

# SISTEMAS COMPLEJOS ADAPTATIVOS Y RECOMENDACIÓN

Jaime Ferrando Huertas

Junio 2021 MIARFID, UPV

## Contents

<b>1</b>	<b>Introducción</b>	<b>2</b>
<b>2</b>	<b>Dataset</b>	<b>2</b>
<b>3</b>	<b>Sistemas de recomendación</b>	<b>2</b>
3.1	Recomendación demográfica . . . . .	2
3.2	Recomendación colaborativa . . . . .	3
3.3	Recomendación híbrida . . . . .	5
<b>4</b>	<b>Web de recomendación</b>	<b>5</b>
4.1	Tecnologías y diseño . . . . .	6
4.2	Registro . . . . .	6
4.3	Recomendación demográfica . . . . .	7
4.4	Recomendación colaborativa . . . . .	8
4.5	Recomendación híbrida . . . . .	8
<b>5</b>	<b>Conclusiones</b>	<b>9</b>
<b>6</b>	<b>Futuros trabajos</b>	<b>9</b>

# 1 Introducción

En esta memoria vamos a tratar distintos sistemas de recomendación y mostrar una implementación web que sirva recomendaciones para películas con los distintos sistemas explicados. Un sistema de recomendación, es una subclase de sistema de filtrado de información que busca predecir la "calificación" o "preferencia" que un usuario le daría a un artículo. Estos sistemas son usados en multitud de áreas, con ejemplos como recomendación de música, ropa, servicios de video. Cada sistema funciona de una manera distinta, pero siempre suelen estar basados en información del usuario como entrada al sistema.

En esta memoria explicaremos sistemas de recomendación basándonos en el dataset de ratings para películas de MovieLens [1].

## 2 Dataset

El dataset de MovieLens que usaremos en este proyecto es un subconjunto del dataset original. Este dataset ha sido proporcionado por los responsables de la asignatura y cuenta con cuatro ficheros **users.txt**, **u1\_base.txt**, **genre.txt** y **items.txt**.

- **Users.txt** contiene las id para cada usuario junto con su edad, sexo y ocupación. Estas características serán especialmente útiles para el recomendador demográfico.
- **Genre.txt** Contiene los distintos géneros de nuestro dataset y su id. En total son 19 géneros.
- **Items.txt** Contiene las ids de géneros correspondientes a cada película de nuestro dataset junto al título de la misma.
- **u1\_base.txt** Datos de las puntuaciones de los usuarios a cada película. Contiene id del usuario, id película y ratio dado por el usuario a esta.

Con este dataset seremos capaces de crear perfiles de usuarios necesarios para nuestros sistemas de recomendaciones.

## 3 Sistemas de recomendación

### 3.1 Recomendación demográfica

Los sistemas demográficos se basan en características demográficas de los usuarios para crear grupos de usuarios y crear recomendaciones para un usuario basándonos en el grupo de usuarios al que pertenece.

# Demographic Recommendation

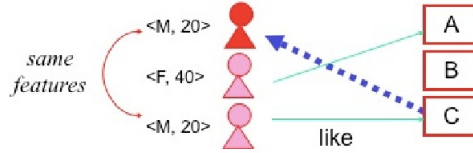


Figure 1: Ejemplo recomendación demográfica

Por ejemplo si una persona A es mujer, ejecutiva y de edad 24 este sistema podría crear un grupo de usuarios (mujer, ejecutivas, 20-30) y crear recomendaciones para la persona A con base en las preferencias de este grupo.

Para nuestro sistema de recomendación demográfica hemos usado las características **sexo**, **edad** y **profesión** como base para crear grupos de usuarios. Una vez el grupo de usuarios está definido usaremos el *voto medio* como métrica para escoger las películas recomendadas para un usuario. Usaremos esta métrica para dar una puntuación personal a cada película para cada usuario. Necesitaremos los siguientes datos:

- $v$ , el número de votos para la película dentro del grupo demográfico del usuario.
- $m$ , mínimo número de votos para considerar la película, 90% del percentil del número de votos para todas las películas votadas dentro del grupo demográfico del usuario.
- $R$ , voto medio de la película dentro del grupo demográfico del usuario.
- $C$ , voto medio de la película en todos los grupos demográficos.

$$P(X) = \frac{v}{(v + M) * R} + \frac{M}{(M + v) * C} \quad (1)$$

## 3.2 Recomendación colaborativa

Este sistema empareja a personas con intereses similares y proporciona recomendaciones basadas en este emparejamiento. Los filtros colaborativos no requieren metadatos de los usuarios como sus contrapartes basadas en características demográficas.

Por ejemplo, si a la persona A le gustan 3 películas, digamos Interstellar, Inception y Predestination, y a la persona B le gustan Inception, Predestination y The Prestige, entonces tienen intereses casi similares. Podemos decir con cierta certeza que a A debería gustarle The Prestige y a B debería gustarle Interstellar.

El algoritmo de filtrado colaborativo utiliza "Comportamiento del usuario" para recomendar elementos. Este es uno de los algoritmos más utilizados en la industria, ya que no depende de ninguna información adicional y puede ser entrenado dado un historial de usuarios e interacciones con los objetos a recomendar.

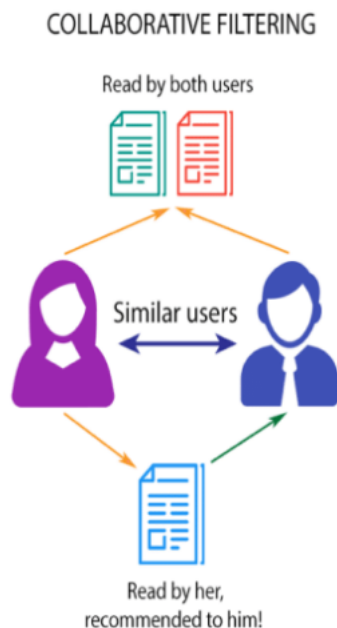


Figure 2: Ejemplo recomendación colaborativa

Para emparejar usuarios similares primero debemos crear una representación vectorial de estos. En nuestro caso hemos creado una matriz de similitudes con las películas puntuadas por cada usuario, donde cada fila de la matriz representa las preferencias de un usuario. Hay otras opciones para la creación de esta matriz como similitudes basadas en preferencias de usuario, pero nosotros decimos usar esta opción, ya que nos reportaba mejores resultados en testing.

Una vez tenemos esta matriz podemos buscar los vecinos cercanos a un usuario  $u$ , existen distintas funciones para calcular distancias/similitud entre dos vectores, pero nosotros hemos escogido el coeficiente de correlación de Pearson. Para calcular el coeficiente de que una película  $i$  sea recomendada a nuestro usuario  $u$  usaremos la siguiente fórmula basada en **KNN**, **K-Nearest Neighbors**:

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} p(u, v) \cdot r_{vi}}{\sum_{v \in N_i^k(u)} p(u, v)} \quad (2)$$

Donde tenemos distintos elementos:

- $u$ , usuario al que estamos creando recomendaciones
- $N_i^k(u)$ , conjunto de usuarios vecinos a  $u$  con un coeficiente de correlación de Pearson positivo y que han puntuado la película  $i$
- $p(u, v) \cdot r_{vi}$ , coeficiente de correlación de Pearson entre nuestro usuario y uno de sus vecinos multiplicado por el rating (puntuación) del vecino para  $i$ .

Con esta fórmula podemos calcular una puntuación para todas las películas  $i$  para un usuario  $u$  y recomendar aquellas con la máxima puntuación.

### 3.3 Recomendación híbrida

En estos sistemas se intenta usar distintos sistemas de recomendación de manera simultánea. Hay multitud de métodos para conseguir una implementación simultánea, pero nosotros hemos optado por uno de los más sencillos. Escogeremos las mejores recomendaciones de cada implementación y las juntaremos en una lista ordenada que será entregada al usuario.

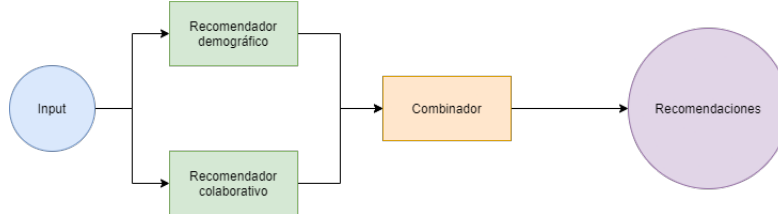


Figure 3: Ejemplo recomendación híbrida

La parte del combinador de la figura 3 es donde se ejecuta la lógica de recomendación híbrida. Para nuestro caso se escogen las mejores recomendaciones de cada sistema de entrada para crear la salida.

## 4 Web de recomendación

En esta sección explicaremos la implementación de la web que hemos creado para servir recomendaciones de películas para este trabajo.

## 4.1 Tecnologías y diseño

Para la web hemos usado **Flask** como servidor backend que genera predicciones y las sirve en formato web con archivos **html**. El servidor backend funciona bajo python y es capaz de interactuar con nuestras implementaciones de cada sistema recomendador. Para el recomendador demográfico las recomendaciones son calculadas en tiempo real debido a que su tiempo de cómputo es ínfimo y nos lo podíamos permitir. Para las recomendaciones colaborativas las pre computamos para cada usuario y son accedidas por el backend para servir las, es decir se calculan de manera "offline".

Para mostrar las películas recomendadas al usuario hemos usado una API pública **OMDb API** para conseguir los posters asociados a cada título. Al principio solo conseguíamos encontrar posters para el 70% de las películas que teníamos, pero tras hacer procesado sobre los títulos proporcionados en el dataset conseguimos subirlo a un 99%.

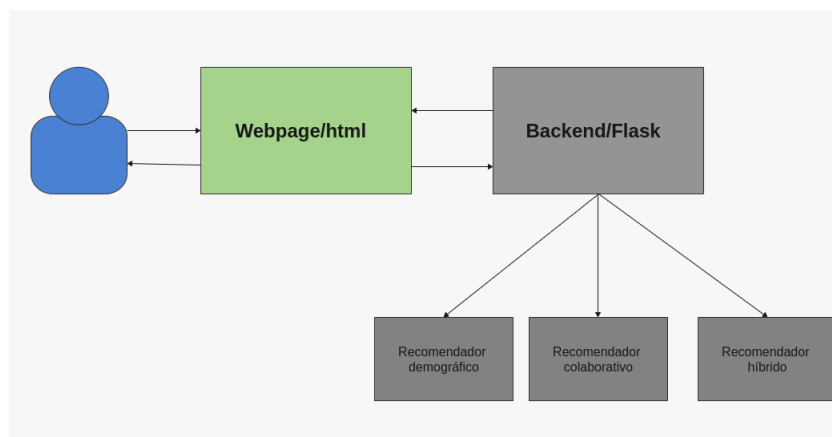


Figure 4: Esquema de web de recomendación

Ahora pasamos a explicar más en detalle las distintas secciones de la web.

## 4.2 Registro

En la sección de registro se permite crear un nuevo usuario dado características demográficas. Ya que el usuario es nuevo no cuenta con ningún historial de películas y solo obtendrá recomendaciones de los sistemas demográficos e híbridos (equivalente en su caso).

## Recomendador de películas SCAR

Selecciona el sistema de recomendador que prefieras

Registro
Demográfico
Colaborativo
Híbrido

Nuevo usuario: Soy Female technician 10-20

Male other 21-30

trabajo en senior executive 31-40

y mi edad esta entre 63,280

Registrar usuario

Usuario registrado con id 1000

Figure 5: Sección de registro

Podemos observar una interfaz que pide características al usuario y devuelve la id del nuevo usuario para su posterior uso.

### 4.3 Recomendación demográfica

En esta sección se sirven las recomendaciones demográficas, podemos hacer uso del **user\_id** para encontrar recomendaciones personalizadas al usuario, podríamos también usar el user\_id de usuarios registrados manualmente.

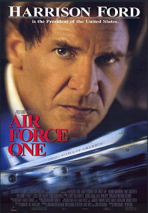
## Recomendador de películas SCAR

Selecciona el sistema de recomendador que prefieras


Registro
Demográfico
Colaborativo
Híbrido

Usuario existente con user\_id


Recomienda!




Ratio 4.39




Ratio 4.31



Ratio 4.31



Ratio 4.31



Ratio 4.31

Figure 6: Sección de registro recomendación demográfica

La interfaz cuenta con pósters para cada película recomendada junto al rating que se le ha asignado al usuario y hace que estas recomendaciones sean las devueltas.

## 4.4 Recomendación colaborativa

En esta sección se sirven las recomendaciones colaborativas, podemos hacer uso del **user\_id** para encontrar recomendaciones personalizadas al usuario, en este caso no podríamos también usar el user\_id de usuarios registrados manualmente.

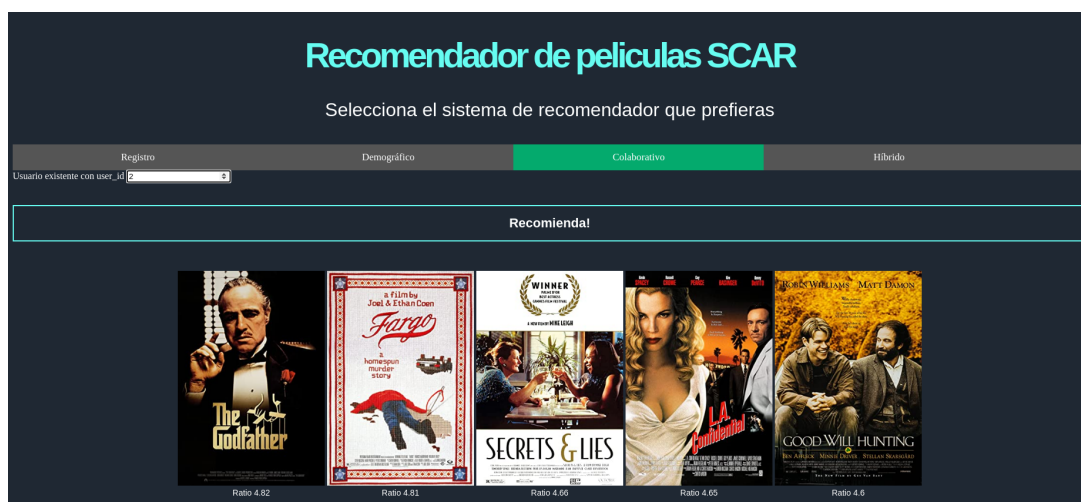


Figure 7: Sección de registro recomendación colaborativa

La interfaz cuenta con pósters para cada película recomendada junto al rating que se le ha asignado al usuario y hace que estas recomendaciones sean las devueltas.

## 4.5 Recomendación híbrida

En esta sección se sirven las recomendaciones colaborativas, podemos hacer uso del **user\_id** para encontrar recomendaciones personalizadas al usuario, en este caso podríamos también usar el user\_id de usuarios registrados manualmente, pero las recomendaciones obtenidas serian equivalentes a las demográficas, ya que los usuarios nuevos no tienen historial de películas y no tienen recomendaciones colaborativas.





Figure 8: Sección de registro recomendación híbrida

La interfaz cuenta con pósters para cada película recomendada junto al rating que se le ha asignado al usuario y hace que estas recomendaciones sean las devueltas.

## 5 Conclusiones

En este trabajo hemos tenido una experiencia práctica de implementación de un sistema de recomendación. Si ya habíamos visto los sistemas de recomendación y la lógica que siguen en clase este trabajo ha servido para aprender una implementación real de los mismos. Personalmente la parte que ha necesitado más trabajo ha sido la creación de la web. Estamos muy contentos con el trabajo, ya que difiere de lo que hemos visto en otras asignaturas y hemos tenido mucha libertad a la hora de implementar el sistema por nuestra cuenta.

## 6 Futuros trabajos

Para futuros trabajos nos gustaría centrarnos más en la lógica de recomendadores, probar sistemas actuales basados en deep learning. Pensamos también que un sistema que tome en cuenta el orden de visionado de películas del usuario también podría tener mejor rendimiento. El gusto de un usuario no es el mismo cuando empieza su "vida" cinematográfica que cuando le estamos recomendados películas, si de alguna forma podemos capturar esta información temporal y dar peso a las interacciones más recientes puede que creemos un sistema recomendador más potente.

Una línea de trabajo también nos gustaría explorar sería implementar algún tipo de feedback loop con el usuario en el que este valorara las recomendaciones

para nosotros ajustar nuestros algoritmos en el lado de backend.

## References

- [1] Grouplens. Movielens, a movie ratings dataset.