

Συναυλίες







Presentación

Álvaro Lozano Murciego

Profesor Contratado Doctor Departamento de Informática y Automática Área: Lenguajes y Sistemas Informáticos Universidad de Salamanca España

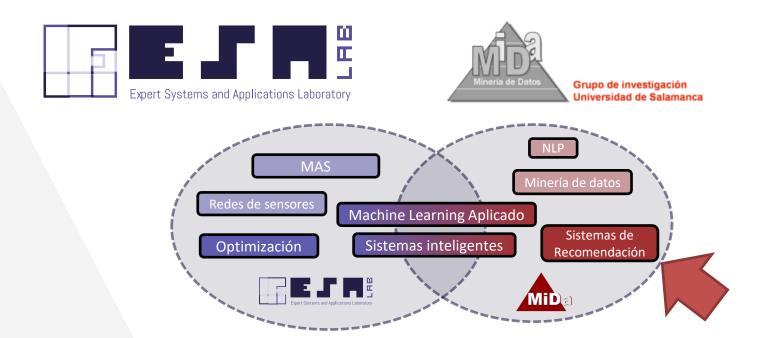


Líneas de investigación y áreas de interés:

- ✓ Machine Learning
- ✓ Recommender Systems
- ✓ Edge Computing
- ✓ Android Development
- ✓ IoT, sensors and MCU



Research interest



Contenido

- 1. Introducción
- 2. Tipos de sistemas de recomendación clásicos
- Sistemas de recomendación sensibles al contexto (CARS)
- 4. Características del dominio de la música
- 5. Trabajos recientes
- 6. ¿Por dónde empiezo? ¡Recursos!

Sistemas de recomendación

Una breve introducción

¿Qué es un sistema de recomendación (RecSys)?

Definición

Los sistemas de recomendación son herramientas que ayudan a un usuario (o grupo de usuarios) que interactúa con un sistema, a elegir ítems de un basto catálogo o espacio de información. (MacNee et. al 2006)

Robin Burke, añade: Busca proporcionar ítems que sean relevantes al usuario de forma personalizada*.



> 180 M Amazon.es



¿Por qué son útiles son los RecSys?

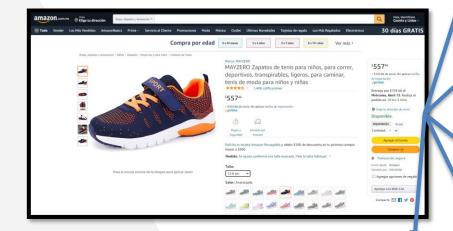
- Por qué muchas *Big Tech* Comanies los utilizan?
- Con la explosión de Internet y de web en las últimas décadas, existen muchos casos en los que los usuarios tienen que tomar una decisión ante un catálogo enorme: YouTube, Spotify, Amazon, etc.
- Mejoran la retención del usuario.
- Es una forma sencilla de analizar el mercado.
- Incremento de las ventas:

35% of Amazon.com, 75% Netflix

Mejoran la **experiencia del usuario** (UX).

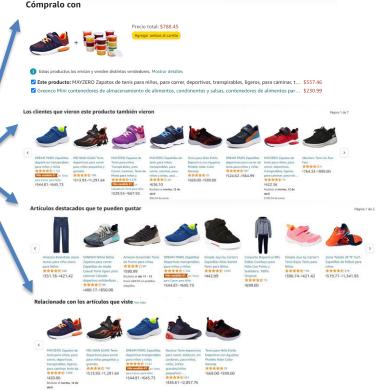


■ Amazon

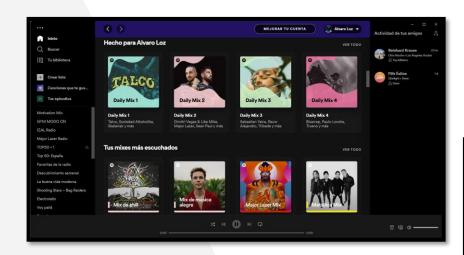


Recomendado en función de tus tendencias de compras

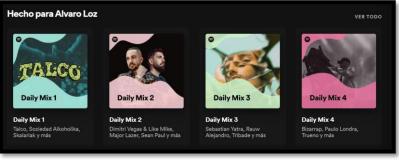




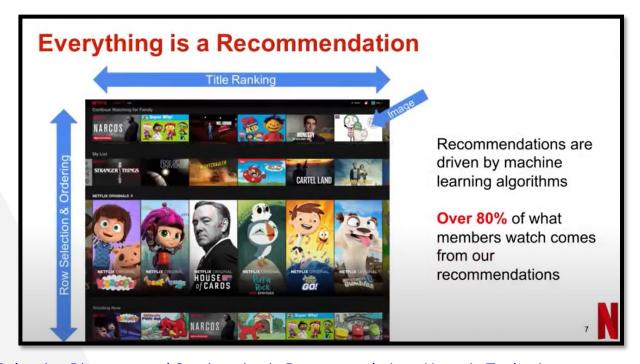
□ Spotify





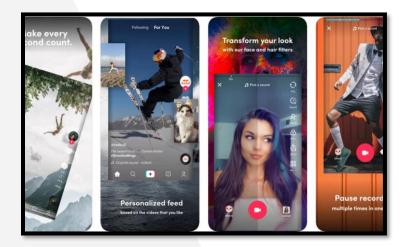


■ Netflix



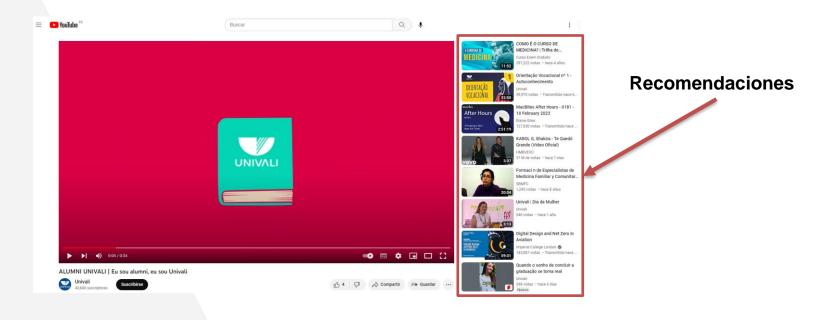
Fuente: Balancing Discovery and Continuation in Recommendations Hossein Taghavi

□ TikTok e Instagram: *Feed* personalizado





■ Youtube



■ Twitter





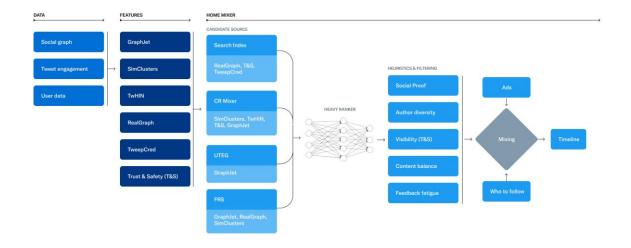
Open source

Twitter's Recommendation Algorithm

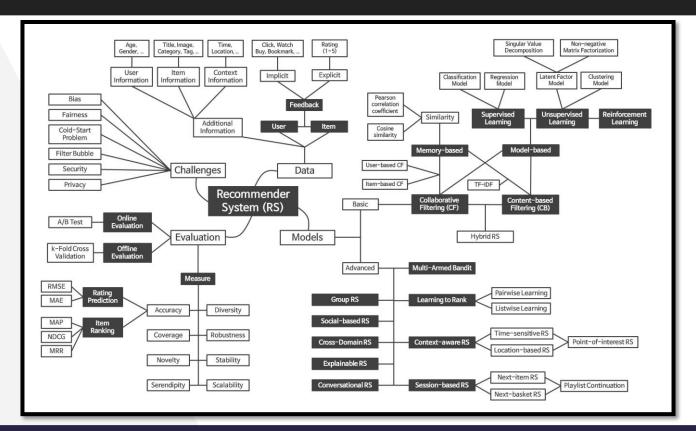
By <u>Twitter</u> Friday, 31 March 2023

□ Twitter:

- Open source code: https://github.com/twitter/the-algorithm
- Explicación: https://github.com/igorbrigadir/awesome-twitter-algo



Mapa del área de los sistemas de recomendación

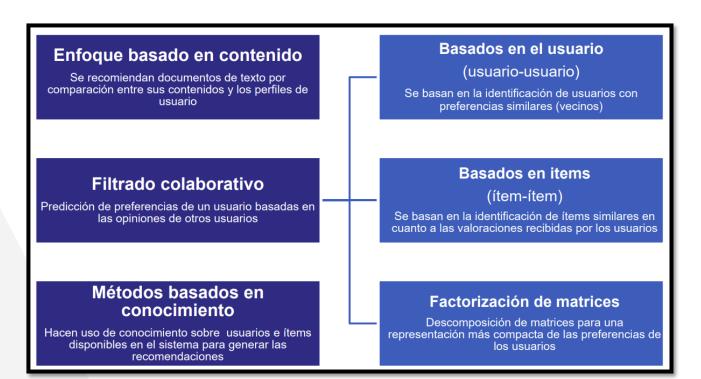


Fuente:

awesome-RecSys

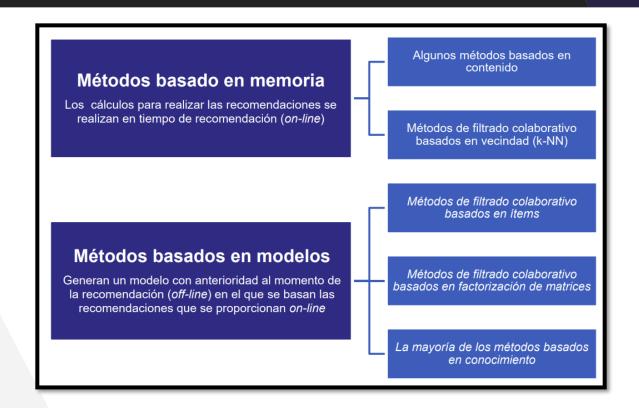
Tipos de sistemas de recomendación (clasificaciones)

CONSIDERANDO LOS DATOS EMPLEADOS



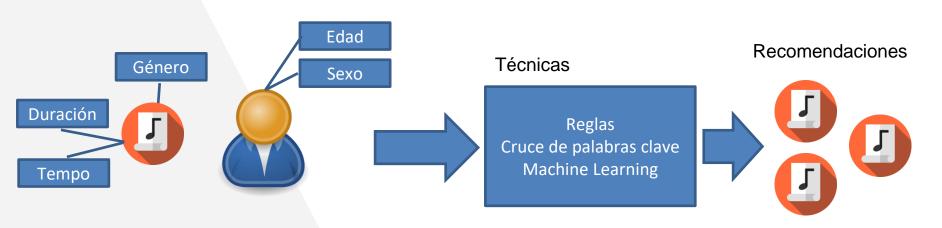
Tipos de sistemas de recomendación (clasificaciones)

CONSIDERANDO EL MODELO



Basados en el contenido

- El usuario recibe recomendaciones de ítems similares a otros por los que manifestó interés en el pasado. Se construyen **perfiles de usuario** empleando:
- Propiedades de los ítems
- Información de los usuarios



Basados en el contenido

Componentes principales:

- Analizador de contenido
- Aprendizaje del perfil del usuario
- 3. Filtrado del contenido

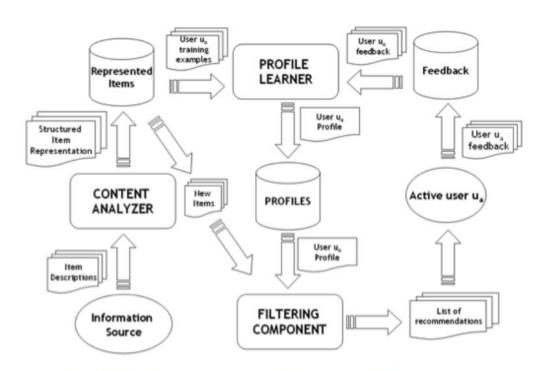


Fig. 3.1: High level architecture of a Content-based Recommender

Basados en el contenido

Pros:

- No es necesaria información de otros usuarios para hacer la recomendación
- Escala fácilmente para un gran número de usuarios
- El modelo captura los intereses específicos del usuario y puede recomendar información de ítems en un nicho muy específico
- Es sencillo hacer un sistema transparente, donde se pueden explicar las recomendaciones.

Contras:

- Es necesario conocer las propiedades de los ítems y la calidad del modelo dependerá de éstas
- Es necesario conocer el interés del usuario, es decir, el modelo tiene poca capacidad para expandir los intereses del usuario.
- Recomendación sobre especializada, tendencia al "filter bubble"



- Las recomendaciones para un usuario se basan **en la opinión de otros usuarios con preferencias similares**
- El usuario tiene que expresar sus valoraciones (**ratings**) sobre diferentes productos a recomendar
- A partir de las valoraciones se calcula la similitud entre usuarios



		Machine Learning Facalitytes		3		
P _s	4	3		Electrical States	5	
8	5		4		4	
8	4		5	3	4	
		3				5
B		4				4
			2	4		5

■ Valoraciones del usuario:

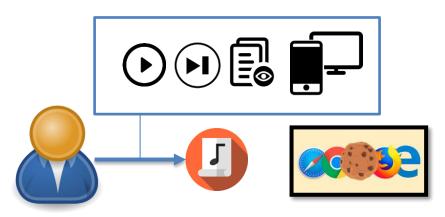
Explícita (active feedback)

Ratings, preferencia mostrada directamente por el usuario.



Implícita (passive feedback)

señal indirecta Log de interacción del usuario:visualización,click, compra, etc.

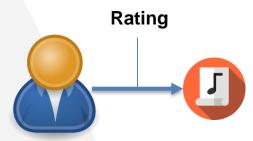


Las recomendaciones para un usuario se basan en la opinión de otros usuarios con preferencias similares

El usuario tiene que expresar sus valoraciones (ratings) sobre diferentes

productos a recomendar

A partir de las valoraciones se calcula la similitud entre usuarios



Matriz Usuario-Ítem Matriz de Ratings

Tipos de técnicas de Filtrado Colaborativo:

Basados en el usuario

(usuario-usuario)

Se basan en la identificación de usuarios con preferencias similares (vecinos)

Basados en items

(item-item)

Se basan en la identificación de ítems similares en cuanto a las valoraciones recibidas por los usuarios

Factorización de matrices

Descomposición de matrices para una representación más compacta de las preferencias de los usuarios

Basados en usuarios (UBCF)

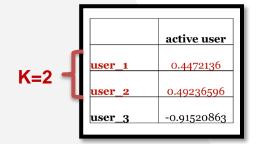
- Se basan en las preferencias del grupo de usuarios más semejante al usuario activo.
- Utilizan todos los registros presentes en la matriz de valoraciones para buscar K usuarios con preferencias semejantes (KNN)
- Es necesaria una función de similitud como por ejemplo:
 - Coeficiente de correlación de Pearson
 - Similitud del coseno
- Se realiza una predicción del rating de un ítem j para el usuario activo.

$$\omega \left(u_{a}, u_{i} \right) = \frac{\sum_{j} (r_{aj} - \overline{r_{a}}) (r_{ij} - \overline{r_{i}})}{\sqrt{\sum_{j} (r_{aj} - \overline{r_{a}})^{2} (r_{ij} - \overline{r_{i}})^{2}}}$$

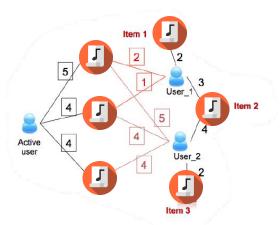
$$pr_{aj} = \bar{r}_a + \frac{\sum_{i=1}^k \omega(u_a, u_i) (r_{ij} - \bar{r}_i)}{\sum_{i=1}^k |\omega(u_a, u_i)|}$$

Basados en usuarios (UBCF) Ejemplo:

- Encontrar los K vecinos más cercanos (KNN) al usuario, empleando una medida de similitud (Por ejemplo Corr. Pearson). Se trata por tanto de un método basado en memoria.
- **2. Predecir el rating** que un usuario dará a un ítem.
- 3. Proporcionar al usuario una lista top-N de recomendaciones.



	User_1	User_2	predicted interest
item_1	2	-	3.262013982
item_2	3	4	3.56331003
item_3	-	2	2.277268083



it	em



Lista top-N

item_3

Basados en ítems (IBCF):

Considera también las similitudes del ítem a recomendar con otros que el usuario ya ha valorado.

- 1. Primero se precalcula las similitudes entre ítems.
- 2. Se realiza la predicción del rating mediante una suma ponderada o una regresión.

Dentro de esta familia de algoritmos se encuentra SLOPE ONE

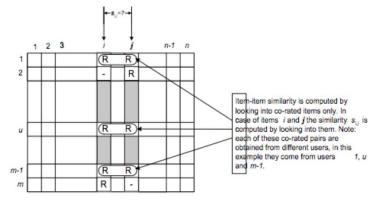
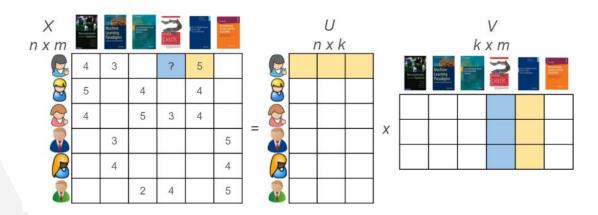


Figure 2: Isolation of the co-rated items and similarity computation



☐ Factorización de matrices:

Consiste en descomponer la matriz de valoraciones que contiene muchos elementos vacíos en dos matrices **menos dispersas**.



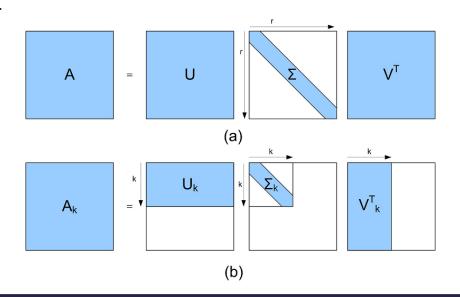
☐ Factorización de matrices

Motivación: ¡matrices de ratings enormes! Amazon, Netflix, etc.

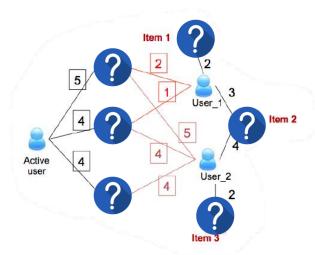
Solución: Reducción de la dimensionalidad.

Métodos:

- ► SVD
- Funk's SVD
- ► NMF
- Etc.

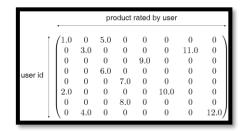


- Ventajas:
- Fácil de implementar.
- Favorece la serendipia.
- Agnóstico al dominio de aplicación.



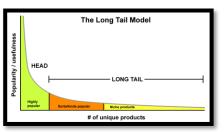
Limitaciones:

- Dispersión (Sparsity)
- Escalabilidad
- Primera valoración (early rater)y arranque en frío (cold start)
- Oveja negra o Grey Sheep
- Long Tail









Sistemas Basados en Conocimiento

Knowledge-Based Systems (métodos basados en conocimiento): Utilizan información dependiente del dominio acerca de los ítems y de los usuarios.

Etapas:

- 1. Construcción de una base de datos que describa las propiedades de los ítems disponibles en el sistema
- 2. Obtención de las preferencias del usuario en base a ítems que haya seleccionado
- 3. Cálculo de las similitudes de otros ítems utilizando alguna medida de distancia entre los ítems que considere los valores de sus atributos (o propiedades).
- 4. Obtención de una lista de ítems interesantes para el usuario
- 5. Recomendación

Context-Aware

RecSys sensibles al contexto (CARS)

CARS

Los sistemas de recomendación sensibles al contexto (*context-aware* recommender systems-CARS) proporcionan recomendaciones basadas en la información habitual y en **información contextual**.

Dependiendo del dominio esta información puede tener un impacto directo en

las preferencias del usuario.

Información contextual:

Tiempo atmosférico

Tiempo: mes, semana, día, hora, etc.

Actividad realizada.

- Localización física.
- Estado anímico.
- Etc.



CARS

El problema de recomendación tradicional (2D)

R: User x Item \rightarrow Rating

Cambia a (MD):

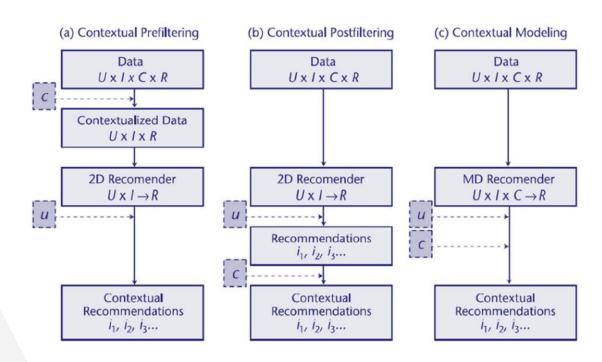
R: User x Item x context \rightarrow Rating

Usuario	Película	Tiempo	Localización	Compañía	Rating
U1	Titanic	Fin de semana	Casa	Novia	4
U2	Titanic	Fin de semana	Casa	Hermano	5
U3	Titanic	Diario	Cine	Amigo	4

114	Titanic	Diario	Caca	Hermano	2
01	Titanic	Diario	Casa	Hermano	•

CARS

Enfoques:



Dominio de la Música

Aspectos propios de este dominio

Aspectos claves del dominio de la música

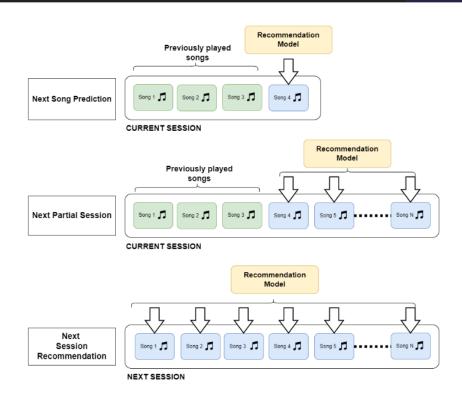
Un ítem puede ser **consumido múltiples veces** en el mismo día a diferencia de otros dominios. El **tiempo de consumo** de un ítem es muy limitado Los usuarios son más tolerantes a una mala recomendación Recomendación a distintos niveles: artista, álbum, canción. El feedback es mayoritariamente implícito. Datasets con una densidad baja (high sparsity) debido al tamaño de los catálogos. Muchos enfoques basados en contenido del área de MIR, sin embargo el usuario en ocasiones busca descubrir nuevas canciones. (Evaluación de la recomendación) El contexto del usuario juega un papel vital en las

preferencias del usuario



Tareas de recomendación en sistemas de streaming

- Siguiente canción
- Siguiente lista parcial
- Siguiente lista de canciones (playlist)

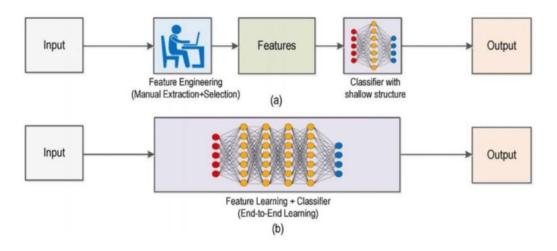


Tendencias actuales RecSys

Trabajos actuales en Music RecSys

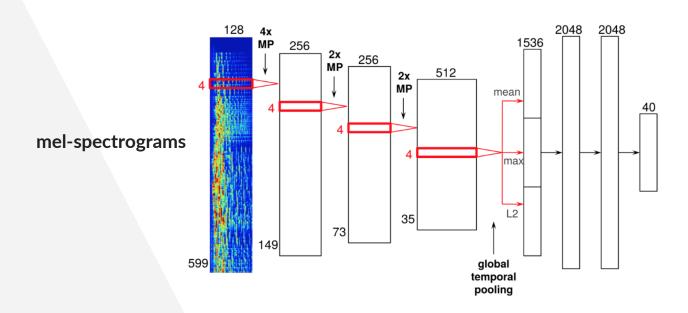
Deep Learning

☐ Deep Learning:



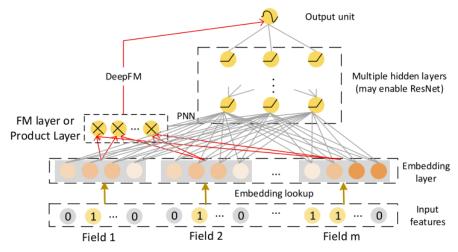
Deep Learning

Extracción de características de la señal en crudo



Deep Learning y Filtrado Colaborativo

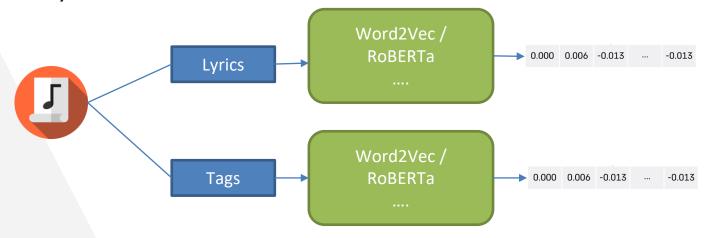
Introducir estas representaciones latentes en modelos de filtrado colaborativo.



xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems

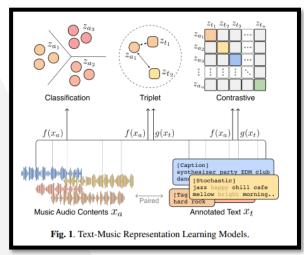
NLP (Embeddings)

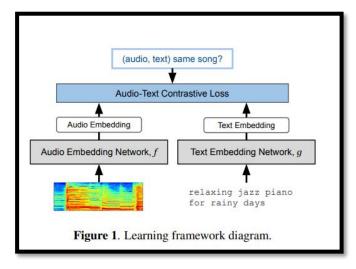
- NLP (Natural Language Processing)
- Conseguir representaciones en forma de *embeddings* de las letras de las canciones empleando modelos como **Word2Vec, BERT, RoBERTa** (entre otros).



NLP (Embeddings)

- NLP (Natural Language Processing)
- Embeddings + Audio





Universal Text-To-Music Retreival: https://arxiv.org/pdf/2211.14558.pdf

MULAN: https://arxiv.org/abs/2208.12415

Transformers and Large Language Models (LLM)

Uso de arquitecturas basadas en Transformers para obtener estos *embeddings*

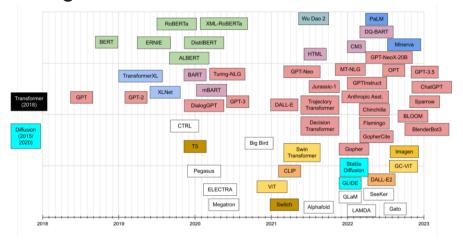
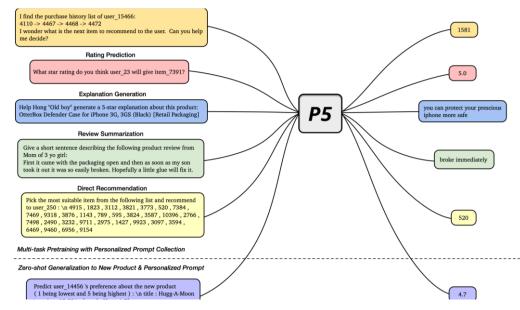


Figure 7: Transformer timeline. Colors describe Transformer family.

Fuente: Xavier Amatriain TRANSFORMER MODELS: AN INTRODUCTION AND CATALOG

Transformers and Large Language Models (LLM)

Recsys based on LLM:



Fuente: https://github.com/jeykigung/P5

¿Por dónde empiezo?

Listado de recursos útiles

Recursos

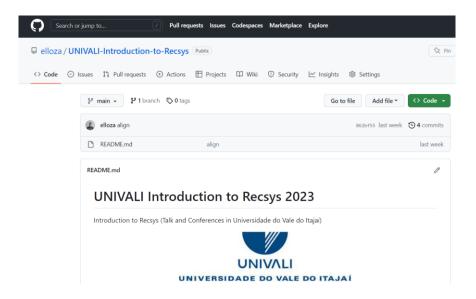
- Cursos (de menor a mayor dificultad):
- Recommender Systems Google
- **Coursera Specialization**
- Recommender systems behind de screen
- Denis Parra Course UPC
- Libros:
 - <u>Recommender Systems Handbook</u> (Libro de referencia teórico)
 - Hands-On Recommendation Systems with Python (Ejemplos prácticos)
- Repositorios Github:
 - Best Practices Recsys (Microsoft)
 - Awesome Recsys





Recursos

Slides presentación: https://github.com/elloza/UNIVALI-Introduction-to-Recsys



iMuchas gracias!



Contacto:

Email: <u>loza@usal.es</u>

Twitter: <a>@_lozanillo_

Website: https://elloza.com/

Github recursos y presentación:

https://github.com/elloza/UNIVALI-Introduction-to-Recsys





