Sistema de Recomendación de Libros por Características

Escuela de Ingeniería
Ingeniería de Ciencias Computacionales
Aprendizaje de Máquina
Professor: Ulises Orozco Rosas
Tijuana, B.C, 2 de Diciembre del 2022

Daniel Agraz
33293
CETYS Universidad
Escuela de Ingeniería
Ciencias Computacionales
daniel.agraz@cetys.edu.mx

David Roldán
27292
CETYS Universidad
Escuela de Ingeniería
Ciencias Computacionales
davide.roldan@cetys.edu.mx

Daniel García
26305
CETYS Universidad
Escuela de Ingeniería
Ciencias Computacionales
danielh@cetys.edu.mx

1. Resumen

El presente trabajo describe una simplificación de los sistemas de recomendación actuales aplicados al ámbito de libros. En él se abordan los objetivos a cumplir para llegar al propósito de recomendar libros basándose en el consumo del usuario. Se ilustra la metodología de investigación a seguir y se presenta un marco teórico con el cual entender el contexto de los sistemas de recomendación. Se muestra la implemmentación del algoritmo aplicado en Python. Se muestra la métrica del Minimum Squared Error para los porcentajes de recomendación que acompañan los resultados obtenidos: que fueron exitosos según los objetivos planteados. Finalmente, se da una conclusión que integra lo visto en el proyecto.

2. Introducción:

Dentro de las aplicaciones móviles que usamos día con día, se encuentra que la mayoría de ellas en especial las de redes sociales, se emplea un sistema de recomendaciones para sugerir contenido que probablemente le guste al usuario, dicha funcionalidad suele estar implícita en el algoritmo del programa, de modo que el usuario no es capaz de manipular esta configuración. La única manera de controlar lo que te recomienda una aplicación es en base a la conducta que el usuario presenta al navegar por la misma.

Este proyecto busca desarrollar un sistema de recomendaciones de libros, a partir de un modelo de aprendizaje supervisado. Inspirados por las tecnologías como DALL-E 2 o Stable Diffusion, se interesa implementar un sistema de recomendación basado en contenido de forma explícita para los libros que consume un lector determinado.

La motivación que se tuvo para seleccionar este proyecto, tiene origen por la necesidad de haber pasado por la situación de buscar una cierta recomendación de un libro a partir de los libros que se leyeron previamente, también sumado a la categoría o género que pertenece el libro.

3. Descripción del Problema

El problema surge a partir de la necesidad de un usuario de poder obtener recomendaciones precisas de libros que le pueden interesar rentar basándose en el historial de rentas que ha realizado. Pues actualmente, en la Biblioteca Luis Fimbres Moreno de la institución educativa de CETYS Universidad Campus Tijuana a pesar de que hay un amplio catálogo de libros del cual escoger, no hay aún un sistema de recomendación que permita al usuario descubrir más libros que se acerquen a sus intereses.

Por lo que se propone, como solución, diseñar e implementar un sistema de recomendación basado en el conocimiento de las características individuales de los libros.

4. Objetivos

4.1. Objetivo Principal

Implementar un algoritmo de recomendación capaz de hacer una sugerencia que se apegue lo más posible a las preferencias del usuario dado su historial de rentas de libros.

4.2. Objetivos Complementarios

- Hacer uso de los conocimientos adquiridos durante el curso de Aprendizaje de Máquina: en especial aquellos relacionados al aprendizaje supervisado.
- Recolectar una buena muestra de datos representativos (en cantidad suficiente) de preferencias generales de libros.
- Proponer un sistema de recomendación sencillo pero que sea ejemplo de las capacidades de estos sistemas de abrir puertas para posiblemente incrementar la satisfacción de los clientes en caso de que sea un servicio, acelerar el contacto que el cliente tiene con múltiples productos o simplemente ser una herramienta adicional a la toma de decisiones.

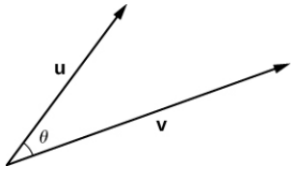
5. Marco Teórico

Los sistemas de recomendaciones están implementados en la mayoría de las cosas que realizamos, en su mayoría no nos percatamos de ellos, hasta muchas veces pueden hasta predecir lo que quieres ver, como es el caso de Youtube. En otros casos de recomendaciones, entraría Amazon donde mediante un algoritmo sugiere posibles artículos relaciones, Spotify donde te sugiere canciones parecidas una vez esta termina de tu lista eleccionada. El significado de un sistema de recomendaciones es una clase de aprendizaje máquina que usa data para ayudar a predecir, reducir y asi encontrar que es lo que esta persona está buscando, entre un valor exponencialmente grande(What is a Recommendation System?, 2021).

Según Banik R. (2018) un sistema de recomendación es ”systems or techniques that recommend or suggest a particular product, service, or entity. However, these systems can be classified into the following two categories, based on their approach to providing recommendations.”

El tipo de sistema de recomendaciones que se busca implementar, es un sistema basado en conocimiento (Knowledge-Based Recommender System) que según Wu J. ”A recommender system is knowledge-based when it makes recommendations based not on a user’s rating history, but on specific queries made by the user” (2019). Este método tiene un acercamiento distinto a lo que se conoce como $coldstartproblem$ que hace referencia a la situación cuando un usuario inicia la aplicación por primera vez, de modo que es complicado empezar a realizar predicciones porque no se tiene registro alguno sobre cuales son sus preferencias en ese momento.
Por lo anterior, para tener una buena precision, se le deberan dar buenas descripciones a los items ya que no se tiene una medida de ”rating” en los datos a emplear, que facilitaria esta tarea, sino que recae enteramente en contenido de los items previamente conocidos.

En cuestion a la operacion encargada del desarrollo de recomendaciones dadas es una operación producto punto de matriz, donde la definición de esta operación multiplicamos la longitud de a por la longitud de b, luego multiplicamos por el coseno del ángulo entre a y b. Sin embargo en python contamos con funciones que nos facilitan estas operaciones. La siguiente muestra la representación de de donde sale la formula.



Otra gran herramienta que nos facilito el trabajo es una simplificacion de bag of words, donde se cambio el funcionamiento a unicamente letras y se llamo bag of letters. La defición del bag of words, se podria definir como un procesamiento de lenguaje natural, donde se añaden palabras en un bolsa, que podria ser una simple lista o un arreglo, dentro de ella se contaran la cantidad de palabras que hay de cada de una ellas, de esta forma tendriamos una arreglo con toda esta cantidad, como se puede apreciar en la imagen posterior.

6. Metodología de la Investigación

1) Se hará la decisión general del ámbito en el que se enfocará la recolección de datos.
2) Se procura una investigación en la literatura de los algoritmos de recomendación más comunes.

3) Se tomará una decisión basados tanto en la cantidad y tipo de datos, necesarios como la dificultad de implementación dadas las limitaciones de tiempo.

4) se hará una búsqueda exhaustiva en datasets públicos según los criterios del algoritmo o si se requiere hacer un web scraping para obtener datos más adecuados a la necesidad.

5) Implementar y probar el algoritmo.

6) Analizar resultados y métricas obtenidas.

7. Implementación

Como se mencionó anteriormente se utilizará un modelo de aprendizaje supervisado, pues a cada libro se le asigna una etiqueta de acuerdo a su género y temática, donde los elementos con más similitudes en común suelen pertenecer a un mismo grupo, esto siendo un problema de clasificación donde a partir de un modelo de entrenamiento se pasan los datos por un filtro que los clasificará como similares al grupo de datos considerado como las preferencias del usuario, y por ende similares a los gustos de un usuario. De forma práctica, lo anterior podría ser implementado usando las librerías de Pandas, Numpy y Matplotlib de Python.

La plataforma en que se llevara a cabo el proyecto será en Google Colaboratory o mejor conocido como Colab, cuyo ambiente de desarrollo será de mucha utilidad para programar el sistema de recomendación, en tiempo de real en equipo.

7.1. Conjunto de Datos

El conjunto de datos que se definió para probar nuestro sistema de recomendaciones, fue el conjunto de datos que se encuentra en la plataforma de Kaggl ,correspondiente al sitio web GoodReads ,con un total de datos acumulados de 10,352 datos, estos corresponden a los mejores libros del siglo XXI que han tenido las mejores reseñas de la plataforma por los usuarios de la misma.

En cuestión al conjunto de datos, fue algo complicado al principio encontrar por los atributos que nos interesa, donde el atributo de género fue algo que no se encontró este atributo en otros conjuntos de datos, además que se acerca frente a tener libros en español por el hecho que lo queremos aplicar al catálogo de la biblioteca de la universidad. La decisión final del conjunto llamado “Goodreads-books”,los atributos con los que cuenta el conjunto se encuentran:

id -> Identificador unico del libro.
title -> Descripción del libro
series -> Series de libros a la que pertenece.
author -> Autor del libro
book link -> Enlace del libro en la plataforma.
genre -> Género literario del libro.
date published ->fecha de publicación.
publisher -> editorial publicadora.
num of pages -> numero de paginas.
lang -> Lenguaje del libro.

7.2. Explicación de codigo

Empezando con las librerias utilizadas en el proyecto fueron las siguiente librerias:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import ast
import itertools

```

Para manipular los datos se descargó el “dataset” de kagglecon extensión csv (comm-separated value) con el propósito de cargar el archivo con más facilidad al colab, no obstante, al abrir el archivo en exce algunas celdas tenían errores de codificación por utilizar el alfabeto en español, además de que los isbn de los libros estaban en formato de notación científica. Todas estas imprecisiones en los datos tienen que pasar un proceso de limpiado de datos, para asegurar el orden y la interpretabilidad del conjunto.

La forma seleccionada para cargar los datos fue la siguiente:

```

bb = pd.read_csv('https://raw.githubusercontent.com/Magnus1515/PruebaGithub/master/Best_Book_21st.csv')
bb = bb.fillna(0)
bb = bb.drop_duplicates(subset='title')
bb.head(2)
```

Donde como se puede observar aplicaríamos la función de pandas para leer archivos csv y posteriormente aplicamos un head() para confirmar que nuestros datos fueron cargados correctamente y mostrados como lo requerimos.

Con los datos cargados, pasaremos a acomodar nuestros datos, para esto nombraremos nuestros atributos así como especificando el tipo de dato que se utilizará, además de que eliminaron espacios vacíos para evitar problemas a futuro, con esto quedaría listo nuestros atributos para ser utilizados.

Habiendo pasado por la etapa de preprocesamiento, ahora toca definir el criterio del recomendado que se encargará de promover determinado libro con respecto a otro. Se tomaron en cuenta dos atributos: Descripción y Género. Descripción se refiere a una breve sinopsis que presente la premisa del libro, de otra manera, el género busca catalogar el género literario del libro o tipo de texto

que contiene. Al final se optó por basar la recomendación de libro basándonos en el género del libro al género del libro, de tal manera que, se recomienden libros tomando en cuenta el género de los libros que has leído y calificado.

El paso siguiente es realizar uno de los pasos más importantes para nuestro sistema de recomendaciones es ingresar los valores de rating o calificación con esto ya tendríamos el input requerido para proceder con los siguientes pasos. Con esto mostramos una tabla con el id, title, author, genre y rating. Todo esto para tener una representación más fácil y cómoda de los datos relevantes, para este instante.

```
 2
 3 inputUser = [
 4             {'title':'The Shack', 'rating':1},
 5             {'title':'The Boy in the Striped Pajamas', 'rating':2},
 6             {'title':'11/22/63', 'rating':5},
 7             {'title':'The Help', 'rating':4},
 8             {'title':'The Fault in Our Stars', 'rating':1}
 9         ]
10 inputBooks = pd.DataFrame(inputUser)
11 inputBooks
```

El siguiente paso es busca el libro y la asigna su id y genero, asi como que se elimina las columnas que no son requeridas para nuestro sistema de recomendación, asi mismo se elimina los duplicados con el mismo id.

```
 2
 3
 4 inputID = bb[bb['title'].isin(inputBooks['title'].tolist())]
 5 inputBooks = pd.merge(inputID, inputBooks)
 6 inputBooks.drop(columns = ['book_link','series','date_published','num_of_page',
 7                            'publisher','award','rating_count', 'review_count',
 8                            'rating_count','rate','lang'], inplace = True )
 9 inputBooks = inputBooks.drop_duplicates(subset='id')
10 inputBooks
```

Para encontrar las coincidencias del género con respecto a otros libros, se tuvo que crear un “Data Frame ”que contenga el género literario de todos los libros del conjunto de datos, además de que se segmentan las categorías de libros en un “bag of letters”, asimismo se tendrá una matriz que contenga todas las letras componen a los géneros de los libros, y si coincide una letra del “bag of letters” con el género del libro se marca como un 1. En el código se remueven columnas innecesarias, y se hace el metodo bag of letters como se comentó, separando todas las letras de todos los géneros, y al pasar por el género de cierto libro marca con uno si ese género tiene una letra y 0 sino Eso se hace para todos los libros del conjunto de datos.

```
 4 BooksWithGenres_df = bb.copy()
 5
 6 for index, row in bb.iterrows():
 7     if row['genre'] == 0:
 8       continue
 9     for genre in row['genre']:
10         BooksWithGenres_df.at[index, genre] = 1
11
12 BooksWithGenres_df = BooksWithGenres_df.fillna(0)
13 BooksWithGenres_df.drop(columns = ['book_link','series','date_published','num_of_page',
14                            'publisher','award','rating_count', 'review_count',
15                            'rating_count','rate','lang'], inplace = True )
16 BooksWithGenres_df.head(2)
```

El atributo de genero es seraparado o divididos por comas y esto lo guardamos en una variable, asi mismo comprobamos su funcionamiento con un output.

```
 4 BooksWithGenres_df['genre'] = BooksWithGenres_df.genre.str.split(',')
 5 BooksWithGenres_df
```

Después se busca en esos titulos de bag of letters los libros, el input del usuario para obtener sus bag of letters. Posterior a ello se buscan estos titulos del bag of letters de los libros, asi con esto conseguimos el input de los libros del usuario junto al bag of letters de sus generos.

```
 2
 3
 4 userBooks = BooksWithGenres_df[BooksWithGenres_df['id'].isin(inputBooks['id'].tolist())]
 5 userBooks
```

En la imagen siguiente, simplemente lo que hacemos nuevamente es eliminar las columnas que no requirimos asi como resetar nuestros ínidces, para así de esta forma contar con nuevos datos.

```
 3 userBooks = userBooks.reset_index(drop = True)
 4 userGenreTable = userBooks.drop(columns = ['id', 'title','author','genre'])
 5 userGenreTable
```

Se muestra como quedaría aplicado el dataframe a la variable $inputBooks$

```
 2
 3 inputBooks['rating']
```

Se obtiene los scores que estos se obtienen a traves de hacer una operación producto punto con la lista de ratings y la matriz de bag of letters de cada libro, como se muesta en la siguiente imagen:

```
 2 userProfile = userGenreTable.transpose().dot(inputBooks['rating'])
 3 userProfile
```

Mostramos la matriz de letras de todo los libros en nuestro conjunto de datos de libros y esto en base a su genero.

```
 2 genreTable = BooksWithGenres_df.set_index(BooksWithGenres_df['id'])
 3 genreTable = genreTable.drop(columns = ['id','title','author','genre'])
 4 genreTable.head
```

Se hace el cálculo del porcentaje de similitud entre los libros de input y los del dataset: multiplica los scores de los inputs y la matriz de bag of letters y los suma para cada cada uno, al final divide esa cantidad entre la suma de los scores del input, con esto se consigue nuestro valor esperado el cual es el valor de coincidencia por asi llamarlo, donde dicha operacion se obtiene entre todos los libros de nuestro conjunto de datos. Esto se realiza con la imagen siguiente.

```
 4 recommendationTableDF = ((genreTable * userProfile).sum(axis=1))/(userProfile.sum())
 5 recommendationTableDF.head()
```

En este penúltimo dato, lo que se realiza únicamente es el ordenamiento de mayor a menor, como muestra del output nosotros manejamos 50 valores estos los acomodamos y de esta variable es donde sacaremos unos datos con mayor indice de conciencia, de ello se seleccionaron solo 5 valores.

```
recommendationTableDF = recommendationTableDF.sort_values(ascending = False)
recommendationTableDF.head(50)
```

Por ultimo paso lo que hacemos es buscar lo datos de los ”id”, que nos salieron con el mayor tasa de coincidencia, de ellos, lo acomodamos con valores que son de interes para el usuario y este output seria nuestro valor final de salida, las 5 mayores recomendaciones segun sus 5 libros seleccionados que se proporcionaron.

```
bb = bb.loc[bb['id'].isin(recommendationTableDF.head(5).keys())]
bb = bb.drop(columns = ['book_link','publisher','num_of_page', 'lang',
                        'review_count', 'rating_count', 'rate', 'award'])
bb.drop(columns = ['series'])
```

8. Descripción de Resultados

El proceso que se empleó para recolectar los resultados del sistema de recomendación propuesto consistió de las siguientes etapas: Primero se definió un $input$, para que simulara la interacción de un usuario con el sistema. Este dato de entrada indica cinco libros dentro de los cuales se califica bajo un atributo de ”rating” que puede ser tomado en una escala tanto del 1 al 5 como del 1 al 10. Más adelante se realizaron otras pruebas que comprueban que modificar este atributo genera una ligera diferencia en la recomendaciones que produce con un mismo input.

En esta segunda fase se ejecuta el algoritmo para realizar la operación matricial de producto punto entre los $ratings$ de los libros del usuario, y el $bagofwords$ que conforma todas las letras que aparecen en el atributo de género del conjunto de datos. Luego se hace una división entre la sumatoria de la matriz de letras multiplicado por el perfil del usuario y la sumatoria del perfil del usuario. Realizando estos pasos se obtiene un dataframe que almacena el porcentaje de coincidencias de todos los libros con respecto al input. Esta coincidencia representa la métrica de similitud que tiene libro por sus géneros respecto al género de los libros del $input$.

Y finalmente, en la útlima etapa, se ordena de mayor a menor el listado de similaridades para obtener los libros que son clasificados por el sistema como los que mas se asemejan a los libros que el usuario de entrada. Normalmente este sistema genera un porcentaje de coincidencia para todo el conjunto de datos, por tanto, solo se extraen los primeros cinco que representan el resultado final.

A partir de este resultado se realizaron tres iteraciones. Como es un sistema de recomendación de libros en base a contenido, se requiere saber por completo todas las características del conjunto de datos, para así, otorgar una recomendación más acertada. También cabe destacar que para estas pruebas se busco utilizar libros conocidos y populares dentro de la cultura júvenil. Otros aspecto a denotar es el hecho de que se usaron libros cuyo título no tuviera números, símbolos o signos de puntuación, esto se hizó con el motivo de conservar la integridad del $bagofletters$ y no involucrar cualquier tipo de cáracter ASCII.

Para la primera iteración se empleó el siguiente $input$. Preferentemente se considera que para dar obtener mejores resultados se pongan dos libros con una calificación alta y otros tres con una calificación por debajo de la mitad.

```
# Se da un input de entrenamiento poniendo titulos y asignan
inputUser = [
            {'title':'The Night Circus', 'rating':1},
            {'title':'The Book Thief', 'rating':2},
            {'title':'Kafka on the Shore', 'rating':5},
            {'title':'Life of Pi', 'rating':4},
            {'title':'The Fault in Our Stars', 'rating':1}
         ]
inputBooks = pd.DataFrame(inputUser)
inputBooks
```

El input se almacena en una lista que maneja varias estructuras de diccionarios. Cada diccionario contiene el título y el $rating$ por libro, de modo, que una ejecutada la celda la librería de pandas la presenta de la siguiente manera.

	title	rating
0	The Night Circus	1
1	The Book Thief	2
2	Kafka on the Shore	5
3	Life of Pi	4
4	The Fault in Our Stars	1

Esta siguiente tabla representa el input del usuario con su respectivogénero la cual se modifica más adelante para así generar el perfil del usuario para catalogar sus gustos.

	id	title	author	genre	rating
0	699	The Book Thief	Markus Zusak	Historical,Historical Fiction,Fiction,Young Ad...	2
1	707	Life of Pi	Yann Martel	Fiction,Fantasy,Classics,Adventure,Contemporar...	4
2	710	The Fault in Our Stars	John Green,Fabiola Stevenson	Young Adult,Romance,Fiction,Contemporary,Reali...	1
3	740	Kafka on the Shore	Haruki Murakami,Philip Gabriel	Fiction,Magical Realism,Fantasy,Cultural,Japan...	5
4	746	The Night Circus	Erin Morgenstern	Fantasy,Fiction,Romance,Historical,Historical ...	1

Una vez que se tiene el perfil del usuario se pasa por la segunda fase del proceso que se explicó previamente y se genera una tabla de recomendaciones como la siguiente.

id	
9989	0.947059
8409	0.941176
5962	0.938235
758	0.938235
5953	0.932353
1805	0.932353
8569	0.932353
4586	0.926471
4049	0.923529
639	0.923529
6331	0.923529
707	0.923529
6090	0.920588
8685	0.920588
7991	0.917647
9062	0.917647
8057	0.917647
8821	0.917647
6838	0.917647
7497	0.917647

Esta tabla de recomendación enlista todos los libros con su porcentaje de similaridad correspondiente, así como el identificador del libro. El identificador hace referencia a la posición que ocupa en el archivo .csv.

	id	title	author	genre	date_published
758	758	The Brief Wondrous Life of Oscar Wao	Junot Díaz	Fiction,Contemporary,Magical Realism,Novels,Li...	September 6th 2007
5953	5953	The Room	Jonas Karlsson,Neil Smith	Fiction,Contemporary,Magical Realism,Fantasy,C...	February 17th 2015
5962	5962	by George: A Novel	Wesley Stace	Fiction,Historical,Historical Fiction,Magical ...	August 25th 2008
8409	8409	The Golden House	Salman Rushdie	Fiction,Contemporary,Literary Fiction,Literatu...	2017
9989	9989	Swamplandia!	Karen Russell	Fiction,Magical Realism,Fantasy,Contemporary,L...	February 1st 2011

Para cerrar con esta primera iteración, estos fueron los resultados finales que arrojo con el input definido al principio. En primera instancia, se puede apreciar que los cinco libros que forman parte de la recomendación final del sistema tienen una gran similitud entre los géneros del libro con mayor rating.

id	
9989	0.945799
5962	0.937669
8409	0.932249
758	0.929539
8569	0.926829

Figure 1. Estos fueron los 5 libros recomendados que arrojo el sismtema. En este listado los libros el porcentaje de coincidencia esta ordenado de mayor a menor

En la figura 1. se observa que el listado de recomendaciones se compone por el identificador del libro y su respectivo porcentaje de coincidencia. En este estado, el resultado es poco interpretable debido a que no conocemos realmente de que libro se esta tratando, por ello, se tuvo que enlazar el atributo id del resultado con el conjunto de datos completo para hallar que libros son los que se están recomendando.
El libro con mayor calificación termino siendo ”Kafka on the Shore”, el cual fue catalogado exactamente con 12 géneros, tales como Ficción, Realismo Mágico, Novela, Ficción Literaria, Ficción Histórica, Literatura entre otros.

A parte se realizó un análisis de las coincidencias reales que hay entre las recomendaciones que dió el sistema con el libro mejor calificado. Dentro de los libros recomendados se encontró que tres libros coincidieron con 7 de los 12 géneros de ”Kafka on the Shore”, 1 libro tuvo 6 de 12 géneros y 1 libro coincidió con 4 géneros.

Para la segunda iteración se probó tener dos libros que tuvieran la máxima calificación a partir de la escala propuesta.

```
# Se da un input de entrenamiento poniendo titulos y asignando ratin
inputUser = [
            {'title':'The Shack', 'rating':1},
            {'title':'The Boy in the Striped Pajamas', 'rating':5},
            {'title':'11/22/63', 'rating':5},
            {'title':'The Help', 'rating':2},
            {'title':'Water for Elephants', 'rating':1}
         ]
inputBooks = pd.DataFrame(inputUser)
inputBooks
```

	title	rating
0	The Shack	1
1	The Boy in the Striped Pajamas	5
2	11/22/63	5
3	The Help	2
4	Water for Elephants	1

Como se esta tratando con un sistema de recomendación en base a contenidos, es importante que los libros que el usuario seleccione sean parte del conjunto de datos y que este escrito de la misma manera a como se presenta en el conjunto de datos.

	id	title	author	genre	rating
0	703	The Help	Kathryn Stockett	Fiction,Historical,Historical Fiction,Historic...	2
1	713	Water for Elephants	Sara Gruen	Fiction,Historical,Historical Fiction,Romance,...	1
2	774	The Shack	William Paul Young	Fiction,Christian,Christian Fiction,Religion,S...	1
3	777	The Boy in the Striped Pajamas	John Boyne	Historical,Historical Fiction,Fiction,Young Ad...	5
4	785	11/22/63	Stephen King	Fiction,Historical,Historical Fiction,Science ...	5

id	
4946	0.965616
283	0.959885
3309	0.957020
3544	0.954155
8172	0.954155
7087	0.948424
6599	0.948424
8335	0.948424
7727	0.942693
516	0.942693
2698	0.942693
5115	0.942693
478	0.939828
269	0.939828
5064	0.939828
8599	0.939828
8411	0.934097
3496	0.934097
6968	0.934097
8310	0.934097
6916	0.934097
3477	0.931232
776	0.931232

De acuerdo a la tabla de recomendaciones que arroja para esta segunda corrida, la tasa de porcentaje de coincidencias es mayor que la pasada, ya que, el primer resultado empieza desde 0.9656%, mientras que con la primera iteración era de 0.9470%. Esto comprende que el sistema tiene la capacidad de recolectar efectivamente todos aquellos libros que cumplan con las características del género de los libros de entrada.

	id	title	author	genre	date_published
283	283	The Past (VanWest, #1)	Kenneth Thomas	Science Fiction,Time Travel,Science Fiction,Sc...	May 20th 2020
3309	3309	Forgive Me, Leonard Peacock	Matthew Quick	Young Adult,Contemporary,Fiction,Realistic Fic...	August 13th 2013
3544	3544	Agatha Heterodyne and the Beetleburg Clank (Gi...	Phil Foglio,Kaja Foglio,Brian Snöddy	Science Fiction,Steampunk,Sequential Art,Graph...	August 12th 2002
4946	4946	America	E.R. Frank	Young Adult,Realistic Fiction,Mental Health,Me...	August 1st 2003
8172	8172	Agatha Heterodyne and the Monster Engine (Girl...	Phil Foglio,Mark McNabb,Kaja Foglio	Science Fiction,Steampunk,Sequential Art,Graph...	December 3rd 2013

Estos terminaron siendo los resultados finales de esta segunda iteración. Esta corrida fue la que mejor resultados tuvo, esto es provocado debido a la calificación de los libros que fueron escritos como datos de entrada.

Para la última iteración, se tomó el mismo acercamiento respecto a la segunda iteración, de modo que en el *input* se tuvieron dos libro con la calificación más alta.

	title	rating
0	The Glass Castle	1
1	Cloud Atlas	5
2	The Graveyard Book	2
3	The Corrections	3
4	The Namesake	5

	id	title	author	genre	rating
0	730	The Glass Castle	Jeannette Walls	Nonfiction,Autobiography,Memoir,Biography,Biog...	1
1	733	Cloud Atlas	David Mitchell	Fiction,Science Fiction,Fantasy,Historical,His...	5
2	742	The Graveyard Book	Neil Gaiman,Dave McKean	Fantasy,Young Adult,Fiction,Horror,Childrens,M...	2
3	747	The Corrections	Jonathan Franzen	Fiction,Contemporary,Novels,Literary Fiction,L...	3
4	764	The Namesake	Jhumpa Lahiri	Fiction,Cultural,India,Contemporary,Literary F...	5

id	
9265	0.941176
8599	0.931765
367	0.931765
7715	0.931765
8335	0.929412
2366	0.927059
776	0.927059
9956	0.924706
484	0.924706
7538	0.924706
260	0.920000
898	0.920000
4841	0.920000
7382	0.920000
8277	0.920000
5813	0.917647
2682	0.917647
639	0.917647
329	0.915294
283	0.915294
2986	0.912941
2273	0.912941
6193	0.912941
4105	0.912941
2528	0.912941

	id	title	author	genre	date_published
367	367	Out Stealing Horses	Per Petterson,Anne Born	Fiction,Historical,Historical Fiction,European...	April 17th 2007
7715	7715	Generosity: An Enhancement	Richard Powers	Fiction,Literature,Science Fiction,Novels,Cont...	September 29th 2009
8335	8335	It's Not About the Tapas: A Spanish Adventure ...	Polly Evans	Travel,Nonfiction,Cultural,Spain,Autobiography...	June 27th 2006
8599	8599	Maya's Notebook	Isabel Allende,Anne McLean	Fiction,Contemporary,Young Adult,Novels,Young ...	April 23rd 2013
9265	9265	The Most Beautiful Thing	Kao Kalia Yang,Khoa Le	Childrens,Picture Books,Family,Childrens,Nonfi...	October 6th 2020

También se comprobó la capacidad bidireccional que tiene el sistema de hacer recomendaciones, de modo que, el resultado de una iteración sea la entrada de otra. Utilizando este acercamiento recursivo con el sistema, se pudo comprobar la relación del atributo de género con los libros, ya que se obtuvieron recomendaciones muy similares.

Ahora para la evaluación de este sistema de recomendación se empleó la métrica MSE (Mean Squared Error) con el porcentaje coincidencia como el valor estimado y se tomó como valor real 1 representando 100% de similaridad por el género del libro.

Para la metodología de esta métrica se trabajó con las primeras 50 recomendaciones que arroja el modelo por iteración. De modo que se aplica la métrica MSE a este número de observaciones.

Después se calculó el promedio del error cuadrático por corrida, obteniendo 0.0074803,0.004171 y 0.0071194, respectivamente. Esto de muestra que el sistema tiene una muy baja tasa de error a la hora de realizar las recomendaciones, tomando en cuenta los primeros 50 resultados.

%Coincidencia	Valor Real		mse	AVG_mse	
0.945799	1		0.0029377	0.0074803	*1era Corrida*
0.937669			0.0038852		
0.932249			0.0045902		
0.929539			0.0049648		
0.926829			0.005354		
0.924119			0.0057579		
0.924119			0.0057579		
0.921409			0.0061765		
0.921409			0.0061765		
0.915989			0.0070578		
0.915989			0.0070578		
0.915989			0.0070578		
0.915989			0.0070578		
0.915989			0.0070578		
0.915989			0.0070578		
0.915989			0.0070578		
0.915989			0.0070578		
0.913279			0.0075205		

%Coincidencia	mse	AVG_mse	
0.965616	0.0011823	0.004171904	*2da Corrida*
0.959885	0.0016092		
0.95702	0.0018473		
0.954155	0.0021018		
0.954155	0.0021018		
0.948424	0.0026601		
0.948424	0.0026601		
0.948424	0.0026601		
0.942693	0.0032841		
0.942693	0.0032841		
0.942693	0.0032841		
0.942693	0.0032841		
0.939828	0.0036207		
0.939828	0.0036207		
0.939828	0.0036207		
0.939828	0.0036207		
0.934097	0.0043432		
0.934097	0.0043432		
0.934097	0.0043432		
0.934097	0.0043432		

%Coincidencia	mse	AVG_mse	
0.941176	0.0034603	0.007194377	*3era Corrida*
0.931765	0.004656		
0.931765	0.004656		
0.931765	0.004656		
0.929412	0.0049827		
0.927059	0.0053204		
0.927059	0.0053204		
0.924706	0.0056692		
0.924706	0.0056692		
0.924706	0.0056692		
0.92	0.0064		
0.92	0.0064		
0.92	0.0064		
0.92	0.0064		
0.92	0.0064		
0.917647	0.006782		
0.917647	0.006782		
0.917647	0.006782		
0.915294	0.0071751		
0.915294	0.0071751		

9. Conclusión

Para concluir, el propósito principal del proyecto fue logrado con éxito: se creó un modelo similar a los sistemas de recomendación actuales en relación a las preferencias de libros de un usuario. Un sistema basado en contenido, que usó el método bag of letters, dando 5 recomendaciones de libros con un muy buen porcentaje de similitud a las preferencias ingresadas, demostrando la efectividad del modelo al realizar en repetidas ocasiones el Minimum Squared Error y obtener valores muy bajos de error. Se espera que los resultados obtenidos en esta investigación sirvan para complementar la literatura existente sobre pruebas realizadas sobre sistemas de recomendación con el uso de aprendizaje de máquina.

Los enlaces al archivo de Google Colab y el video de 3 minutos presentando el código, se encuentra adjuntado en la sección de referencias.

10. Bibliografía

1) Wu, J. (2022). Knowledge-Based Recommender Systems: An Overview. Retrieved 13 September 2022, from https://medium.com/@jwu2/knowledge-based-recommender-systems-an-overview-536b63721dba

2) Banik, R. (2018). Hands-On Recommendation Systems with Python (1st ed.). Packt Publishing.

3) NVIDIA Data Science Glossary. 2021. What is a Recommendation System?. [online] Available at: ¡https://www.nvidia.com/en-us/glossary/data-science/recommendation-system/¿ [Accessed 13 September 2022].

4) Link al Colab con el código de la implementación: Colab

11. Video explicativo

Link al video de explicación de la implementación final: VideoProyectoFinal