Universidad Pontificia Comillas

Bases de datos

Proyecto final
2023-2024

Carlos Martínez Cuenca, 202211532
Lydia Ruiz Martínez, 202213363

Resumen

En esta memoria se presenta el proyecto final de la asignatura de Bases de Datos, realizado por Carlos Martínez Cuenca y Lydia Ruiz Martínez.

La resolución del proyecto se ha implementado en Python. Este proyecto consta de los siguientes elementos de desarrollo software:

- Módulos: `configuracion.py`, `load_data.py`, `menu_visualizacion.py`, `queries.py`, `neo4JProyecto.py`, `inserta_dataset.py` y `load_data_PBi.py`.
- 1 archivo: `visualizacion.pbix`.
- 1 fichero de texto: `requirements.txt`, que especifica las dependencias (librerías de python no estándar) del proyecto.

Índice

1. Diseño de datos

Cada uno de los ficheros origniales de los datos constan de 9 columnas: “reviewerID”, “asin”, “reviewerName”, “helpful”, “reviewText”, “overall”, “summary”, “unixReviewTime” y “reviewTime”.

Para almacenar los datos de todos estos ficheros crearemos dos bases de datos, una en MySQL y otra en MongoDB, ambas seguirán el nombre de “Reviews”.

Las columnas “reviewerID” y “reviewerName” proporcionan un identificador al usuario que realizó la reseña del producto, entre estas dos columnas existe una dependencia funcional completa por lo que almacenaremos en MySQL una tabla que contenga cada identificador y su nombre de usuario. Esta tabla que llamaremos “reviewers” nos permite tener un registro sencillo de todos los usuarios que han realizado alguna reseña sobre cualquier producto, además, tendrá como clave principal la columna “reviewerID”.

La columna “asin’ representa un identificador de cada uno de los productos de los ficheros de datos, sin embargo, este identificador puede repetirse en ficheros distintos (como es el caso de B00002DHEV en el fichero de “Video Games” y “Toys and Games”), por lo que para que cada producto tenga un verdadero identificador crearemos uno propio. Además, para seguir un buen registro de cada uno de los productos, almacenaremos el nuevo identificador, el asin y tipo de producto (si se trata de música, videojuegos, etc) en una nueva tabla de MySQL llamada “items”. La clave principal de esta tabla será el identificador único que hemos creado.

La información del resto de columnas hemos decidido almacenarla en MongoDB, cada uno de los ficheros tendrá su propia colección dentro de una base de la base de datos y seguirá el mismo nombre que el fichero. En cada elemento de las colecciones se almacenarán las columnas: “reviewerID”, “asin”, “helpful”, “reviewText”, “overall”, “summary”, “unixReviewTime” y “reviewTime”. Todas estas columnas son propias de cada review y tiene más sentido almacenarlas en una base de datos no relacional. Cada instancia tendrá un nuevo identificador único el cual es creado automáticamente por MongoDB.

Siguiendo esta estructura, tenemos en MySQL una manera eficaz de identificar los usuarios y los productos, y en MongoDB tenemos cada una de las características de las reviews almacenadas por tipo de producto.

1.1. Esquema relacional de MySQL

- REVIEWERS(<u>ID</u>, Name)
- PRODUCTS(<u>ID</u>, Type)
- ITEMS(<u>ID</u>, Asin, Type)
    - Type FK REFERS PRODUCTS(ID)

1.2. Datos de MongoDB

Cada tipo de objeto tiene una colección en MongoDB asociada con estos datos:

- reviewerID (String)
- asin (String)
- helpful (Array)
- overall (Int)
- summary (String)
- reviewText (String)
- reviewTime (Timestamp)
- unixReviewTime (Int)

2. Aplicación para el acceso y visualización de datos

En esta parte del proyecto se ha creado un fichero Python llamado “menu_visualizacion.py” para poder obtener diferentes plots de visualización de diferentes datos. Se ha implementado un menú en python mediante la librería gráfica Tkinter. Cabe destacar que también se ha usado otro archivo llamado “queries.py” que contiene todas las funciones de las consultas de MongoDB y de los plots que se llaman en “menu_visualizacion.py”.

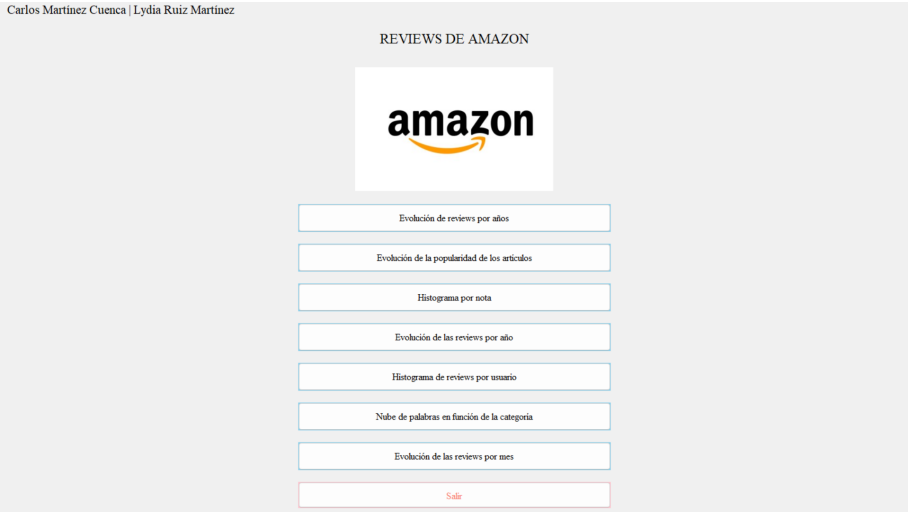

Figura 1: Interfaz gráfica del menú

La primera opción del menú muestra la evolución de las reviews por año en función del tipo de producto.

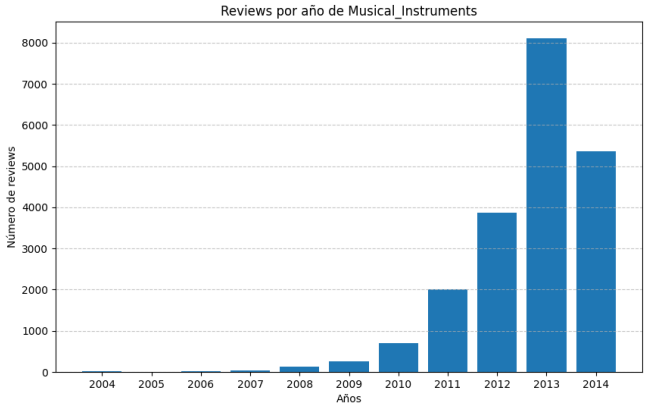

Figura 2: Evolución de las reviews

La segunda opción del menú muestra la evolución de los artículos en función del tipo de producto, ordenados de mayor a menor popularidad.

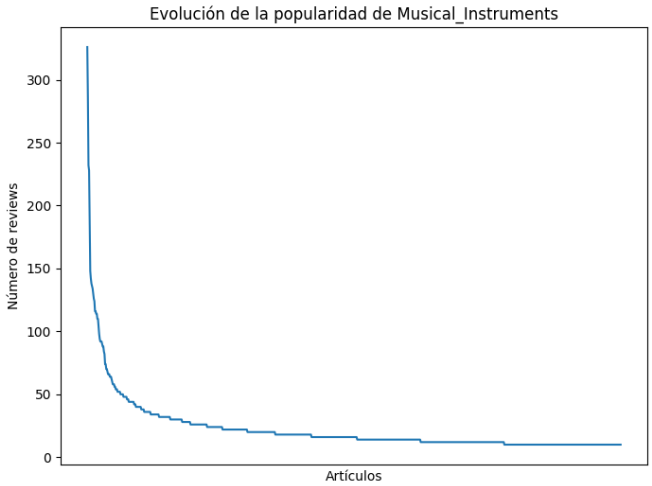

Figura 3: Evolución de los artículos

La tercera opción del menú obtiene un histograma por número de nota en función del tipo de producto.

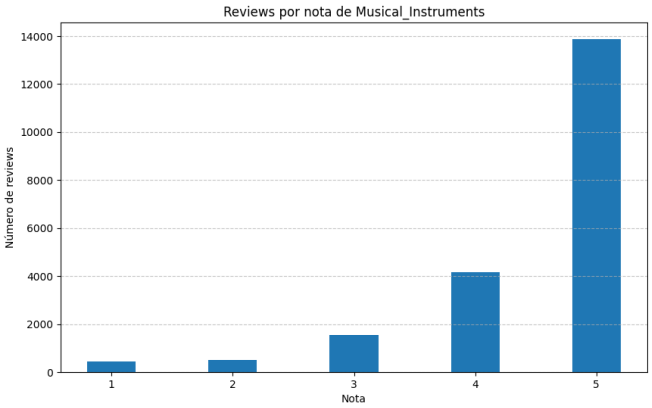

Figura 4: Histograma por nota

La cuarta opción del menú muestra la evolución de las reviews a lo largo del tiempo en función del tipo de producto.

Figura 5: Evolución de las reviews

La quinta opción del menú muestra un histograma de reviews por usuario.

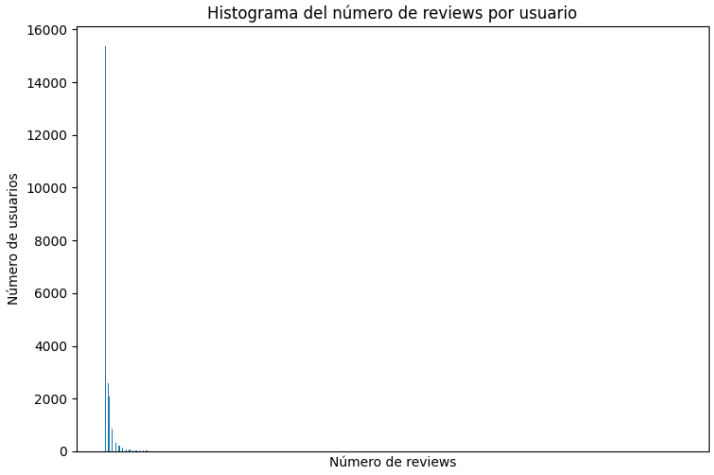

Figura 6: Histograma de reviews

La sexta opción del menú una nube de palabras en función del tipo de producto.

Figura 7: Circuito de acondicionamiento del servomotor

La última opción del menú ha sido de elección propia y se trata de un histograma de reviews por mes.

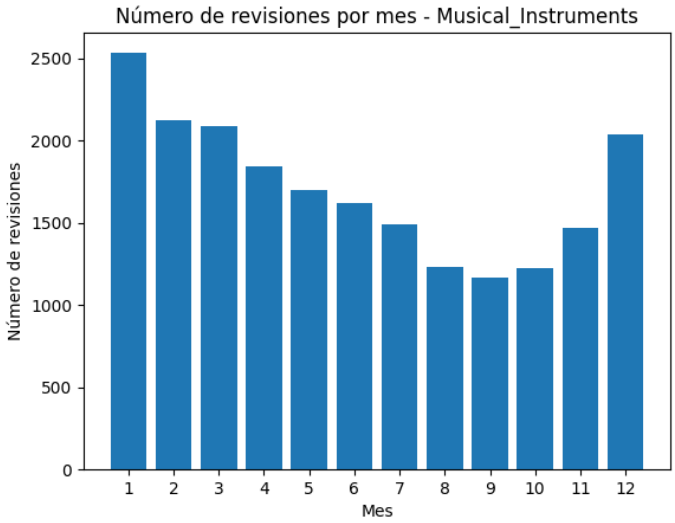

Figura 8: Circuito de acondicionamiento del servomotor

El menú se muestra en la interfaz hasta que se selecciona la opción de salir.

3. Neo4J

Esta parte se ha implementado en un fichero llamado “neo4JProyecto.py” que contiene un menú con las siguientes opciones.

La primera opción obtiene similitudes entre usuarios y muestra los enlaces en Neo4J. A continuación se adjunta la imagen de las similitudes entre los 30 usuarios obtenida en Neo4J. El número de usuarios entre los que obtener las similitudes es fácilmente configurable mediante el archivo “configuracion.py”.

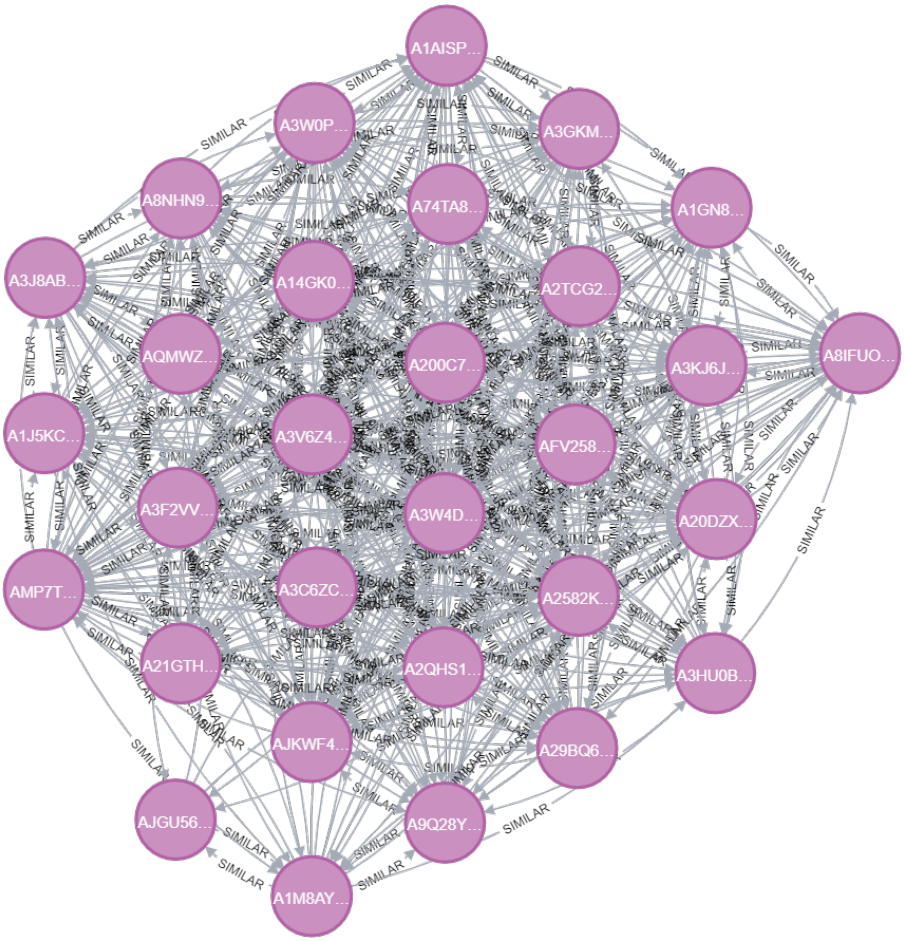

Figura 9: Primera opción de Neo4j

La segunda opción permite obtener enlaces entre usuarios y artículos. En nuestro caso se adjunta una imagen para tres artículos aleatorios del tipo Digital_Music.

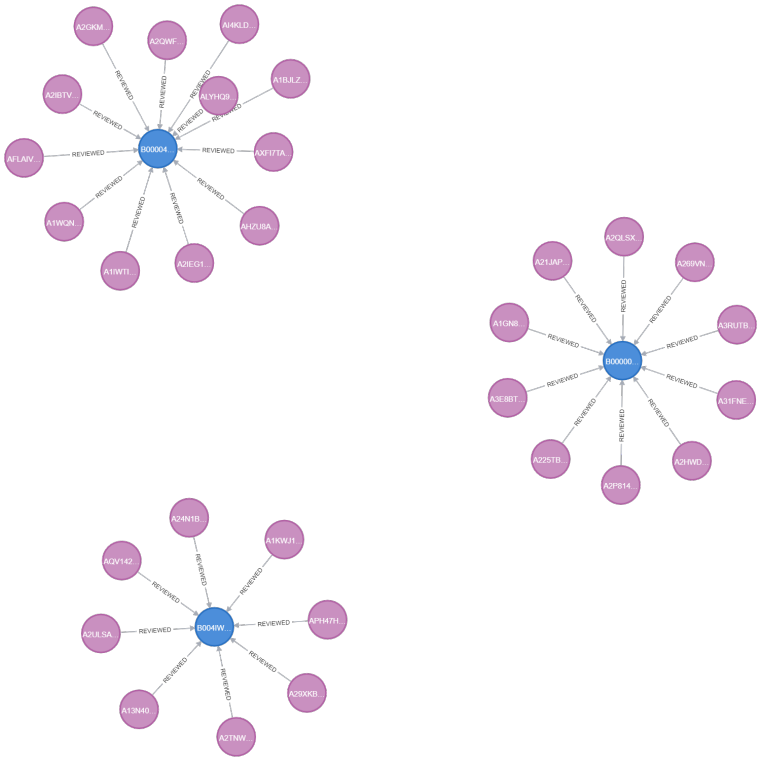

Figura 10: Segunda opción de Neo4j

La tercera opción permite obtener algunos usuarios que han visto más de un determinado tipo de artículo. En nuestro caso se han seleccionado los primeros 400 usuarios ordenados por el nombre.

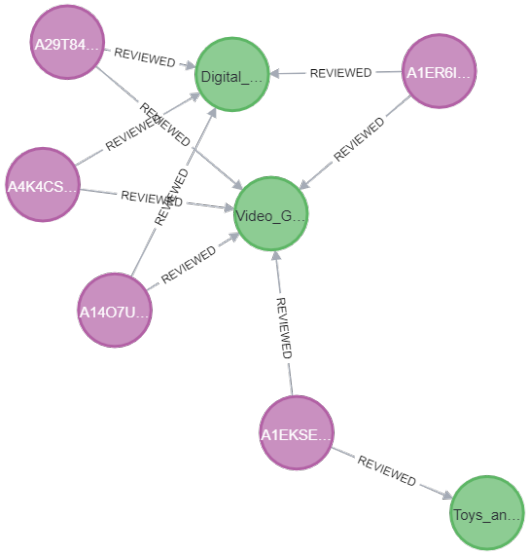

Figura 11: Tercera opción de Neo4j

Finalmente, la cuarta opción permite obtener artículos populares y artículos en común entre usuarios. Para ello se han seleccionado los 5 artículos más populares de entre los que tienen menos de 40 reviews. Estos se muestran junto con todos los usuarios que los han puntuado y los enlaces entre usuarios con artículos puntuados en común.

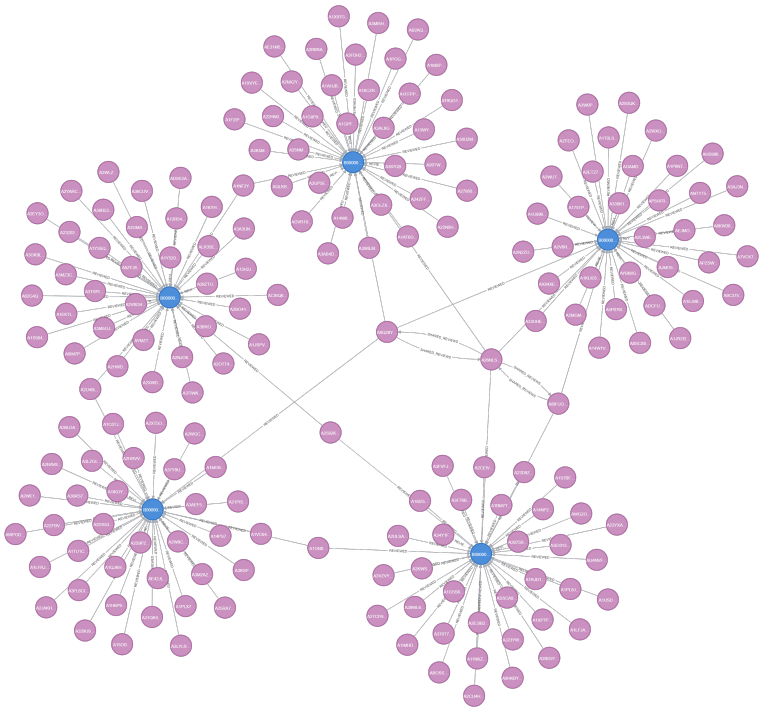

Figura 12: Cuarta opción de Neo4j

4. Nuevos datos

Para llevar a cabo este proyecto es necesario un archivo denominado "inserta_dataset.py", el cual permite incorporar nueva información a la base de datos. Para lograrlo, es necesario manejar los datos que serán ingresados tanto en MySQL como en MongoDB por separado.

La inserción de datos en MongoDB resulta relativamente simple, ya que implica incorporar todas las reseñas, simplemente omitiendo cierta información, como los nombres de los revisores o el tipo de producto. Por otro lado, la obtención de datos de MySQL es algo más complicada, ya que es necesario revisar los usuarios que ya han sido añadido en la tabla de REVIEWERS. Además, es necesario incluir los nuevos productos en la lista junto con su tipo correspondiente (el cual también se añade a la tabla de tipos de productos).

Para utilizar este código, es necesario cambiar la ruta del fichero que se desea introducir en la base de datos (este se encuentra en la primera línea del main del fichero). Además, este fichero debe ser un fichero json y seguir la misma estructura que el resto de ficheros que ya han sido añadidos.

5. PowerBI y relación con Machine Learning

En esta sección se exponen los resultados obtenidos en PowerBI al replicar las gráficas obtenidas en el segundo apartado. Para este apartado se ha usado el archivo "load_data_PBi.py" para cargar los datos con otra estructura en MongoDB.

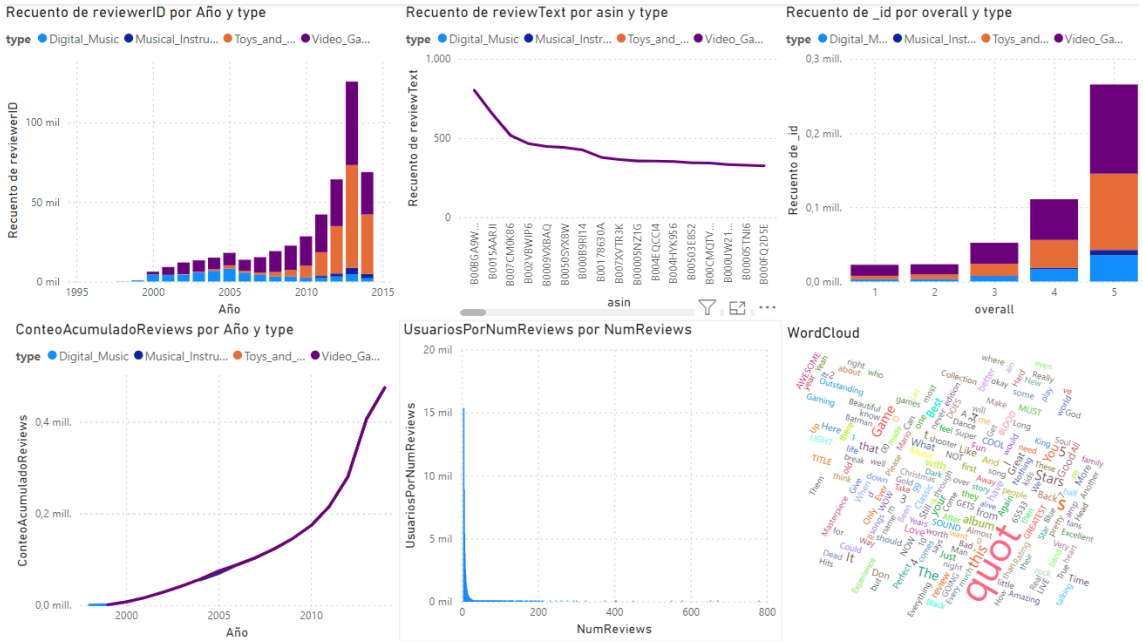

Figura 13: Visualización en PowerBI

Ahora vamos a explicar cómo podríamos diseñar un modelo de Machine Learning capaz de efectuar recomendaciones a usuarios de artículos que no hayan consumido previamente.

Primero, recopilamos datos de usuarios y productos. Estos incluyen identificaciones de usuario (*reviewerID*), identificaciones de producto (*asin*), calificaciones (*overall*), texto de reseñas (*reviewText*), y datos sobre la utilidad de la reseña (*helpful*). Este último aspecto puede indicar qué tan influyentes o valoradas son ciertas reseñas, lo cual puede ser crucial para ponderar sus efectos en las recomendaciones.

El texto de las reseñas se procesa usando técnicas de procesamiento de lenguaje natural. Podemos aplicar TF-IDF para medir la importancia de una palabra en relación con el corpus de documentos o utilizar embeddings de palabras para capturar contextos y semánticas subyacentes. Esto convierte el texto en una forma numérica que puede ser analizada por algoritmos de Machine Learning.

Aplicamos Análisis de Componentes Principales (PCA) para reducir la dimensionalidad de los datos vectorizados de las reseñas. Esto ayuda a simplificar los datos, lo que puede ser vital para mejorar los tiempos de entrenamiento y el rendimiento del modelo sin sacrificar información crítica. Escogemos los componentes que retienen la mayor varianza de los datos.

Utilizamos los componentes principales como características en dos enfoques de recomendación:

- Filtrado Colaborativo: Aquí, empleamos los componentes principales para detectar similitudes entre usuarios o productos. Si dos usuarios han dado reseñas similares a un conjunto de productos, entonces es probable que tengan gustos similares, lo que nos permite recomendar productos que uno ha valorado positivamente al otro.

- Filtrado Basado en Contenido: Utilizamos los componentes principales de las reseñas para recomendar productos con descripciones similares a aquellas que un usuario ha calificado positivamente en el pasado.

El modelo se evalúa con un conjunto de prueba para determinar su eficacia en recomendar artículos. Ajustamos el modelo según sea necesario usando métricas como la precisión, el *recall*, y el F1-*score* para mejorar su capacidad de predecir lo que un usuario preferirá pero que aún no ha consumido.

Una vez optimizado, el modelo se implementa en un sistema de producción para ofrecer recomendaciones personalizadas. Los usuarios reciben sugerencias basadas en su historial de interacciones y preferencias, mejorando la experiencia del usuario y potencialmente aumentando la satisfacción y retención del cliente.

Cabe destacar que las ideas y conceptos descritos en este apartado están inspirados en el libro "An Introduction to Statistical Learning" (ISLP), que es una fuente reconocida en el campo del aprendizaje estadístico.