

# WHAT TO WATCH

## WEBIR 2017

### Grupo 01:

María Lucía Barrero	4.412.531-1
Bernabé González	4.863.201-5
Joaquín Lejtregger	4.550.703-3

### Docente:

Libertad Tansini

# Contenido

<b>Introducción</b>	<b>3</b>
<b>Problema</b>	<b>3</b>
<b>Enfoque de la solución</b>	<b>3</b>
<b>Diseño</b>	<b>4</b>
<b>Arquitectura</b>	<b>5</b>
<b>Implementación</b>	<b>6</b>
<b>Funcionalidades y uso</b>	<b>7</b>
<b>Evaluación y resultados</b>	<b>9</b>
<b>Conclusiones</b>	<b>11</b>
<b>Trabajo Futuro</b>	<b>12</b>
<b>Referencias</b>	<b>13</b>

## Introducción

La gran cantidad y variedad de información disponible en la web hace difícil la elección en base a los gustos de cada persona. De esta necesidad surgen los Sistemas de Recomendación.

El proyecto “**WHAT TO WATCH**” consiste en un programa recomendador de películas para mirar, basándose en la información obtenida de las películas, las preferencias conocidas del usuario y las de los usuarios con similitudes.

El desarrollo del mismo permitió ahondar en los conceptos de extracción de información de la web y filtrado colaborativo.

## Problema

El problema abarca la necesidad de un usuario de filtrar entre la gran cantidad de películas disponibles en la red, algunas recomendadas acordes a sus preferencias, sus gustos y su perfil.

Es decir, se busca crear un Sistema Recomendador de películas, que filtre los resultados que van a interesarle al usuario.

## Enfoque de la solución

Para solucionar el problema planteado, en lugar de hacer un ranking promedio de cada película, fue utilizado filtrado colaborativo basándose en el algoritmo SVD.

El filtrado colaborativo consiste en un método para hacer predicciones automáticas sobre el interés de un usuario utilizando la información conocida sobre las preferencias que otros usuarios colaboradores tomaron en el pasado. Es decir, dado un usuario 1, si tiene gustos en común con el usuario 2, entonces es más probable que en otro aspecto el usuario 1 prefiera algo también elegido por el usuario 2 que por un usuario al azar.

En el caso del problema planteado, cada vez que se accede al sistema y se califican películas, la información de (usuario, película, ranking) es almacenada. Esto permite, a partir de esa lista ubicar los usuarios vecinos para poder hacer las predicciones de las películas que le gustan a un nuevo usuario ingresado. Estas predicciones son específicas para el usuario dado, pero se obtienen usando la información del resto de los usuarios.

La elección del algoritmo SVD, utilizado por *Netflix*, se debe a que obtiene muy buenas recomendaciones, pero como desventaja se tiene que demora mucho tiempo en poder entrenarse para realizar las recomendaciones.

En el sistema se tiene una matriz de *rankings* con los votos de los usuarios sobre las películas (*movies*). Al realizar una descomposición SVD de la misma, se obtienen tres matrices U (*usuarios \* k*), S y V (*movies \* k*) lo que implica que la matriz U tiene *k\_factores* que identifican al usuario y la matriz V tiene los mismos *k\_factores* que identifican a las películas. Esto permite identificar los factores que caracterizan a los usuarios y las películas del sistema y de esta forma obtener solamente las matrices de los factores característicos de usuarios y películas, lo que permite el filtrado colaborativo.

## Diseño

El sistema cuenta con:

- Interfaz gráfica web del usuario, desarrollada en React.
- Servidor de backend, desarrollado en Python.
- Base de datos propia, en MongoDB.

Un usuario accede a la aplicación y se le brinda una lista de películas para que califique. Se le solicita un mínimo de cinco calificaciones pero puede seguir haciéndolo, inclusive puede solicitar que sean cargadas más películas.

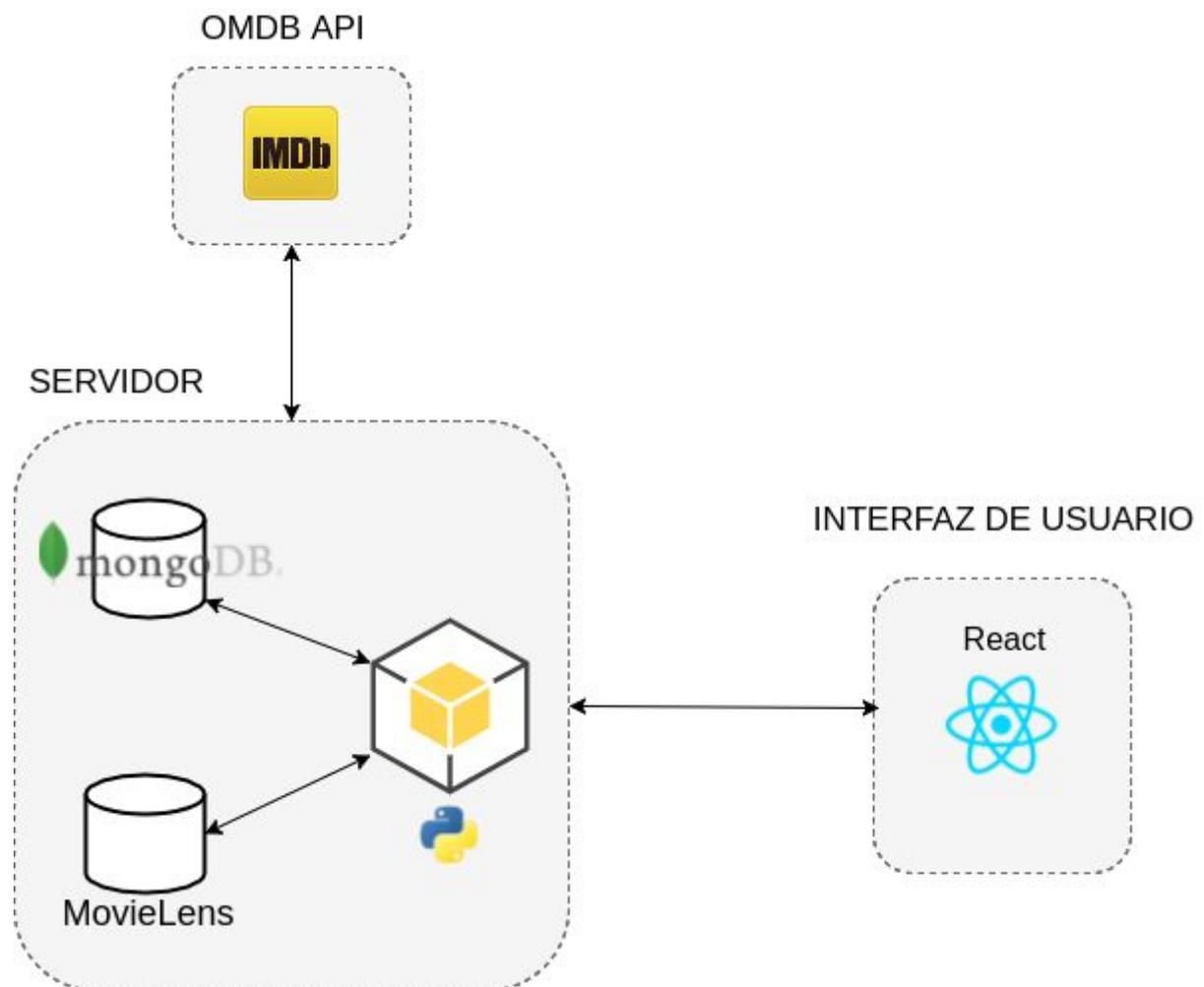
Luego, los puntajes asignados son almacenados en la base de datos y se entrena el algoritmo, combinando los resultados de otros usuarios con la base de datos de MovieLens, para recomendar las películas.

Para la elección de las películas a recomendar fue realizado un filtrado por reglas para seleccionar solamente las películas que hayan sido estrenadas después del año 2000.

De esta forma se evita que el sistema recomiende clásicos que ya fueron vistos por la mayoría de la gente y puede enfocarse en un conjunto más diverso de películas, que están menos establecidas como clásicos.

## Arquitectura

Para lograr el diseño previamente explicado se presenta la siguiente arquitectura:



## Implementación

El usuario, al entrar a la página, recibe una lista de películas (estrenadas después del 2000) seleccionadas de forma semi-randómica desde la API de OMDb (basándose también en la popularidad de las películas).

Se le solicita al usuario que califique las películas, y mientras eso ocurre, se guardan sus decisiones en memoria.

Una vez que el usuario ya puntuó las películas que vio, el sistema almacena los datos en la base de datos de mongoDB, para mejorar el algoritmo.

Como datos para el entrenamiento del algoritmo, se utilizan los datos de la base de datos mongo (interna), combinados con la base de datos de MovieLens utilizando la librería "Pandas" que permite administrar datos de forma eficiente. Pandas permite homogeneizar los datos que se tienen de ambas bases de datos, para poder usar el conjunto completo para entrenar el algoritmo.

Para el filtrado colaborativo se utiliza principalmente la librería Surprise para Python. Esta librería permite implementar diversos algoritmos para este tipo de recomendación. En particular, fue elegido el SVD.

La librería implementa filtrado colaborativo basado en usuarios (en contraposición con el basado en ítems). Y permite calcular la probabilidad de que a cierto usuario le gusten las películas de la base de datos. A partir de esto se toman los 10 "vecinos más cercanos", que son las películas de la base de datos interna que mejor puntuadas están.

Una vez que obtenidas estas 10 películas, es utilizada la tabla de MovieLens que mapea un id de película con un id de IMDB.

Con ese id de IMDB de la película, se solicita, utilizando la API de OMDb, información sobre las películas (nombre, poster, etc) y son mostradas al usuario.

## Funcionalidades y uso

Para acceder a la funcionalidad de recibir recomendaciones simplemente deben acceder a la aplicación web.

En el momento en que un nuevo usuario ingresa, se le asocia un identificador en el sistema y le despliega películas al azar dentro de las cien más populares publicadas a partir del año 2000, para que el usuario califique (positivamente o negativamente) al menos cinco películas.



Una vez que estos datos son ingresados, el sistema recomienda películas que aún no hayan sido calificadas por el usuario. Además, el sistema entrena con los nuevos datos entonces continúa mejorando su filtrado colaborativo cada vez que un nuevo usuario lo utiliza.



## Evaluación y resultados

Para evaluar el sistema fue solicitado probar el programa a diversas personas, con distintos perfiles. Se les pidió que calificaran películas y luego fueron consultados si las recomendaciones les parecían aceptables.

En general fue recibido buen feedback en cuanto a la calidad de la recomendación, sin embargo, un aspecto que las personas que testearon consideraron que debía mejorar fue el hecho de que el sistema toma demasiado tiempo en entrenar el algoritmo y buscar la información de las películas, antes de devolver las recomendaciones.

Los sistemas de recomendaciones no son fáciles de evaluar debido a que solo la persona que realizó las calificaciones puede determinar si los resultados son aceptables o no.

De todas formas, se buscó determinar si los resultados tienen sentido con las calificaciones ingresadas.

Primero un usuario calificó las siguientes películas:

Positivamente: Star wars, El origen, Harry potter y la camara secreta, La isla, Atrapame si puedes

Negativamente: Todo poderoso, Los increíbles, Up

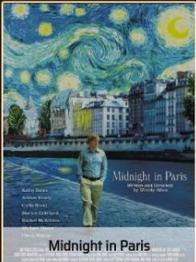
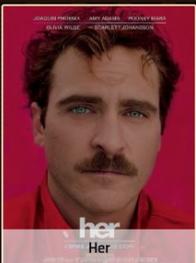
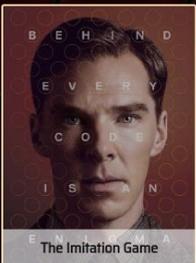
Según las calificaciones ingresadas se puede ver que al usuario no le gustan mucho las películas animadas, y le gustan las sagas como Harry Potter o Star wars. Además la mayoría son del género suspenso o drama.

Por supuesto que el género, director o actores no deberían determinar en sentido estricto los resultados, es allí donde se encuentra la potencia del filtrado colaborativo, de poder relacionar películas más allá de sus características.

Luego de obtener los resultados, se encuentran dos películas de Harry Potter y ninguna película animada. Los resultados tienen sentido con las calificaciones ingresadas.

**ENCONTRAMOS LAS PELÍCULAS PERFECTAS PARA TI**

Listo! Las siguientes películas son compatibles con tus gustos!

 <p>Little Miss Sunshine</p>	 <p>The Prestige</p>	 <p>Harry Potter and the Half-Blood Prince</p>
 <p>Hachi: A Dog's Tale</p>	 <p>Midnight in Paris</p>	 <p>Harry Potter and the Deathly Hallows: Part 2</p>
 <p>Her</p>	 <p>Interstellar</p>	 <p>The Imitation Game</p>

**VOVER A INTENTARLO**

## Conclusiones

El filtrado colaborativo basado en usuarios es una estrategia que resulta muy útil en este tipo de problemas, ya que en el ámbito del entretenimiento la hipótesis de que “usuarios similares buscan resultados similares”

Por el tipo de algoritmo planteado, no es posible evitar el reentrenamiento del algoritmo cuando un usuario ingresa al sistema. Como contrapartida se obtienen mejores resultados, ya que el algoritmo se va perfeccionando constantemente.

## Trabajo Futuro

Creemos que la aplicación tiene mucho espacio para mejorar, a continuación se detallan algunos aspectos:

1. Implementar un sistema de usuarios y persistir el algoritmo de filtrado colaborativo para reducir el tiempo de espera en la experiencia del usuario.
2. Una vez que la base de datos crezca y el sistema de usuarios esté implementado, configurar cuándo volver a entrenar el algoritmo para las recomendaciones a los usuarios ya registrados. Por ejemplo, cada cierta cantidad de inserts en la base de datos, o cada cierto tiempo (tres veces por día).
3. Subdividir la base de datos por género para que el usuario pueda recibir recomendaciones más específicas.
4. Agregar un buscador que le permita al usuario encontrar las películas que desee calificar.
5. Una vez que la base de datos de usuario sea más grande, prescindir de MovieLens y utilizar únicamente una base de datos propia.
6. Combinar con algún servicio de streaming/youtube para ver la película o ver un trailer de esta.

## Referencias

Surprise - <http://surpriselib.com/>  
Python - <https://www.python.org/>  
MovieLens - <https://movielens.org/>  
OMDb - <http://www.omdbapi.com/>  
IMDb - <http://www.imdb.com/>  
Flask - <http://flask.pocoo.org/>  
Pandas - <https://pandas.pydata.org/>