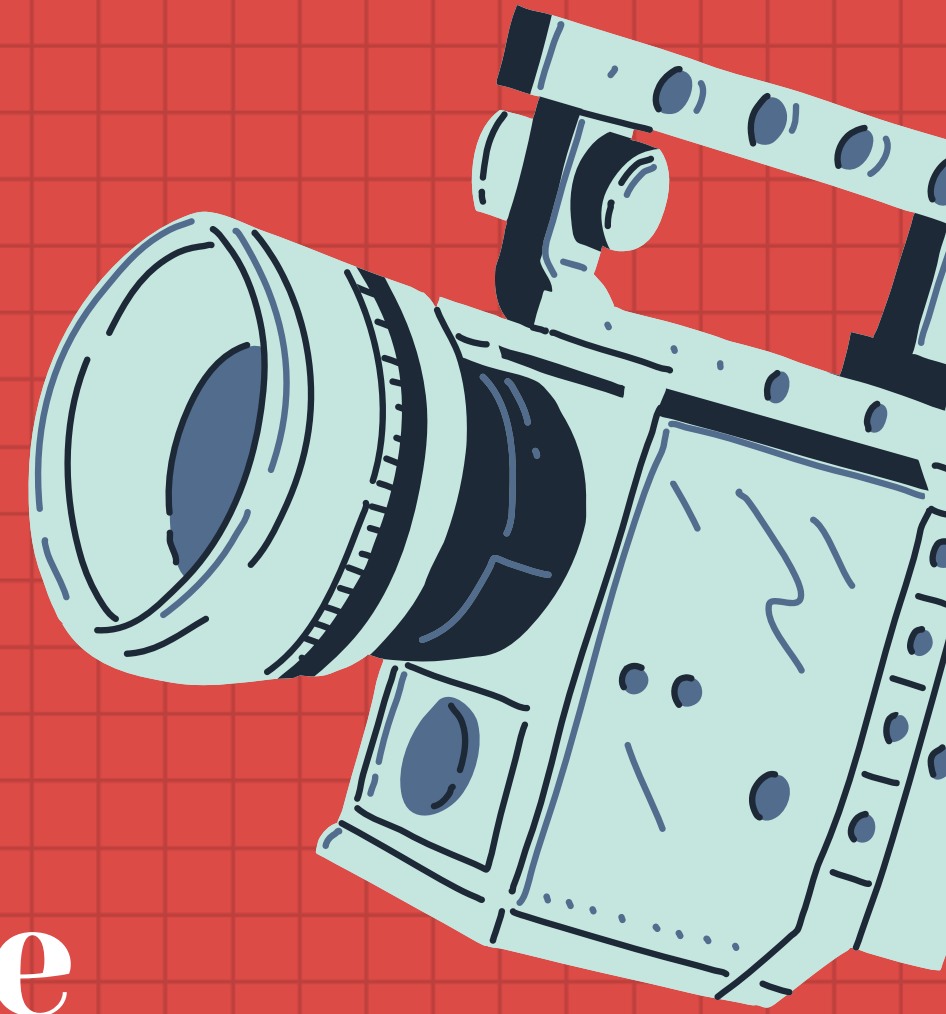
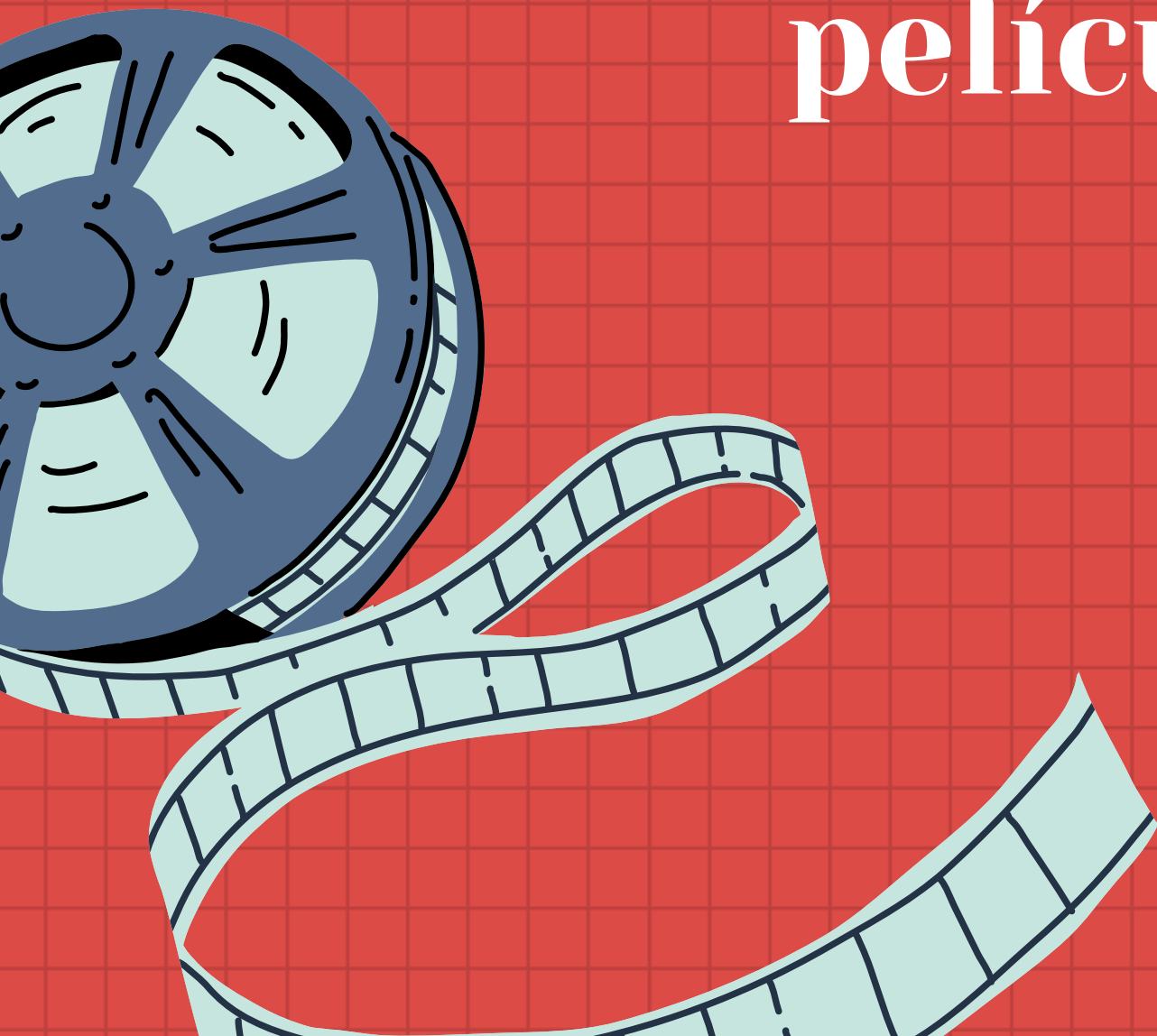


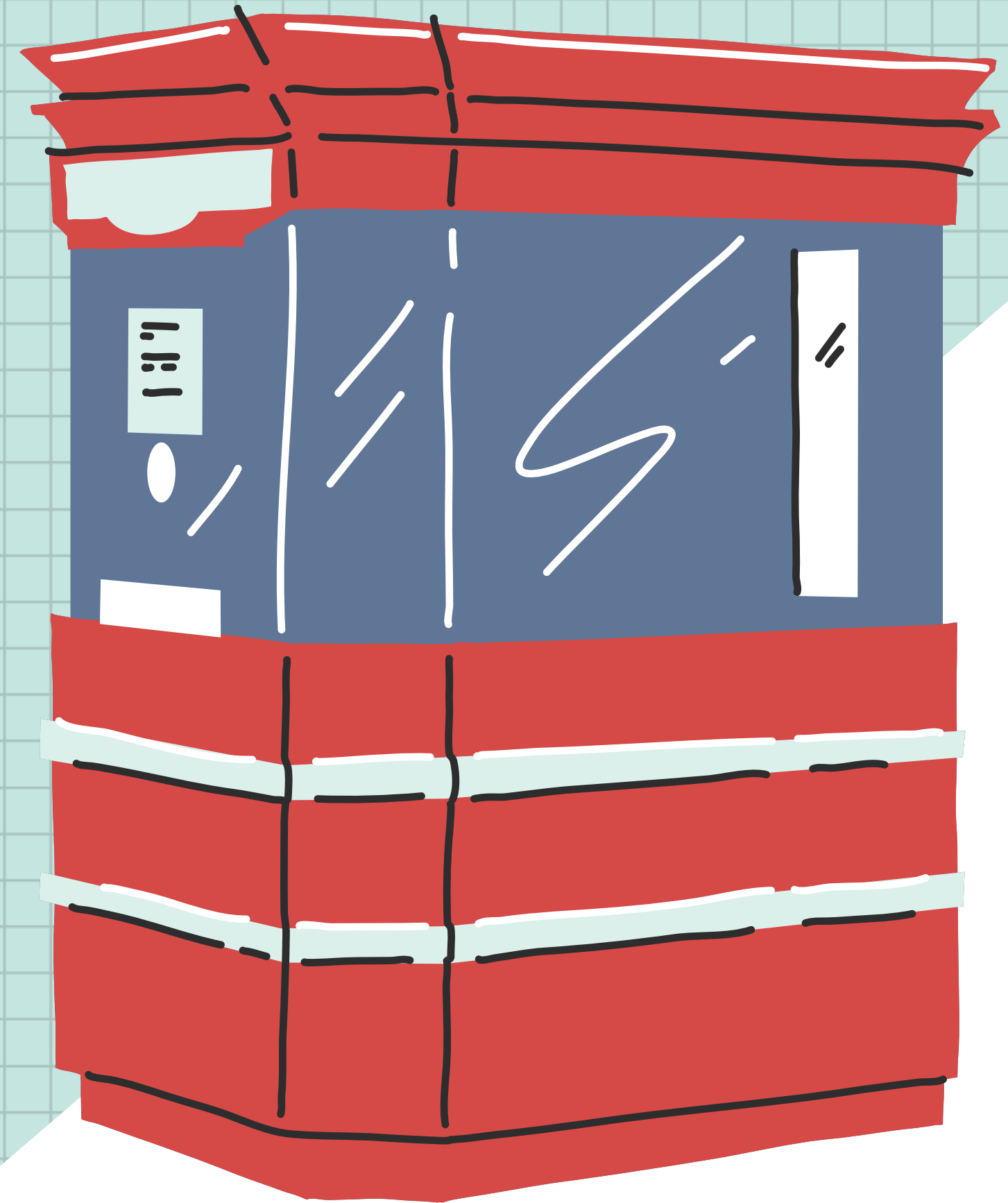
Recomendación de películas usando Machine Learning

Jorge Velásquez

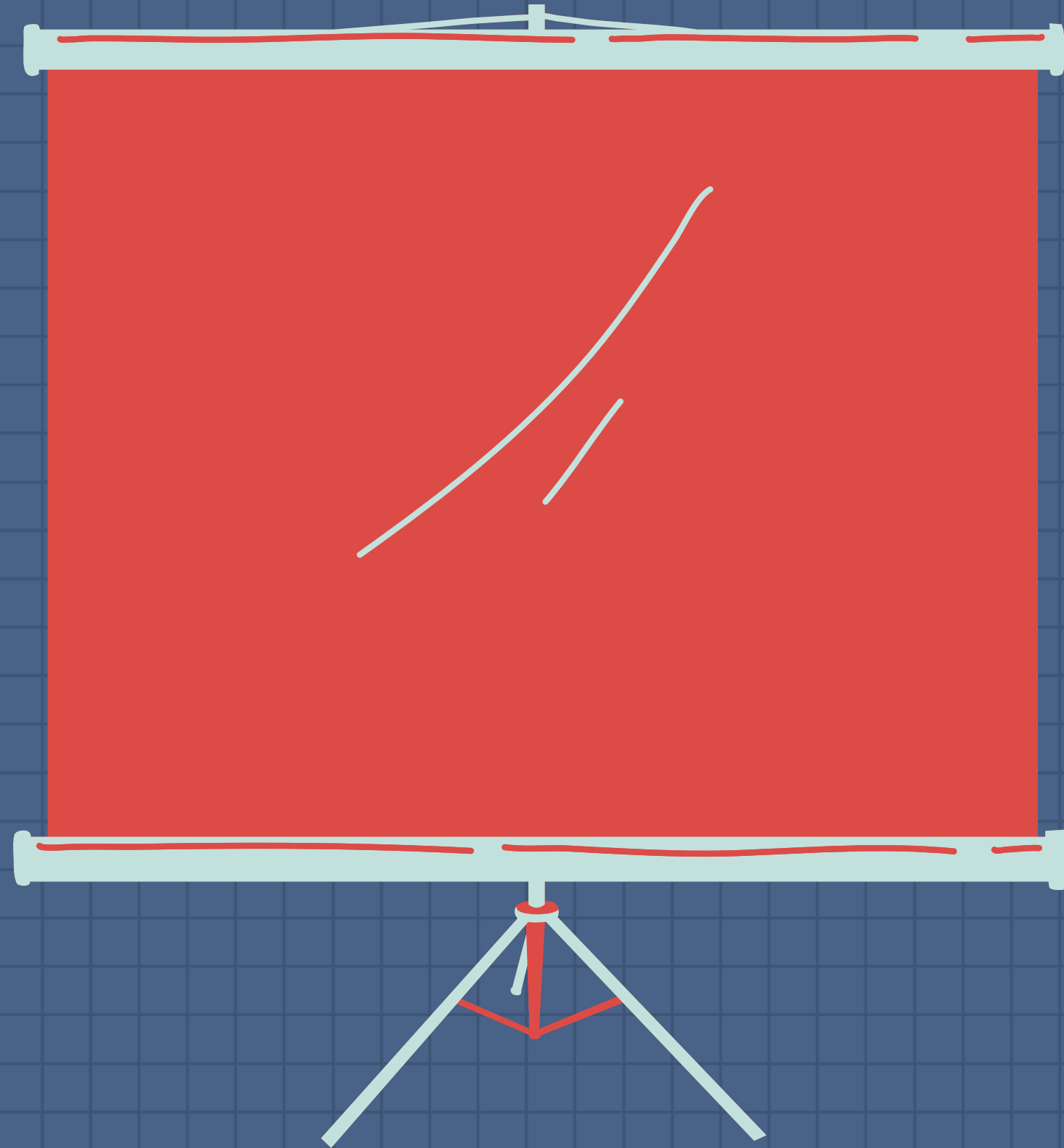


Contents

- 1 Definición del problema
- 2 Modelo Two-Tower
- 3 Implementación
- 4 Resultados y conclusiones



× ×



Definición del problema

× ×

Definición del problema

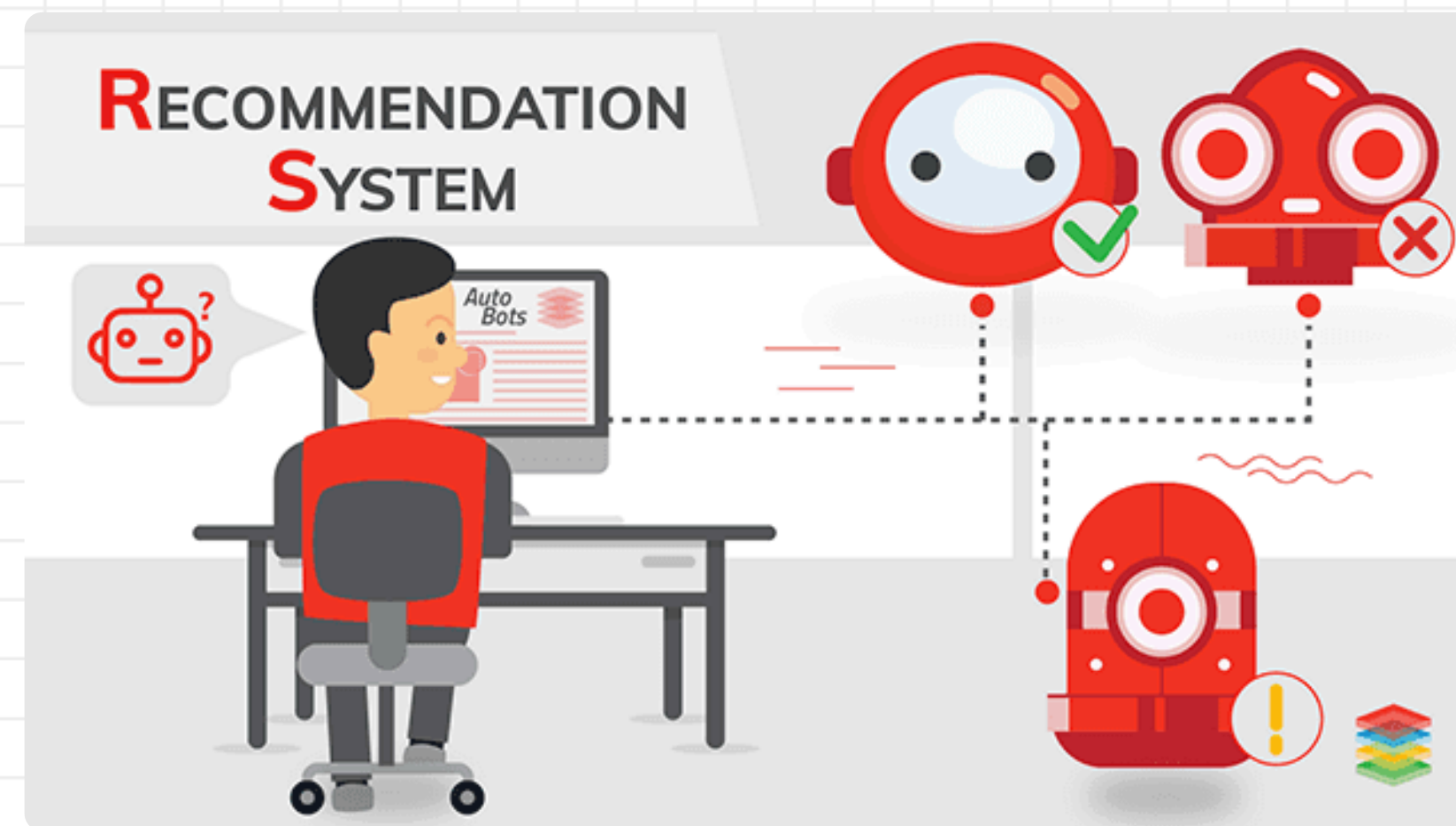
La industria cinematográfica
ofrece una gran cantidad de
opciones



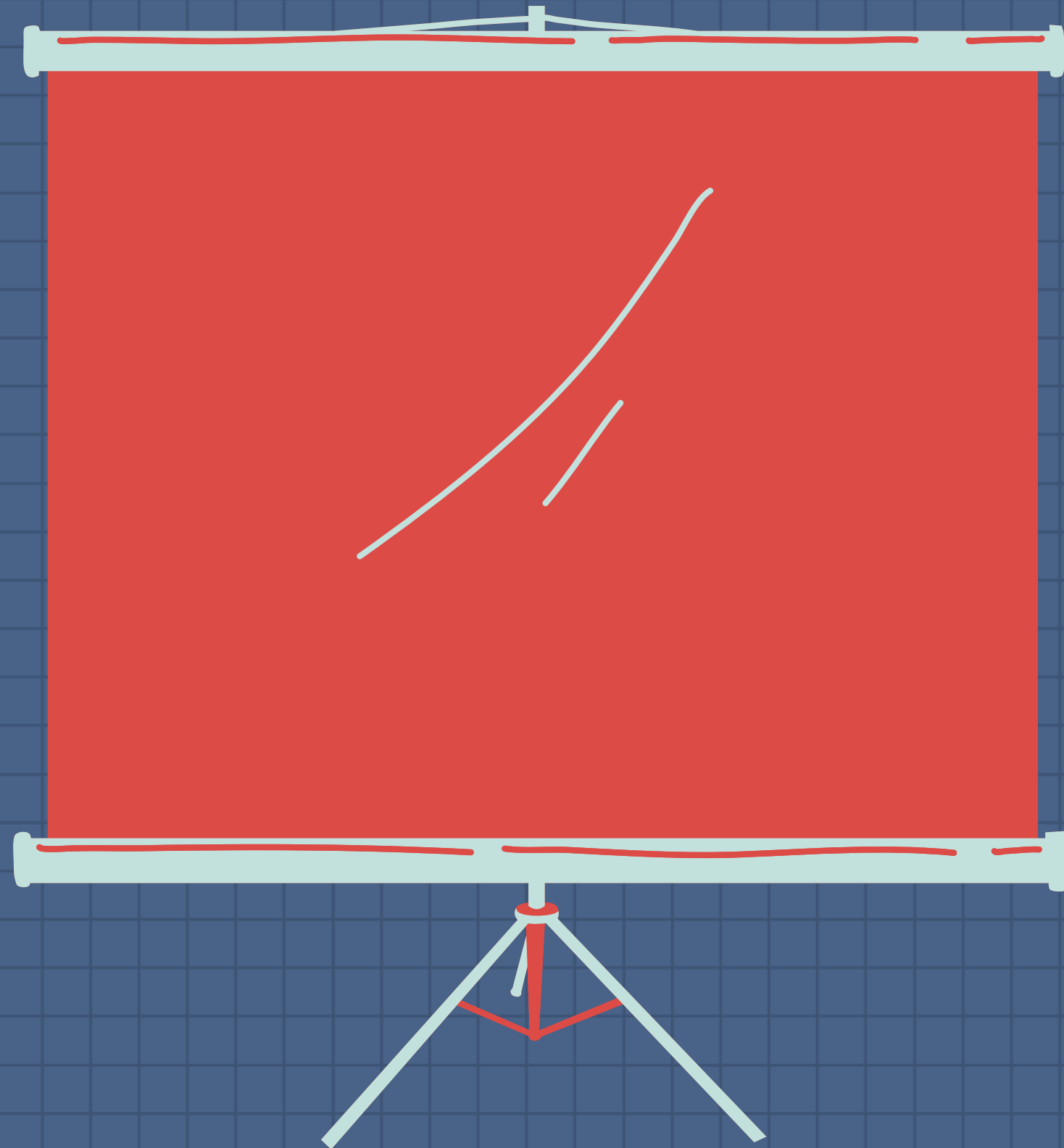
Cómo escoger la mejor?

Definición del problema

Las nuevas herramientas de
Machine Learning ofrecen una
solucion



× ×



Modelo Two-Tower

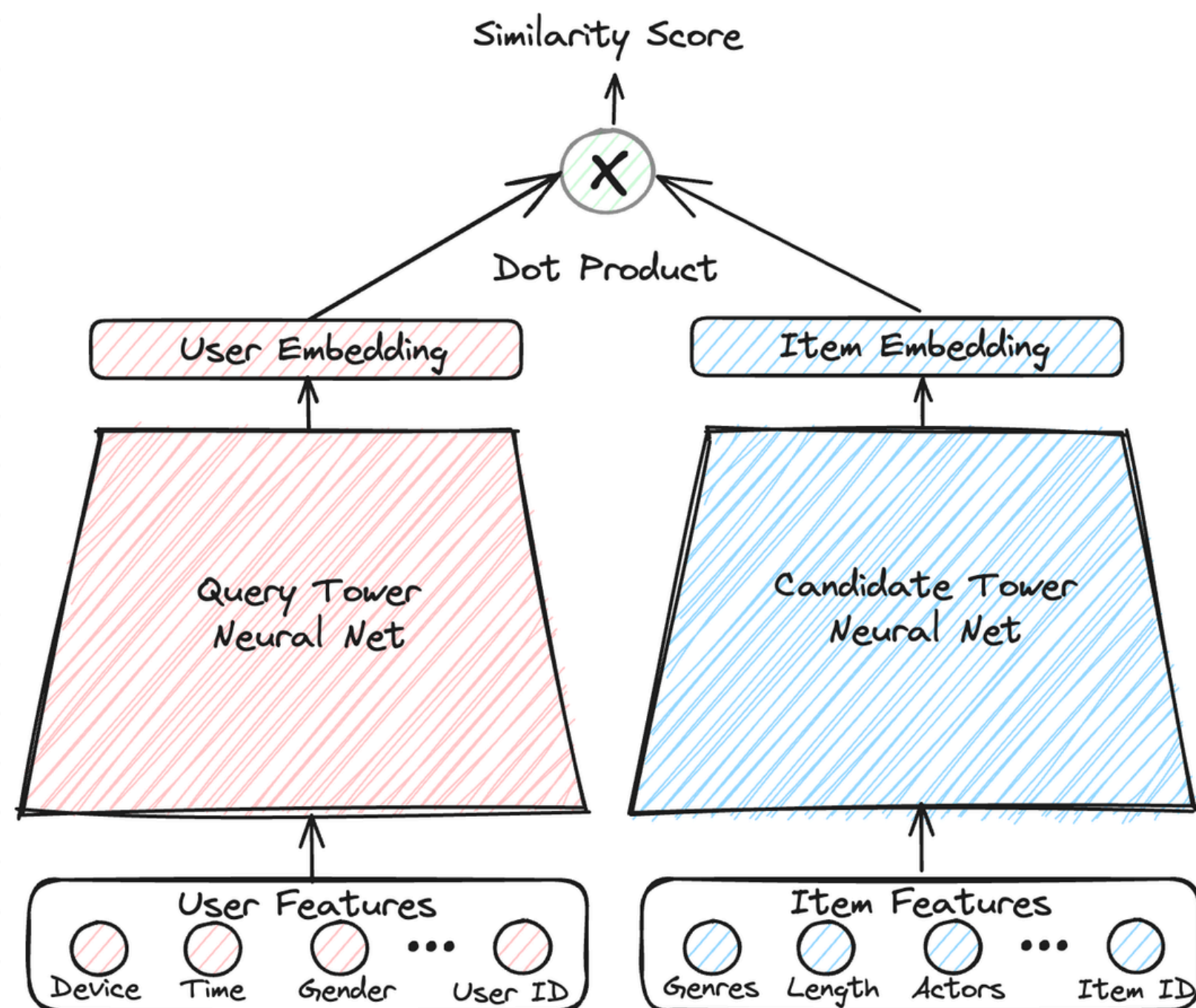
× ×

Modelo Two-Tower

Una red neuronal ("torre") para cada entidad

Una representación (embedding) para cada entidad

Un puntaje que nos permite hacer predicciones



Modelo Two-Tower



Entrenamos el modelo para
escoger un subconjunto de
candidatos de todas las películas



Modelo Two-Tower



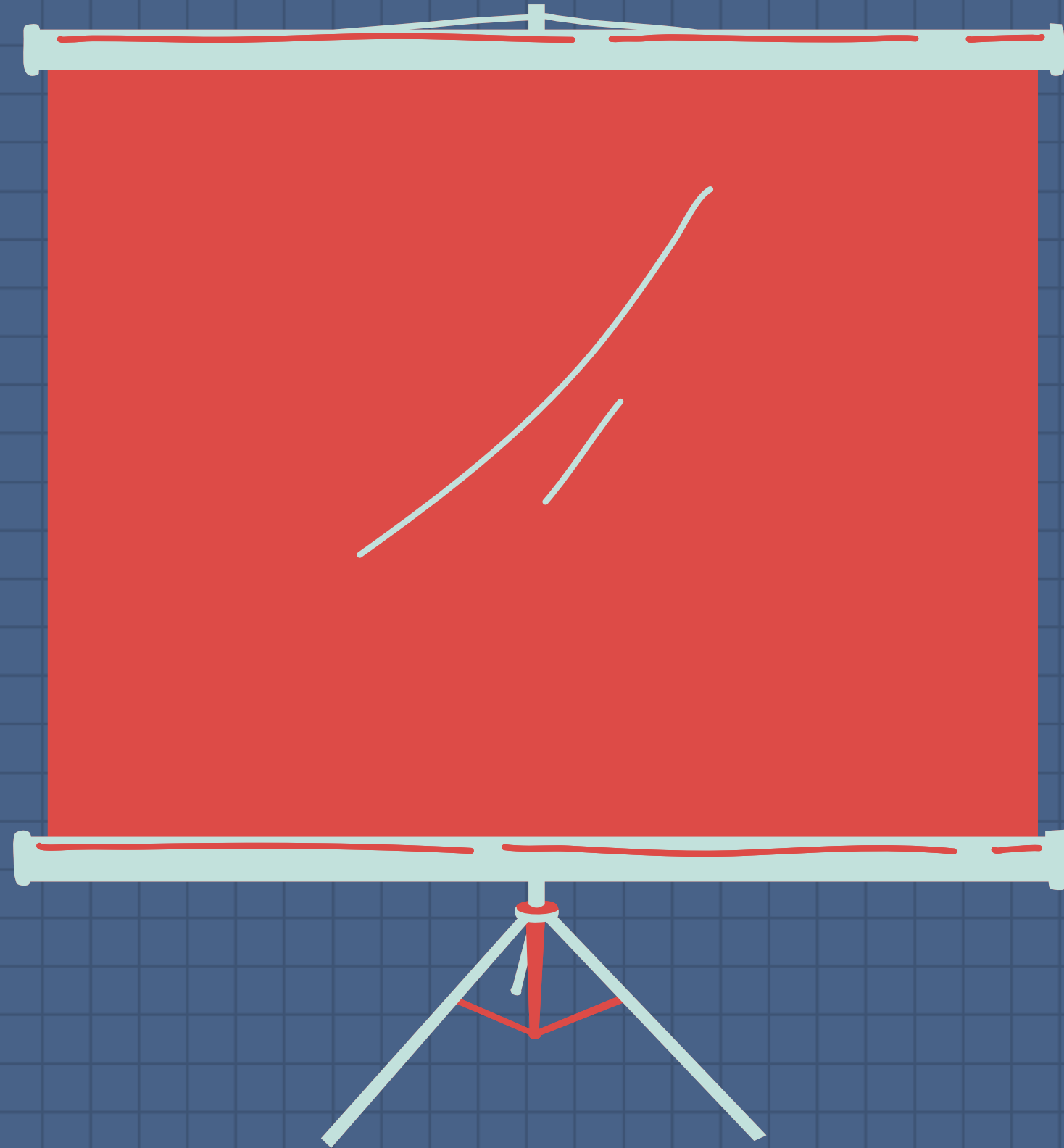
Embeddings

$$u : \mathcal{X} \times \mathbb{R}^d \mapsto \mathbb{R}^k, v : \mathcal{Y} \times \mathbb{R}^d \mapsto \mathbb{R}^k$$

Puntuación

$$s(x, y; \theta) = \langle u(x, \theta), v(y, \theta) \rangle$$

× ×



Implementación

× ×

Implementación



Base de datos de 100 mil datos



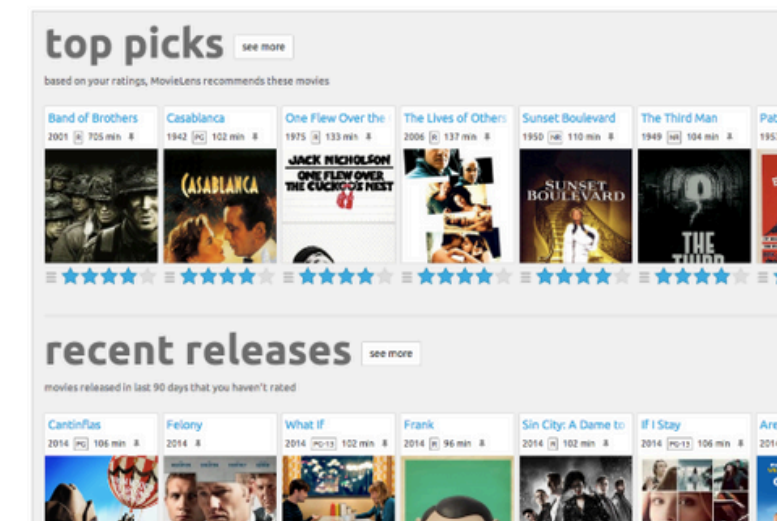
movielens

Non-commercial, personalized movie recommendations.

[sign up now](#) or [sign in](#)

recommendations

MovieLens helps you find movies you will like. Rate movies to build a custom taste profile, then MovieLens recommends other movies for you to watch.



Implementación

Base de datos de 100 mil datos

$$\dim(\mathbb{R}^k) = 32$$

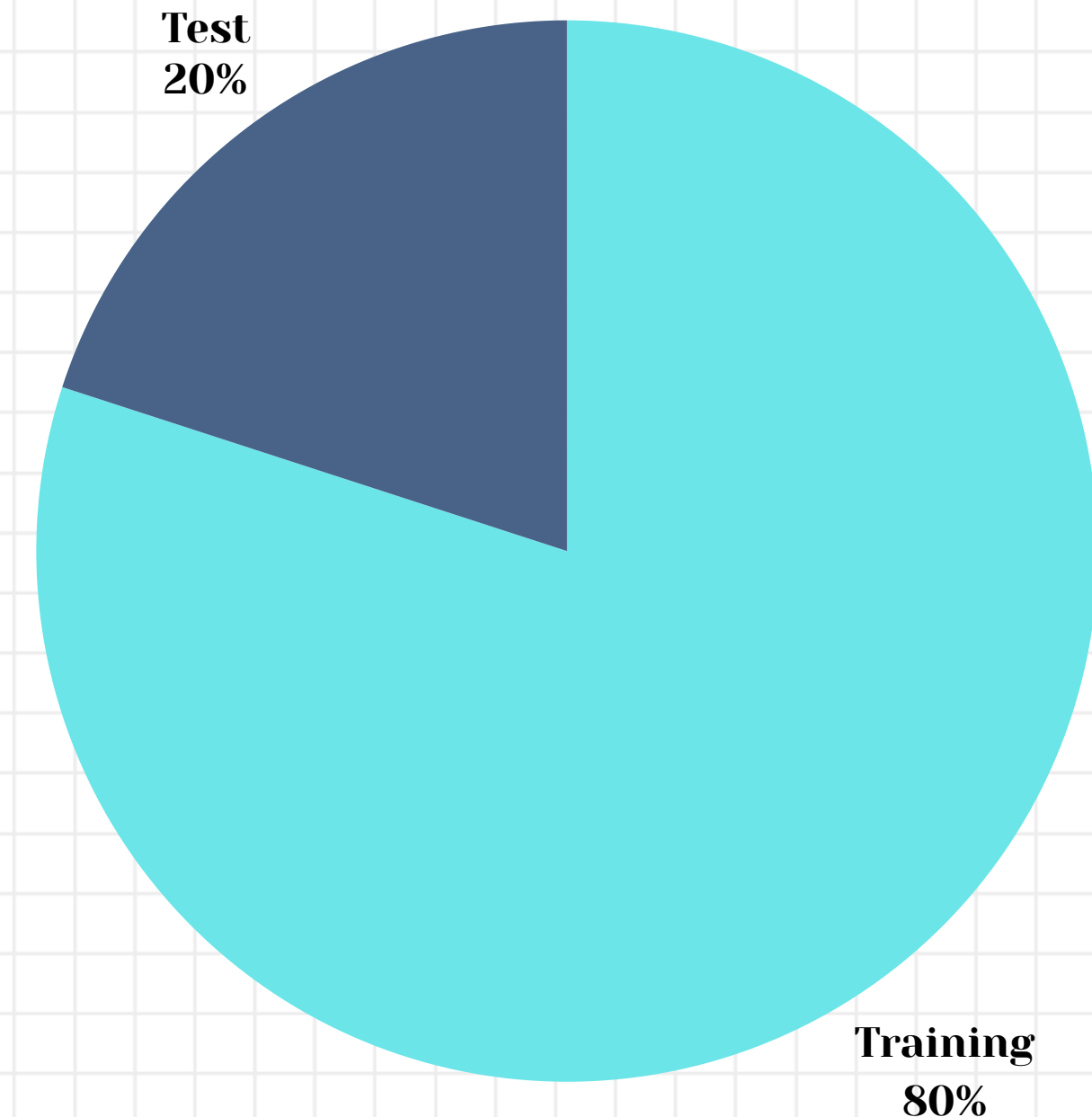
×

×

30208 parámetros para el
embedding de usuarios

53280 parámetros para el
embedding de películas

Base de datos de 100 mil datos



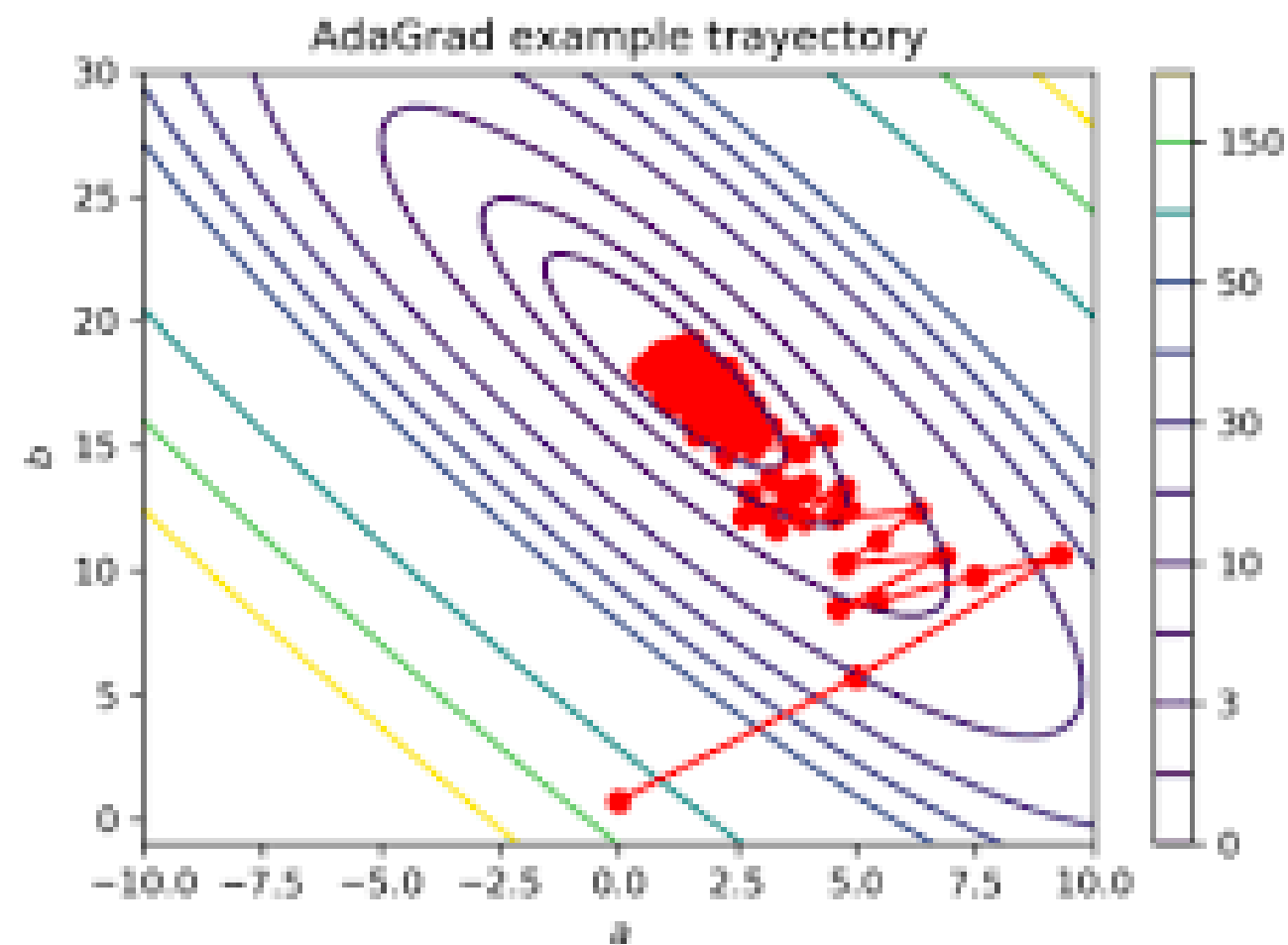
Implementación

$$\mathcal{L}(u, v) = -\log(\sigma(\langle u, v \rangle)) + \sum_{v' \notin S} \log(1 - \sigma(\langle u, v' \rangle))$$

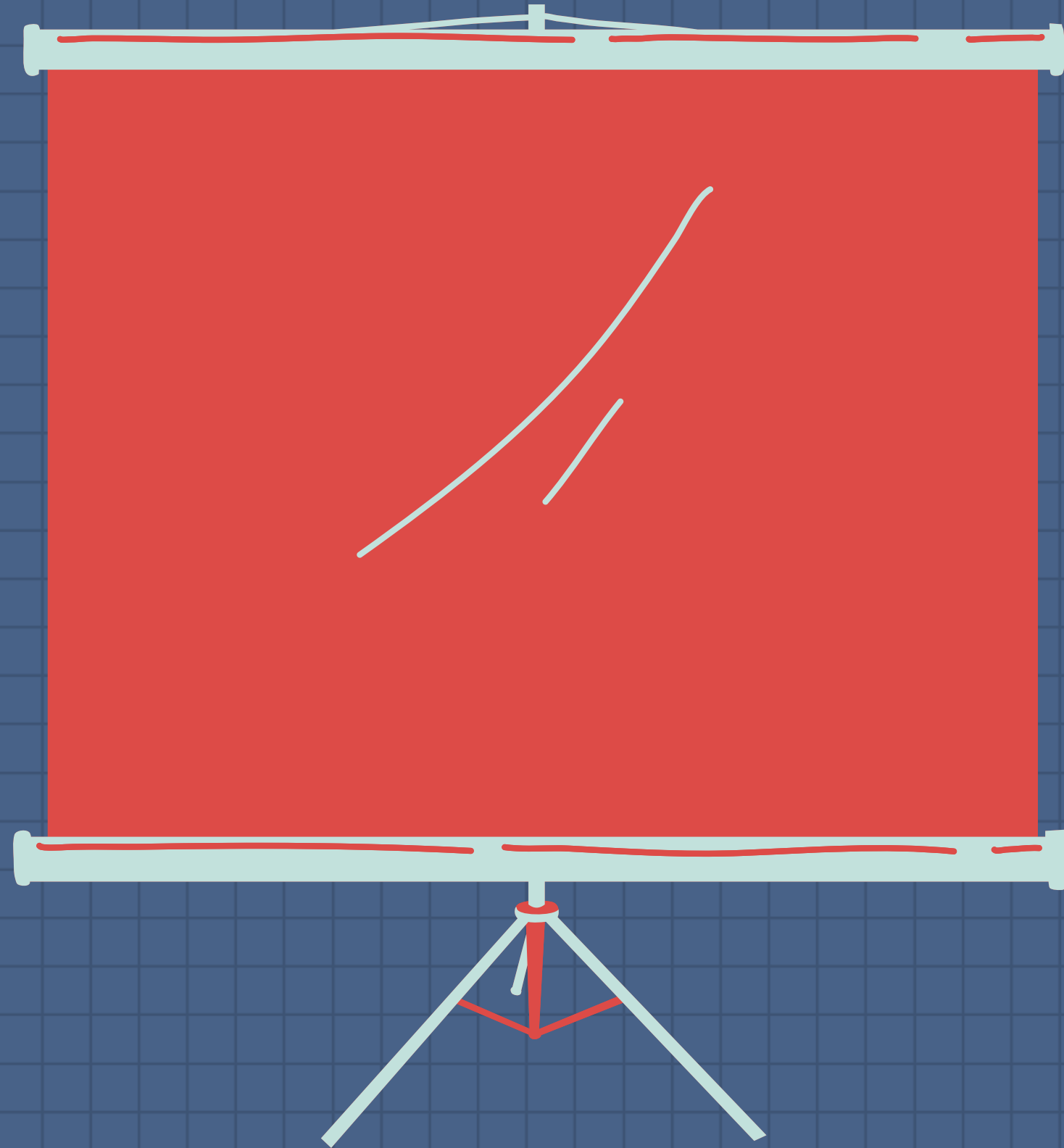
- u es el embedding del usuario.
- v es el embedding de la película que el usuario ha visto.
- v' representa los embeddings de las películas que el usuario no ha visto (negativas).
- $\langle u, v \rangle$ denota el producto punto entre los embeddings del usuario y de la película.
- σ es la función sigmoide.
- S es el conjunto de películas que el usuario ha calificado positivamente.

Implementación

La tarea es maximizar
recuperación top k



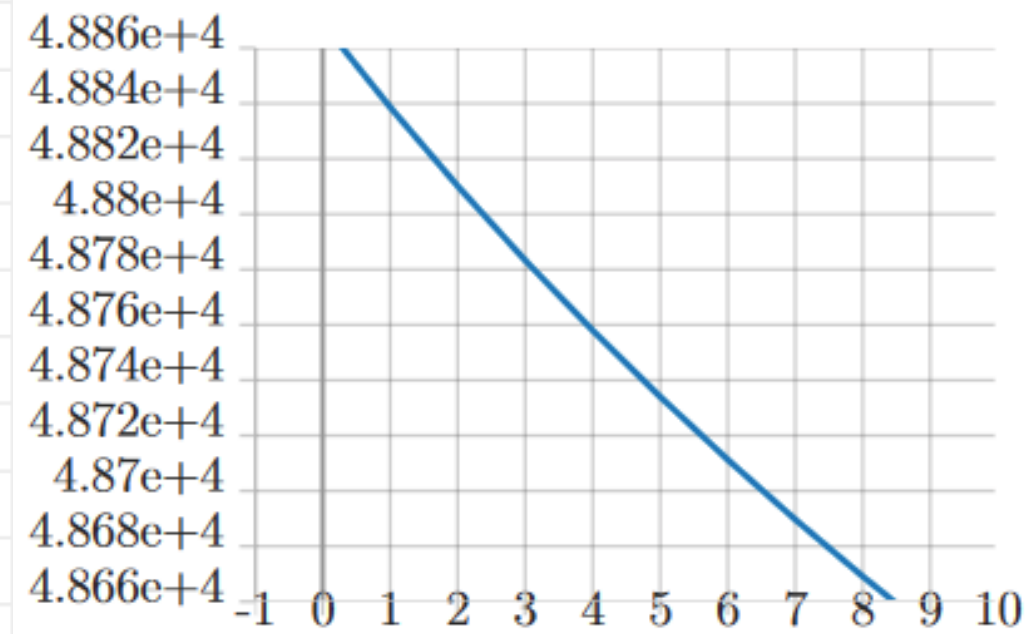
× ×



Resultados y conclusiones

× ×

Resultados



A mayor k , mayor probabilidad de encontrar la película

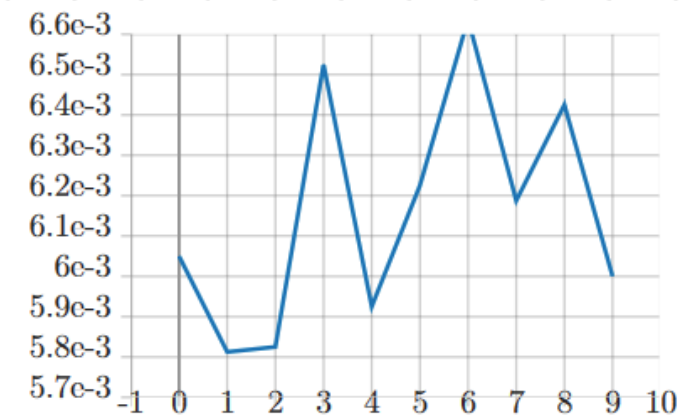


Figura 3: Recuperación top-1

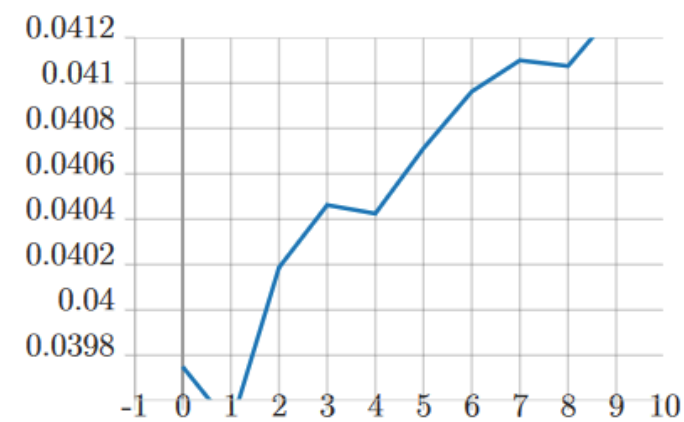
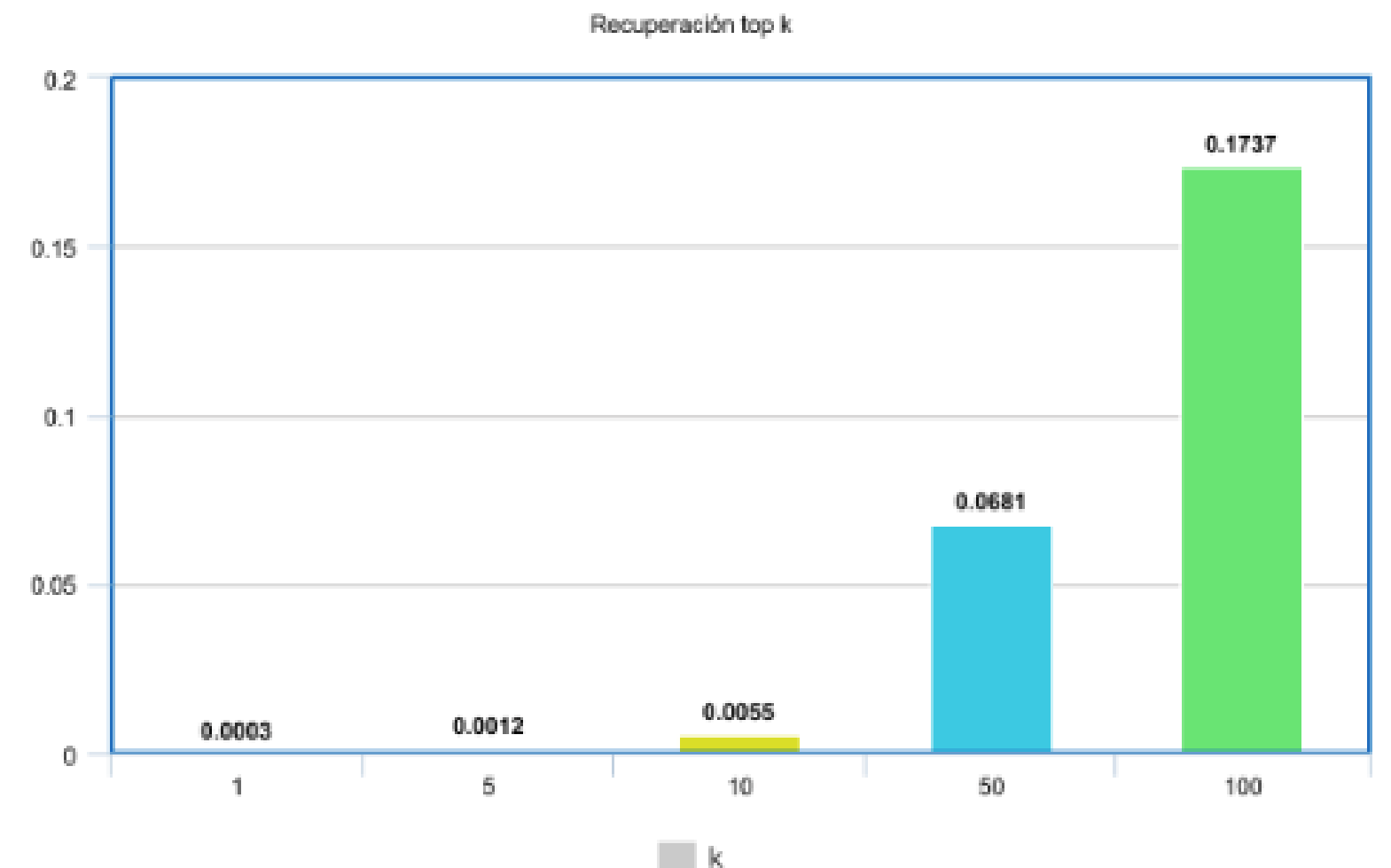
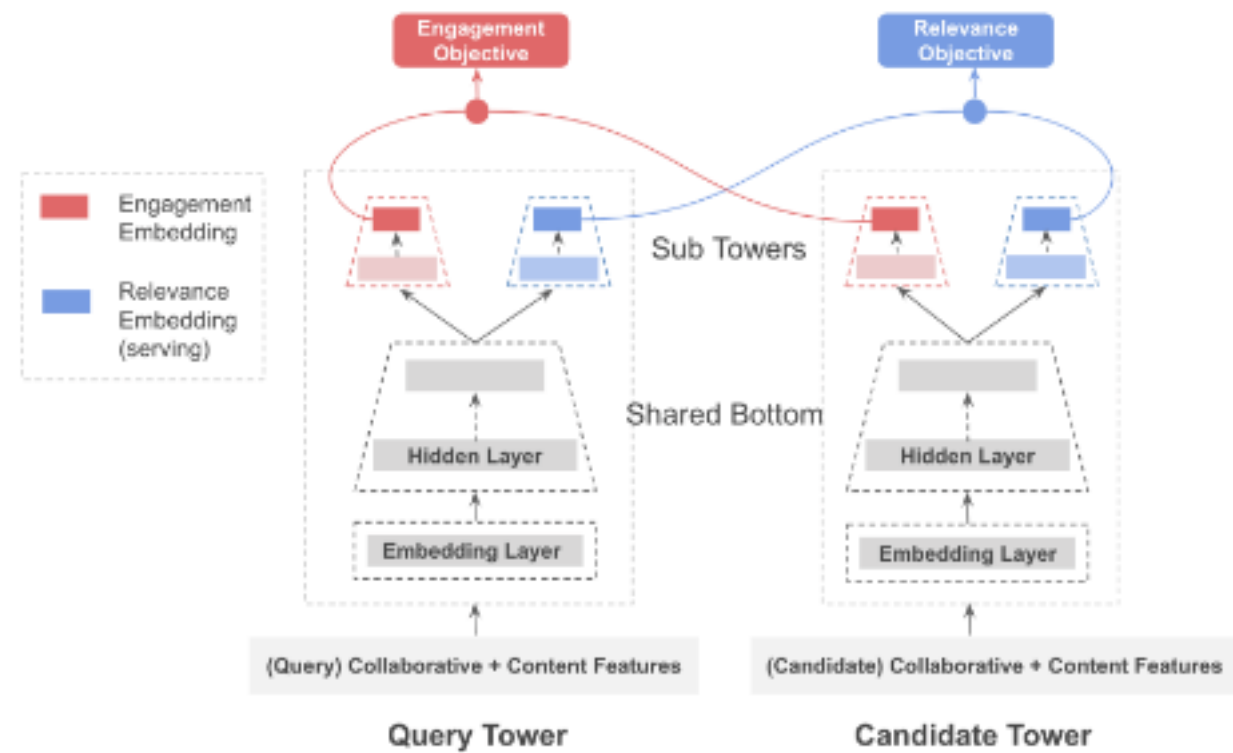


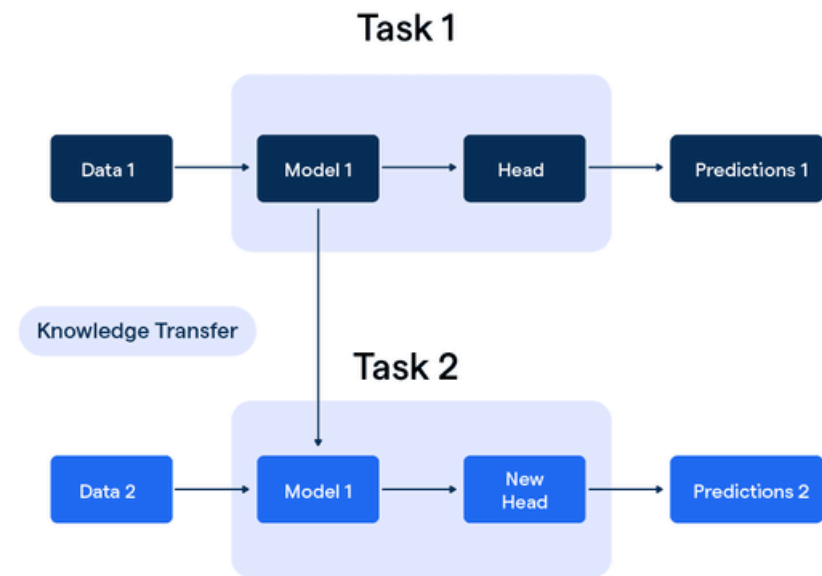
Figura 4: Recuperación top-5



Algunas posibles mejoras



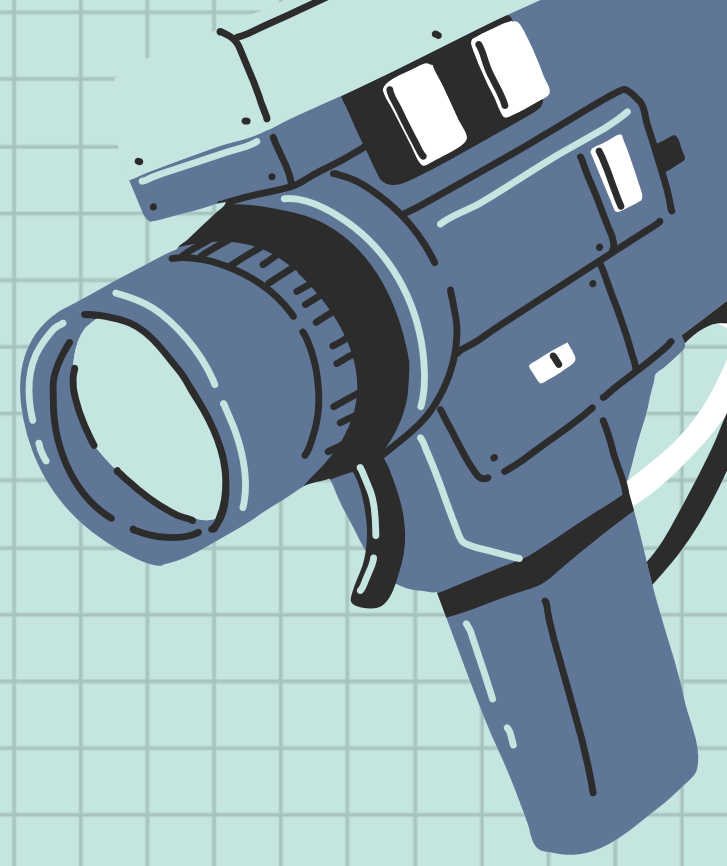
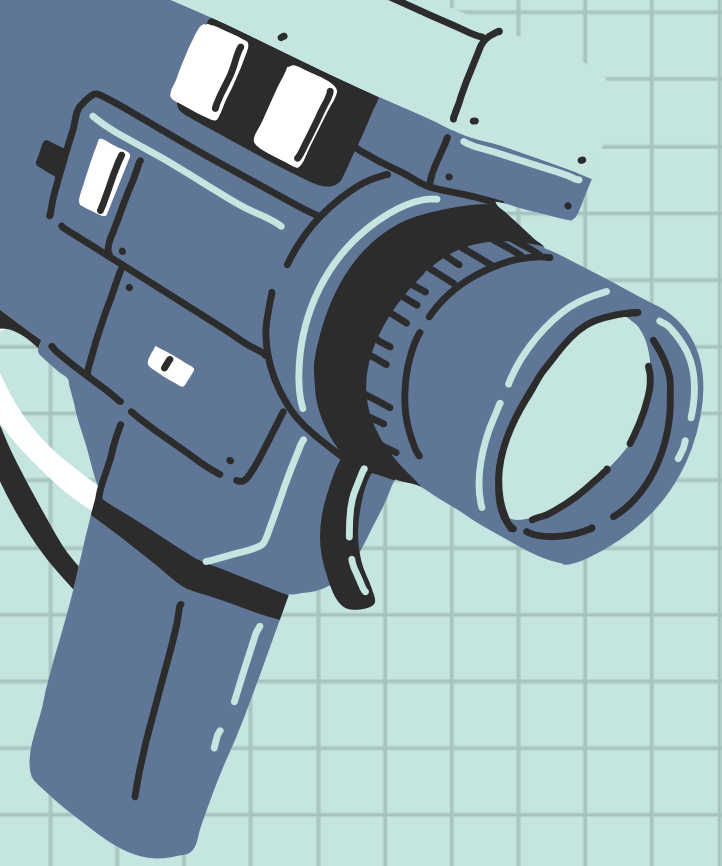
Transfer Learning



Conclusiones

- El modelo Two-Tower es útil para generar una lista de posibles películas a recomendar
- Es robusto ante grandes conjuntos de datos
- Se puede extender su funcionalidad con mas tareas
- Se puede complementar con modelos de clasificación





Gracias!