

Práctica 2: Desarrollo de un Sistema de Recomendación basado en Filtrado Colaborativo

Manuel Orantes Taboada

manuelorantes96@gmail.com

morantes96@correo.ugr.es

77150692K

15 de abril de 2021

Índice

1	Descripción	3
2	Desarrollo de la práctica	3
2.1	CrearTablaSimilitud.py	3
2.2	recomendador.py	3
2.3	Extra	3
3	Guía de usuario	4
4	Bibliografía consultada	4

1. Descripción

La práctica consiste en hacer un recomendador de películas a partir de la base de datos MovieLens. Es una base de datos que tiene 100000 valoraciones de 943 usuarios sobre 1682 películas. Se usarán estas valoraciones para crear un sistema que recomiende películas a un usuario siguiendo un sistema de vecino cercano. Para ello, se le pedirá al usuario que valore primero 20 películas aleatorias, según las cuales se pondrán recomendar películas después.

2. Desarrollo de la práctica

La práctica se ha realizado con el lenguaje python. Se han seguido las diapositivas del temario vistas en clase.

En primer lugar, hay una serie de líneas para parsear los archivos, y de este modo tener los datos en arrays y diccionarios de python.

Tenemos dos archivos .py. Veamos para que sirven ambos:

2.1. CrearTablaSimilitud.py

Este archivo será el encargado de:

- Recoger los datos del usuario, preguntándole por 20 películas que deberá puntuar de 1 a 5.
- Crear la tabla de similitud, de todos los usuarios con el usuario que puntúa las 20 películas.

Una vez que se han obtenido los resultados del usuario, se utiliza la similitud del coseno, vista en las transparencias de la clase. Por último, se exportan los datos en el fichero tablaSimilitud.txt, aunque realmente es un archivo .csv. Además, también se crea otro archivo, peliculasVisualizadas.txt (que también es un csv), en el que se guardarán los títulos de las películas vistas por el usuario (las 20 que se usaron al principio).

2.2. recomendador.py

En este archivo se leerán los datos, tanto proporcionados por MovieLens como los obtenidos en el paso anterior. Se usa la tabla de similitudes para calcular la puntuación de cada una de las películas, según las diapositivas del tema, y después se añaden, las que no haya visto el usuario. Por último, se ordenan según la puntuación.

2.3. Extra

Para finalizar, buscando información por Internet encontré un programa que usa también el vecino cercano, pero que lo hace sobre las películas en sí. Busca las películas más similares a otra dada. Me pareció buena idea incorporar esta idea como extra, del modo que lo que hago es lo siguiente: En el paso anterior, me quedo con las 7 películas mejor

valoradas. Después, completaré hasta 10 de la siguiente forma: de la película mejor valorada, cogeré las dos más similares a esta, siempre que no estén ya añadidas y no las haya visto el usuario. Cogeré también la más similar a la segunda película mejor valorada.

3. Guía de usuario

El uso es muy sencillo, solo hace falta tener los paquetes necesarios de python instalados (numpy , pandas, math, operator, scipy). Después, ejecutaremos CrearTablaSimilitud.py y responderemos a las preguntas introduciendo números del 1 al 5 (enteros y ambos incluidos). Este es el ejemplo de ejecución que yo he realizado:

```
manuel-aircury@manuel-aircury ~/Master/GIW/Recomendador $ python3 crearTablaSimilitud.py
Cats Don't Dance (1997)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 3
Two or Three Things I Know About Her (1966)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 1
Glass Shield, The (1994)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 2
Looking for Richard (1996)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 3
Jean de Florette (1986)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 1
Walk in the Clouds, A (1995)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 2
Winter Guest, The (1997)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 3
Quartier Mozart (1992)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 3
Loch Ness (1995)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 1
New Age, The (1994)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 2
Further Gesture, A (1996)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 2
George of the Jungle (1997)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 3
Pocahontas (1995)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 5
For Whom the Bell Tolls (1943)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 3
Race the Sun (1996)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 2
Tombstone (1993)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 1
Mallrats (1995)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 2
Golden Earrings (1947)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 3
Walkabout (1971)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 1
Slingshot, The (1993)
¿Con cuantos puntos (de 1 a 5) puntuas a esta película? 1
```

Después, bastará con ejecutar recomendador.py, que devolverá una lista de las 10 películas recomendadas para tí.

```
manuel-aircury@manuel-aircury ~/Master/GIW/Recomendador $ python3 recomendador.py
1: Star Wars (1977)
2: Return of the Jedi (1983)
3: Fargo (1996)
4: Raiders of the Lost Ark (1981)
5: Empire Strikes Back, The (1980)
6: Silence of the Lambs, The (1991)
7: Pulp Fiction (1994)
8: Independence Day (ID4) (1996)
9: Star Trek: First Contact (1996)
10: English Patient, The (1996)
```

4. Bibliografía consultada

Además de basarme en la propia práctica, he usado este tutorial:

<https://hendra-herviawan.github.io/Movie-Recommendation-based-on-KNN-K-Nearest-Neighbors.html>