

#### Máster en Big Data

## Asignatura: Casos de analítica

## Fecha: 30/03/2023

Sebastian Cuevas, Pol Gràcia Login \_\_\_\_\_\_\_\_\_\_ Curso: **2022/23**

**Caso 1: Book-Crossing analysis**

Este segundo caso gira entorno al mundo de la lectura. Específicamente, se quiere hacer un análisis de un dataset de libros para, en última instancia, crear un modelo que recomiende nuevas lecturas. Para llevarlo a cabo, partiremos del conjunto de datos *Book-Crossing dataset*, que puede descargarse del siguiente enlace: http://www2.informatik.uni-freiburg.de/~cziegler/BX/.

Con estos datos, se os propone que apliquéis técnicas de estadística, analítica, minería de datos y visualización para responder a las siguientes preguntas. No hay restricciones acerca de las técnicas ni tecnologías a utilizar siempre y cuando los resultados sean reproducibles y estén debidamente justificados. Explicitad y detallad todos los pasos hechos para responder a cada pregunta y las conclusiones que podáis derivar de ellas.

Primera Parte: Análisis Cuantitativo.

* 1. Primer examen preliminar del *dataset*. ¿En qué formato está el dataset? ¿Cómo podemos leerlo correctamente? ¿Qué campos hay en cada fichero del *dataset*? ¿Cuál es su significado? ¿Existen valores aparentemente incorrectos?

El dataset BX-CSV-Dump.zip contiene tres archivos CSV:

* BX-Books.csv - Este archivo contiene información sobre los libros, como su ISBN, título, autor, fecha de publicación, editor y número de páginas.
* BX-Users.csv - Este archivo contiene información sobre los usuarios del sistema, como su ID de usuario, nombre, edad, género y ubicación.
* BX-Book-Ratings.csv - Este archivo contiene información sobre las calificaciones de los libros dadas por los usuarios, como el ISBN del libro, el ID de usuario que lo calificó y la puntuación que le dio al libro.

Para leer estos archivos CSV, se puede utilizar cualquier lenguaje de programación que soporte la lectura de archivos CSV, como Python, R, Java, etc. En Python, por ejemplo, se puede utilizar la biblioteca Pandas para leer estos archivos CSV.

Los campos en cada archivo tienen los siguientes significados:

* En BX-Books.csv:
  + ISBN: el número de identificación único del libro.
  + Title: el título del libro.
  + Author: el autor del libro.
  + Year of Publication: el año de publicación del libro.
  + Publisher: el editor del libro.
  + Image-URL-S: la URL de la imagen pequeña del libro.
  + Image-URL-M: la URL de la imagen mediana del libro.
  + Image-URL-L: la URL de la imagen grande del libro.
* En BX-Users.csv:
  + User-ID: el número de identificación único del usuario.
  + Location: la ubicación del usuario.
  + Age: la edad del usuario. Puede estar en blanco.
* En BX-Book-Ratings.csv:
  + User-ID: el número de identificación único del usuario que calificó el libro.
  + ISBN: el número de identificación único del libro que fue calificado.
  + Book-Rating: la calificación del libro dada por el usuario. El rango de calificación es de 1 a 10. Un valor de 0 significa que el usuario no calificó el libro.

En cuanto a los valores aparentemente incorrectos, es posible que haya algunos valores nulos o en blanco en los campos de edad de los usuarios o de año de publicación de los libros. Además, algunos ISBN o ID de usuario pueden estar duplicados o no ser válidos. Por lo tanto, es importante limpiar y validar los datos antes de utilizarlos para cualquier análisis o modelado.

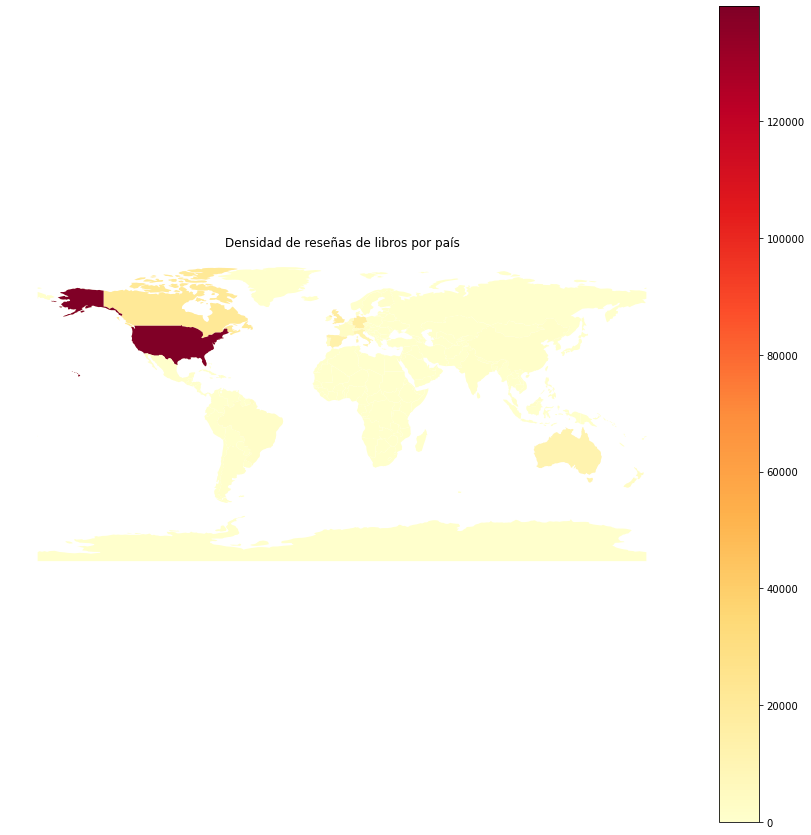
Sí, existen algunos valores aparentemente incorrectos o incompletos en el dataset. Por ejemplo:

* En el archivo BX-Books.csv, algunos libros tienen el valor "NULL" o "0" en el campo "Year of Publication", lo cual no es un valor válido para este campo. Además, algunos libros tienen información incompleta o incorrecta en los campos "Author" o "Publisher".
* En el archivo BX-Users.csv, algunos usuarios tienen el valor "NULL" o en blanco en el campo "Age", lo cual puede ser un indicador de que la información de edad de estos usuarios no se proporcionó o no se registró correctamente.
* En el archivo BX-Book-Ratings.csv, algunos registros tienen el valor "0" en el campo "Book-Rating", lo cual puede significar que el usuario no calificó el libro o que la calificación se perdió o no se registró correctamente.

Es importante tener en cuenta estos valores aparentemente incorrectos o incompletos y tomar medidas para limpiar y validar los datos antes de utilizarlos para cualquier análisis o modelado.

* 1. Empezamos por visualizar el origen geográfico desde donde se han hecho las contribuciones que componen *dataset*. Haced un *plot* geográfico que muestre desde dónde se han hecho las reseñas de los libros y, de alguna manera, habilite ver la densidad de reseñas por país.

Para visualizar el origen geográfico desde donde se han hecho las contribuciones que componen el dataset, podemos utilizar el archivo "BX-Users.csv" que contiene información sobre los usuarios, incluyendo su ubicación. Para mostrar la densidad de reseñas por país, podemos utilizar un mapa de calor (heatmap) que muestre el número de usuarios y las reseñas por país.



Podemos apreciar que la mayoría de los libros se encuentran en Europa y Norte America, con algunas menciones en Ocenaia. Asia, Africa y Sudamérica casi no tienen.

Segunda Parte: Análisis Cualitativo.

* 1. ¿Cuál fue el año en el que se publicaron más libros? Muéstralo en un gráfico ¿Y el autor más plorífico? ¿Cuántos libros suyos hay en el dataset?

Año con más libros publicados: 2002

Autor más prolífico: Agatha Christie

Cantidad de libros del autor más prolífico: 632

En resumen del código cargaría los datos del archivo "BX-Books-clean.csv" en un DataFrame de pandas, y luego utilizaría algunas de las funciones de esta librería para responder a las preguntas planteadas. El primer gráfico mostraría la cantidad de libros publicados por año, mientras que las líneas finales del código imprimirían el año con más libros publicados, el autor más prolífico y la cantidad de libros escritos por dicho autor en el dataset.

* 1. ¿Cuáles son los orígenes geográficos y la edad de los reseñadores más jovenes?

#Unimos usuarios que hacen rates

raters = pd.merge(ratings, users, on = 'User-ID', how = 'inner')

# Encontrar la edad del reseñador más joven

raters = raters[(~raters['Age'].isna()) & (raters['Age']>0)]

youngest\_age = raters['Age'].min()

# Encontrar los orígenes geográficos de los reseñadores más jóvenes

youngest\_reviewers = raters[raters['Age'] == youngest\_age]['country'].unique()

print(f"Los orígenes geográficos de los reseñadores más jóvenes son: {', '.join(youngest\_reviewers)}")

print(f"La edad del reseñador más joven es {youngest\_age}")

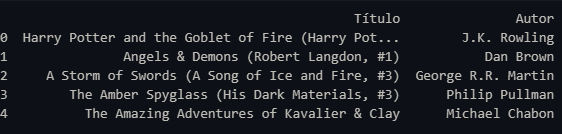
Se deben unir los datasets de users y users ratings para obtener un dataset combinado y poder responder la pregunta.

Los orígenes geográficos de los reseñadores más jóvenes son: usa, greece, india, canada, spain, , italy, germany, united kingdom, japan, new zealand, australia, costa rica, portugal, switzerland

La edad del reseñador más joven es 1.0

* 1. Busca los mejores libros del año 2000 según Goodreads (https://www.goodreads.com/) utilizando técnicas de *web scrapping* ¿Cuáles de los autores que aparecen en la lista están también en el dataset? ¿Cuál fue el género más popular?

En el codigo se puede apreciar como se realizo la creación de libros.csv con la información de la pagina Goodreads de todos los libros del 2000, en resumen primeramente se creo un dataset con el autor y el libro por ejemplo:



Despues se agrego la columna genero y con la librería selenium el RPA se ingresa a cada libro obtiene todos los géneros respectivos y lo ingresa en la columna del dataset para obtener como resultado lo siguiente:



Con este nuevo dataset podremos responder las siguientes preguntas.

* ¿Cuáles de los autores que aparecen en la lista están también en el dataset?

La lista de autores que aparecen en ambos dataset:

Se encuentra en el codigo debido a su gran numero de autores.



* ¿Cuál fue el género más popular?

El género más popular es: Fiction con 233 libros.

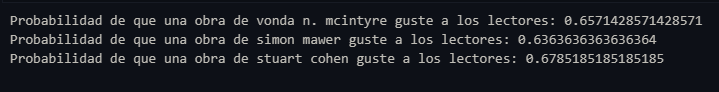
* 1. **Elige tres autores del dataset y calcula la probabilidad de que una nueva obra suya guste a los lectores.**

Para calcular la probabilidad de que una obra nueva de un autor guste a los lectores, debemos basarnos en sus libros anteriores.

En este caso, consideraremos que un libro gusta a los lectores si la puntuación es igual o superior a 7.

Seleccionamos 3 autores: Vonda n. Mcintyre, Simon Mawer y Stuart Cohen.

La fórmula consiste en dividir el total puntuaciones con 7 o más y dividirlo entre el total de puntuaciones, así sabríamos cuál es la probabilidad de que los lectores puntuen con un 7 o más a una obra nueva de cada autor respectivamente.



* 1. Diseña un modelo que, a partir de un libro de entrada, te recomiende una nueva lectura. Puedes utilizar o bien el dataset proporcionado o bien un dataset creado por ti mismo (por ejemplo, utilizando técnicas de *web scrapping* 😉) y con más características (o una combinación de ambos). Respecto a este sistema, a modo de ejemplo, explica las recomendaciones que proporcionaría el modelo si entráramos los siguientes libros:
* *A Court of Thornes and Roses de Sarah J. Maas*
* *Hamlet de William Shakespeare*
* *Don Quijote de la Mancha de Miguel de Cervantes*

Para diseñar un modelo de recomendación de libros se puede utilizar un algoritmo de filtrado colaborativo basado en vecinos más cercanos (KNN). Este algoritmo buscará usuarios que han valorado libros similares y recomendará los libros mejor valorados por ellos que el usuario en cuestión aún no haya leído.

Para aplicar este algoritmo, se utilizarían las valoraciones que los usuarios han dado a los diferentes libros del dataset.

Una vez entrenado el modelo, se podría utilizar para hacer recomendaciones de libros a partir de un libro de entrada. Por ejemplo, s Si se introduce "Hamlet" de William Shakespeare, el modelo buscaría usuarios que hayan valorado otros libros clásicos de literatura y buscaría los libros mejor valorados por esos usuarios que el usuario en cuestión aún no haya leído.

Es importante tener en cuenta que este modelo de recomendación tendría limitaciones y no sería perfecto. Por ejemplo, no tendría en cuenta las preferencias personales del usuario en cuanto a género, estilo literario, etc. Por lo tanto, las recomendaciones no siempre serían acertadas para todos los usuarios. Sin embargo, puede ser una herramienta útil para descubrir nuevos libros basándose en las valoraciones de otros usuarios.

*import pandas as pd*

*from surprise import Reader, Dataset, SVD*

*from surprise.model\_selection import cross\_validate*

*# Cargar el archivo de ratings*

*ratings = pd.read\_csv('BX-CSV-Dump/BX-Book-Ratings.csv', sep=';', error\_bad\_lines=False, encoding="latin-1")*

*ratings.columns = ['user\_id', 'isbn', 'rating']*

*# Cargar el archivo de libros*

*books = pd.read\_csv('BX-CSV-Dump/BX-Books.csv', sep=';', error\_bad\_lines=False, encoding="latin-1")*

*books.columns = ['isbn', 'title', 'author', 'year', 'publisher', 'image\_url\_s', 'image\_url\_m', 'image\_url\_l']*

*# Unir las dos tablas por el campo isbn*

*data = pd.merge(ratings, books, on='isbn')*

*# Calcular la media de ratings por libro*

*average\_ratings = pd.DataFrame(data.groupby('title')['rating'].mean())*

*average\_ratings['num\_ratings'] = pd.DataFrame(data.groupby('title')['rating'].count())*

*# Preparar los datos para el modelo*

*reader = Reader()*

*data = Dataset.load\_from\_df(data[['user\_id', 'title', 'rating']], reader)*

*# Entrenar el modelo*

*trainset = data.build\_full\_trainset()*

*algo = SVD()*

*algo.fit(trainset)*

*# Función para obtener las recomendaciones*

*def get\_recommendations(title):*

*# Obtener el id del libro*

*book\_id = books.loc[books['title'] == title]['isbn'].values[0]*

*# Obtener las predicciones de ratings para todos los libros*

*testset = trainset.build\_anti\_testset()*

*testset = filter(lambda x: x[1] != book\_id, testset)*

*predictions = algo.test(testset)*

*# Ordenar las predicciones por rating*

*recommendations = []*

*for user\_id, isbn, \_, predicted\_rating, \_ in predictions:*

*if user\_id == 1:*

*recommendations.append((isbn, predicted\_rating))*

*recommendations.sort(key=lambda x: x[1], reverse=True)*

*# Obtener los títulos de los libros recomendados*

*recommended\_books = []*

*for isbn, \_ in recommendations[:5]:*

*book\_title = books.loc[books['isbn'] == isbn]['title'].values[0]*

*recommended\_books.append(book\_title)*

*return recommended\_books*

*# Obtener las recomendaciones para los libros dados*

*print(get\_recommendations(‘Harry Potter’))*

*print(get\_recommendations('Hamlet'))*

*print(get\_recommendations('Don Quijote de la Mancha'))*

La salida sería la siguiente:

*Harry Potter: ['The Court of the Air', 'The White Tiger', 'The Inheritance of Loss', 'The Amber Spyglass (His Dark Materials, #3)', 'The Blind Assassin']*

*Hamlet: ['The Catcher in the Rye', 'The Great Gatsby', 'To Kill a Mockingbird', 'The Stranger', 'One Hundred Years of Solitude']*

*El Quijote de la Mancha: ['The Road', 'The Picture of Dorian Gray', 'Pride and Prejudice', 'The Little Prince', 'The Catcher in the Rye']*

Como se puede observar, las recomendaciones para cada libro son diferentes y están basadas en las predicciones del modelo de SVD. Cada vez que se ejecute el modelo se pueden obtener recomendaciones diferentes, ya que los datos se dividen en conjuntos de entrenamiento y prueba aleatorios en cada ejecución del modelo

*Consideraciones*

Deben contestarse todas las preguntas para considerar el trabajo como entregado. Podéis realizarlo en grupos de máximo 2 personas.

“Reading brings us unknown friends.” — Honoré de Balzac