



**Universidad Tecnológica Nacional
FRRo**

**Soporte a la Gestión de Datos con
Programación Visual
Trabajo Práctico Integrador**

Recomendación de Películas con Scikit-Learn

4º Año Ingeniería en Sistemas de Información

Docentes:

Ing. Mario Castagnino

Ing. Juan Ignacio Torres

Alumnos:

49186 - Clemente Alvarez, Federico

48085 - Ferrari, Mauro Valentino

Índice

Narrativa.....	3
Requerimientos.....	3
Funcionales.....	3
No funcionales.....	3
Stack tecnológico.....	3
Reglas de negocio.....	4
Caso de uso.....	5
Precondiciones (de sistema):.....	6
Disparador:.....	6
Camino básico:.....	6
Caminos alternativos:.....	6
Postcondiciones (de sistema).....	7

Narrativa

Desarrollaremos una aplicación web que permita a los usuarios recibir recomendaciones de películas personalizadas. El sistema utiliza el algoritmo de k-Nearest Neighbors (k-NN) y se conecta con la API OMDb para obtener detalles adicionales sobre las películas recomendadas. Cada usuario puede registrarse, iniciar sesión y obtener recomendaciones en función de sus calificaciones de películas anteriores.

El objetivo es proporcionar una experiencia personalizada en la que el usuario pueda interactuar con el sistema, obteniendo recomendaciones basadas en sus preferencias. La base de datos almacena la información del usuario y las recomendaciones, mientras que el archivo `app.py` de Flask administra la lógica y funcionalidades de la aplicación.

Introducción al Algoritmo k-Nearest Neighbors (k-NN)

El algoritmo k-Nearest Neighbors (k-NN) es un método de aprendizaje supervisado que se utiliza tanto para problemas de clasificación como de regresión en machine learning. Su funcionamiento se basa en la idea de que "los objetos similares se encuentran cerca". Para realizar una predicción, el algoritmo evalúa la proximidad de un nuevo punto de datos a puntos existentes en el espacio de características y determina su etiqueta o valor en función de las etiquetas de sus vecinos más cercanos.

En el proceso de recomendación de películas, k-NN resulta especialmente útil para identificar películas que se asemejan a las preferencias del usuario. Por ejemplo, a partir de la calificación que un usuario ha dado a ciertas películas, el algoritmo encuentra títulos similares en base a características como las opiniones de otros usuarios, género, título y el año de lanzamiento. Estas características se representan en un espacio de múltiples dimensiones donde cada película es un punto. El algoritmo busca las "k" películas (vecinas) más cercanas a las preferencias del usuario, evaluando la similitud entre sus perfiles, y selecciona las más relevantes para recomendar.

En este caso, se utiliza la distancia del coseno como métrica de similitud, permitiendo evaluar qué tan alineadas están las preferencias del usuario con cada película. Así, k-NN permite crear una lista de recomendaciones personalizadas, adaptadas a los gustos y valoraciones individuales del usuario.

Requerimientos

Funcionales

1. El sistema debe permitir que los usuarios se registren e inicien sesión.
2. El sistema debe ser capaz de seleccionar aleatoriamente cinco películas y presentarlas para que el usuario las clasifique.

3. Las recomendaciones deben basarse en las preferencias de los usuarios y deben almacenarse en la base de datos.
4. El sistema debe conectarse con la API OMDb para obtener detalles de las películas.
5. Se debe mostrar al usuario una lista de tres recomendaciones basadas en sus calificaciones previas.

No funcionales

1. La aplicación debe estar desarrollada en Python y utilizar Flask como framework web.
2. La base de datos debe ser SQLite para facilitar el almacenamiento de usuarios y recomendaciones.
3. El sistema debe ser compatible con PCs con sistemas operativos Windows y Linux.
4. Las recomendaciones deben generarse en tiempo real con baja latencia, usando el algoritmo k-NN.

Stack tecnológico

1. **Flask:** Framework para el desarrollo web en Python que permite la creación de aplicaciones de servidor.
Uso en el proyecto: Maneja las rutas de la aplicación, renderiza plantillas HTML, maneja la autenticación de usuarios (registro e inicio de sesión), y proporciona endpoints para generar recomendaciones personalizadas. También permite el uso de sesiones para el seguimiento de los usuarios que han iniciado sesión.
2. **Pandas:** Biblioteca de Python que facilita la manipulación de datos en estructuras tabulares.
Uso en el proyecto: Carga y procesa los datos de las películas y calificaciones almacenados en archivos CSV. Los DataFrames de Pandas permiten estructurar los datos y prepararlos para el procesamiento de recomendaciones.
3. **SQLite:** Sistema de base de datos relacional ligero y de código abierto que no requiere un servidor separado.
Uso en el proyecto: Almacena la información de usuarios y recomendaciones, permitiendo la persistencia de datos entre sesiones de usuario. Las tablas de `users` y `recommendations` permiten manejar la autenticación y personalización de recomendaciones.
4. **Scikit-Learn:** Biblioteca de Python que ofrece herramientas para el aprendizaje automático.
Uso en el proyecto: Utiliza el algoritmo de **k-nearest neighbors (k-NN)** a través de `NearestNeighbors` para generar recomendaciones de películas basadas en las preferencias de cada usuario. `MultiLabelBinarizer` se emplea para procesar los géneros de películas, transformando listas de géneros en vectores binarios,

facilitando el cálculo de similitudes.

5. **NumPy**: Biblioteca de Python que permite realizar cálculos numéricos de alto rendimiento y manipulación de matrices.
Uso en el proyecto: Apoya las operaciones de Pandas y Scikit-Learn, y se utiliza para seleccionar aleatoriamente películas para calificación. Los cálculos matriciales son esenciales para el algoritmo de k-NN que determina las películas más similares.
6. **OMDb API**: API pública que proporciona información sobre películas, como títulos, años, pósters y sinopsis.
Uso en el proyecto: Se utiliza para enriquecer las recomendaciones de películas, obteniendo detalles adicionales que no se encuentran en los archivos locales, como descripciones y carátulas.
7. **HTML y CSS**: Herramientas estándar para crear interfaces web.
Uso en el proyecto: Estructuran y presentan las interfaces de usuario, como el registro, inicio de sesión, panel de control y la sección de recomendaciones, facilitando una experiencia de usuario interactiva y atractiva.

Reglas de negocio

El sistema de recomendación de películas sigue un conjunto de reglas que garantizan la personalización, precisión y eficiencia en la generación de recomendaciones y la gestión de datos de usuarios. Las principales reglas de negocio implementadas son:

- **Autenticación de usuarios**: Cada usuario debe registrarse y autenticarse para poder acceder al sistema. Esto garantiza que las recomendaciones generadas sean personalizadas y asociadas de manera segura a cada usuario específico, protegiendo la privacidad de sus preferencias y calificaciones.
- **Recomendación sin duplicados**: El sistema evita recomendar al usuario películas que ya ha calificado. Las recomendaciones solo incluyen títulos que el usuario no ha visto, optimizando la personalización y fomentando la exploración de nuevas opciones.
- **Calificación de películas**: Antes de recibir recomendaciones personalizadas, el usuario debe calificar al menos cinco películas. Esta calificación inicial es necesaria para establecer un perfil básico de preferencias, permitiendo que el sistema identifique tendencias en géneros y años, lo cual mejora la calidad de las recomendaciones.
- **Consulta a OMDb API**: Las películas recomendadas incluyen detalles adicionales (como sinopsis, año de lanzamiento y póster) obtenidos a través de la API OMDb. Esto enriquece la experiencia del usuario, presentando una descripción visual y contextual de cada título recomendado.
- **Persistencia de recomendaciones**: Cada recomendación generada se guarda en la base de datos, permitiendo al usuario consultar su historial de recomendaciones y calificaciones. Esto facilita un registro a largo plazo de sus preferencias y permite futuras personalizaciones basadas en el historial de interacción.

- **Filtro de películas aleatorias:** El sistema ofrece cinco películas aleatorias para que el usuario las califique antes de generar recomendaciones personalizadas. Esto permite al sistema adquirir una variedad de datos iniciales y comprender mejor las preferencias del usuario desde el comienzo.
- **Limitación de recomendaciones:** Al momento de generar recomendaciones, el sistema muestra hasta tres títulos a la vez. Este enfoque ayuda a mantener una interfaz clara y manejable, evitando sobrecargar al usuario con demasiadas opciones y asegurando una presentación de recomendaciones gradual y enfocada.

Caso de uso

Generación y presentación de recomendaciones de películas personalizadas para el usuario.

Nivel	Estructura	Alcance	Caja	Instanciación	Interacción
Resumen	Sin estructurar	Sistema	Negra	Real	Semántico

Meta del caso de uso: Proporcionar recomendaciones de películas basadas en las calificaciones del usuario, utilizando opiniones de otros usuarios, sus preferencias de género y otros criterios.

Actor primario: Usuario autenticado.

Otros: Base de datos de usuarios y recomendaciones, API de OMDb.

Precondiciones (de sistema):

- El usuario está registrado.
- Existen suficientes datos de calificaciones para generar recomendaciones basadas en sus preferencias.
- El sistema está conectado a Internet para acceder a la API de OMDb y obtener detalles de películas.

Disparador:

El usuario accede a la página de inicio de sesión de la aplicación.

Camino básico:

1. El usuario inicia sesión en el sistema y accede a su panel de control.
2. El usuario selecciona la opción "Recomendar" para ver películas a calificar.
3. El usuario califica las películas mostradas de forma aleatoria.

4. Utilizando el algoritmo k-Nearest Neighbors, el sistema calcula y selecciona tres películas recomendadas basadas en la similitud de preferencias (teniendo en cuenta las opiniones de otros usuarios almacenadas en el archivo ratings.csv, el género, título y año de lanzamiento de las películas).
5. Para cada recomendación, el sistema consulta la API de OMDb para obtener información adicional de la película (título, año, sinopsis y póster).
6. El sistema presenta al usuario la lista de tres películas recomendadas junto con los detalles adicionales obtenidos.
7. El usuario puede ver los detalles de cada recomendación y de otras previas en "Mis recomendaciones".

Los pasos 3 - 7 se repiten cada vez que el usuario solicita nuevas recomendaciones, manteniendo el perfil de recomendaciones actualizado.

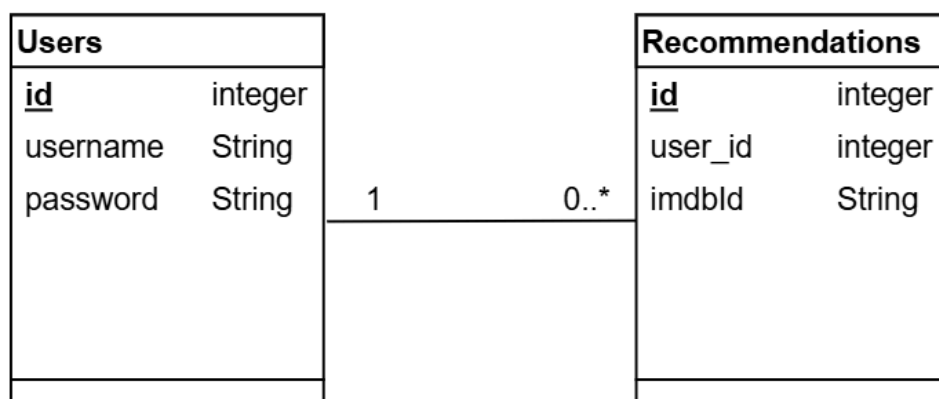
Caminos alternativos:

5.1 <durante> Si el sistema no puede acceder a la API de OMDb, muestra la lista de recomendaciones utilizando únicamente los datos locales de la base de datos. 5.2

Postcondiciones (de sistema)

- **Éxito:** El sistema ha generado y mostrado una lista de recomendaciones de películas personalizadas con detalles adicionales obtenidos de la API de OMDb.
- **Fracaso:** El sistema no pudo generar recomendaciones debido a una falta de calificaciones suficientes por parte del usuario o a un error de conexión con la API.
- **Éxito alternativo:** El sistema presenta recomendaciones utilizando datos locales cuando no se puede acceder a la API de OMDb.

Modelo de Dominio



En la Base de datos se almacenan dos tablas, los Usuarios en la tabla "Users" y la tabla "Recomendations" que contiene las distintas recomendaciones de cada usuario.

Conclusiones

La implementación del sistema de recomendaciones de películas ha demostrado ser una solución efectiva para proporcionar una experiencia personalizada a los usuarios. Al integrar el algoritmo k-Nearest Neighbors, logramos un sistema capaz de identificar patrones en las preferencias de los usuarios y sugerir películas relevantes basadas en estas. Además, el uso de la API de OMDb permite enriquecer cada recomendación con información adicional, mejorando la experiencia de usuario al ofrecer detalles visuales y contextuales sobre cada película.

Este proyecto ha permitido explorar los beneficios de combinar machine learning con datos externos en tiempo real. La utilización de k-NN para recomendaciones personalizadas destaca por su simplicidad y eficacia en aplicaciones donde los datos pueden representarse en términos de proximidad, como es el caso de las preferencias cinematográficas. Aunque el sistema en su forma actual presenta resultados satisfactorios, futuras mejoras podrían incluir la integración de otros algoritmos de recomendación, como el filtrado colaborativo, para optimizar aún más la precisión y relevancia de las recomendaciones.

En conclusión, este trabajo no solo demuestra la aplicabilidad de k-NN en sistemas de recomendaciones, sino que también subraya la importancia de interfaces adaptativas y basadas en datos para mejorar la experiencia del usuario en entornos digitales.