



RECOMENDAR

EL ARTE DE HACER ALGORITMOS DE
RECOMENDACIÓN

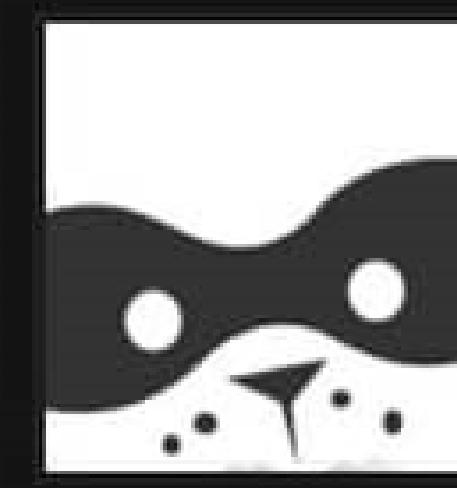
¿QUIENES SOMOS?



BAKKEN, LOUISE



GUGLIELMI, FELIX



RASIA, MARTÍN



RODRIGUEZ , CARLOS



ALVAREZ , JUAN



SISTEMAS DE RECOMENDACIÓN

- ¿Que son los sistemas de recomendación?
- Tipos de sistemas de Recomendación
 1. Sistemas de Popularidad
 2. Sistemas de Contenido
 3. Sistemas Colaborativos
 - 3.1. Filtrado Colaborativo
- Algoritmos de Sistemas de Recomendación
- Beneficios de los Sistemas de Recomendación

CASO DE ANÁLISIS

Seleccionamos un *dataset* proporcionado para un evento de selección de personal de Flow.

Para realizar el desarrollo del algoritmo trabajamos con dos *datasets*:

Primer *dataset*: cuenta con información de los ID de cuenta, usuario y contenido visualizado por cada usuario, también incluye información adicional referida al tipo de dispositivo desde donde se vio el contenido, hora de inicio y fin de la reproducción.

Segundo *dataset*: cuenta con información del contenido disponible en la plataforma de flow, contiene el detalle de ID del contenido y características/descripción del mismo.





OBJETIVO

Nuestro principal objetivo es lograr el desarrollo de un algoritmo de recomendación de películas que recomiende al usuario de la plataforma de *straming* que sean lo mas acertadas para cada cliente.

Para medir la precisión de nuestro algoritmo decidimos usar el *Mean Average Precision* para cada uno de los algoritmos implementados, para considerar exitosa la implementación del algoritmo debe superar el 0,014.



ETAPAS DEL PROYECTO

DATA PROCESSING

RECOMENDER
MODEL

EVALUACIÓN Y
POST PROCESSING

DESPLIEGUE

DATA PROCESSING

Limpieza del dataset:

- Limpieza de registros duplicados y NaN.
- Exclusión de los registros que no pudimos asociar con algún contenido.





DATA PROCESSING

Se crearon tres instancias de data wrangling, segun fuera la implementacion del algoritmo a realizar.

INSTANCIA FEATURE IMPORTANCE

- Creación de la variable target
- Creación de una columna con los minutos que incurrió el usuario en cada contenido
- Creación de de matriz de visualizaciones y características.

INSTANCIA DE GRID SEARCH

- Construcción de clases para componer rating por contenido visto.
- Construcción de clases que construyan matriz de visualizaciones

INSTANCIA TEXT MINING

- Composición de variable target "texto" por contenido.
- Eliminación de signos de puntuación, transformación de palabras a raíces y eliminación de *stopwords*.
- Armado de matriz TF-IDF y de *features*.

RECOMENDER MODEL

En la implementación, se utilizaron dos (2) algoritmos de recomendación:

- Light FM: es una implementación en Python de una serie de algoritmos de recomendación populares para la retroalimentación implícita y explícita, incluyendo la implementación eficiente de las pérdidas de clasificación BPR y WARP. Esta fue nuestra primera selección puesto que es un algoritmo de rápida implementación (a través de la estimación de modelos multihilo), y produce resultados de alta calidad.
- SURPRISE (*Simple Python Recommendation System Engine*): es una implementación de *scikit learn* basado en datos de calificación explícitos, no admite valoraciones implícitas ni información basada en el contenido.





EVALUACIÓN Y POST PROCESSING

Para verificar la precisión del algoritmo utilizamos un método de evaluación llamado *Mean Average Precision* (MAP) que es una métrica utilizada para evaluar los modelos de recomendación, los valores son calculados de 0 a 1.

La formula MAP esta basado en las siguientes métricas:

- Precisión.
- Ponderación por orden de recomendación

RESULTADOS

1

TEXT MINNING + Light FM

MAP = 0,283

2

Light FM

MAP = 0,281

3

SURPRISE

MAP = 0,001

ALGORITMO GANADOR

TEXT MINNING + LIGHT FM



DESPLIEGUE

Link de la notebook:

<https://colab.research.google.com/drive/109Aa1sEQWNATsKhOuYe6PTHeT98-2TPz?usp=sharing>





iMUCHAS GRACIAS!