Universidad
Internacional
Menéndez Pelayo

Máster Universitario en Investigación en Inteligencia Artificial

Curso 2020-2021

Sistemas de Recomendación

Recomendación para grupos en Python

26 de abril de 2021

Laura Rodríguez Navas
DNI: 43630508Z
e-mail: rodrigueznavas@posgrado.uimp.es

Índice

1. Introducción

2. Conjunto de datos

El conjunto de datos que se ha usado en esta práctica se encuentra disponible públicamente para su descarga en el siguiente enlace: https://www.kaggle.com/abhikjha/movielens-100k/download. Este conjunto de datos llamado *ml-latest-small*, describe las calificaciones (entre 1 y 5 estrellas) y la actividad del etiquetado de MovieLens [1], un servicio de recomendación de películas. Concretamente, el conjunto de datos contiene 100836 clasificaciones y 3683 etiquetas de 9742 películas. Los datos fueron creados por 610 usuarios que fueron seleccionados al azar. Cada usuario clasificó al menos 20 películas y está representado por una identificación.

Una vez descargado el conjunto de datos veremos que está contenido en los archivos *links.csv*, *movies.csv*, *ratings.csv* y *tags.csv*. Pero para el desarrollo de esta práctica solo se utilizan los archivos *movies.csv* y *ratings.csv*. A continuación cargamos cada archivo dentro de su dataframe (ver Definición 1) con el uso de la librería pandas [2].

```
movies_df = pd.read_csv('dataset/movies.csv')
ratings_df = pd.read_csv('dataset/ratings.csv')
```

Observamos el contenido de *movies_df* para ver como ha quedado organizado:

movieId	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	Father of the Bride Part II (1995)	Comedy

Tabla 1: Contenido del dataframe *movies_df* inicial.

Cada película tiene un único identificador, un título con su año de estreno y diferentes géneros. Como los años contienen caracteres *unicode* y para que no haya problemas más adelante, los sacaremos de la columna de los títulos y los ubicaremos en su propia columna que nombraremos *year*. Para ello, primero utilizamos una expresión regular para encontrar los años guardados entre paréntesis, y con la función *extract* de la librería pandas los extraemos de la columna de los títulos para guardarlos en su propia columna. Después borramos los años de la columna de los títulos y con la función *strip* nos aseguramos de sacar los espacios finales que pudieran haber.

```
regular_expression = r'\((.*?)\)'
movies_df['year'] = movies_df.title.str.lower().str.extract(regular_expression)
movies_df['title'] = movies_df.title.str.replace(regular_expression, '', regex=True)
movies_df['title'] = movies_df['title'].apply(lambda x: x.strip())
```

Vemos el resultado:

movieId	title	genres	year
1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	1995
2	Jumanji	Adventure|Children|Fantasy	1995
3	Grumpier Old Men	Comedy|Romance	1995
4	Waiting to Exhale	Comedy|Drama|Romance	1995
5	Father of the Bride Part II	Comedy	1995

Tabla 2: Contenido del dataframe *movies_df* con la columna *year*.

Definición 1. *Un DataFrame es una estructura de datos etiquetada bidimensional que acepta diferentes tipos datos de entrada organizados en columnas. Se puede pensar en un DataFrame como una hoja de cálculo o una tabla SQL.*

Sacamos la columna de los géneros, ya que no los usaremos para este sistema de recomendaciones.

```
movies_df = movies_df.drop('genres', 1)
```

Y así queda el dataframe *movies_df* final:

movieId	title	year
1	Toy Story	1995
2	Jumanji	1995
3	Grumpier Old Men	1995
4	Waiting to Exhale	1995
5	Father of the Bride Part II	1995

Tabla 3: Contenido del dataframe *movies_df* final.

Bibliografía

[1] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), December 2015.

[2] Jeff Reback, Wes McKinney, jbrockmendel, Joris Van den Bossche, Tom Augspurger, Phillip Cloud, gfyoung, Sinhrks, Simon Hawkins, Matthew Roeschke, Adam Klein, Terji Petersen, Jeff Tratner, Chang She, William Ayd, Shahar Naveh, Marc Garcia, Jeremy Schendel, Andy Hayden, Daniel Saxton, Vytautas Jancauskas, Ali McMaster, Pietro Battiston, Skipper Seabold, patrick, Kaiqi Dong, chris b1, h vetinari, Stephan Hoyer, and Marco Gorelli. pandas-dev/pandas: Pandas 1.1.5, December 2020.