Proyecto de Recomendación de Películas

# 1. Aprendizaje

Para este proyecto utilicé el conjunto de datos MovieLens 100k, que contiene 100.000 valoraciones realizadas por 943 usuarios sobre 1682 películas. Este dataset fue procesado para generar dos enfoques de recomendación: filtrado colaborativo y filtrado basado en contenido.  
  
- Filtrado colaborativo (SVD): Se entrenó un modelo basado en descomposición en valores singulares (SVD) usando la librería `surprise`. Este modelo predice valoraciones para películas no vistas por un usuario, basándose en patrones de comportamiento de otros usuarios.  
- Filtrado basado en contenido: Se representaron las películas como vectores de géneros (codificación one-hot) y se calculó la similitud del perfil de un usuario (promedio de películas que le gustaron) con las demás películas usando similitud de coseno.  
  
El código de entrenamiento y exploración está disponible en el repositorio de GitHub: https://github.com/usuario/mlip-project

# 2. Comparación de Modelos

Para comparar ambos modelos se utilizaron las siguientes cuatro métricas:  
  
1. Precisión de predicción  
 - Métrica: RMSE y MAE  
 - Datos: División 80/20 de train/test sobre MovieLens 100k  
 - Operacionalización: evaluación de valoraciones predichas vs reales  
  
2. Costo de entrenamiento  
 - Métrica: Tiempo de entrenamiento (segundos)  
 - Datos: Medición de tiempo de ejecución al entrenar cada modelo  
 - Operacionalización: uso de time.time() al inicio y fin del entrenamiento  
  
3. Costo de inferencia  
 - Métrica: Tiempo en milisegundos para generar recomendaciones para un usuario  
 - Datos: Medición de tiempo por petición HTTP en el API  
 - Operacionalización: logging desde Flask + Kafka  
  
4. Tamaño del modelo  
 - Métrica: Peso en disco (KB/MB)  
 - Datos: Archivo `.pkl` del modelo  
 - Operacionalización: uso de `os.path.getsize()`

Tabla de comparación de modelos:

|  |  |  |
| --- | --- | --- |
| Métrica | Filtrado Colaborativo (SVD) | Filtrado Basado en Contenido |
| RMSE | 0.936 | N/A |
| MAE | 0.738 | N/A |
| Tiempo de entrenamiento | Bajo (unos segundos) | Muy bajo |
| Tiempo de inferencia | < 800 ms | < 100 ms |
| Tamaño del modelo | ~1 MB | No requiere modelo |

Modelo elegido: Se selecciona el modelo de filtrado colaborativo para producción por su capacidad para predecir valoraciones y aprender patrones latentes de preferencia.

# 3. Servicio de Predicción

El servicio fue implementado usando Flask y expuesto en el endpoint `/recommend/<user\_id>`. Al recibir una petición:  
- Carga el modelo SVD desde archivo `.pkl`  
- Identifica las películas no vistas por el usuario  
- Predice valoraciones y ordena las películas por preferencia estimada  
- Devuelve hasta 20 IDs de películas separados por comas  
  
El servicio responde en menos de 800ms y emite logs en Kafka con el siguiente formato:  
<timestamp>,<user\_id>,recommendation request <hostname>, status 200, result: <ids>, <responsetime>  
  
El código está disponible en `src/recommender\_api.py`