

UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO
FACULTAD DE CIENCIAS
INTELIGENCIA ARTIFICIAL PROYECTO 03:
SISTEMA DE RECOMENDACIONES

Guzmán Mosco, Mario Alexis

Martínez Mendoza, Miguel Ángel Torres Bucio, Miriam

27 de Octubre 2019

1 Introducción

Los sistemas de recomendación aparecieron debido al gran incremento de los datos en la web y en las distintas aplicaciones que la gente usa día con día. Gracias a la gran cantidad de información a la cuál los usuarios tiene acceso, han sido capaces de obtener datos utiles, Sin embargo, el problema con ésto es que como las bases de datos son muy extensas son incapaces de abstraer informacion util y adecuada para ellos de una manera fácil y rápida, a esto se le conoce como desbordamiento de datos, dicho problema no es causado solo por el volumen de los datos, si no tambien por que se posee información no deseada o irrelevante para el usuario.

Es por eso que nace la necesidad de crear un sistema que sea capaz de hacer esa busqueda de manera automática y con solo unos clicks. En este proyecto abordamso 2 tipos de sistemas de recomendación:

Los primeros sistemas de recomendación comenzaron a eliminar la información inútila este tipo de sistemas se les conoce como **filtrado colaborativo**, además de éste tipo de filtrado, los investigadores crearon un sistema personalizado para hacer recomendaciones, el cual se centra en cada usuario, según sus preferencias los sistemas de recomendación proporcionan un servicio o información favorable para el usuario, a éste tipo de sistemas se les conoce como **Filtrado basado en contenido**.

Actualmente, la importancia de la recomendación de información está aumentando en distintos entornos es por eso que muchos sitios web comenzaron a desarrollar y utilizar la tecnología de recomendación para proporcionar servicios personalizados para el usuario un claro ejemplo de esto es Amazon, Netflix, entre otros.

Como ya se mencionó anteriormente, los usuarios prefieren los sistemas de recomendación porque les ayudan a ahorrar tiempo para buscar información y

obtener los mejores documentos, productos o servicios. A pesar del gran éxito de los sistemas de recomendación, los investigadores se han enfrentado a diversos problemas uno de ellos y quizá uno de los más grandes es el hecho de que no siempre se hacen las mejores recomendaciones para sus usuarios, es aquí donde entran los investigadores de **machine learning**, quienes se enfocan en mejorar dichos sistemas para así hacer recomendaciones más efectivas y que en realidad vallan acorde a la información y gustos del cliente.

2 Definición del problema

* Descripción del problema y por qué es necesario un sistema de recomendación

En este proyecto se busca desarrollar un sistema que basandose en los gustos del usuario sea capaz de recomendar distintas aplicaciones de la tienda de Android, Play Store, para ésto se empleará la técnica de **filtrado basado en contenido**, usando como punto de partida los gustos del usuario y se aplicará el método de aprendizaje $k - means$, usando el framework de desarrollo web *Django*, un ambiente virtual en *anaconda* y el lenguaje de programación *Python*

* Descripción del modelado del problema

- (a) Características del perfil del usuario: El sistema no necesita mucha información acerca del usuario, solo son necesarios los nombres de las aplicaciones que le gustan, para a partir de ahí hacer la búsqueda y mas adelante hacer la recomendación

Ejemplo:

El usuario indica que le gustan las aplicaciones: Zombie Tsunami, Bubble shooter 2 y Geometry Dash World

- (b) Características que debe cumplir el producto: A diferencia del usuario, con el producto si necesitamos conocer mucha más información, en este caso usaremos una base de datos con 6 características en específico, esto para poder hacer una recomendación más precisa, necesitamos conocer:

- i. Nombre de la aplicación: Nombre oficial con el que se encuentra la app en la Play Store
- ii. Categoría:¹ Las categorías ayudan a los usuarios a buscar las aplicaciones más relevantes en Play Store agrupándolas según las características que comparten.
- iii. Rating: Calificación asignada por los usuarios
- iv. Reviews: Cantidad de comentarios que los usuarios han hecho sobre la app
- v. Peso: Peso en *mb*
- vi. Descargas: Cantidad de veces que la app ha sido descargada e instalada por los usuarios

- vii. Contenido:² Permite mediante un número clasificar las aplicaciones con respecto a la idoneidad para el público en términos de distintos temas.

Ejemplo:

Nombre	Categoría	Rating	Reviews	Peso	descargas	Contenido
Zombie Tsunami	18	4.4	4920817	73	100000000	9
Bubble shooter 2	18	4.3	23005	28	5000000	10
Geometry Dash Word	18	4.6	760628	63	10000000	10

Table 1: Ejemplo

¹Google Play Store cuenta con un total de 33 categorías con las que separan su contenido, para poder usar esa información en k-means, se le asignó un valor numérico arbitrario a cada categoría

CATEGORÍA	VALOR	CATEGORÍA	VALOR
Art and design	1	Lifestyle	17
Auto and vehicles	2	Game	18
Beauty	3	Family	19
Books	4	Medical	20
Business	5	Social	21
Comics	6	Shopping	22
Communication	7	Photography	23
Dating	8	Sports	24
Education	9	Travel	25
Entertainment	10	Tools	26
Events	11	Personalization	27
Finance	12	Productivity	28
Food and drink	13	Parenting	29
Health and fitness	14	Weather	30
House and home	15	Videoplayers	31
Libraries	16	News	32
		Navigation	33

Table 2: Categorías de Google Play Store

²Al igual que con las categorías, al contenido se le asignó un valor numérico para poder trabajar.

CONTENIDO	VALOR
Everyone	10
Everyone 10+	9
Teen	8
Mature 17+	4
Adults only 18+	3
Unrated	1

Table 3: Contenidos de Google Play Store

3 Descripción de la propuesta e implementación

- * **Descripción del esquema de representación del conocimiento** (justificar el tipo de esquema de representación y su relación con el proyecto) (¿Cuál es el esquema que se utiliza para representar el conocimiento? y ¿Cuáles son las propiedades de ese esquema?, ¿Qué tipo esquema de representación de conocimiento es (declarativo o procedimental, etc.)? ¿Qué estrategias se pueden usar para manipular este tipo de conocimiento?)

- * **Descripción de la forma de operar el esquema de representación de conocimiento de acuerdo con el objetivo del proyecto**

Como se mencionó anteriormente, para la resolución de éste problema se propone utilizar **K-means Clustering**, basandonos en los gustos de los usuarios, es por eso que se usará el **filtrado basado en contenido** para poder encontrar las aplicaciones que más se acerquen a sus gustos. (Explicar que es kmeans, filtrado basado en contenido y porque lo haremos basado en usuarios)

- K-means clustering

Como su nombre lo menciona, k-means es un método de agrupamiento (clustering), el cual se usa para separar datos en k grupos distintos en el cual los elementos que comparten características semejantes se encuentran dentro del mismo cluster, para hacer ésta distinción, k-means calcula la distancia entre los datos, los datos que comparten características entre sí son los que tendrán menor distancia entre ellos.

Funcionamiento: Lo primero que k-means necesita saber es el número de grupos en los que se va a separar todo el conjunto de datos, una vez que se tiene la información de cuantos k clusters se quiere, se procede a colocar k puntos aleatorios, a éstos puntos se les conoce como **centroides**

- Filtrado basado en contenido

- * **Entorno de trabajo REAS**

- Agente: Sistema de recomendaciones
- Rendimiento: óptimo en tiempo y espacio, proporcionar al usuario la recomendación adecuada.
- Entorno: usuario, aplicación web

- Actuadores: K-means y todas sus funciones dependientes
- Sensores: Interfáz gráfica, teclado y ratón

* **Propiedades del entorno**

- Entorno: Aplicación web
- Observable: Totalmente, la app tiene acceso a todos los componentes de la interfáz gráfica, así como a la base de datos.
- Determinista / Estocástico: Estocástico, pues la decisión futura (hacer una recomendación) depende de las aplicaciones que el usuario seleccionó anteriormente.
- Episódico / Secuencial: Secuencial, pues dependiendo de las aplicaciones que seleccionó como sus favoritas, afectarán el cálculo de su recomendación.
- Discreto / Continuo: Discreto, pues se debe esperar a que el usuario seleccione sus gustos para poder hacer la recomendación.
- Estático / Dinámico: Estático, el entorno de trabajo no cambia, siempre se usa el mismo dataset
- Agente: Multiagente, hay una interacción entre el usuario y el software

* **Complejidad teórica de la solución** (Demostrar formalmente)

* **Instrucciones para operar el sistema**

* **Comportamiento del agente**

Ejemplo con respecto a los objetivos

4 Conclusiones

* **Complejidad real de la solución**

* **Qué tan adecuada es la recomendación**

* **Ventajas y desventajas**

5 Bibliografía

- * [https://es.wikipedia.org/wiki/Clasificaci%C3%B3n_por_edades_\(cine\)](https://es.wikipedia.org/wiki/Clasificaci%C3%B3n_por_edades_(cine))
- * <https://42matters.com/docs/app-market-data/android/apps/google-play-categories>
- * <https://estrategiastrading.com/k-means/>
- * MOVIE RECOMMENDATION WITH K-MEANS CLUSTERING AND SELF-ORGANIZING MAP METHODS, Eugene Seo and Ho-Jin Choi, Department of Computer Science, Korea Advanced Institute of Science and Techonology (KAIST), 119 Munjro, Yuseong, Daejeon, 305-732, Korea