PROYECTO INTEGRADOR FINAL

Reporte Final

Data Science – Digital House

2022

GRUPO 3

INTEGRANTES:

Guglielmi, Felix

Rasia, Martín

Rodriguez Elorza, Carlos

Bakken, Louise

Alvarez Hurtado, Juan José

Introducción

El tema elegido es el análisis de las preferencias de los usuarios que visualizan contenido en la plataforma de streaming Flow con el objetivo de brindar sugerencias de otros títulos, tanto a ellos como a otros usuarios.

Es de conocimiento general la proliferación en los últimos años de las plataformas de contenido digital audiovisual (Ej: Plataforma de Netflix en 2007, Amazon Prime 2006). Inicialmente, el hábito de consumo estaba caracterizado por usuarios que accedían a las plataformas para visualizar un contenido específico. Dichas plataformas en el afán de retener y aumentar la cantidad de usuarios y la fidelidad de los mismos, incursionaron en fomentar una potencial demanda adicional de otros títulos a través de herramientas de recomendaciones específicas.

Una implementación conceptual de dichos sistemas sería a través de la identificación y clasificación de grupos de usuarios en función del contenido visualizado, que posteriormente al ser correlacionado con una base de datos del contenido disponible, permitan reconocer títulos apropiados para los distintos grupos.

Para este trabajo, se utilizaron datos de la plataforma de streaming de Flow los cuales fueron liberados al público para una competencia de Dathaton de Telecom durante la celebración del Campus Party Latam 2021. Flow ofrecía las siguientes descargas para poder ensayar un prototipo de herramienta de recomendaciones:

- datasets con información histórica de visualizaciones por cuenta de cliente y usuario específico (donde se identifica el título o denominación del contenido que se visualizó);
- dataset con la oferta de títulos disponible y las características de los mismos (por ej: año de emisión, país de origen) y su categorización (drama, comedia, documental, etc.).

Índice

1. Dataset Utilizados

La información de base para el desarrollo del trabajo está constituida por los dos datasets:

1.1. Dataset 1:

Dataset con los registros de más de 3.5 millones de visualizaciones de contenidos de Flow, de una muestra aleatoria de más de 113 mil perfiles de la aplicación. A continuación, se detallan las variables del dataset:

- customer_id: código de identificación de cada cliente de Flow (puede tener asociados uno o más account_id)
- account_id: código de identificación de cada perfil de Flow (se corresponde con un único customer_id)
- device_type: indica el tipo de dispositivo desde el que se efectuó la visualización. Las categorías posibles son:
    - CLOUD: cliente web
    - PHONE: teléfono celular
    - STATIONARY: smart TV
    - STB: decodificador Flow
    - TABLET: tablet
- asset_id: código de identificación de cada título disponible en la plataforma
- tunein: fecha y hora de inicio de la visualización
- tuneout: fecha y hora de finalización de la visualización
- resume: variable dummy que indica si se reanuda un consumo anterior del mismo asset_id

	customer_id	account_id	device_type	asset_id	tunein	tuneout	resume
2750315	82636	102343	STB	19668.0	2021-02-22 13:45:00.0	2021-02-22 13:47:00.0	0
2543824	77824	96738	STB	4360.0	2021-02-07 00:37:00.0	2021-02-07 01:45:00.0	1
3621980	111044	25047	STB	11013.0	2021-03-09 17:23:00.0	2021-03-09 17:53:00.0	1
3275991	98527	15485	STB	10478.0	2021-02-19 14:14:00.0	2021-02-19 14:18:00.0	1
621778	21956	43750	STB	14337.0	2021-03-03 18:39:00.0	2021-03-03 19:35:00.0	0
2355274	72909	91399	STB	12111.0	2021-01-14 10:15:00.0	2021-01-14 10:17:00.0	0

Sólo presentan 29 valores nulos device_type y 22 valores nulos asset_id. Además se identificaron 76 registros de visualizaciones repetidos.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
customer_id	3657801.0				56975.0	30907.0	0.0	31687.0	57595.0	82502.0	112339.0
account_id	3657801.0				59027.0	32375.0	0.0	31422.0	60782.0	87524.0	113880.0
device_type	3657772.0	5	STB	2993554							
asset_id	3657779.0				16672.0	9666.0	0.0	8318.0	16419.0	25128.0	33143.0
tunein	3657801.0	128267	2021-03-21 20:58:00.0	284							
tuneout	3657801.0	128552	2021-03-21 20:58:00.0	131							
resume	3657801.0				0.0	0.0	0.0	0.0	0.0	1.0	1.0

1.2. Dataset 2:

Dataset con los datos de los más de 33 mil títulos disponibles en la plataforma de streaming. Las variables incluidas son:

- asset_id: código de identificación de cada título disponible en Flow
- content_id: código de identificación que agrupa los distintos asset_id asociados a un mismo contenido (por ejemplo, cada episodio de una misma serie tiene su propio asset_id, mientras que la serie se identifica con un content_id único)
- title: nombre del título
- reduced_title: nombre del título reducido
- episode_title: título del episodio (válido para contenidos asociados a series)
- show_type: tipo de show, las categorías son autorreferenciales del tipo "serie", "película", "TV", "tutorial" con excepción de "Rolling" que indica que se trata de una serie incompleta y "Web" la cual remite a contenidos pensados íntegramente en formato digital
- released_year: año de lanzamiento
- country_of_origin: país de origen expresado con el código de dos letras ISO 3166
- category: categoría o género al que pertenece el contenido, puede haber una o más, por ej: "Documental/Animales" o "Infantil/Comedia" (221 códigos)
- keywords: palabras clave o *tags* asociadas al contenido, puede haber una o más, por ej: "Actualidad,Periodístico,Política" (4450 códigos)
- description: descripción (sinopsis)
- reduced_desc: descripción (sinopsis) reducida
- cast_first_name: nombre y apellido de los actores y actrices principales (4301 códigos)
- credits_first_name: nombre y apellido del director o directora (3259 códigos)
- run_time_min: duración total, expresada en minutos
- audience: audiencia *target, por ej: "General", "Teens", "Juvenil" (10 códigos)*
- made_for_tv: variable *dummy* que indica si el contenido fue hecho para TV
- close_caption: variable *dummy* que indica si el contenido posee subtítulos
- sex_rating: variable dummy *que* indica si el contenido tiene escenas de sexo explícitas
- violence_rating: variable *dummy* que indica si el contenido tiene escenas de violencia explícitas
- language_rating: variable *dummy* que indica si el contenido posee lenguaje que puede ser considerado ofensivo o inapropiado
- dialog_rating: variable *dummy* que indica si el contenido posee diálogos que pueden ser considerado ofensivos o inapropiados
- fv_rating: variable *dummy* que indica si el contenido tiene rating de FV, que corresponde a público infantil con violencia ficticia

- pay_per_view: variable *dummy* que indica si se trata de un alquiler
- pack_premium_1: variable *dummy* que indica si se trata de un contenido exclusivo del pack premium 1
- pack_premium_2: variable *dummy* que indica si se trata de un contenido exclusivo del pack premium 2
- create_date: fecha de creación del título
- modify_date: fecha de modificación del título
- start_vod_date: fecha desde la cual el título se encuentra disponible en la plataforma
- end_vod_date: fecha de finalización de la disponibilidad del título en la plataforma

	asset_id	content_id	title	reduced_title	episode_title	show_type	released_year	country_of_origin	category
4257	16334	86.0	11/25 - Bienvenidos a bordo	Bienvenidos_25-11	25 Noviembre 2020	TV	2020.0	AR	Entretenimiento/Competencia
6244	8456	929.0	T:2 Ep:16 Crudo y sin censura	Crudo_y_sin__T2_E16	Episodio 16	TV	2008.0	US	Documental/Animales
32203	17037	48.0	T:2 Ep:28 Bia	Bia_T2_E28	Episodio 28	TV	2020.0	AR	Infantil/Comedia
15790	26245	2034.0	T:2 Ep:01 Run Coyote Run	T:2 Ep:01 Run Coyote Run	La mafia china	Serie	2018.0	US	Comedia/Aventura
28024	32182	139.0	01/02 - TN Internacional	TN_Internacio_02-01	2 Enero 2021	TV	2021.0	AR	Interés General/Periodístico
5155	1529	385.0	Ep:021 Separadas	Separadas_UY_E021	El camino	Serie	2020.0	AR	Romance/Drama

Los campos que presentan valores incompletos de relevancia son:

- 12554 valores nulos en credits_first_name.
- 8732 valores nulos en cast_first_name
- 4147 valores nulos episode_title

No se encontraron registros de títulos repetidos.

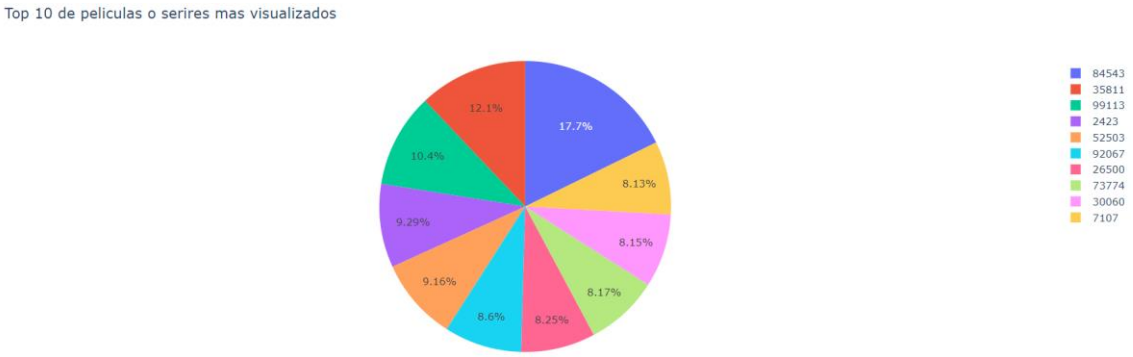
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
asset_id	33144.0				16572.0	9568.0	0.0	8286.0	16572.0	24857.0	33143.0
content_id	33123.0				1372.0	1153.0	0.0	387.0	1019.0	2160.0	4371.0
title	33144.0	28587	Star Trek: En la oscuridad	8							
reduced_title	33144.0	28280	Un_misterio_para_Au	18							
episode_title	28997.0	15773	Episodio 1	328							
show_type	33140.0	8	TV	15681							
released_year	33144.0				2016.0	6.0	1942.0	2015.0	2018.0	2019.0	2021.0
country_of_origin	33140.0	56	US	13583							
category	33144.0	221	Infantil/Dibujos Animados	5491							
keywords	33142.0	4450	Actualidad,Periodístico,Política	735							
description	33142.0	20738	Eylül fue abusada por su padrastro, pero su ma...	430							
reduced_desc	33144.0	17908	Eylül fue abusada por su padrastro, pero su ma...	422							
cast_first_name	24412.0	4301	İpek Karapınar, Özgür Çevik, Biran Damla Yılma...	356							
credits_first_name	20590.0	3259	Serkan Birinci	430							
run_time_min	33144.0				48.0	35.0	0.0	22.0	43.0	67.0	308.0
audience	33143.0	10	General	13658							
made_for_tv	33144.0	2	N	32707							

close_caption	33144.0	1	N	33144
sex_rating	33144.0	1	N	33144
violence_rating	33144.0	1	N	33144
language_rating	33144.0	1	N	33144
dialog_rating	33144.0	1	N	33144
fv_rating	33144.0	1	N	33144
pay_per_view	33144.0	2	N	32291
pack_premium_1	33144.0	2	N	30867
pack_premium_2	33144.0	2	N	31217
create_date	33144.0	33112	2021-03-12T21:10:49.0Z	4
modify_date	33144.0	30735	2019-11-05T14:42:57.0Z	4
start_vod_date	33144.0	1525	2020-12-15T00:00:00.0Z	2673
end_vod_date	33144.0	1377	2020-12-14T23:59:59.0Z	2339

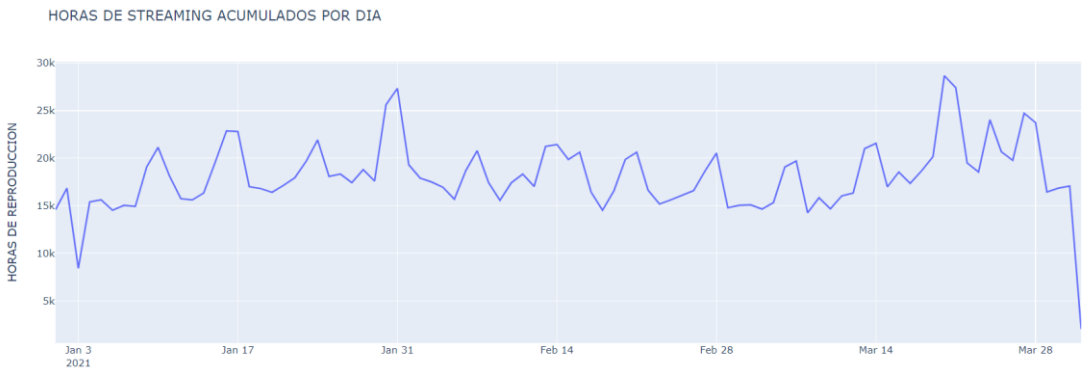
1.3. Análisis Información Contenida

Analizando la información de los Dataset, se concluye la siguiente información:

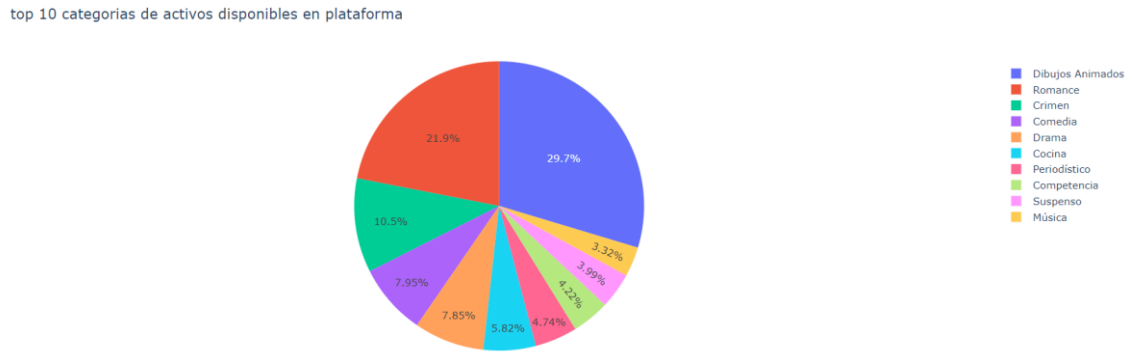
A continuación se muestra la distribución en horas del tiempo acumulado de los 10 títulos más reproducidos. El título principal se lleva 925 hrs lo que representa un 17.7% de lo acumulado. Los 10 principales títulos representan 5226 hrs de reproducción. El tiempo total de reproducción de títulos de todo el set de visualizaciones es 1644470 hrs (esto implica que las 10 principales sólo representan un 3.2 % del tiempo de visualización).



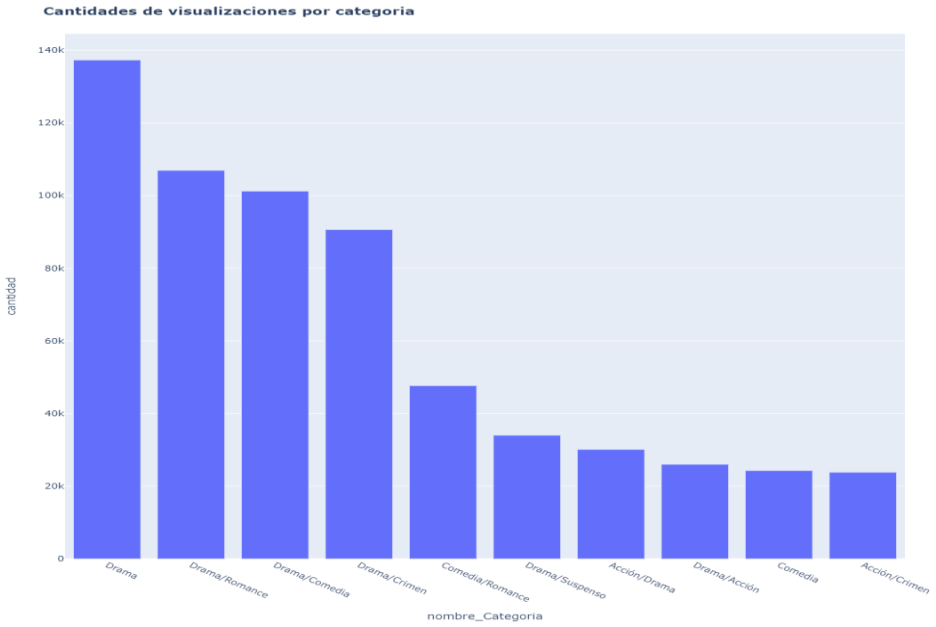
Con respecto a la utilización de los títulos, a continuación puede observarse las horas de reproducción por día, siendo el valor medio 18071 horas de reproducción diarias.



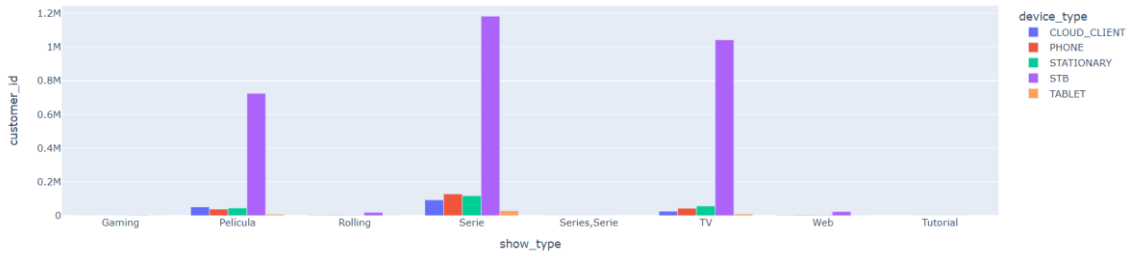
A continuación se muestran las 10 categorías principales en cantidades de títulos. La principal es ¨*Dibujos Animados* ¨con 5491 títulos, lo que representa un 29.7% del total de las 10 principales. El total de títulos es 33000, y las 10 principales categorías representan 18500.



Finalizando el análisis de categoría, se muestra un gráfico de barras con las cantidades de visualizaciones por categoría, donde se observa una clara preponderancia al Drama.



Con respecto al tipo de dispositivo más utilizado por categoría, a continuación se muestran las agrupaciones de usuarios según las distintas categorías y el tipo de dispositivo utilizado. Se destacan los Smart TV como principal dispositivo o plataforma utilizado.



2. Data Processing

2.1. Limpieza

Para realizar la limpieza de los datasets se realizaron los siguientes procesos:

- Limpieza de registros duplicados y NaN (*Not a Number*)
- Exclusión de los registros que no podían asociar con algún contenido

Para el dataset de visualizaciones, posteriormente a remover los registros con valores nulos o NaN, se prosiguió a generar una columna contabilizando la duración en minutos de dicha visualización. Lo más relevante de la limpieza en el dataset de visualizaciones fue el filtrado de los registros a las cuales no se le podía asociar un contenido.

Para el dataset de títulos no se encontraron valores duplicados necesarios de remover. Posteriormente, sólo se mantuvieron las columnas que no son de tipo exclusivo texto (como la descripción o el nombre del actor principal) ya que las features de interés para el modelo de recomendación deben ser numéricas o dummies y la transformación a dummies de variable de texto solo podía hacerse con con dimensionalidad reducida. Luego se realizó una eliminación de los títulos con valores nulos o NaN en las columnas preseleccionadas, en especial en la columna content_id, dado que no podrían ser posteriormente asociados a visualizaciones.

2.2. Data Wrangling

Se desarrollaron tres instancias de data wrangling para la aplicación del algoritmo a realizar:

Primera 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.

Segunda Instancia “Grid Search”:

- Construcción de clases para componer rating por contenido visto.

- Construcción de clases que construyan matriz de visualizaciones

Tercera 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.

3. Recommender Model

Para llevar a cabo la implementación de nuestro modelo de recomendación, elegimos dos algoritmos llamados Light FM y SURPRISE, dichos algoritmos son muy populares y no conllevan una complejidad excesiva a la hora de su implementación. A continuación se describe el proceso que se llevó a cabo para la puesta en marcha de dichos algoritmos.

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 (*Bayesian Personalised Ranking*) y WARP (*Weighted Approximate-Rank Pairwise*). 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.

Primera implementación sin text mining:

Para la implementación del algoritmo se llevó a cabo una preparación del set de entrenamientos, que consistió en *fitear* las columnas id_unico, content_id y puntuación, que son aquellas que consideramos más importantes a la hora de desarrollar la recomendación.

Se construyó una matriz de interacciones del modelo con las variables elegidas.

Se utilizó descenso gradiente estocástico asíncrono para minimizar el costo computacional del algoritmo.

Por último se guardó la primera implementación en un archivo PKL.

Segunda implementación con text mining:

A efectos de mejorar los resultados alcanzados en la primera implementación del modelo Light FM, decidimos usar técnicas de text mining, para mejorar las métricas predictoras del primer algoritmo.

La aplicación de esta herramienta se realizó de la siguiente manera:

- Se creó una columna llamada “Texto” que contiene la información de las siguientes columnas:
    - released_year
    - country_of_origin
    - keywords
    - audience
    - category
    - show_type
- Se definió una función que va a tokenizar, se aplicará stem y se sacará las stopword de la columna “Texto”.

Se aplicaron los pasos descritos anteriormente para el adecuamiento del dataset.

SURPRISE (*Simple Python Recommendation System Engine*): es una implementación de scikit learn basado en datos de calificación explícitos, no admite ni información basada en el contenido. Esta es nuestra segunda selección debido a su facilidad de uso y confiabilidad de resultados.

Para la implementación del algoritmo utilizamos criterios lógicos similares a la implementación del Light FM.

Se creó una matriz, donde incluimos el elemento a recomendar, ya que este debe estar dentro el set de datos para evitar la pérdida de información significativa, elegimos recomendar id de títulos y no de contenidos para poder comparar los resultados con los que nos dio los modelos con Light FM.

Se renombraron las columnas que íbamos a utilizar en el modelo de la siguiente forma:'id_unico':'user_id','idunicoDeContenido':'item_id', 'puntuacion':'rating'.

Se cargó de SVD y GridSearchCV, ambos pertenecientes a la librería de Surprise, para la generación de grillas. Generamos una grilla de parámetros donde se prueban distintas combinaciones, para mejorar las métricas del modelo, se utilizó lo siguiente:

> *epochs: es la cantidad de pasadas sobre el dataset que hará el algoritmo empleando descenso por el gradiente biased: usar parámetros de sesgo o no;*
>
> *lr_all: learning rate para todos los parámetro;*
>
> *reg_all: término de regularización para todos los parámetros (lambda)*

Evaluamos FCP (factor de pares concordantes) y rmse (error cuadrático medio) para quedarnos con un criterio de composición de rating, imitando un pipeline, corriendo para cada ejecución de GridSearchCV clases diferentes que componían valoraciones con tiempo o cantidad de visualizaciones, dando como mejor resultado para el modelo final, los rating compuesto por cantidad de visualizaciones (fcp: 0,949,rmse:0,568).

5. Evaluación y Post Processing

Mean Average Precision (MAP)[1]

Las predicciones fueron evaluadas utilizando la métrica de mean average precision (MAP). MAP es una medida estándar para comparar la lista completa de los veinte contenidos recomendados con los contenidos efectivamente consumidos. Dado un conjunto de Q recomendaciones, MAP constituye la media de la precisión promedio (average precision) de cada recomendación:

$$MAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q}$$

donde Q es la cantidad de recomendaciones y AP(q) es la precisión promedio de la recomendación q.

La precisión promedio (AP) es el promedio de los valores de precisión en todos los rangos donde se encuentran los contenidos efectivamente consumidos, considerados como ítems relevantes:

$$AP = \frac{\sum_{k=1}^{n} P(k) \times rel(k)}{cantidad\ de\ items\ relevantes}$$

donde n es la cantidad de elementos recomendados, P(k) es la precisión alcanzada hasta la posición k de la lista y rel(k) es una función indicadora que vale 1 cuando el ítem de la posición k resulta relevante (es decir, es un contenido efectivamente consumido), y 0 en caso contrario.

Si todos los contenidos vistos en el set de test se encuentran ordenados consecutivamente entre las primeras recomendaciones para un usuario en particular, la AP tendrá un valor de 1. En el extremo opuesto, si no hay ningún contenido relevante entre las recomendaciones, la AP valdrá 0.

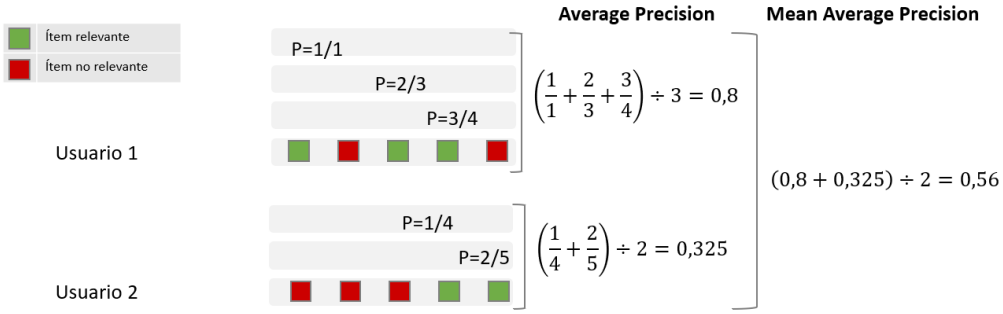
Para obtener la MAP, los valores de AP se promedian aritméticamente sobre el conjunto total de recomendaciones.

Esta métrica se basa en las siguientes premisas:

[1] Extraído del archivo proporcionado por Flow para la Competencia, link de referencia:

- Relevancia del contenido: se considera relevante un contenido que se haya visto al menos una vez en el set de test (excluyendo los contenidos ya vistos en train y las novedades del catálogo del mes al que corresponden los datos de testeo)
- Posición en la lista: el orden en que se presentan las recomendaciones influye en el score obtenido (es deseable que los ítems relevantes se encuentren al inicio y no al final de la lista)

A modo de ejemplo:



Formula: https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52
Ejemplo: https://medium.com/swlh/rank-aware-recsys-evaluation-metrics-5191bba16832

Resultados y Conclusiones

Los resultado alcanzados se muestran a continuación:

Light FM:

Primera implementación:

Se evaluó el algoritmo vía dos métodos *precision* (total de aciertos) y MAP(Mean Average Precision). La *precision* sobre train nos dio 0.14637455 y el MAP nos dio 0.281, el valor baseline implementado por la competición era de 0.014.

Segunda implementación:

Se evaluó el algoritmo vía dos métodos *precision* y MAP, el *precision* sobre train nos dio 0.14671066 y el MAP nos dio 0.283, notamos una leve mejoría en los valores, por lo tanto, utilizaremos este modelo como el definitivo.

Surprise:

Se evaluó la implementación del algoritmo utilizando el MAP en el conjunto de testeo, el valor obtenido fue de 0.001, muy inferior al obtenido por el Light FM, por esta razón, rechazamos la implementación de este algoritmo.

6. Despliegue

Para el despliegue de nuestro algoritmo, desarrollamos una API en heroku, la cual establece un endpoint en su servidor, que nos sirve para consultar los resultados a los que llega el algoritmo que desarrollamos, al ser consultado por la recomendación a unos de los usuarios.

El link donde se puede hace la consulta es el siguiente:
https://github.com/C3r3l4k/TP4_Sistemas_De_Recomendacion_DS_DH