**Sistema de Recomendación de Películas**

**Entregables**

* Este documento actualizado con la información requerida en cada ítem. Indicar fuentes bibliográficas utilizadas.
* El dump de la BD recommendations.db (para versión 5.23) que queda luego de ejecutar todo lo pedido. Verificar que el dump subido puede cargarse sin problemas desde cero.
* Si para el ítem a.2 usaron algún archivo, subirlo también.

**Parte A**

La BD **recommendations.db,** importada en clase, se completará con más información proveniente del archivo **newfile.csv** (bajar de Campus).

Dicho archivo contiene **usuarios y su calificación a ciertas películas.** Las películas a las que ellos referencian **no necesariamente existen en recommendations.db**.

Cada línea del csv contiene:

* **userID**: string identifica a los usuarios los cuales están anonimizados
* **ttmovieID**: string compuesto por “tt” + IMDB ID de una película
* **imdbID** : string que contiene el IMDB ID de una película ( o sea, la parte numérica del atributo anterior)
* **rating**: string que contiene la calificación que userID le dio a la película
* **review\_date**: string que contiene la fecha en que el usuario emitió su rating.

Ejemplo de línea:

userID,ttmovieID,imdbID,rating,review\_date

ur4592644,tt0120884,0120884,10,16 January 2005

La siguiente expresión, si bien cargaría los datos del csv a la base recommendation.db no lo haría en forma correcta porque hay múltiples usuarios repetidos en el archivo (porque emitieron varios ratings) y además porque no está creando el eje del usuario a las películas ya existentes.

Esta sentencia estaría**, erróneamente**, creando un usuario distinto por cada fila en el csv y colocaría el dato de la película y su rating como propiedades embebidas:

:auto LOAD CSV WITH HEADERS  FROM 'file:///newfile.csv' AS row

CALL {

  WITH row

  CREATE (c:User {userID: row.userID, ttmovieID: row.ttmovieID, imdbID: row.imdbID, rating: row.rating, review\_date: row.review\_date})

} IN TRANSACTIONS  OF 5000 ROWS

Más allá de eso, LOAD CSV precisa que (ver detalles <https://neo4j.com/docs/cypher-manual/current/clauses/load-csv/>)

1. el archivo debe estar en el directorio “import” (o subdirectorios a partir de él) donde se descompactó Neo4j (Ej: D:\software\neo4j-community-5.23.0\import). No path absoluto.
2. En neo4j.conf se setea el directorio default **server.directories.import=import** (siempre relativo). Si hace falta, descomentar la línea y levantar la BD.

**Se pide:**

**a.1)** Escribir **la o las sentencias Cypher (en la secuencia en que hay que ejecutarlas)** que permiten cargar las filas del CSV en **recommendationDB en forma correcta**. Tener en cuenta que los usuarios con mismo userID no deben pertenecer a varios vértices. Además, debe crearse un eje con su respectivo rating desde el usuario a las películas existentes en recommendations.db y no debe ser propiedades del nodo usuario. Considerar, si es necesario, conversiones a tipos de datos correctos en la creación de nodos y ejes. Si es pertinente, se pueden crear índices.

**Respuesta a.1**

**a.2)** Si existieran películas referidas por usuarios que no existen en recommendations.db tomar alguna de las siguientes decisiones y escribirla en el informe claramente:

Opción a.2.1) No colocar esas películas porque no se les conoce más que un ID (no se puede recomendar porque no se posee ni el título). En ese caso, indicar cuáles son las sentencias Cypher que hay que ejecutar para que no queden movies sin información de títulos.

Opción a.2.3) Completar con información a esas películas que no existían en recommendations.db (enriquecer con title, etc). Indicar todas las sentencias Cypher para lograr esto y si hace falta proveer un archivo adicional, entregarlo.

También pueden combinarlas, es decir, una cierta cantidad de películas las enriquecen y otras no las incluyen.

**Respuesta a.2**

**a.3)** Exportar en este momento en formato dump la BD con los cambios previos. Escribir la sentencia de línea de comandos utilizada para generar el dump

**Respuesta a.3**

**Parte B**

Con los datos del archivo csv ya cargados se pide:

**b.1)** **Crear un nuevo usuario con el nombre del grupo** (que representa las preferencias de algún/os integrante/s del grupo, por supuesto que es un usuario ficticio y no necesariamente tiene que contener las verdaderas preferencias de los integrantes. Es nuestro **usuario objetivo**) el cual debe calificar una colección de las películas en la BD. Escribir la/s sentencia/s Cypher para producir ese usuario y sus preferencias.

**Respuesta b.1**

**b.2)** Se deberá utilizar ese **nuevo usuario objetivo creado en el ítem b.1** y encontrar **usuarios similares** basándose en los **ratings a las películas coincidieron en mirar**.

Sin embargo, tener en cuenta que, si no hay un número interesante de películas coincidentes, no se obtendrá un buen resultado (excluir candidatos con coincidencia en bajo números de películas). Ej: usuario1 vio 100 películas, usuario2 vio 30 películas pero solo tienen 2 películas coincidentes, la recomendación no va a ser buena porque no tiene preferencias similares. Analizar y proponer ese “umbral” conveniente.

Para detectar los usuarios **más similares** utilizar (ver la descripción en Cypher. No son procedimientos): euclideanDistance, euclidean, pearson, overlap, jaccard and cosine. Si alguna de las métricas no puede aplicarse, indicarlo claramente indicando el por qué.

Finalmente, se debe producir la lista de las películas que se le recomendaría al nuevo usuario objetivo porque los K usuarios más similares las han visto y él aún no (indicar el valor K elegido y justificar su elección). Deben aparecer ordenadas correctamente. Considerar que si una película aparece muchas veces (recomendada por varios usuarios) su rating debe calcularse como el promedio de estos para posicionarla en la respuesta.

Escribir todas las sentencias Cypher que hay que ejecutar para obtener las similitudes y recomendaciones pedidas para cada una de las métricas:

**Respuesta a.2**

* **para métrica euclideanDistance**
* **para métrica euclidean**
* **para métrica Pearson**
* **para métrica overlap**
* **para métrica jaccard**
* **para métrica cosine**

**b.3)** Para poder evaluar cuál de las métricas anteriores resultó mejor hay que evaluarlas lo más objetivamente posible. Proceder de la siguiente manera:

Elegir uno o varios usuarios y quitarle alguna de sus puntuaciones a cierta película/s. Indicar con Cypher qué se eliminó.

Ejecutar los algoritmos anteriores y medir alguna o varias métricas (usando la información que han eliminado): por ejemplo recall, precisión, error cuadrático medio, etc. Realizar un análisis de los resultados obtenidos y ver si aparece en la recomendación la/s eliminadas que sabíamos que la había elegido.

**Respuesta b.3**