



Gestión de Información en la Web

Desarrollo de un Sistema de Recomendación basado en Filtrado Colaborativo

Irene Béjar Maldonado

76066018G

irenebejar@correo.ugr.es

23/3/2020

Índice

Introducción	3
Descripción de la práctica	3
MoviesAndUsers	3
RecommenderSystem	4
Instrucciones para la ejecución	5
Referencias	6

1. Introducción

En esta práctica se construirá, partiendo de cero y en el lenguaje preferido, un sistema de recomendación de películas basado en filtrado colaborativo (usuario -usuario).

La práctica se realizará con la colección MovieLens, compuesta por 100.000 valoraciones de 943 usuarios sobre 1682 películas.

Para tal fin, se desarrollará una aplicación que muestre al usuario 20 películas al azar y éste las evalúe asignándole un valor de 1 estrella, indicando que no le gusta nada, hasta 5 estrellas (es una de sus películas favoritas). Una vez obtenidas las valoraciones del usuario, el sistema procederá a calcular el vecindario, es decir, el grupo de usuarios que más se parecen a él en cuanto a las películas vistas y a las valoraciones dadas. Finalmente, la aplicación predecirá la valoración para el usuario activo de todas las películas que han visto sus vecinos y que no ha visto el usuario activo, y le mostrará aquellas predichas con cuatro o cinco estrellas.

2. Descripción de la práctica

El código de la práctica se ha desarrollado en **Python** usando las siguientes bibliotecas:

- **Pandas** → Una biblioteca para el análisis y tratamiento de datos. ("Pandas - Python Data Analysis Library" n.d.)
- **PyInquirer** → Una biblioteca para realizar interfaces interactivas por terminal. (CITGuru n.d.)

La práctica incluye dos archivos que tienen la siguiente funcionalidad: (Luna n.d.)

- **MoviesAndUsers.py** → Incluye toda la logística para cargar los dataset y hacer los cálculos sobre los datos. (Mwiti 2018)
- **RecommenderSystem.py** → Contiene la interfaz de la aplicación.

A continuación se detallará más los métodos de cada clase.

2.1. MoviesAndUsers

Esta clase incluye todos los métodos necesarios para cargar los dataset y realizar los cálculos necesarios. Los métodos más importantes de esta clase son los siguientes:

- **__init__** → Constructor de la clase. Aquí se indican la ruta de los ficheros de los dataset.
- **read_files** → Carga los dataset *u.data* y *u.item* y elimina las columnas que no son necesarias. También crea un nuevo dataset que contiene la media de las valoraciones de cada usuario.
- **rank_movies** → Devuelve 20 películas al azar para que las valore un usuario.

- **coef_pearson** → Calcula el coeficiente de pearson entre dos usuarios.
- **nearest_neighbors** → Calcula los k vecinos más cercanos de un usuario.
- **recommendation** → Hace una recomendación al usuario que se pasa como argumento. Hay que indicar el tamaño de vecindario que se quiere usar para realizar la recomendación.

Para más detalles acerca de los métodos se puede consultar el fichero *MoviesAndUsers.py* que incluye documentación más extensa.

2.2. RecommenderSystem

Este fichero es un script que lanza la interfaz por terminal para interactuar con el usuario. Se ha usado la biblioteca **PyInquirer** que nos permite declarar una serie de preguntas y nos captura los datos de las respuestas que se introduzcan.

Las opciones de las que dispone la interfaz son, bien, crear un nuevo usuario para el sistema que tendrá que evaluar 20 películas que se muestran al azar, o bien, usar un usuario del sistema proporcionando su ID.

Ejemplo valorando películas:

```
? Do you want to make a new user?  Yes
? What score do you give to Schizopolis (1996)?  Four
? What score do you give to Othello (1995)?  Three
? What score do you give to Poison Ivy II (1995)?  Two
? What score do you give to Some Folks Call It a Sling Blade (1993)?  (Use arrow
> Five
  Four
  Three
  Two
  One

? What score do you give to Breakfast at Tiffany's (1961)?  Five
? What score do you give to All About Eve (1950)?  Five
We recommend you...:

Silence of the Lambs, The (1991)

Young Frankenstein (1974)

Monty Python and the Holy Grail (1974)

Sting, The (1973)

Raiders of the Lost Ark (1981)
```

Ejemplo usando un usuario del sistema:

```
? Do you want to make a new user? No
? Give me a user ID 85
We recommend you...:

Edge, The (1997)

Rock, The (1996)

Hoodlum (1997)

Eve's Bayou (1997)

187 (1997)
```

Para más detalles se puede consultar el fichero *RecommenderSystem.py* que incluye más documentación.

3. Instrucciones para la ejecución

1. Descomprimir el fichero BejarMaldonadolreneP4.zip
2. Estando en el directorio BejarMaldonadolrene/RecommenderSystem creamos un entorno virtual con:

```
python3 -m venv env
```

3. Activamos el entorno virtual con:

```
source env/bin/activate
```

4. Instalamos las dependencias necesarias:

```
pip install -r requirements.txt
```

5. Ejecutamos la interfaz:

```
python3 RecommenderSystem.py
```

Referencias

CITGuru. n.d. "CITGuru/PyInquirer." GitHub. Accessed March 23, 2020.

<https://github.com/CITGuru/PyInquirer>.

Luna, Juan Manuel Fernández. n.d. *Tema 5 Sistemas de Recomendación*. Accessed March 23, 2020.

Mwiti, Derrick. 2018. "How to Build a Simple Recommender System in Python." Medium. Towards Data Science. October 3, 2018.

<https://towardsdatascience.com/how-to-build-a-simple-recommender-system-in-python-375093c3fb7d>.

"Pandas - Python Data Analysis Library." n.d. Accessed March 23, 2020.

<https://pandas.pydata.org/>.