



Primera parte, proyecto 2



Francisco Rosal - 18676
Gian Luca Rivera - 18049
Algoritmos y Estructura de Datos
Sección 20

INTRODUCCIÓN

Existe una problemática actual que es el saber que queremos. Vivimos en una época donde la información y data se han proliferado de gran manera y es tanta que es imposible llegar a toda y descubrirla. Sin embargo, hay herramientas que nos pueden ayudar a navegar entre toda esta información de diferentes maneras y hacemos uso de estas incluso sin saberlo. Una de estas herramientas la usamos diariamente, al usar nuestras redes sociales, navegar por internet e incluso al entrar a nuestra plataforma favorita de reproducción de películas y series como Netflix.

Este sistema que utilizamos tan constantemente es llamado: sistema de recomendación y es un algoritmo que forma parte de los sistemas de filtrado de información. Este trata de predecir el gusto o valor que tendría algo para algún usuario, dando recomendaciones de cosas que a él usuario le agradarían; funcionando como cuando un amigo te recomienda una película.

La recomendación está cada vez más presente en la tecnología de uso diario que está directamente relacionada con la información. Los sistemas de recomendación son los sistemas de software que permiten a un usuario obtener propuestas de productos en base a sus gustos. Con estos sistemas, se le proporciona al usuario una herramienta efectiva para conseguir lo que busca en menos tiempo.

ALGORITMOS EXISTENTES

Los sistemas de recomendación, normalmente, funcionan a base de las valoraciones o reseñas que los usuarios les dan a ciertos productos de alguna plataforma, sin embargo, hay otro tipo de plataformas que no funcionan exactamente así ya que no se tiene información directa de los usuarios. En este caso, los sistemas se basan de las interacciones que los usuarios realizan con los productos que proporciona la plataforma (y son precisamente estas las plataformas que utilizan grafos para representar las interacciones). Algunos algoritmos que existen son:

- Collaborative Filtering: Este método se basa en la recopilación y el análisis de una gran cantidad de información sobre los comportamientos, actividades o preferencias de los usuarios y en la predicción de lo que les gustará a los usuarios en función de su similitud con otros usuarios. Una ventaja de este método es que no se basa en el contenido analizable de la máquina, por lo que, es capaz de recomendar con precisión elementos complejos sin requerir una comprensión del elemento en sí.
- Content Based Filtering: Se basa en una descripción del elemento y un perfil de las preferencias del usuario. Las palabras clave se utilizan para describir los elementos y se construye un perfil de usuario para indicar el tipo de elemento que gusta a este usuario. Estos algoritmos intentan recomendar elementos que osn similares a los que un usuario le gustó en el pasado por lo que varios artículos

se comparan con artículos que previamente fueron calificados por el usuario y se recomiendan los elementos de mejor coincidencia.

- Hybrid Recommendation Systems: Este algoritmo combina los dos algoritmos anteriores y se ha demostrado que es más efectivo en algunos casos. La implementación puede ser variada ya que se pueden hacer predicciones basadas en el contenido y basadas en la colaboración por separado y luego se combinan, se agregan capacidades basadas en la colaboración o sino unificando los enfoques en un solo modelo. Este algoritmo supera problemas como el arranque en frío y la escasez.
- KNN: Este algoritmo clasifica cada dato nuevo en el grupo que corresponda, según tenga k vecinos más cerca de un grupo o de otro. El algoritmo calcula la distancia de un elemento nuevo a cada uno de los elementos ya existentes y los ordena de menor a mayor para seleccionar al grupo al que pertenece.

Plataformas con sistemas de recomendación:

- Netflix: El sistema de netflix ofrece recomendaciones personalizadas para ayudar a encontrar series y películas del interés de cada usuario. Cada vez que se ingresa a Netflix, el sistema de recomendaciones encuentra una serie o película del agrado del usuario, para calcular la probabilidad de que si es del interés del usuario se basan en factores como: la interacción del usuario con el servicio (historial de visualización y calificaciones que ha dado el usuario), información sobre títulos, género, categorías, actores, años de lanzamiento, hora del día en la que se utiliza la plataforma, dispositivo que se utiliza, durante cuánto tiempo es utilizado el sistema y por medio de actividades de otros usuarios que tengan gustos similares al usuario. Su sistema de recomendaciones no incluye información como la edad o el género.

DESIGN THINKING

EMPATÍA

Para saber qué sistema de recomendación es el más adecuado a los usuarios, se realizaron dos encuestas: la primera consiste en tres preguntas sencillas en donde le preguntamos al usuario acerca de que tema le gustaría tener un sistema de recomendación. La encuesta se miraba de la siguiente manera:

1. ¿Sabe que es un sistema de recomendación?

Si
 No

2. Actualmente, ¿Utilizas un sistema de recomendación?

Si
 No

3. Preferirías un sistema de recomendación de películas, libros, canciones, recetas?

Películas Recetas
 Libros Otro
 Canciones

0 de 3 respondidas

La segunda encuesta se utilizó para poder llenar la base de datos con los gustos que tienen los usuarios y así poder crear relaciones entre ellos. Esta encuesta tenía el siguiente formato:

Sistema de recomendación de películas

* 1. Ingrese su nombre y apellido

* 2. ¿En que te basas para escoger una película?

Género
 Director
 Actores
 Casa productora
 Calificación
 Premios

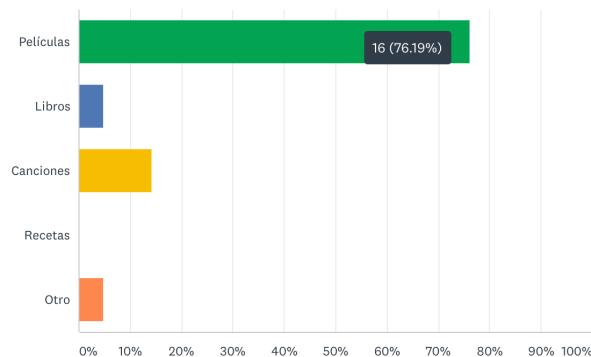
* 3. Ingrese sus películas favoritas (Mínimo 5)

DEFINICIÓN

Según la primera encuesta que se pasó en la fase de empatía, los resultados acerca de qué sistema de recomendación prefieren los usuarios es el de películas ya que el 76.16% de los usuarios optaron por esta respuesta como se muestra en la siguiente gráfica:

Preferirías un sistema de recomendación de películas, libros, canciones, recetas?

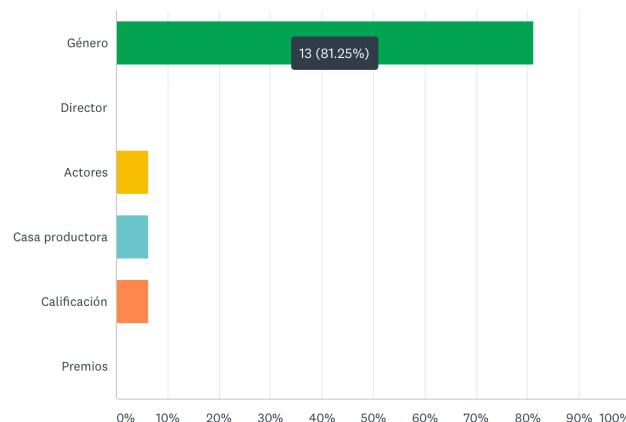
Answered: 21 Skipped: 0



Según la segunda encuesta de la fase de empatía, los usuarios eligen una película según su género sobre todas las otras opciones como se muestra en la siguiente gráfica:

¿En que te basas para escoger una película?

Answered: 16 Skipped: 0



IDEA

- Crear un sistema de recomendación de películas con el algoritmo content based filtering en donde la recomendación principal se va a basar por medio del género de la película.
- Crear un sistema de recomendación de películas con el algoritmo híbrido en donde se recomiende según los actores que presente la película
- Crear un sistema de recomendaciones de películas que se base en los gustos de otros amigos para sugerirle al usuario

- Crear un sistema de recomendaciones para películas en donde sugerirá las películas de su género favorito que hayan sido nominados a algún tipo de premio relacionado al cine.

PROTOTIPADO

Al ingresar el programa y seleccionar la primera opción se verá así:

```
Bienvenido al sistema de recomendación de películas
-----
Ingrresa tu usuario:
>> Luca RB

¡Hola Luca RB!
Seleccione una opción para continuar:
1. Mostrar películas que me gustan
2. Añadir una película que me gusta
3. Mostrar películas que me podrían gustar
4. Salir
>> 1

- Star Wars Episodio 3
- Star Wars Episodio 6
- Interestelar
- Aquaman
- Pasante de moda
- El Hobbit
- Star Wars Episodio 2
```

Al ingresar el programa y seleccionar la segunda opción se verá así:

```
Bienvenido al sistema de recomendación de películas
-----
Ingrresa tu usuario:
>> Luca RB

¡Hola Luca RB!
Seleccione una opción para continuar:
1. Mostrar películas que me gustan
2. Añadir una película que me gusta
3. Mostrar películas que me podrían gustar
4. Salir
>> 2

Ingrresa el nombre de la película que te gusta:
>> Han Solo: Una historia de Star Wars
Tu película se ha agregado con éxito!
```

Al ingresar el programa y seleccionar la tercera opción se verá así:

```
Bienvenido al sistema de recomendación de películas  
-----  
  
Ingresa tu usuario:  
>> Luca RB  
  
¡Hola Luca RB!  
Seleccione una opción para continuar:  
1. Mostrar películas que me gustan  
2. Añadir una película que me gusta  
3. Mostrar películas que me podrían gustar  
4. Salir  
>> 3  
  
- Star Wars Episodio 4  
- Gravity  
- Wonder Woman  
- Aquaman  
- El Diablo Viste de Prada  
- El Hobbit  
- Piratas del Caribe
```

Al ingresar el programa y seleccionar la cuarta opción se verá así:

```
Bienvenido al sistema de recomendación de películas  
-----  
  
Ingresa tu usuario:  
>> Luca RB  
  
¡Hola Luca RB!  
Seleccione una opción para continuar:  
1. Mostrar películas que me gustan  
2. Añadir una película que me gusta  
3. Mostrar películas que me podrían gustar  
4. Salir  
>> 4  
Gracias por utilizar el sistema de recomendación de películas.
```

TESTEO

Esta fase se encuentra en los documentos .txt dentro de la carpeta testing del repositorio y se muestran de la siguiente manera (la foto es la del entrevistado):

TESTING

Pablo Rosal:
Algoritmo: Prioridad

Peliculas que le gustan:
Movie 3: Iron Man 3
Movie 5: Avengers Age of Ultron

Entonces:
Movie 1: Avengers 1
Movie 2: Ant Man
Movie 4: La vida es bella
Actor 1: Chris Hemsworth
Actor 2: Robert Downey Jr.

Buscar pelicula similar por genero:
Avengers 1

Cantidad de similitudes de genero:
Action, SuperHeroes = 2

Cantidad de similitudes de actor:
Chris Hemsworth, Robert Downey Jr., Scarlett Johansson... = 10

(Otras categorias son omitidas para el testing)
Otra categoria = 3

Obtener el valor de Match:

$$(0.5(2) + 0.25(10) + 0.25(3)) / 2 + 10 + 3 = 0.28$$

Se ordena y se retorna

**TESTING**

Javier Castañeda:
Algoritmo: Prioridad

Peliculas que le gustan:
Movie 3: Avengers Infinity War
Movie 5: Avengers Endgame

Entonces:
Movie 1: Capitan America Civil War
Movie 2: Thor Ragnarok
Movie 4: Iron Man 3
Actor 1: Chris Evans
Actor 2: Mark Ruffalo

Buscar pelicula similar por genero:
Capitan America Civil War

Cantidad de similitudes de genero:
Action, SuperHeroes = 2

Cantidad de similitudes de actor:
Chris Hemsworth, Robert Downey Jr., Scarlett Johansson... = 10

(Otras categorias son omitidas para el testing)
Otra categoria = 3

Obtener el valor de Match:

$$(0.5(2) + 0.25(10) + 0.25(3)) / 2 + 10 + 3 = 0.28$$

Se ordena y se retorna



TESTING

Gabriela Lopez:
Algoritmo: Prioridad

Peliculas que le gustan:
Movie 3: Mamma Mia
Movie 5: Loco y estupido amor

Entonces:
Movie 1: Sing, you sinners
Movie 2: Hairspray
Movie 4: Alguien tiene que ceder
Actor 1: Meryl Streep
Actor 2: Steve Carell

Buscar pelicula similar por genero:
Sing, you sinners

Cantidad de similitudes de genero:
Comedia musical, comedia romantica = 2

Cantidad de similitudes de actor:
Steve Carell, Ryan Goslin, Emma Stone... = 10

(Otras categorias son omitidas para el testing)
Otra categoria = 3

Obtener el valor de Match:

$$(0.5(2) + 0.25(10) + 0.25(3)) / 2 + 10 + 3 = 0.28$$

Se ordena y se retorna



TESTING

Fernando Castañeda:
Algoritmo: Prioridad

Peliculas que le gustan:
Movie 3: La maldicion de la llorona
Movie 5: Soy leyenda

Entonces:
Movie 1: Annabelle
Movie 2: La monja
Movie 4: I am omega
Actor 1: Will Smith
Actor 2: Marisol Ramirez

Buscar pelicula similar por genero:
Annabelle

Cantidad de similitudes de genero:
Novela, Terror sobrenatural = 2

Cantidad de similitudes de actor:
Willow Smith, Alice Braga, Linda Cardellini... = 10

(Otras categorias son omitidas para el testing)
Otra categoria = 3

Obtener el valor de Match:

$$(0.5(2) + 0.25(10) + 0.25(3)) / 2 + 10 + 3 = 0.28$$

Se ordena y se retorna



TESTING

Luis Pedro Valenzuela:
Algoritmo: Prioridad

Películas que le gustan:
Movie 3: The intern
Movie 5: About time

Entonces:
Movie 1: Último viaje a las Vegas
Movie 2: El secreto de mi éxito
Movie 4: Realmente amor
Actor 1: Anne Hathaway
Actor 2: Rachel McAdams

Buscar película similar por género:
Último viaje a las vegas

Cantidad de similitudes de género:
Drama, Fantasía, Romance, Comedia = 4

Cantidad de similitudes de actor:
Anne Hathaway, Robert De Niro, René Russo... = 10

(Otras categorías son omitidas para el testing)
Otra categoría = 3

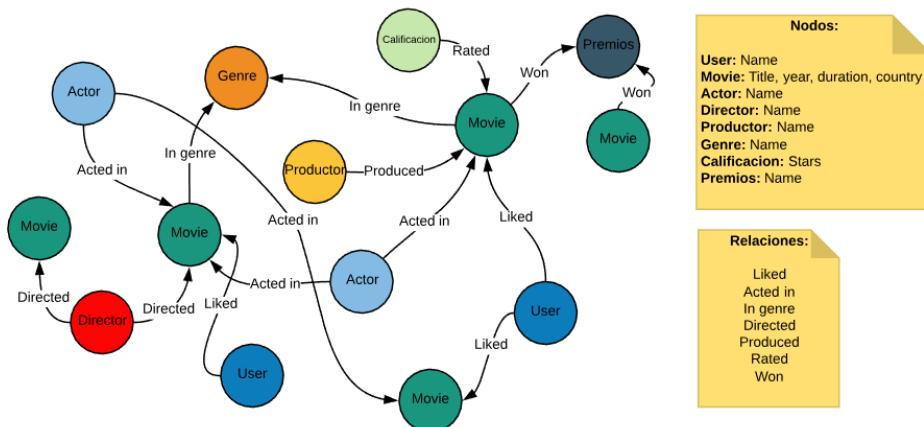
Obtener el valor de Match:
 $(0.5(2) + 0.25(10) + 0.25(3)) / 2 + 10 + 3 = 0.28$

Se ordena y se retorna



DISÑO DE LA BASE DE DATOS

El diseño de la base de datos de la cual se va a basar el programa será así:



Como se puede observar en el diseño, un usuario es un nodo que se une a otro nodo que es el de la película. Una película se conecta a otros nodos como el género, actores, productor, director, calificación y premios. Las relaciones se realizan de la manera siguiente: a un usuario le gusta una película → Un director dirige una película → Un actor actúa en una película → Una película gana un premio.

PSEUDOCODIGO

**Documento algoritmoDescriptivo.txt

Algoritmo de recomendacion:

Nuestro algoritmo estará basado en Content-Based Filtering, en el cual el usuario escoge películas que le gusten y a partir del contenido de la información se recomiendan otras películas.

Se buscaran similitudes basada en generos segun las películas ya vistas por el usuario ya que es lo que mayor afecta en la decision de las personas al buscar una película.
Al filtrar por genero se obtendra un numero que indica la cantidad de similitudes existentes con las películas de gusto del usuario.
Se hace igualmente con los actores, productora, calificacion, y director, reduciendo cada vez mas la cantidad de opciones.
Se muestran los valores en comun de cada conjunto por separado.
Al finalizar se obtiene el puntaje de coincidencia haciendo una suma de los valores obtenidos, cada uno multiplicado por un numero el cual es la prioridad que se le da a ese valor, a esta suma se le divide la suma de los valores obtenidos.
Se ordena la lista obtenida por ese puntaje de coincidencia y se retorna.

```
# Algoritmo
MATCH (u:User {name: "User name"})-[r:LIKED]->(m:Movie)

MATCH (m)-[:IN_GENRE]->(g:Genre)<-[:IN_GENRE]-(rec:Movie) WHERE NOT EXISTS( (u)-[:LIKED]->(rec) )
WITH u, m, rec, COUNT(g) AS genreSelection
MATCH (m)-[:ACTED_IN]-(a:Actor)-[:ACTED_IN]->(rec)
WITH u, m, rec, genreSelection, COUNT(a) AS actorSelection
OPTIONAL MATCH (m)-[:DIRECTED]-(d:Director)-[:DIRECTED]->(rec)
WITH u, m, rec, genreSelection, actorSelection, COUNT(d) AS directorSelection

MATCH (m)-[:IN_GENRE]->(mo:Movie)
WITH u, m, rec, genreSelection, actorSelection, directorSelection, COLLECT(mo.name) AS conjunto1
MATCH (rec)-[:IN_GENRE]->(re:Movie)
WITH u, m, rec, genreSelection, actorSelection, directorSelection, conjunto1, COLLECT(re.name) AS conjunto2

RETURN u.name AS User, rec.title AS Recommendation, rec.year AS Year, conjunto1, conjunto2, genreSelection, actorSelection, directorSelection, ((0.5*genreSelection)+(0.25*actorSelection)+(0.25*directorSelection))/ (genreSelection+actorSelection+directorSelection) AS score
ORDER BY score DESC LIMIT 50
```

Se utilizará el indice de Jaccard que es un numero entre 0 y 1 que indica la similitud de dos conjuntos. Este se calcula dividiendo el tamaño de la intersección de dos conjuntos por la unión de los dos conjuntos. Para esto se obtienen las películas de agrado de un usuario y se compara con todas las películas para obtener las que tienen las mismas propiedades; a este resultado se le conoce como intersección de dos conjuntos. Se muestran los valores en comun de cada conjunto por separado. Y luego se obtiene la unión de estos conjuntos que es la suma del conjunto 1 con los elementos del conjunto 2 que no tiene el conjunto 1. Se calcula el indice de Jaccard y se ordena por este.

```
# Algoritmo
MATCH (u:User {name: "User name"})-[r:LIKED]->(m:Movie)
MATCH (m:Movie)-[:IN_GENRE]->(filtered)-[:ACTED_IN|:DIRECTED]->-(rec:Movie) WHERE NOT EXISTS( (u)-[:LIKED]->(rec) )
WITH m, rec, COUNT(filtered) AS intersection

MATCH (m)-[:IN_GENRE]->(mo:Movie)
WITH m, rec, intersection, COUNT(mo.name) AS conjunto1
MATCH (rec)-[:IN_GENRE]->(re:Movie)
WITH m, rec, intersection, conjunto1, COUNT(re.name) AS conjunto2

WITH m, rec, intersection, conjunto1, conjunto2, COUNT(x IN conjunto2 WHERE NOT x IN conjunto1) AS union
RETURN m.title, rec.title, conjunto1, conjunto2, ((1.0*intersection) / SIZE(union)) AS jaccard ORDER BY jaccard DESC LIMIT 50
```

CONCLUSION

Los dos algoritmos que se crearon son funcionales ya que recomiendan películas por medio de un score match que se compara con otra película que los usuarios ya hayan marcado como favoritas. El primer algoritmo prioriza el valor del genero y actores que es en lo que nos dio resultado en la encuesta que se basan mas los usuarios. El segundo algoritmo tomo como conjunto las prioridades de dos películas y saca la unión e intersección de estas películas que luego por el método de Jaccard se obtiene la similitud entre estas.

LINK GITHUB

<https://github.com/FR98/Proyecto2Datos>

BIBLIOGRAFÍA

Martinez, M. C. (2017). SISTEMAS DE RECOMENDACIÓN BASADOS EN TÉCNICAS DE PREDICCIÓN DE ENLACES PARA JUECES EN LÍNEA.
Obtenido de Trabajo Fin De Master en Ingeniería Informática:
<https://eprints.ucm.es/43975/1/Sistemas%20de%20Recomendaci%25C3%25B3n%20para%20Jueces%20en%20l%C3%A1nea.pdf>

Obasados%20en%20técnicas%20de%20predicción%20de%20enlaces%
20para%20jueces%20en%20l%C3%ADnea%20-%20Marta%20Caro%
20Mart%C3%ADnez.pdf

Pepa, S. M. (2016). *Novedad y diversidad en la recomendación de contactos en redes sociales*. Obtenido de Doble Máster en Ingeniería Informática e Investigación e Innovación en Tecnologías de la Información y las Comunicaciones :
https://repositorio.uam.es/bitstream/handle/10486/670409/Sofia_Martina_Pepa.pdf?sequence=1&isAllowed=y

Velez-Langs, O., & Santos, C. (2006). *Sistemas Recomendadores: Un enfoque desde los algoritmos genéticos* . Obtenido de UNMSM:
http://sisbib.unmsm.edu.pe/bibvirtualdata/publicaciones/indata/vol9_n1/a04.pdf

dinngo. (S.F). *Design Thinking en Español*. Obtenido de Design Thinking en Español:
<http://www.designthinking.es/inicio/>

DATAASPIRANT. (2015). *AN INTRODUCTION TO RECOMMENDATION ENGINES*.
Obtenido de DataCconomy: <http://dataconomy.com/an-introduction-to-recommendation-engines/>

Analytics, A. &. (2017). *El algoritmo K-NN y su importancia en el modelado de datos*. Obtenido de AnaliticaWeb:
<https://www.analiticaweb.es/algoritmo-knn-modelado-datos/>