variables objetivo en cada uno de los clusters.

También es deseable que se pueda obtener una descripción de los clusters obtenidos en función de los intereses mayoritarios de los miembros del cluster. Es decir, ¿es posible clasificar a los usuarios en chicas, buenos estudiantes, deportistas, antisociales?, etc...

La práctica se evaluará de la siguiente forma:

1) **(2 puntos)** Ejecución de los pasos de preparación de datos hasta poder empezar a generar el modelo de clustering.
2) **(4 puntos)** Creación de varios modelos de clustering.
3) **(4 puntos)** Explicación e interpretación de los resultados experimentales obtenidos. Se valoraran los comentarios debidamente justificados y la claridad de los mismos así como lo riguroso y exhaustiva que sea la validación del modelo.

Normativa de realización, entrega y evaluación de la práctica:

- La práctica se realizará y entregará por grupos compuestos de 2 personas como máximo.
- La práctica se realizará en R/RStudio, utilizando el software instalado en el aula o bien una versión idéntica.
- La entrega deberá hacerse mediante Blackboard antes del 25 de diciembre de 2015 a las 23:59 horas (hora peninsular en España).
- La entrega se compondrá de un único fichero ZIP, que contendrá lo siguiente:
    a. Un fichero de texto llamado “prac_1_Nombres_Grupo.R” que contendrá el código R necesario para realizar la práctica.
    b. Un documento explicativo de los resultados obtenidos.
- **En ningún caso se debe entregar el modelo entrenado o variables guardadas a disco.**
- Al hacer entrega del fichero comprimido, éste se habrá renombrado con el nombre y el primer apellido de los alumnos integrantes del grupo, escritos en letras mayúsculas y separados mediante un guión bajo ‘_’. Ejemplo: JOSE_FONT_FEDERICO_CASTANEDO.zip
- La corrección de la práctica se hará ejecutando los ficheros proporcionados de manera secuencial y evaluando el resultado obtenido.
- Cualquier sospecha de copia entre dos o más prácticas derivará en la calificación de 0 para todos los alumnos involucrados.