

Programación Avanzada

IIC2233 2024-2

Hernán Valdivieso - Daniela Concha - Francisca Ibarra - Lucas Van Sint Jan - Francisca Cattan

Anuncios

1. Hoy es la última clase para aprender nuevas cosas.
 2. Mañana se entrega la T4, ¡No olviden pushear códigos funcionales!
 3. La otra semana será el cierre del curso. Tendremos 2 salas para eso.
 4. Quedan solo 2 ECA más por responder.
-

Tópicos Avanzados 2

Sistemas Recomendadores

Vamos a explorar los conceptos claves de esta aplicación, junto con algunos algoritmos.

Luego los pondremos en práctica con un caso de recomendación de anime.

Sistemas Recomendadores

Sistemas que sugieren ítems (como series, películas, productos) a los usuarios en función de sus preferencias y/o comportamientos pasados.

Ejemplos comunes:

- *Netflix*: Recomendaciones de series y películas.
- *Spotify*: Descubrimiento de música personalizada.
- *Amazon*: Productos sugeridos.

Sistemas Recomendadores

Conceptos que utilizaremos durante la clase:

- **Usuario**: entidad a quien le vamos a recomendar.
- **Ítem**: entidad que deseamos recomendar.
- **Interacción**: todo lo que el usuario hace al ítem (*like, dislike, rating, ver, etc.*).
- **Metadata**: información adicional que describe al usuario o al ítem.

Cold Start

Problema **común** de los sistemas cuando no hay suficiente información inicial para hacer buenas recomendaciones.

- *Cold Start de Usuario Nuevo*: no se sabe sus gustos y/o interacciones.
- *Cold Start de Ítem Nuevo*: no hay datos de si es bueno o no, o qué tipo de usuario lo consume.

🤔 ¿Qué harían ustedes para solucionar estos problemas? 🤔

Tipos de sistemas recomendadores

Se pueden clasificar según diferentes criterios:

- **Tipo de Personalización:** es igual para todos (sin personalizar) o se ajusta al usuario (personalizado).
- **Fuente de Datos utilizada:** interacciones, contenidos o híbrido.
- **Nivel de Interacción con el Usuario:** solicitar datos iniciales, inferir según las acciones del usuario o utilizar algún algoritmo que no se ajuste a las preferencias del usuario.

Algoritmos no personalizados

- Algoritmos que **no** tienen en cuenta quién es el usuario objetivo.
- Útiles para tratar el *cold start* de usuarios nuevos
- Algunos algoritmos son:

Algoritmo	¿Qué recomienda?	Ventaja	Desventaja
Most popular	Ítem más consumido	Eficiente de ejecutar e implementar	No considera la opinión de los usuarios sobre el ítem
Ítem average	Ítem con mayor rating promedio	Considera información de otros usuarios	Puede no coincidir con el gusto del usuario
Random	Cualquier ítem	Explorar nuevos ítems	Es fácil dar un ítem malo

Algoritmos personalizados

Filtrado Colaborativo basado en rating

1. Se utiliza la opinión de los demás que comparten una serie en común para predecir la opinión del usuario objetivo sobre los demás ítems.

“A” le dió un 4 a Gintama y un 5 a Spy x Family.

Por lo tanto, dado que “B” le dió un 4 a Spy x Family, entonces le dará un 3 a Gintama.

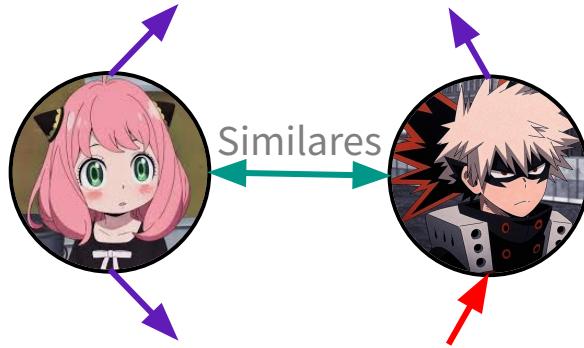


Algoritmos personalizados

Filtrado Colaborativo basado en similaridad

1. Se busca a un **usuario** con gustos **similares** al usuario objetivo.
2. Se recomienda un ítem que le gustó al usuario encontrado y no consumió el usuario objetivo.

“A” y “B” les gusta Detective Conan. Dado que “A” también le gustó Kaito Kid, entonces recomendamos esa serie a “B”.



Algoritmos personalizados

Filtrado Basado en Contenido

1. Se busca a un **ítem** con metadata **similar** a un ítem ya consumido.
2. El metadato a utilizar queda a criterio de los datos disponible (sinopsis, géneros, autor, etc.)

“A” le gusta Dragon Ball. Entonces se buscará otra serie similar, en este caso, del mismo autor: Sand Land.

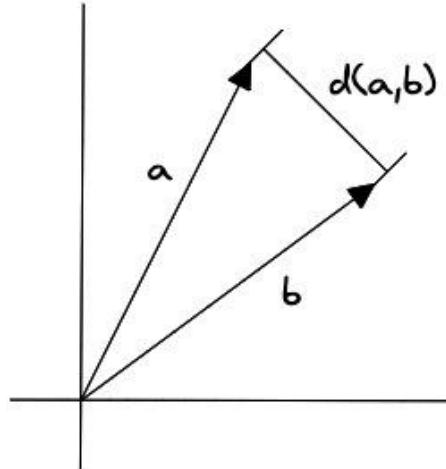


Similares

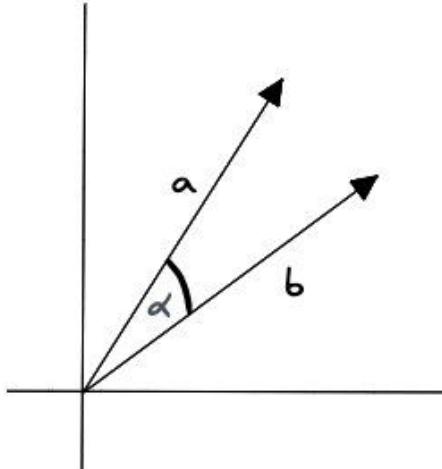
 ¿Cómo determinamos qué es similar? 

¿Cómo determinamos qué es similar?

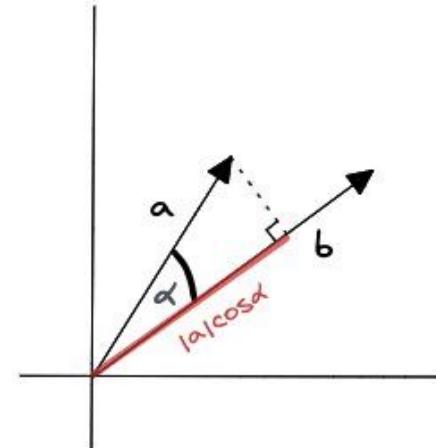
- Existen muchas formas de verificar similaridad. Algunas concidas son:



Euclidean Distance



Cosine Similarity



Dot Product



¿Cómo sé si recomiendo bien?



¿Cómo sé si recomiendo bien? 🤔

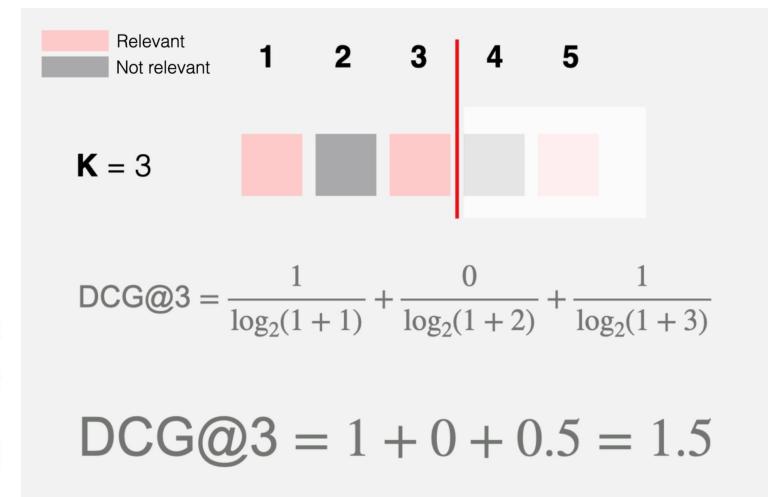
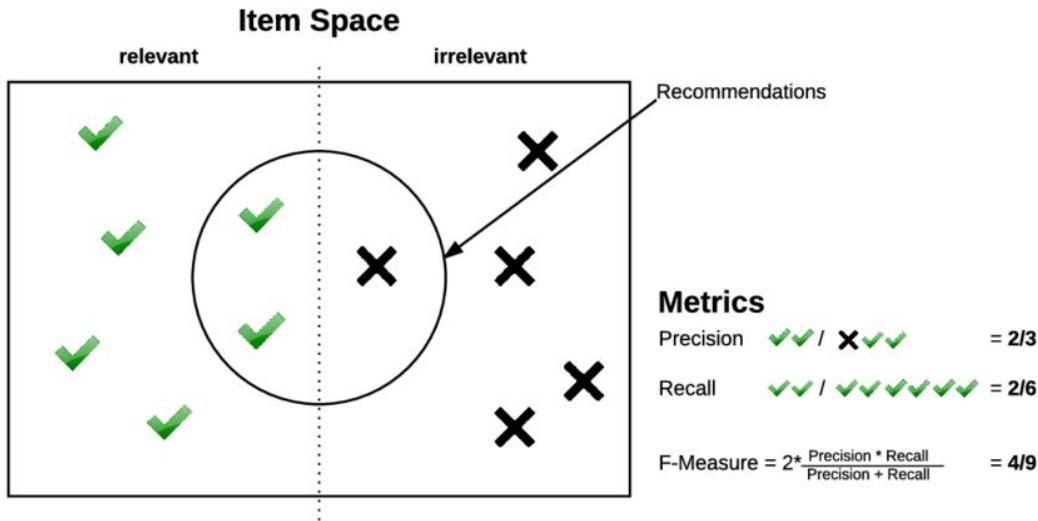
- **Rating:** Dado un usuario (u) y los ítems (i), se calcula el error entre el *rating* predicho (\hat{r}_{ui}) y el rating real (r_{ui}).

$$MAE = \frac{\sum_{i=1}^n |\hat{r}_{ui} - r_{ui}|}{n} \quad MSE = \frac{\sum_{i=1}^n (\hat{r}_{ui} - r_{ui})^2}{n}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{r}_{ui} - r_{ui})^2}{n}}$$

¿Cómo sé si recomiendo bien? 🤔

- **Recomendación:** se evalúa si los ítems recomendados fueron relevantes para el usuario. El orden puede afectar el puntaje.



Sistemas recomendadores y ¿*Deep Learning*?

El uso de *Deep Learning* no reemplaza los algoritmos, los complementa para que sean más precisos, personalizados y adaptables.

- Captura patrones no lineales y **relaciones avanzadas** entre usuarios e ítems.
- Es capaz de **combinar tipos de datos**: texto, imágenes, audio y temporalidad en un solo modelo.
- **Escalable** a millones de datos.
- Permite **combinar algoritmos** en un mismo modelo.
- Puede crear *embeddings* más ricos en contenido.

Sistema de recomendación?

El uso de
sean más

ra que

- Captura de imágenes para que los sistemas de recomendación sepan más sobre los ítems.
- Es capaz de recomendar ítems en un orden jerárquico.
- **Escalabilidad** y eficiencia.
- Permite la creación de **embeddings**.
- Puede crear **embeddings** más ricos en contenido.



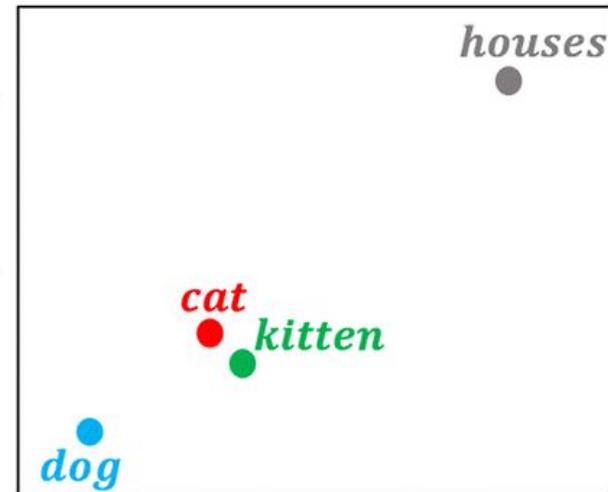
Embeddings?!!

(Extra) ¿Qué es un *Embedding*?

- Vector con únicamente números que "describe" las características de una entidad.

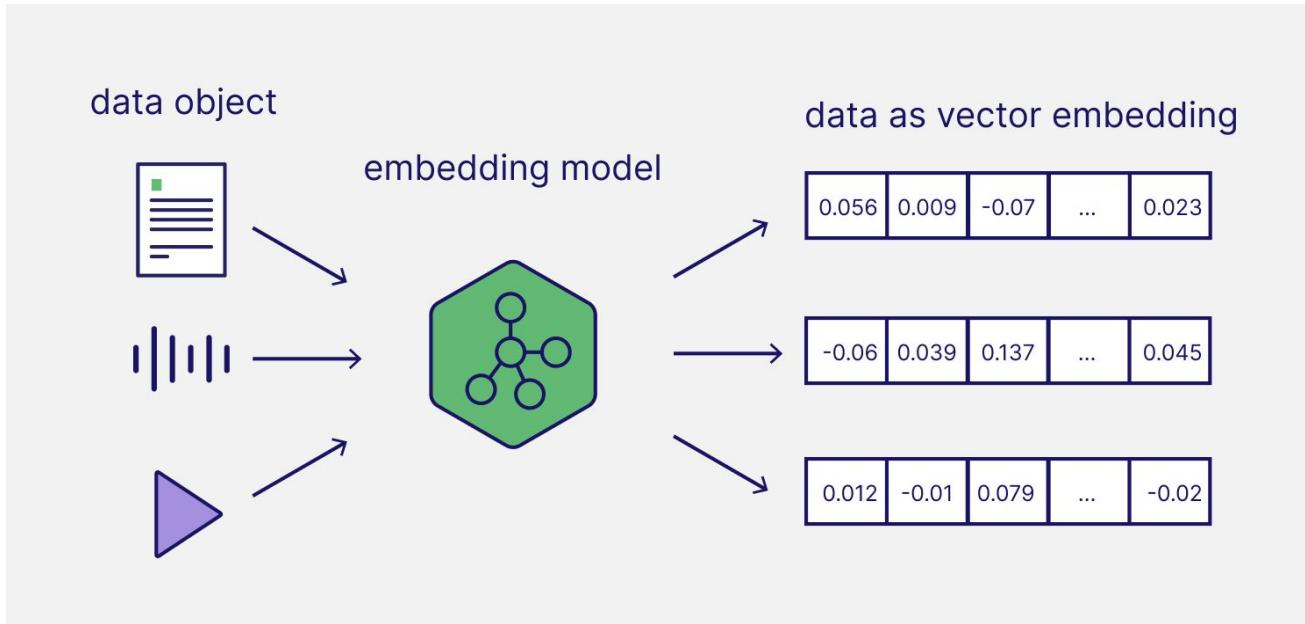
	living being	feline	human	gender	royalty	verb	plural
<i>cat</i> →	0.6	0.9	0.1	0.4	-0.7	-0.3	-0.2
<i>kitten</i> →	0.5	0.8	-0.1	0.2	-0.6	-0.5	-0.1
<i>dog</i> →	0.7	-0.1	0.4	0.3	-0.4	-0.1	-0.3
<i>houses</i> →	-0.8	-0.4	-0.5	0.1	-0.9	0.3	0.8

Dimensionality
reduction of
word
embeddings
from 7D to 2D



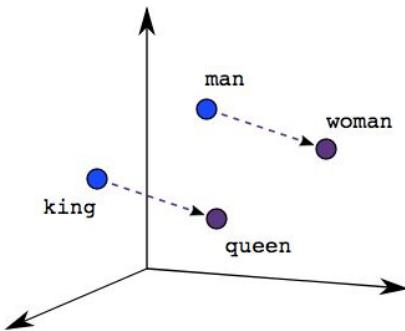
(Extra) ¿Qué es un *Embedding*?

- Con modelos de *Deep Learning* podemos generar estos *embeddings* con características que el humano a veces no aprecia.

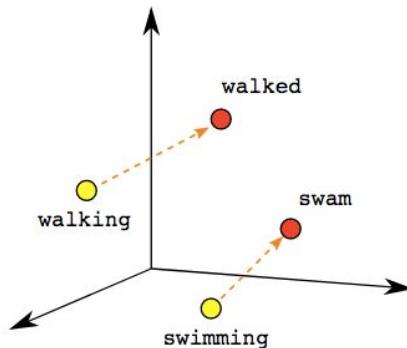


(Extra) ¿Qué es un *Embedding*?

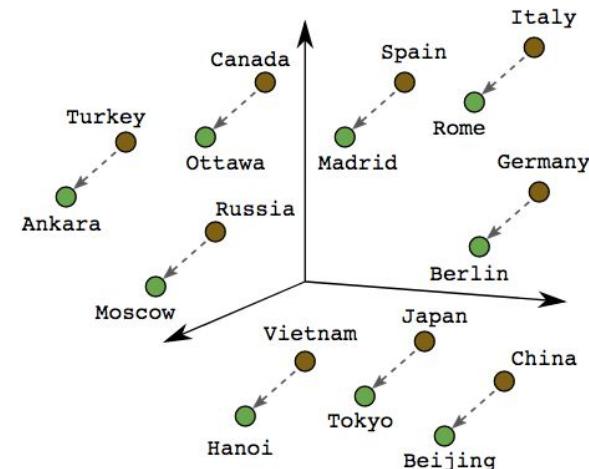
- Lo ideal es que operaciones entre los *embeddings* tengan un "sentido".



Male-Female



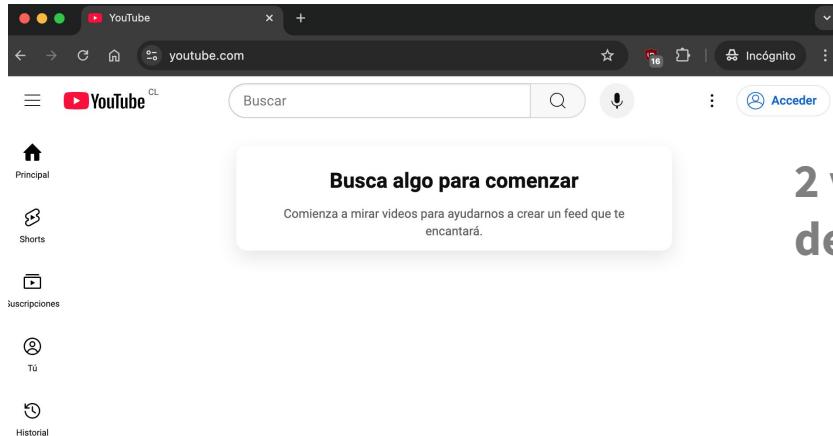
Verb Tense



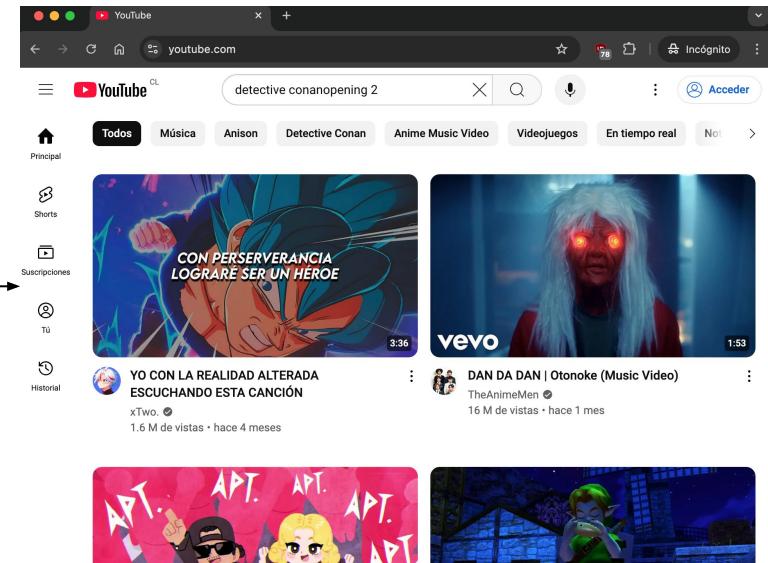
Country-Capital

Ejemplo - Youtube

- ¿Cómo recomienda videos esta gran plataforma?

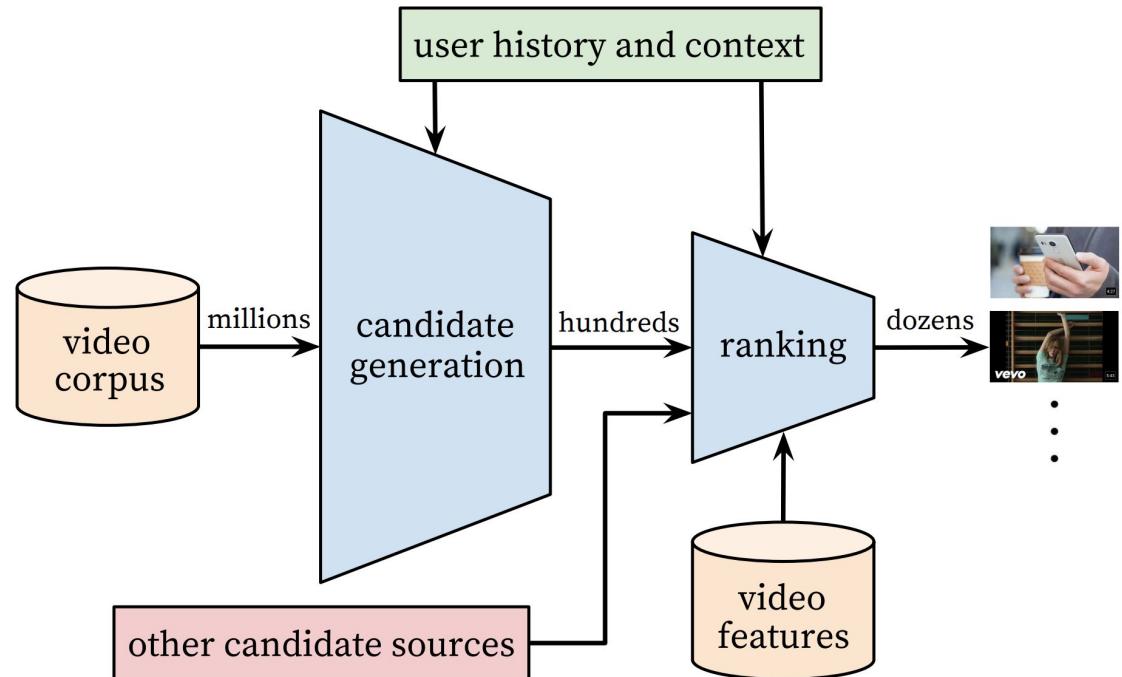


2 videos
después



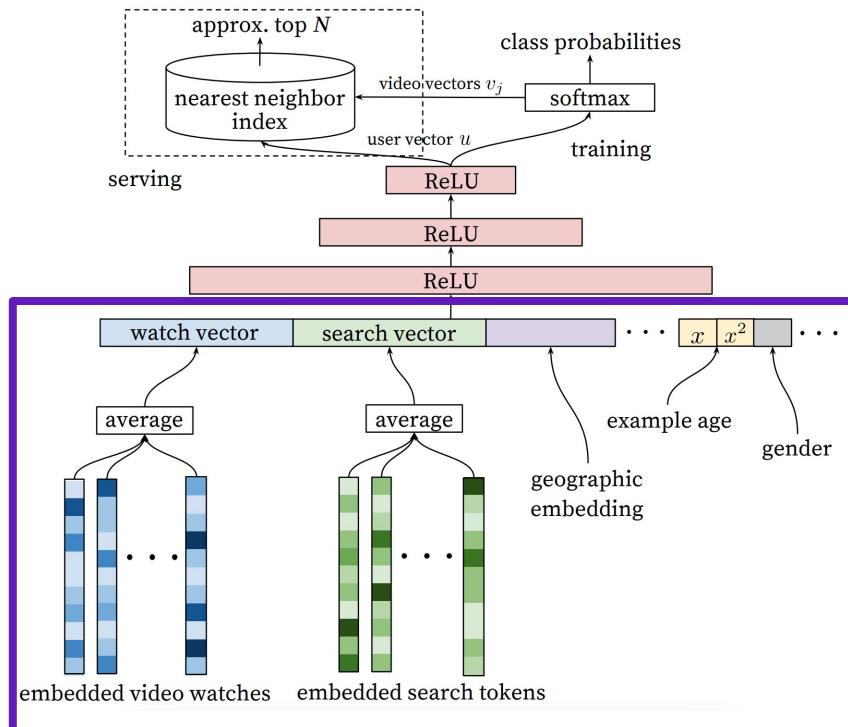
Ejemplo - Youtube (Panorama)

- Vamos a conocer cómo hacía youtube hasta el 2019 para recomendar videos.
- La nueva forma todavía no está publicada (secreto empresarial).



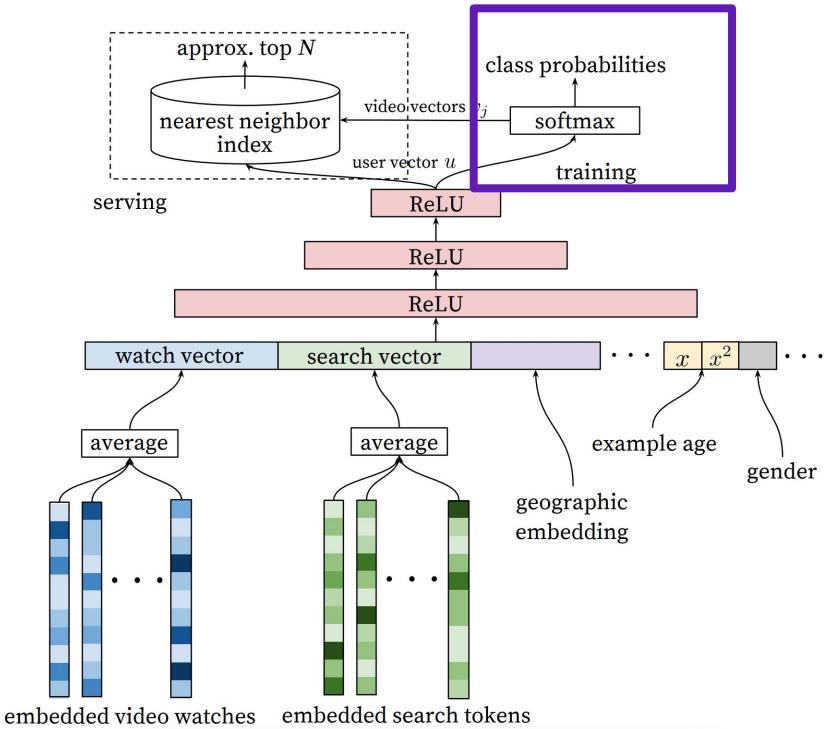
Ejemplo - Youtube (Parte 1)

- Se crea un gran *embedding* del usuario construido a partir de:
 - Promedio de los *embeddings* de los videos que vió.
 - Promedio de los *embeddings* de las búsquedas que hizo.
 - Información contextual.



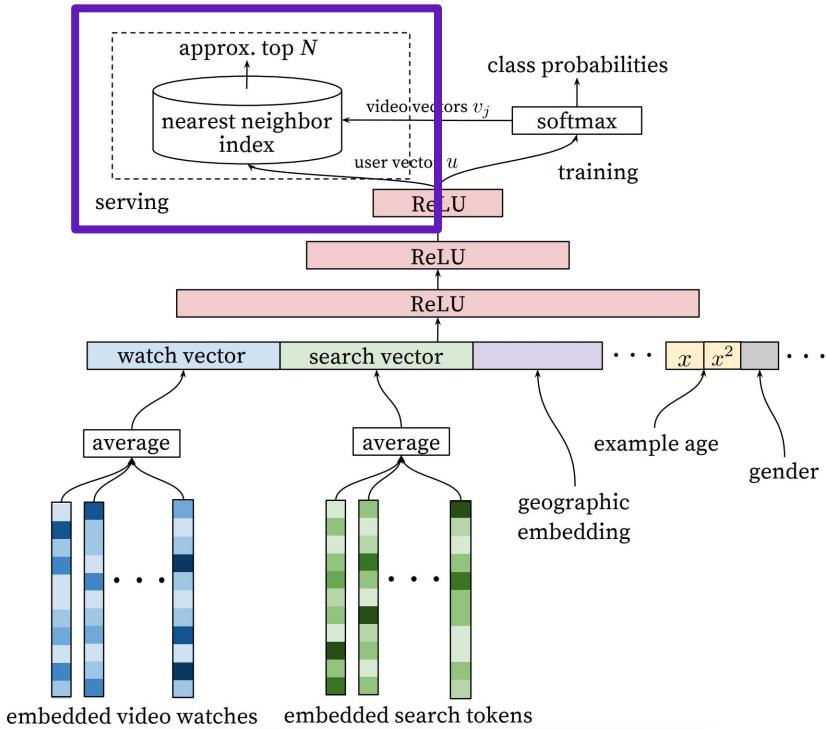
Ejemplo - Youtube (Parte 1)

- Se intenta predecir la probabilidad de ver cada video de Youtube.
- El modelo va mejorando para representar de forma más fiel al usuario y cada video.



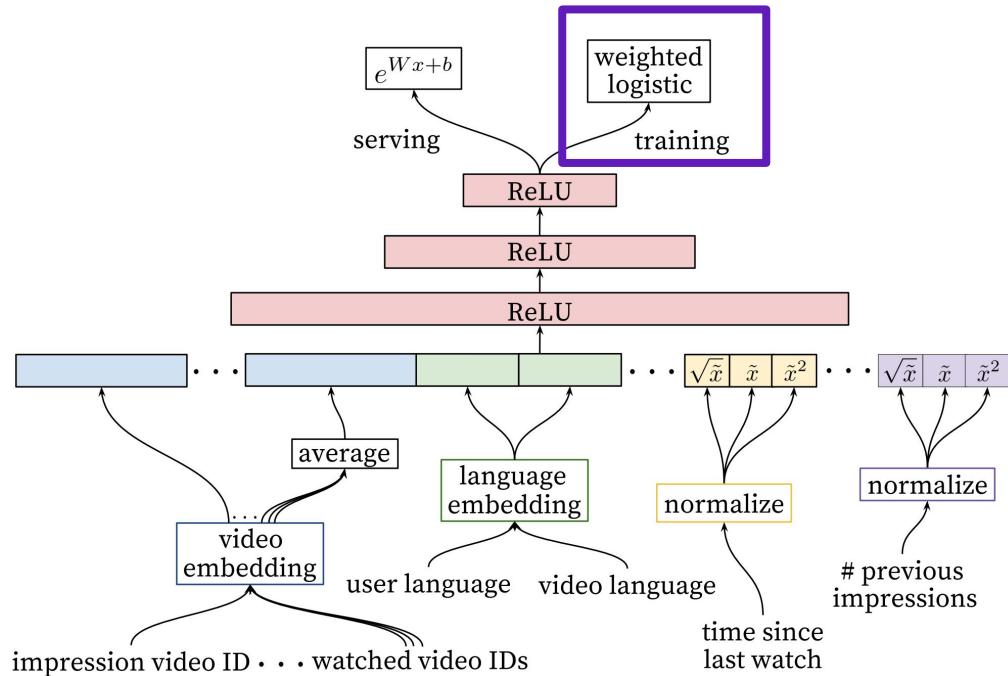
Ejemplo - Youtube (Parte 1)

- Al momento de “servir” (recomendar), dado un usuario, selecciona solo los videos con una alta posibilidad de ver.



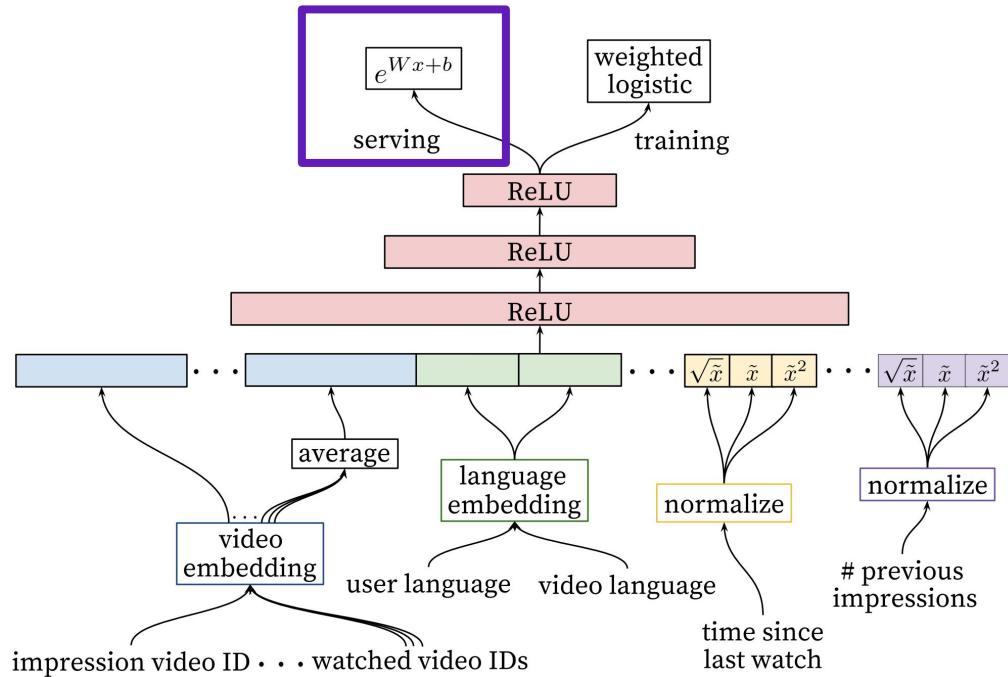
Ejemplo - Youtube (Parte 2)

- Dado un conjunto acotado de videos y más datos por video, el modelo se entrena intentando obtener las probabilidades de que un usuario vea un video durante un tiempo determinado.

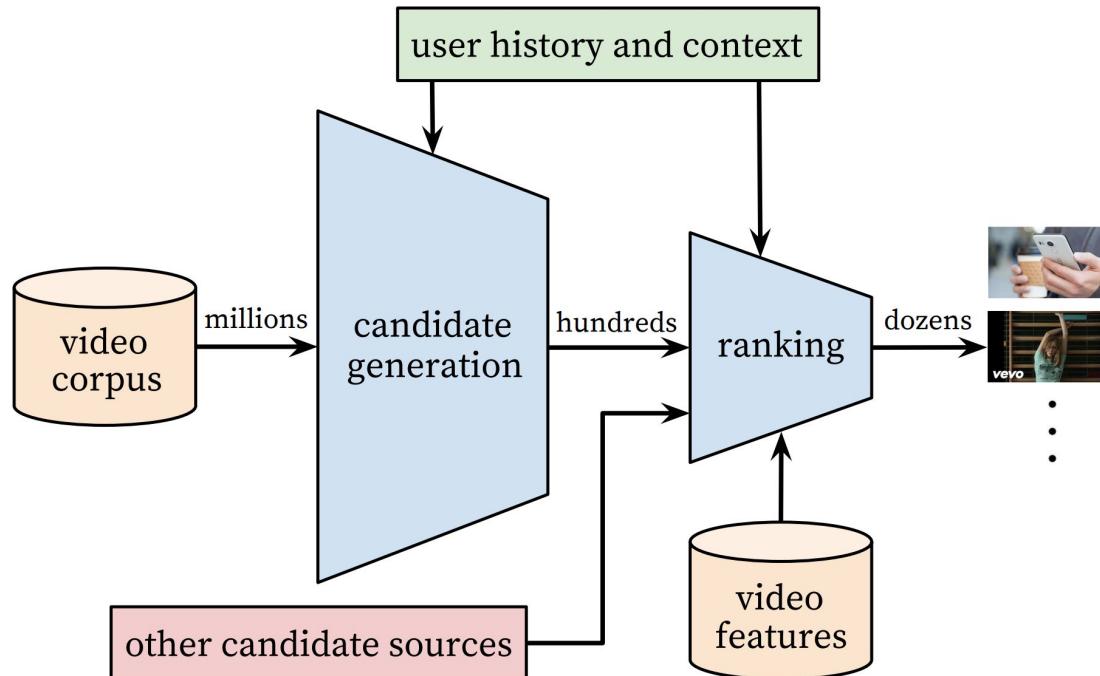


Ejemplo - Youtube (Parte 2)

- Al momento de “servir” (recomendar), se aplica una función exponencial (e^x) para transformar la *probabilidad de ver el video un tiempo t en el tiempo de visualización esperado.*
- Los videos son ordenandos según el tiempo de visualización.



Ejemplo - Youtube (Resumen)



Tópicos Avanzados 2

Sistemas Recomendadores

Demo aplicada con MyAnimeList

MyAnimeList

- Plataforma Web con información de animes, mangas y más.
- Como usuario, puedes registrar los animes que estás viendo, ya viste, dejaste de ver, etc. Además de asignar un puntaje entre 1 y 10 a cada anime.
- Usada fuertemente fuera de Japón.

MyAnimeList

Hide Ads | Panel Settings | Hernan4444 ▾

Search Anime, Manga, and more...

My Panel

MALxJapan - More than just anime-

Find authentic Japanese merch with colleize

Visit MALxJapan

My Statistics

Anime Entries	732 Add entries
Manga Entries	263 Add entries
AnimeList Views	69
MangaList Views	65
Profile Views	283 My profile
Signature Clicks	0

Upcoming Friend Birthdays

No friend birthdays are upcoming.

Recent Friend Updates

The Art of Reincarnation add
Reading 118/? - Scored - Yesterday, 4:40 PM by lily416

Comment

MAD MAD add
Reading 13/? - Scored - Yesterday, 4:40 PM by lily416

Comment

Settings See more

My Recently Active Clubs

You haven't joined any clubs.

Top Airing Anime

Dandadan Re:Zero Kara Hajimeru Isekai Seikatsu 3rd Season UZUMAKI Blue Lock vs. U-20 Japan Bleach: Sōka no Soukoku-tan

Bleach: Sōka no Soukoku-tan

MyAnimeList

Vamos a probar 4 modelos:

- *Most Popular.*
- Basado en Contenido usando algoritmo de Vecinos Cercanos.
- Filtrado Colaborativo usando algoritmo de Vecinos Cercanos.
- Híbrido (Filtrado Colaborativo y Basado en Contenido) utilizando redes neuronales.

Most Popular

1. Para entrenar, solo necesitamos darle un peso a cada anime según la cantidad de usuarios que vió dicho anime.
2. Para recomendar, ordenamos la lista de mayor a menor peso.

Basado en Contenidos

1. Crearemos un *embedding* por anime utilizando su **metadata**.
2. Creamos un modelo KNN que dado 1 *vector*, diga cuales son los *embeddings* más cercanos.
3. Buscamos los animes asociados a dichos *embeddings*.

Filtrado Colaborativo

1. Crearemos un *embedding* por anime utilizando las **interacciones de los usuarios**.
2. Creamos un modelo KNN que dado 1 *vector*, diga cuales son los *embeddings* más cercanos.
3. Buscamos los animes asociados a dichos *embeddings*.

Modelo Híbrido

1. Cada anime será representado como un *embedding* a partir de su metadata.
2. Cada usuario será representado como el promedio de los *embeddings* de los animes ya vistos.
3. Ambos vectores se usarán en un modelo de *Deep Learning* para intentar **predecir el *rating* del anime.**
4. Para recomendar, solo necesitamos predecir el *rating* de cada anime según el vector del usuario que nos llegue y determinar los animes con mayor *rating*.

MyAnimeList



¡Vamos a la demo!



[Google Colab: Semana 16 - Tópicos Avanzados 2 \(Sistemas recomendadores\)](#)

Desafíos actuales

Desafíos actuales

- Escalabilidad



Muchos datos



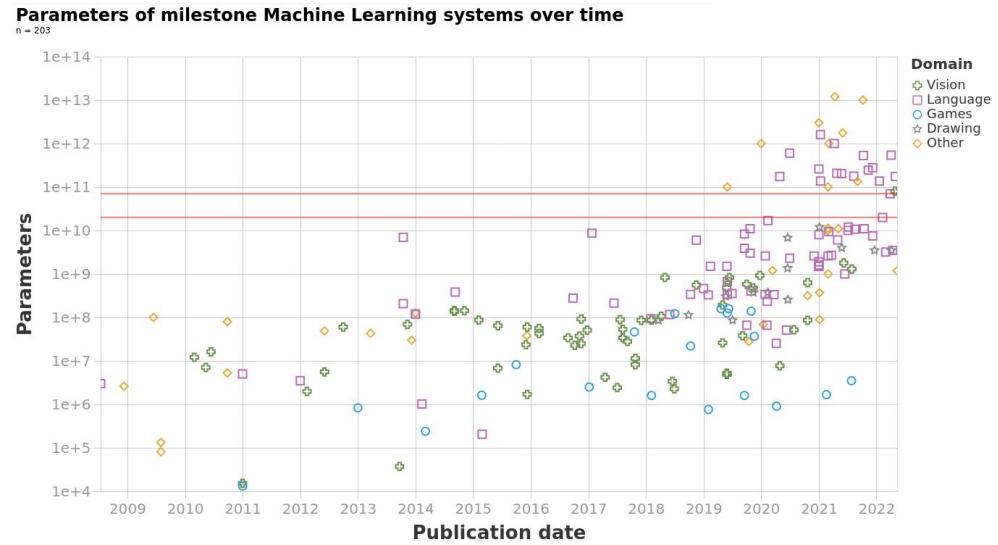
Miles de usuarios

Desafíos actuales

- Interpretabilidad



¿Por qué recomendaste eso?



Modelos con millones de parámetros

Desafíos actuales

- Diversidad y novedad



Dar recomendación variada y relevante

No ser obvio con la recomendación

Desafíos actuales

- Sesgos en los datos



Desbalance de datos



Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
Microsoft	94.0%	79.2%	100%	98.3%	20.8%
FACE++	99.3%	65.5%	99.2%	94.0%	33.8%
IBM	88.0%	65.3%	99.7%	92.9%	34.4%

¡Modelos con resultados sesgados donde no corresponde!

Cierre de la clase

Recomendación: Tomen IIC3633 - Sistemas Recomendadores

- Aprenderán en profundidad todo lo visto hoy y muchas cosas más.
- Tendrán evaluaciones que acercará la computación a problemas reales.
- Oportunidad de acercarse al ámbito académico de una forma guiada.

Repositorio de este semestre: [GitHub - PUC-RecSys-Class/RecSysPUC-2024-2](https://github.com/PUC-RecSys-Class/RecSysPUC-2024-2)

Programación Avanzada

IIC2233 2024-2

Hernán Valdivieso - Daniela Concha - Francisca Ibarra - Lucas Van Sint Jan - Francisca Cattan