

Anexo XIII. Recomendador de canciones

En este documento vamos a explicar cómo hemos programado el recomendador y cómo utilizarlo. Este recomendador va a consistir en un programa que recibirá varios parámetros, relacionados con la canción que nos gusta, el clustering y el dataset que utilizamos para realizar el estudio.

Primero de todo vamos a mostrar el código correspondiente del recomendador:

```
library(cluster)

recommend = function (song, dataset, clustering){
  centroids = clustering$centers
  includes = setdiff(colnames(song), "id")
  datamean = colMeans(dataset[,includes])
  datasd = apply(dataset[,includes], 2, FUN = sd)
  song[includes] = scale(song[includes], center = datamean, scale =
datasd)
  candidates = as.data.frame(rbind(song[includes], centroids))
  rownames(candidates) = c("Song", paste("Cluster",
1:nrow(clustering$centers)))

  closeness = as.data.frame(as.matrix(get_dist(candidates)))

  #print(closeness)

  closest = min(closeness[1,-1])
  chosen_cluster = which(closeness[1,-1] == closest)

  dataset_sc = dataset
  dataset_sc[, -1] = scale(dataset[, -1])

  songs_cluster = dataset_sc[clustering$cluster == chosen_cluster,]
  all_together = as.data.frame(rbind(song, songs_cluster))
  finally = as.data.frame(as.matrix(get_dist(all_together[, -1])),
row.names = all_together$id)
  colnames(finally) = all_together$id

  index = which(finally[[song$id]] == min(finally[[song$id]][-1]))
  chosen_one_id = colnames(finally)[index]
  chosen_one = dataset[dataset$id == chosen_one_id,]

  return (chosen_one)
}
```

Ahora que ya hemos visto el código, observamos que los parámetros que recibe la función son los siguientes:

- **Song**: registro dónde se encuentran los atributos de la canción que nos gusta (de la que queremos la recomendación).
- **Dataset**: la variable en la que se encuentran todas las canciones con las que hemos realizado el clustering.
- **Clustering**: hace referencia al resultado del clustering, que es un tipo 'imanes' en este caso.

Una vez hemos llamado a la función pasándole estos parámetros, la función realiza los siguientes pasos:

- Guardamos en una variable llamada 'centroids' los centroides del clustering que le hemos pasado.
- En la variable 'includes' almacenaremos los nombres de las columnas numéricas.
- Calcular en la siguiente línea la media del dataset para cada uno de estos atributos y después la desviación típica de la misma forma.
- Una vez ya tenemos las medias y sd, escalamos y centramos los datos de la nueva canción (SONG) con dichos valores.

Esos pasos nos han servido para preparar los datos recibidos de la nueva canción y separar los centroides en una única variable. Tras haber realizado esto, pasamos a analizar distancias:

- Juntamos todos los datos, los de song y los centroides, en una sola variable llamada 'candidates' en un objeto 'dataframe', además les añadimos nombres a las filas: 'song' para la fila de la nueva canción y 'cluster X' (siendo X un número) para señalar de qué cluster es cada centroide.
- Lo siguiente es almacenar en un objeto 'dataframe' las distancias de los datos guardados en 'candidates'.
- Al ya tener las distancias, comprobamos qué centroide es el más cercano a la song y lo guardamos en una nueva variable.

Tras tener elegido el centroide más cercano, que es el que está guardado en la variable 'chosen_cluster', realizaremos unos últimos cálculos de distancias con el objetivo de encontrar la canción más similar a la nuestra como explicamos a continuación:

- Escalamos el dataset excepto la columna 1 que contiene las 'id'.
- Nos guardamos las canciones que pertenecen al cluster cuyo centroide es el elegido. Después las unimos con la canción SONG y se guarda en un objeto tipo 'dataframe'.
- Volvemos a calcular las distancias pero esta vez de las canciones guardadas en el anterior punto, almacenándolas como 'dataframe' y poniendo como nombre de las filas y columnas los ID de cada canción.

Por último, cómo ya tenemos todas las distancias necesarias para llevar a cabo la recomendación, pasamos a describir los pasos finales:

- Creamos una variable con el índice de la canción más cercana a la SONG.
- Buscamos dicho índice entre los nombres de las columnas de la matriz de distancias para saber el ID de la canción.
- Finalmente, como ya tenemos el identificador de la canción a recomendar, podemos buscarla en el dataset inicial y devolver todos los atributos correspondientes a esa canción, que sería la solución del programa.

En resumen, simplemente pasando unos pocos parámetros, que como ya hemos indicado son los datos de la nueva canción, el dataset utilizado en el clustering y el resultado de dicho clustering, conseguimos todos los atributos de la canción que se recomienda en base a la dada como input (inclusive el nombre del artista, el título de la composición...).

Nota: Para que NO haya ningún tipo de problema con el programa, los atributos de la 'SONG' y del 'DATASET' deberán de seguir este orden: 'id', 'acousticness', 'instrumentalness', 'speechiness', 'loudness'. De esa forma no habrá ningún error de compatibilidad o de recomendaciones no acertadas.