Bookflix

Documentación técnica - ABP

BOOKFLIX

Autores:

Daniel Fabián Rodríguez Lorenzo
Tatiana María Quintas Rodríguez
Miguel Ángel Seara Losada
David Simón Nóvoa

Índice

Contenido

Índice de ilustraciones

Ejemplo de Carga y Tratamiento de Datos

Objetivo

Demostrar la carga, exploración y gestión de datos (CRUD) usando el dataset de libros de Bookflix.

Tanto el dataset, como el notebook se encuentran en el siguiente repositorio:

- Akanenosketch/Bookflix-ABP

Cargar dataset en un DataFrame

Primero cargamos el dataset data.csv en un DataFrame de pandas para poder manipularlo y analizarlo de manera eficiente.

Nota: Primero es necesario cargar manualmente el archivo al notebook

Un DataFrame es una estructura tabular que permite realizar operaciones como filtrar, ordenar, agregar columnas o calcular estadísticas. Mostramos una vista previa de los datos para comprobar que se han cargado correctamente.

```
import pandas as pd
df = pd.read_csv('data.csv')
df
```

	isbn13	isbn10	title	subtitle	authors	categories	thumbnail	description	published_year	average_rating	num_pages	ratings_count
0	9780002005883	0002005883	Gilead	NaN	Marilynne Robinson	Fiction	http://books.google.com/books/content?id=KQZCP...	A NOVEL THAT READERS and critics have been eag...	2004.0	3.85	247.0	361.0
1	9780002261982	0002261987	Spider's Web	A Novel	Charles Osborne;Agatha Christie	Detective and mystery stories	http://books.google.com/books/content?id=gA5GP...	A new 'Christie for Christmas' -- a full-lengt...	2000.0	3.83	241.0	5164.0
2	9780006163831	0006163831	The One Tree	NaN	Stephen R. Donaldson	American fiction	http://books.google.com/books/content?id=OmQaw...	Volume Two of Stephen Donaldson's acclaimed se...	1982.0	3.97	479.0	172.0
3	9780006178736	0006178731	Rage of angels	NaN	Sidney Sheldon	Fiction	http://books.google.com/books/content?id=FKo2T...	A memorable, mesmerizing heroine Jennifer -- b...	1993.0	3.93	512.0	29532.0
4	9780006280897	0006280897	The Four Loves	NaN	Clive Staples Lewis	Christian life	http://books.google.com/books/content?id=XhQ5X...	Lewis' work on the nature of love divides love...	2002.0	4.15	170.0	33684.0
...	...	...	...	...	...	...	...	...	...	...	...	...
6805	9788185300535	8185300534	I Am that	Talks with Sri Nisargadatta Maharaj	Sri Nisargadatta Maharaj;Sudhakar S. Dikshit	Philosophy	http://books.google.com/books/content?id=Fv_JP...	This collection of the timeless teachings of o...	1999.0	4.51	531.0	104.0
6806	9788185944609	8185944601	Secrets Of The Heart	NaN	Khalil Gibran	Mysticism	http://books.google.com/books/content?id=XcrVp...	NaN	1993.0	4.08	74.0	324.0
6807	9788445074879	8445074873	Fahrenheit 451	NaN	Ray Bradbury	Book burning	NaN	NaN	2004.0	3.98	186.0	5733.0
6808	9789027712059	9027712050	The Berlin Phenomenology	NaN	Georg Wilhelm Friedrich Hegel	History	http://books.google.com/books/content?id=Vy7Sk...	Since the three volume edition ofHegel's Philo...	1981.0	0.00	210.0	0.0
6809	9789042003408	9042003405	'I'm Telling You Stories'	Jeanette Winterson and the Politics of Reading	Helena Grice;Tim Woods	Literary Criticism	http://books.google.com/books/content?id=2IVyR...	This is a jubilant and rewarding collection of...	1998.0	3.70	136.0	10.0

Ilustración 1.- Output de la carga de Datasets

Añadir un nuevo libro

En esta celda definimos un nuevo libro con toda su información: ISBN13, ISBN10, título, subtítulo, autores, categoría, imagen, descripción, año de publicación, valoración media, número de páginas y número de valoraciones.

Luego, utilizamos **pd.concat** para añadir este libro al DataFrame existente. Finalmente, mostramos las últimas filas del DataFrame para comprobar que el libro se ha añadido correctamente.

Esta operación es útil para realizar un registro de nuevos libros en nuestro dataset sin sobrescribir los existentes.

```
# Definir el nuevo libro
nuevo_libro = {
      "isbn13": 9789999999999,
      "isbn10": "9999999999",
      "title": "El Arte de Programar",
      "subtitle": "Edición Avanzada",
      "authors": "Donald Knuth",
      "categories": "Informática",
      "thumbnail": "https://ejemplo.com/arte_programar.jpg",
      "description": "Guía completa sobre estructuras y
algoritmos.",
      "published_year": 2025,
      "average_rating": 4.9,
      "num_pages": 950,
      "ratings_count": 4200 }

# Añadirlo al DataFrame
df = pd.concat([df, pd.DataFrame([nuevo_libro])],
ignore_index=True)

# Comprobar que se añadió
df.tail(2)
```

	isbn13	isbn10	title	subtitle	authors	categories	thumbnail	description	published_year	average_rating	num_pages	ratings_count
6809	9789042003408	9042003405	'I'm Telling You Stories'	Jeanette Winterson and the Politics of Reading	Helena Grice;Tim Woods	Literary Criticism	http://books.google.com/books/content?id=2IVyR...	This is a jubilant and rewarding collection of...	1998.0	3.7	136.0	10.0
6810	9789999999999	9999999999	El Arte de Programar	Edición Avanzada	Donald Knuth	Informática	https://ejemplo.com/arte_programar.jpg	Guía completa sobre estructuras y algoritmos.	2025.0	4.9	950.0	4200.0

Ilustración 2.- Output de añadir un libro nuevo

Modificar información de un libro

En esta celda actualizamos los datos de un libro existente, buscando primero su ISBN13.

Modificamos el título y la valoración media (average_rating) para reflejar cambios en la edición o en las valoraciones.

Mostrar las columnas modificadas nos permite verificar que los cambios se aplicaron correctamente, sin afectar otros libros.

```
# Ver el libro antes de modificarlo
df.loc[df["isbn13"] == 9789999999999, ["title", "average_rating"]]
```

	title	average_rating
6810	El Arte de Programar	4.9

Ilustración 3.- Output previo a la modificación de la información

```
# Modificar campos
df.loc[df["isbn13"] == 9789999999999, "title"] = "Gilead (Edición Revisada)"
df.loc[df["isbn13"] == 9789999999999, "average_rating"] = 4.2
# Ver los cambios
df.loc[df["isbn13"] == 9789999999999, ["title", "average_rating"]]
```

	title	average_rating
6810	Gilead (Edición Revisada)	4.2

Ilustración 4.- Output tras la modificación de los datos

Filtrar libros por categoría

Podemos seleccionar todos los libros de una categoría específica para analizarlos más a fondo o visualizarlos por separado.

En este ejemplo filtramos todos los libros cuya categoría es "Informática".

Esto es útil para trabajar con subconjuntos de datos y realizar análisis específicos de cada tipo de libro.

```
# Filtrar libros de la categoría "Informática"
libros_informatica = df[df["categories"] == "Informática"]
libros_informatica
```

	isbn13	isbn10	title	subtitle	authors	categories	thumbnail	description	published_year	average_rating	num_pages	ratings_count
6810	9789999999999	9999999999	Gilead (Edición Revisada)	Edición Avanzada	Donald Knuth	Informática	https://ejemplo.com/arte_programar.jpg	Guía completa sobre estructuras y algoritmos.	2025.0	4.2	950.0	4200.0

Ilustración 5.- Output del filtrado de libros por categoría

Eliminar un libro

A veces necesitamos eliminar registros del dataset, por ejemplo, si un libro se ha retirado o se ha añadido por error.

Primero mostramos cuántos libros hay antes de la eliminación.

Luego filtramos el DataFrame para excluir el libro con el ISBN especificado.

Finalmente, mostramos cuántos libros quedan después de la eliminación para comprobar que se ha borrado correctamente.

```
# Ver cuántos libros hay antes
print("Libros antes:", len(df))

# Eliminar el libro por ISBN
df = df[df["isbn13"] != 9789999999999] # El ISBN del libro añadido
antes

# Ver cuántos quedan
print("Libros después:", len(df))
```

```
Libros antes: 6810
Libros después: 6810
```

Ilustración 6.- Output del borrado de un libro

Ordenar libros por valoración media

Es útil ordenar los libros según su valoración media (average_rating) para identificar los más recomendados.
Podemos ordenar de mayor a menor para ver los libros mejor valorados o de menor a mayor para detectar los peor valorados.
La función **sort_values()** permite especificar la columna por la que ordenar y si queremos orden ascendente o descendente.

```
# Ordenar de mayor a menor valoración
df_sorted = df.sort_values(by="average_rating", ascending=False)
df_sorted.head(10) # Mostrar los 10 mejores valorados
```

	isbn13	isbn10	title	subtitle	authors	categories	thumbnail	description	published_year	average_rating	num_pages	ratings_count
6738	9781932206081	1932206086	Insights	Talks on the Nature of Existence	Frederick Lenz	Spiritual life	http://books.google.com/books/content?id=NOX2P...	In 1983, when Rama - Dr. Frederick P. Lenz rec...	2003.0	5.0	304.0	1.0
1441	9780310249870	0310249872	Fanning the Flame	Bible, Cross & Mission : Meeting the Challenge...	Christopher J. H. Wright	Religion	http://books.google.com/books/content?id=U-tpk...	The Bible, the cross, and the mission – by lea...	2003.0	5.0	336.0	1.0
6671	9781890995522	1890995525	The Diamond Color Meditation	Color Pathway to the Soul	John Diamond	Health & Fitness	http://books.google.com/books/content?id=1ChsH...	The Diamond Color Meditation presents an inspi...	2006.0	5.0	74.0	5.0
6720	9781930901353	1930901356	The Irish Anatomist	A Study of Flann O'Brien	Keith Donohue	Biography & Autobiography	http://books.google.com/books/content?id=baEJA...	The most full length critical and biographical...	2002.0	5.0	222.0	1.0
4281	9780738511672	0738511676	Middlesex Borough	NaN	NaN	History	http://books.google.com/books/content?id=c7a7U...	Protected by the Watchung Mountains on the nor...	2003.0	5.0	128.0	2.0
5972	9781551052700	1551052709	Ecuador Nature Guide	Southwest Forests : Sozoranga Forest Project	Christopher D. Jiggins	Botanique	http://books.google.com/books/content?id=1JjbG...	The guide provides information on 76 species o...	2000.0	5.0	96.0	1.0
5398	9780851621814	0851621813	The Complete Theory Fun Factory	NaN	Katie Elliott;Ian Martin	Juvenile Nonfiction	http://books.google.com/books/content?id=RoX8A...	(Boosey & Hawkes Scores/Books). Contains the m...	1996.0	5.0	96.0	1.0
4284	9780738539560	0738539562	Lake Orion	NaN	James E. Ingram;Lori Grove	History	http://books.google.com/books/content?id=OioLW...	Orion Township, established in 1835, became a ...	2006.0	5.0	128.0	0.0
4306	9780739844328	0739844326	Bill Gates	Computer Legend	Sara Barton-Wood	Juvenile Nonfiction	http://books.google.com/books/content?id=Fr814...	Presents the life of Bill Gates, from his chil...	2001.0	5.0	48.0	0.0
3580	9780567044716	0567044718	Colossians and Philemon	NaN	Robert McL Wilson	Religion	http://books.google.com/books/content?id=61CEy...	For over one hundred years International Criti...	2005.0	5.0	512.0	1.0

Ilustración 7.- Output de la ordenación de libros por valoración media

Contar libros por categoría

A veces queremos conocer cuántos libros tenemos en cada categoría.

Esto nos ayuda a entender la distribución de nuestro catálogo y a identificar categorías más o menos representadas.

Usamos value_counts() sobre la columna categories, que devuelve un recuento de cuántas veces aparece cada categoría en el DataFrame.

```
# Contar libros por categoría
df["categories"].value_counts()
```

categories	count
Fiction	2588
Juvenile Fiction	538
Biography & Autobiography	401
History	264
Literary Criticism	166
...	...
Death (Fictitious character : Gaiman)	1
Astronomers	1
Epic literature	1
Girls	1
Africa, East	1

568 rows × 1 columns

Ilustración 8.- Output del conteo de libros por categoría

Estadísticas descriptivas

Podemos obtener estadísticas básicas como media, mínimo, máximo y desviación estándar de columnas numéricas como num_pages o average_rating.

```
# Estadísticas del número de páginas y valoración media
df[["num_pages", "average_rating"]].describe()
```

	num_pages	average_rating
count	6768.000000	6768.000000
mean	348.269947	3.933323
std	242.469252	0.331343
min	0.000000	0.000000
25%	208.000000	3.770000
50%	304.000000	3.960000
75%	420.000000	4.130000
max	3342.000000	5.000000

Ilustración 9.- Output de las estadísticas

Buscar libros de un autor específico

Podemos filtrar todos los libros escritos por un autor determinado.

En este ejemplo buscamos libros de "George Orwell".

Algunos valores de la columna authors pueden estar vacíos (Por eso usamos na=False en NaN). str.contains() para evitar errores y que las filas vacías no se incluyan en el resultado.

```
# Filtrar libros de J.K. Rowling, ignorando los NaN
df[df["authors"].str.contains("George Orwell", na=False)]
```

	isbn13	isbn10	title	subtitle	authors	categories	thumbnail	description	published_year	average_rating	num_pages	ratings_count
902	9780141183725	0141183721	Keep the Aspidistra Flying	NaN	George Orwell	Fiction	http://books.google.com/books/content?id=0oNp8...	London 1934. Gordon Comstock, copywriter for t...	2000.0	3.88	277.0	11220.0
913	9780141185163	0141185163	Orwell in Spain	the full text of Homage to Catalonia, with ass...	George Orwell	Fiction	http://books.google.com/books/content?id=uVNpA...	Including Homage to Catalonia, Orwell's profou...	2001.0	4.33	416.0	203.0
1031	9780143036357	0143036351	Why I Write	NaN	George Orwell	Language Arts & Disciplines	http://books.google.com/books/content?id=6l_qP...	Throughout history, some books have changed th...	2005.0	4.03	120.0	5874.0
1073	9780151010264	0151010269	Animal Farm and 1984	NaN	George Orwell	Fiction	http://books.google.com/books/content?id=h1Klm...	George Orwell's classic satire on totalitarian...	2003.0	4.28	400.0	140015.0
1144	9780156421171	0156421178	Homage to Catalonia	NaN	George Orwell	History	http://books.google.com/books/content?id=MXwx-...	Presents the British novelist's firsthand repo...	1952.0	4.15	232.0	27619.0
1156	9780156701761	0156701766	The Orwell Reader	Fiction, Essays, and Reportage	George Orwell	Literary Collections	http://books.google.com/books/content?id=DfQlq...	Selections reveal the development of Orwell's ...	1961.0	4.37	480.0	265.0
2061	9780375415036	0375415033	Essays	NaN	George Orwell	Literary Collections	http://books.google.com/books/content?id=Bm1yQ...	Presents a collection of essays from George Or...	2002.0	4.33	1369.0	598.0
2660	9780436350238	0436350238	The Complete Works of George Orwell	NaN	George Orwell	NaN	NaN	NaN	1986.0	4.11	230.0	20.0
3099	9780452284234	0452284236	1984	NaN	George Orwell	London (England)	http://books.google.com/books/content?id=Ocq6h...	Portrays a terrifying vision of life in the fu...	2003.0	4.17	339.0	9298.0

Ilustración 10.- Output de la búsqueda de libros por autor

Ejemplo de funcionamiento del sistema de recomendación de Bookflix

El Notebook donde se localiza toda la información mostrada a continuación se encuentra en:

- Bookflix-ABP/Notebook_Bookflix_Recomendador.ipynb at main · Akanenosketch/Bookflix-ABP

El objetivo es demostrar el sistema de recomendación haciendo uso del dataset de libros de Bookflix.

Procesado de los datos para recomendación

Para nuestro sistema de recomendación, obtendremos como resultado final en este Notebook una matriz de NxN, siendo N el número de libros de nuestro dataset.

Para dicha matriz, el valor matriz[x][y] será la distancia entre las descripciones de los libros "x" e "y", por lo que los valores del array matriz[x] serán las distancias entre el libro X y los N libros existentes, por lo que se pueden consultar los valores más bajos para encontrar libros con descripciones similares. En nuestra aplicación, los valores de la matriz de distancias se almacenarán en la base de datos, de forma que el proceso para calcular las distancias solo se necesita ejecutar cuando se añaden nuevos libros.

Para generar la matriz de distancias, primero realizamos un preprocesado sobre el texto, mediante la ***tokenización*** de la descripción original, la eliminación de ***stopwords*** y la ***stemmización*** de los tokens.

Estos textos preprocesados se convierten en una ***Bag of Words*** **(BoW)**, que asocia para cada texto preprocesado las palabras que aparecen y su frecuencia. Dicha frecuencia se calcula con el algoritmo ***TF-IDF***.

Finalmente, se usa la BoW para crear la matriz de distancias, calculando la distancia coseno entre vectores del BoW.

Carga de datos

Primero cargamos el dataset data.csv en un DataFrame de pandas para poder manipularlo y analizarlo de manera eficiente.

Nota: Primero es necesario cargar manualmente el archivo al notebook

Mostramos una vista previa de los datos para comprobar que se han cargado correctamente.

```
import pandas as pd

df_og = pd.read_csv('data.csv')
df_og
```

	isbn13	isbn10	title	subtitle	authors	categories	thumbnail	description	published_year	average_rating	num_pages	ratings_count
0	9780002005883	0002005883	Gilead	NaN	Marilynne Robinson	Fiction	http://books.google.com/books/content?id=KQZCP...	A NOVEL THAT READERS and critics have been eag...	2004.0	3.85	247.0	361.0
1	9780002261982	0002261987	Spider's Web	A Novel	Charles Osborne;Agatha Christie	Detective and mystery stories	http://books.google.com/books/content?id=gA5GP...	A new 'Christie for Christmas' -- a full-lengt...	2000.0	3.83	241.0	5164.0
2	9780006163831	0006163831	The One Tree	NaN	Stephen R. Donaldson	American fiction	http://books.google.com/books/content?id=OmQaw...	Volume Two of Stephen Donaldson's acclaimed se...	1982.0	3.97	479.0	172.0
3	9780006178736	0006178731	Rage of angels	NaN	Sidney Sheldon	Fiction	http://books.google.com/books/content?id=FBzZ1...	A memorable, mesmerizing heroine Jennifer -- b...	1993.0	3.93	512.0	29532.0
4	9780006280897	0006280897	The Four Loves	NaN	Clive Staples Lewis	Christian life	http://books.google.com/books/content?id=XhQ5X...	Lewis' work on the nature of love divides love...	2002.0	4.15	170.0	33684.0
...	...	...	...	...	...	...	...	...	...	...	...	...
6805	9788185300535	8185300534	I Am that	Talks with Sri Nisargadatta Maharaj	Sri Nisargadatta Maharaj;Sudhakar S. Dikshit	Philosophy	http://books.google.com/books/content?id=Fy_JP...	This collection of the timeless teachings of o...	1999.0	4.51	531.0	104.0
6806	9788185944609	8185944601	Secrets Of The Heart	NaN	Khalil Gibran	Mysticism	http://books.google.com/books/content?id=XcVp...	NaN	1993.0	4.08	74.0	324.0
6807	9788445074879	8445074873	Fahrenheit 451	NaN	Ray Bradbury	Book burning	NaN	NaN	2004.0	3.98	186.0	5733.0
6808	9789027712059	9027712050	The Berlin Phenomenology	NaN	Georg Wilhelm Friedrich Hegel	History	http://books.google.com/books/content?id=Vy7Sk...	Since the three volume edition of Hegel's Philo...	1981.0	0.00	210.0	0.0
6809	9789042003408	9042003405	'I'm Telling You Stories'	Jeanette Winterson and the Politics of Reading	Helena Grice;Tim Woods	Literary Criticism	http://books.google.com/books/content?id=2fVyR...	This is a jubilant and rewarding collection of...	1998.0	3.70	136.0	10.0

6810 rows × 12 columns

Ilustración 11.- Carga del dataset

Preprocesado del dataset

En esta celda se preprocesa el texto almacenado en el campo "description" y se almacena en una nueva columna del DataFrame con nombre "processed_desc".

El preprocesamiento incluye los siguientes pasos:

- ***Tokenización*** del texto (mediante la librería ***nltk***): Se divide el texto en palabras (tokens), de forma que se obtiene un array compuesto por las palabras que constituyen del texto, separándolas mediante los delimitadores habituales (espacios, comas, puntos...).
- Eliminación de ***stopwords*** (usando de referencia las stopwords por defecto de la librería ***nltk***): Consiste en eliminar palabras que no aportan significado al texto (stopwords), como pueden ser artículos, pronombres y preposiciones. Las stopwords a eliminar son dependientes del idioma, por lo que el dataset usado debe estar en Ingles.
- ***Stemmizacion del texto*** (usando el algoritmo ***PorterStemmer***): Se eliminan variaciones de palabras provocadas por conjugación de verbos y por uso de plurales y géneros. Para conseguirlo, se obtiene la raíz semántica de todas las palabras.

Al finalizar el preprocesamiento, se muestran los datos para verificar que el proceso se ha realizado con éxito, y se muestra la descripción y el texto procesado del primer libro para comparar.

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer

import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('punkt_tab') # Added to resolve a LookupError

ps = PorterStemmer()

processed_desc = []

# Fill NaN values in 'description' column with empty strings
df_og['description'] = df_og['description'].fillna('')

for row in df_og.itertuples():
    text = word_tokenize(row[8]) ## indice de la columna que contiene el texto
    ## Remove stop words
    stops = set(stopwords.words("english"))
    text = [ps.stem(w) for w in text if not w in stops and w.isalnum()]
    text = " ".join(text)

    processed_desc.append(text)

df_preprocessed = df_og
df_preprocessed['processed_desc'] = processed_desc

df_preprocessed
```

	isbn13	isbn10	title	subtitle	authors	categories	thumbnail	description	published_year	average_rating	num_pages	ratings_count	processed_desc
0	9780002005883	0002005883	Gilead	NaN	Marilynne Robinson	Fiction	http://books.google.com/books/content?id=KQZCP...	A NOVEL THAT READERS and critics have been eag...	2004.0	3.85	247.0	361.0	a novel that reader critic eagerli anticip dec...
1	9780002261982	0002261987	Spider's Web	A Novel	Charles Osborne;Agatha Christie	Detective and mystery stories	http://books.google.com/books/content?id=gA5GP...	A new 'Christie for Christmas' -- a full-lengt...	2000.0	3.83	241.0	5164.0	a new christma novel adapt acclaim play charl ...
2	9780006163831	0006163831	The One Tree	NaN	Stephen R. Donaldson	American fiction	http://books.google.com/books/content?id=OmQaw...	Volume Two of Stephen Donaldson's acclaimed se...	1982.0	3.97	479.0	172.0	volum two stephen donaldson acclaim second tri...
3	9780006178736	0006178731	Rage of angels	NaN	Sidney Sheldon	Fiction	http://books.google.com/books/content?id=FKo2T...	A memorable, mesmerizing heroine Jennifer -- b...	1993.0	3.93	512.0	29532.0	a memor mesmer heroin jennif brilliant beauti ...
4	9780006280897	0006280897	The Four Loves	NaN	Clive Staples Lewis	Christian life	http://books.google.com/books/content?id=XhQ5X...	Lewis' work on the nature of love divides love...	2002.0	4.15	170.0	33684.0	lewi work natur love divid love four categori ...
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6805	9788185300535	8185300534	I Am that	Talks with Sri Nisargadatta Maharaj	Sri Nisargadatta Maharaj;Sudhakar S. Dikshit	Philosophy	http://books.google.com/books/content?id=Fv_JP...	This collection of the timeless teachings of o...	1999.0	4.51	531.0	104.0	thi collect timeless teach one greatest sage i...
6806	9788185944609	8185944601	Secrets Of The Heart	NaN	Khalil Gibran	Mysticism	http://books.google.com/books/content?id=XcrVp...		1993.0	4.08	74.0	324.0	
6807	9788445074879	8445074873	Fahrenheit 451	NaN	Ray Bradbury	Book burning	NaN		2004.0	3.98	186.0	5733.0	
6808	9789027712059	9027712050	The Berlin Phenomenology	NaN	Georg Wilhelm Friedrich Hegel	History	http://books.google.com/books/content?id=Vy7Sk...	Since the three volume edition ofHegel's Philo...	1981.0	0.00	210.0	0.0	sinc three volum edit ofhegel philosophi subje...
6809	9789042003408	9042003405	'I'm Telling You Stories'	Jeanette Winterson and the Politics of Reading	Helena Grice;Tim Woods	Literary Criticism	http://books.google.com/books/content?id=2fVyR...	This is a jubilant and rewarding collection of...	1998.0	3.70	136.0	10.0	thi jubil reward collect winterson scholarship...

6810 rows × 13 columns

Ilustración 12.- Preprocesado de la tabla

```
df_preprocessed.iloc[0]['description']
```

```
'A NOVEL THAT READERS and critics have been eagerly anticipating for over a decade, Gilead is an astonishingly imagined story of remarkable lives. John Ames is a preacher, the son of a preacher and the grandson (both maternal and paternal) of preachers. It’s 1956 in Gilead, Iowa, towards the end of the Reverend Ames’s life, and he is absorbed in recording his family’s story, a legacy for the young son he will never see grow up. Haunted by his grandfather’s presence, John tells of the rift between his grandfather and his father: the elder, an angry visionary who fought for the abolitionist cause, and his son, an ardent pacifist. He is troubled, too, by his prodigal namesake, Jack (John Ames) Boughton, his best friend’s lost son who returns to Gilead searching for forgiveness and redemption. Told in John Ames’s joyous, rambling voice that finds beauty, humour and truth in the smallest of life’s details, Gilead is a song of celebration and acceptance of the best and the worst the world has to offer. At its heart is a tale of the sacred bonds between fathers and sons, pitch-perfect in style and story, set to dazzle critics and readers alike.'
```

```
df_preprocessed.iloc[0]['processed_desc']
```

```
'a novel that reader critic eagerli anticip decad gilead astonishingli imagin stori remark live john ame preacher son preacher grandson matern patern preacher it 1956 gilead iowa toward end reverend ame life absorb record famili stori legaci young son never see grow haunt grandfath presenc john tell rift grandfath father elder angri visionari fought abolitionist caus son ardent pacifist he troubl prodig namesak jack john ame boughton best friend lost son return gilead search forgiv redempt told john ame joyou rambl voic find beauti humour truth smallest life detail gilead song celebr accept best worst world offer at heart tale sacr bond father son style stori set dazzl critic reader alik'
```

Creación del Bag of Words

Una vez se ha preprocesado las descripciones de los libros, se representarán los textos obtenidos como Bag of Words (BoW).
Un BoW representa cada texto como un vector de tamaño igual al número de palabras distintas de todos los textos, donde cada elemento del vector es la frecuencia de aparición de dicha palabra en el texto.
Es decir, un BoW es conceptualmente una matriz NxP, donde N es el número de textos procesados (es decir, el número de libros del dataset) y P es el número de palabras totales que existen en el conjunto de textos procesados.
Todo el preprocesamiento del texto es necesario para evitar que el tamaño del BoW sea excesivo, y que se considere que una palabra aparece con menor frecuencia de la real por el uso de distintas formas de esta.
Para el cálculo de las frecuencias de cada palabra, se usa el algoritmo TF-IDF, que otorga más frecuencia a las palabras que aparezcan menos a lo largo de todos los textos.
Las fórmulas para el cálculo del TF-IDF para una palabra x en un texto t son las siguientes:
TF-IDF(x,d) = TF(x,d) * IDF(x).
TF(x,d) = apariciones de la palabra x en el documento d / número de palabras en el documento d.
IDF(x) = número de documentos / número de documentos con ocurrencias de x.

```
from sklearn.feature_extraction.text import TfidfVectorizer


bagOfWordsModel = TfidfVectorizer()
bagOfWordsModel.fit(df_preprocessed['processed_desc'])
textsBoW=
bagOfWordsModel.transform(df_preprocessed['processed_desc'])
textsBoW.shape
print(textsBoW)
```

```
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 227452 stored elements and shape (6810, 20913)>
  Coords        Values
  (0, 250)      0.09614188085354987
  (0, 516)      0.09400861519692674
  (0, 535)      0.08589839991972426
  (0, 557)      0.06766722385815023
  (0, 879)      0.07479361895851114
  (0, 989)      0.40847532189651714
  (0, 1124)     0.09400861519692674
  (0, 1194)     0.08253235645246104
  (0, 1358)     0.09400861519692674
  (0, 1534)     0.09400861519692674
  (0, 1552)     0.0620657257234804
  (0, 2058)     0.047965849724653745
  (0, 2243)     0.09825371527712391
  (0, 2562)     0.07337362807614864
  (0, 2644)     0.10686300228406859
  (0, 3411)     0.06399725246249192
  (0, 3444)     0.05642420673791942
  (0, 4620)     0.10437650748040324
  (0, 4933)     0.07272342928947313
  (0, 4984)     0.06120770822307784
  (0, 5239)     0.0550199318651205
  (0, 5891)     0.08926444338698745
...
  (6809, 16296) 0.30848845745018066
  (6809, 17989) 0.32947681918460286
  (6809, 18567) 0.162668031160442
  (6809, 20484) 0.41459069018501404
```

Ilustración 13.- Bag of Words

Creación de la matriz de distancias

Finalmente, creamos la matriz de distancias, que será una matriz NxN (N siendo el número de libros) donde el valor de matriz[x][y] será la distancia calculada entre las descripciones de los libros correspondientes a la posición X e Y del dataset. Para el cálculo de la distancia se calcula la distancia entre los vectores correspondientes en el BoW, usando la métrica de coseno.

```
from sklearn.metrics import pairwise_distances


distance_matrix=
pairwise_distances(textsBoW,textsBoW ,metric='cosine')
print(distance_matrix.shape)
```

```
(6810, 6810)
```

Ilustración 14.- Tamaño de la matriz de diferencias

Funcionamiento del sistema de recomendación

Se presenta un ejemplo de uso del sistema de recomendación.

Se obtiene un ISBN de un libro sobre el que basar la recomendación, en nuestro caso "The Lord of the Rings" con ISBN 9780007124015 (existen múltiples versiones del libro con el mismo título en el dataset).

A continuación, se recupera la posición del libro en el dataset, y se recupera la fila correspondiente a dicho libro en la matriz de distancias.

Se ordenan las distancias de la fila, y se recuperan las 10 distancias de menor valor (indica mayor similitud entre las descripciones de los libros).

Nótese que se recuperan las posiciones 1:11 de la lista de top_scores, esto se debe a que la primera posición (la menor distancia) siempre será el propio libro buscado, y que en las recomendaciones existen libros con el mismo título, pero que son diferentes.

```
searchTitle = "The Lord of the Rings" #Libro base para las
recomendaciones
indexOfTitle =
df_preprocessed[df_preprocessed['title']==searchTitle].index.values
[0]


distance_scores = list(enumerate(distance_matrix[indexOfTitle]))
ordered_scores = sorted(distance_scores, key=lambda x: x[1])
top_scores = ordered_scores[1:11]
top_indexes = [i[0] for i in top_scores]
df_preprocessed['title'].iloc[top_indexes]
```

	title
6414	The Fellowship of the Ring
3745	The Fellowship of the Ring
2500	The Lord of the Rings
1780	The Fellowship of the Ring
3741	The Hobbit, Or, There and Back Again
5622	The Return of the King
80	The Return of the King
1779	The Fellowship of the Ring
1352	The Hobbit, Or, There and Back Again
3709	The Two Towers

Ilustración 15.- Recomendaciones generadas a partir de la búsqueda

Ejemplo de análisis de sentimientos

El Notebook donde se localiza toda la información mostrada a continuación se encuentra en: Bookflix-ABP/Notebook_Bookflix_Análisis_De_Sentimientos.ipynb at main · Akanenosketch/Bookflix-ABP

El objetivo es implementar un sistema de análisis de sentimientos para clasificar reseñas en positivas o negativas.

Introducción

Para nuestro sistema de valoración, usaremos un algoritmo de clasificación supervisado. Para entrenar y probar nuestro algoritmo, usaremos 2 datasets preparados anteriormente: **training_data.xlsx** y **testing_data.xlsx**, que son distintos datasets pero que tienen la misma estructura. Usaremos las columnas review y label, siendo la primera el texto a valorar y la segunda una etiqueta que indica al algoritmo si el texto es positivo (1) o negativo (0).

Para ambos datasets, inicialmente realizaremos un preprocesado de los textos para que sea más fácil trabajar con ellos, aplicando la *_tokenización_*, la eliminación de *_stopwords_* y la *_stemmización_* de los tokens. Los textos obtenidos del preprocesado se representarán mediante una Bag Of Words (BoW).

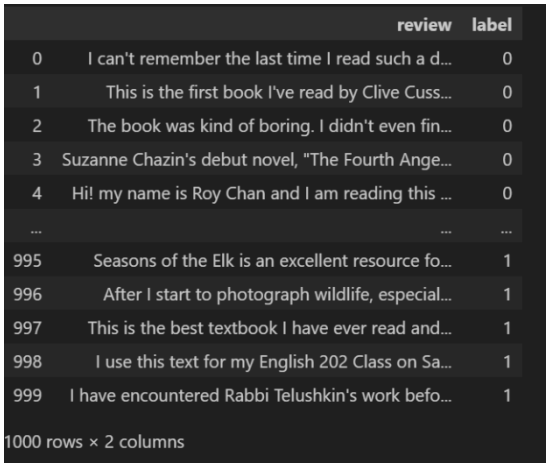
Cargar datos de entrenamiento

Inicialmente cargamos el dataset training_data.xlsx, que contiene información de aproximadamente 1200 reseñas de libros.

Nota: Primero es necesario cargar manualmente el archivo al notebook

Mostramos una vista previa de los datos para comprobar que se han cargado correctamente, y comprobamos cuantos elementos hay positivos y negativos (1 = positivo, 0 = negativo).

```
import pandas as pd
trainingData = pd.read_excel('training_data.xlsx')
#Usaremos solo 1000 reseñas
trainingData = trainingData.head(1000)
trainingData
```

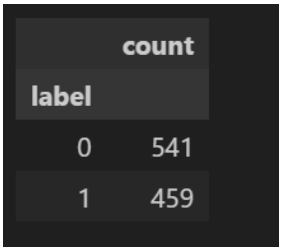


	review	label
0	I can't remember the last time I read such a d...	0
1	This is the first book I've read by Clive Cuss...	0
2	The book was kind of boring. I didn't even fin...	0
3	Suzanne Chazin's debut novel, "The Fourth Ange...	0
4	Hi! my name is Roy Chan and I am reading this ...	0
...	...	...
995	Seasons of the Elk is an excellent resource fo...	1
996	After I start to photograph wildlife, especial...	1
997	This is the best textbook I have ever read and...	1
998	I use this text for my English 202 Class on Sa...	1
999	I have encountered Rabbi Telushkin's work befo...	1

1000 rows × 2 columns

Ilustración 16.- Elementos empleados en el entrenamiento y su asignación

```
trainingData['label'].value_counts()
```



label	count
0	541
1	459

Ilustración 17.- Datos totales de entrenamiento

Preprocesamiento de los datos

En esta celda se preprocesa el texto almacenado en el campo "review" y se almacena en un nuevo DataFrame con nombre "preprocessedData", que contendrá la información del DataFrame original más una columna "preprocessedText".

El preprocesamiento incluye los siguientes pasos:

1. ***Tokenización* del texto** (mediante la librería ***nltk***): Se divide el texto en palabras (tokens), de forma que se obtiene un array compuesto por las palabras que constituyen del texto, separándolas mediante los delimitadores habituales (espacios, comas, puntos...).
2. **Eliminación de *stopwords*** (usando de referencia las stopwords por defecto de la librería ***nltk***): Consiste en eliminar palabras que no aportan significado al texto (stopwords), como pueden ser artículos, pronombres y preposiciones. Las stopwords a eliminar son dependientes del idioma, por lo que el dataset usado debe estar en Ingles.
3. ***Stemmizacion del texto*** (usando el algoritmo ***PorterStemmer***): Se eliminan variaciones de palabras provocadas por conjugación de verbos y por uso de plurales y géneros. Para conseguirlo, se obtiene la raíz semántica de todas las palabras.

Al finalizar el preprocesamiento, se muestran los datos para verificar que el proceso se ha realizado con éxito.

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer

import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('punkt_tab') # Added to resolve a LookupError

ps = PorterStemmer()

preprocessedText = []

for row in trainingData.itertuples():
    text = word_tokenize(row[1]) ## indice de la columna que contiene el texto
    ## Remove stop words
    stops = set(stopwords.words("english"))
    text = [ps.stem(w) for w in text if not w in stops and w.isalnum()]
    text = " ".join(text)

    preprocessedText.append(text)

preprocessedData = trainingData
preprocessedData['preprocessedText'] = preprocessedText

preprocessedData
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
```

	review	label	preprocessedText
0	I can't remember the last time I read such a d...	0	i ca rememb last time i read dire book i howl ...
1	This is the first book I've read by Clive Cuss...	0	thi first book i read clive cussler i doubt i ...
2	The book was kind of boring. I didn't even fin...	0	the book kind bore i even finish read i also t...
3	Suzanne Chazin's debut novel, "The Fourth Ange...	0	suzann chazin debut novel the fourth angel fol...
4	Hi! my name is Roy Chan and I am reading this ...	0	hi name roy chan i read i 7th grade i decid re...
...	...	...	...
995	Seasons of the Elk is an excellent resource fo...	1	season elk excel resourc anyon interest studi ...
996	After I start to photograph wildlife, especial...	1	after i start photograph wildlif especi larg m...
997	This is the best textbook I have ever read and...	1	thi best textbook i ever read way i could unde...
998	I use this text for my English 202 Class on Sa...	1	i use text english 202 class saipan mixtur cha...
999	I have encountered Rabbi Telushkin's work befo...	1	i encount rabbi telushkin work revis text agia...

1000 rows × 3 columns

Ilustración 18.- Resultado del preprocesado de los datos

Creación del Bag of Words

Una vez se ha preprocesado las reseñas de los libros, se representarán los textos obtenidos como Bag of Words (BoW).

```
from sklearn.feature_extraction.text import TfidfVectorizer


bagOfWordsModel = TfidfVectorizer()
bagOfWordsModel.fit(preprocessedData['preprocessedText'])
textsBoW=
bagOfWordsModel.transform(preprocessedData['preprocessedText'])
textsBoW.shape
```

```
(1000, 8332)
```

Entrenamiento de un algoritmo de clasificación

Emplearemos SVM (Support Vector Machine), que es un algoritmo de aprendizaje automático supervisado.

```
from sklearn import svm
svc = svm.SVC(kernel='linear') #Modelo de clasificación

X_train = textsBoW #Documentos
Y_train = trainingData['label'] #Etiquetas de los documentos
svc.fit(X_train, Y_train) #Entrenamiento
print("Finished")
```

```
Finished
```

Obtención y evaluación de las predicciones

Inicialmente se obtienen las predicciones y se almacenan en el array predictions, y a continuación se evalúan distintas métricas, como pueden ser la precisión (cuantas predicciones fueron correctas, es decir, mide cuantos falsos positivos ocurren), la sensibilidad o recall (cuantos elementos del tipo X se han conseguido predecir, es decir, mide cuantos falsos negativos ocurren), o la f1-score (evaluación conjunta de precisión y sensibilidad).

```
from sklearn.metrics import classification_report

X_test = textsBoWTest #Reviews de testing a evaluar
#Obtención y almacenamiento de las predicciones del clasificador
predictions = svc.predict(X_test)

Y_test = testingData['label'] #Etiquetas reales de los documentos
print (classification_report(Y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.69      0.85      0.76       448
           1       0.85      0.69      0.76       552

    accuracy                           0.76      1000
   macro avg       0.77      0.77      0.76      1000
weighted avg       0.78      0.76      0.76      1000
```

Ilustración 19.- Resultados del aprendizaje