Sistema de Recomendación con Aprendizaje por Refuerzo Integrando

Ratings

# Introducción

Los sistemas de recomendación son fundamentales en plataformas de streaming de contenido, comercio electrónico, redes sociales, etc. Mejorar la personalización y la precisión de las recomendaciones es crucial para aumentar la satisfacción del usuario y la retención en la plataforma.

Este proyecto propone implementar un sistema de recomendación utilizando técnicas de aprendizaje por refuerzo (RL) que integre data real de los usuarios y calificaciones del contenido (1 a 5 estrellas).

Para ello se usará parte de la data de Netflix Prize data disponible en [Kaggle](https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data) filtrando desde 1 hasta el id\_customer 100.000 teniendo en cuenta la siguiente anotación en el repositorio "*CustomerIDs range from 1 to 2649429, with gaps. There are 480189 users.",* el objetivodel modelo es generar sugerencia del siguiente contenido a ver dada la puntuación media de usuarios a los contenidos vistos el mismo día que el contenido actual (dado que no se tiene detalle de la hora exacta en la que se vio el contenido). Teniendo como objetivo obtener la mayor recompensa posible hasta recomendar un contenido con puntuación media menor a 3 estrellas.

# Planteamiento del Problema

El reto es desarrollar un sistema de recomendación que pueda adaptarse dinámicamente a las preferencias cambiantes de los usuarios y a la vez reconciliar diferentes tipos de retroalimentación que pueden ser contradictorios, como un rating bajo acompañado de múltiples visualizaciones. La solución debe aprender de la interacción continua con los usuarios para mejorar la relevancia de las recomendaciones ofrecidas.

# Objetivos

* + **Maximizar la relevancia de las recomendaciones:** Ajustar las sugerencias para alinearlas con las preferencias explícitas e implícitas de los usuarios (las cuales pueden cambiar con el tiempo, por ende, no pueden ser estáticas).
  + **Incrementar la interacción del usuario:** Aumentar las interacciones de los usuarios con las recomendaciones a través de feedback positivo.
  + **Mejorar la precisión de la predicción:** Reducir los casos de recomendaciones no relevantes ajustando continuamente el modelo a través del aprendizaje por refuerzo.

# Métricas de Éxito

Podrían utilizarse métricas tales como tasa de clicks, tasa de Likes vs. Dislikes, y feedback de los usuarios para medir la eficacia del sistema. Sin embargo, esto se considera como una fase que está fuera del alcance de este proyecto, y por lo tanto no será incluido.

# Diseño de la Solución Ambiente

El ambiente simula la interacción en una plataforma donde cada usuario interactúa con el sistema de recomendación. Los estados son definidos por el historial de interacción de los usuarios, con el cual se pasa de un contenido a otro con determinada probabilidad (siendo esta la probabilidad de recomendarlo). De esta manera estando en un Contenido x1 se plantean otros contenidos seleccionando como el siguiente a recomendar aquel con mayor probabilidad, dicha probabilidad cambia con el feedback recibido por el agente tras las interacciones (escalado como 1 si el rating medio es 5 estrellas, 0 si es 3 estrellas y -1 si es 1 estrella).

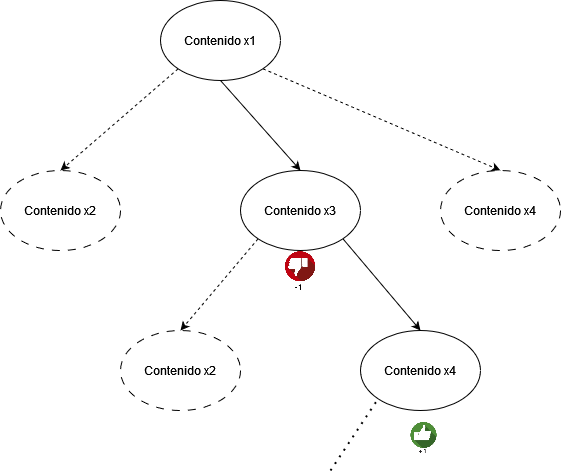


Ilustración 1 Cambios de estados (Recomendaciones)

# Estados

Dado lo anterior, cada estado estará determinado por el contenido dentro de la base de datos, teniendo como restricción no recomendar el mismo contenido que se vio, dentro de estos el agente tiene como objetivo recomendar la mayor cantidad de contenidos recibiendo refuerzos positivos durante n transiciones, evitando el estado terminal de recomendar un contenido con puntuación media inferior a 3 estrellas o 0 en la recompensa escalada.

# Acciones

El agente puede recomendar cualquier contenido que el usuario no haya visto todavía. La decisión se basa en la política aprendida, que busca maximizar las recompensas futuras según las interacciones previas, por ende el cambio constante de estado contenido actual a contenido recomendado.

# Recompensas

La función de recompensa integra rating dado como 1 a 5 estrellas, por ende esta variable del conjunto de datos se escala siendo 1 si el rating medio es 5 estrellas (máximo refuerzo positivo), 0 si es 3 estrellas y -1 si es 1 estrella (máximo refuerzo negativo), junto con todos sus valores intermedios.

# Agente de Aprendizaje

Utilizamos Q-learning con una tabla de valores Q para cada acción posible (estar en un Contenido x1 y pasar a un Contenido x2). El agente aprende de las recompensas recibidas para actualizar sus predicciones sobre qué acciones maximizarán las recompensas futuras dado el estado actual.

# Implementación

La implementación se realiza en Python utilizando bibliotecas como NumPy para las operaciones matemáticas y Pandas para la manipulación de datos. El sistema se prueba inicialmente con una muestra de los datos.

Tras ejecución de 2000 episodios y la siguiente configuración de parámetros, le toma 9 minutos y 46 segundos ejecutar el entrenamiento.

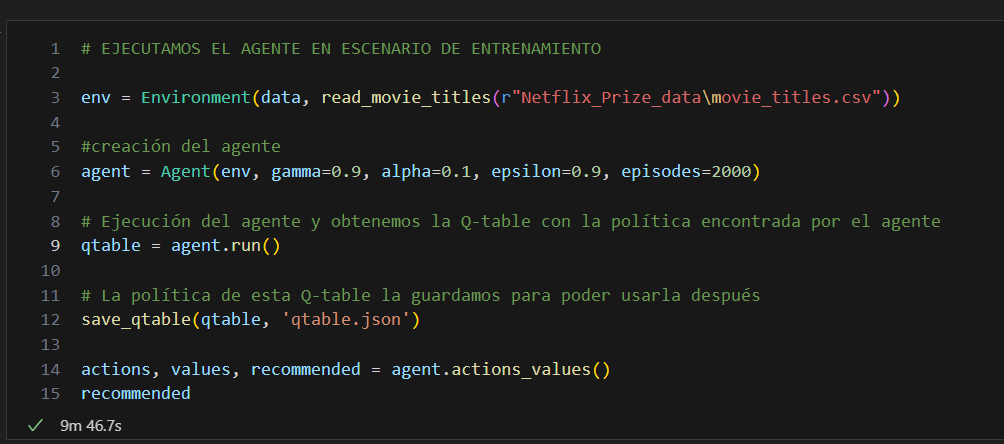


Ilustración 2 Parámetros de entrenamiento

Lo anterior ejecutado en un pc con Windows 10, procesador Ryzen 7 y 16 GB de ram, con lo cual se espera obtener el siguiente resultado:

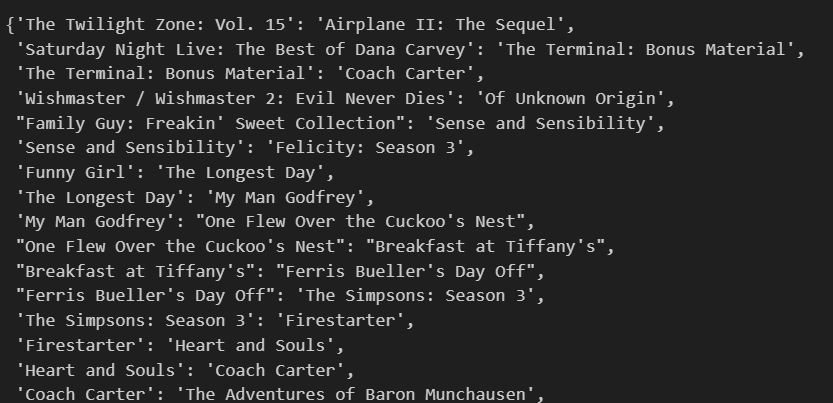


Ilustración 3 Comportamiento esperado del agente

La figura anterior muestra el comportamiento esperado del agente, retornar un diccionario el cuál como clave tiene el contenido actual y como valor el contenido con el máximo valor de la Q-table para recomendar, de igual manera el agente retorna esta información en forma de ids y de los valores aprendidos por el agente.

# Evaluación y Mejoras

La efectividad del sistema se podría evaluar mediante pruebas A/B, comparando la nueva solución con un sistema de recomendación basado en métodos tradicionales, sin embargo estaría fuera del alcance actual.