Filtrado Basado en Contenido

Juan de Jesús Aguilar Solano
Maestría en Ciencia de Datos
Facultad de Ciencias Físico Matemáticas
Universidad Autónoma de Nuevo León

Karla Cureño Vega
Maestría en Ciencia de Datos
Facultad de Ciencias Físico Matemáticas
Universidad Autónoma de Nuevo León

Abril Grisel Guevara Cedillo
Maestría en Ciencia de Datos
Facultad de Ciencias Físico Matemáticas
Universidad Autónoma de Nuevo León

***Resumen*—Este documento muestra el reporte de la implementación del algoritmo de filtrado de contenido utilizando un conjunto de datos extraído mediante una API y preprocesado para el mejor desempeño del modelo.**

I. Introducción

La tecnología digital cada vez se integra más a nuestra vida cotidiana, durante los últimos años las transacciones vía internet se han incrementado de manera considerable, tan solo de 2015-2020 el porcentaje de usuarios en México que realizan transacciones se incrementó en un 20 % [1], al día de hoy la cifra debe ser mucho mayor debido a la pandemia surgida durante el 2019.

Estas transacciones van ligadas a las compras online, de las cuales pueden destacarse la compra de artículos en sitios web como Amazon o Mercado Libre y la compra de membresías para adquirir contenido digital como videojuegos o películas mediante servicios de streaming, esto hace que la inversión en plataformas digitales se haya convertido en algo bastante atractivo.

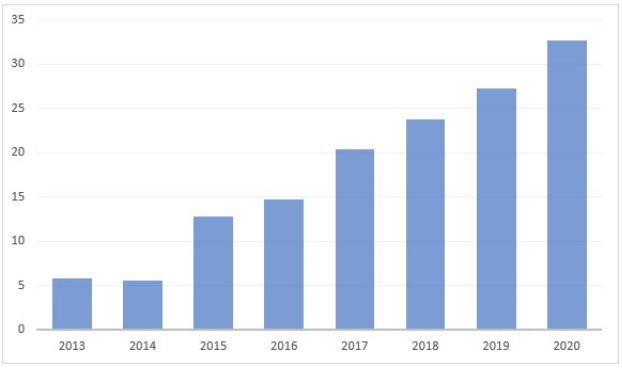


Figura 1. Proporción de usuarios que realizan transacciones vs usuarios de internet.

Para este trabajo se realizará un algoritmo para el filtrado basado en contenido para plataformas de servicio de streaming de películas, el cual consiste en la recomendación de 10 películas basado en el historial de la última película vista.

El proyecto describirá los pasos para el desarrollo de éste, desde la adquisición de datos mediante la descripción del uso de la API (Anexo), la transformación de datos para su uso en el modelo, abordaje de las problemáticas surgidas durante la manipulación de los datos y por último, la prueba del modelo y formas de poder mejorar el algoritmo acorde al propósito que tenga éste.

II. Filtrado Basado en Contenido

El filtrado basado en contenido (FBC) es una de las técnicas de recomendación más exitosas, esta técnica utiliza los atributos de los objetos que el usuario ha consumido, visto o en los que ha mostrado interés para sugerir nuevos objetos con atributos similares basado en el nivel de correlación entre el objeto observado y el objeto sugerido.

Existen diferentes métricas utilizadas para medir el nivel de correlación entre existente entre dos objetos, éstas son llamadas medidas de similitud [2] y algunas de las más usuales son:

II-A. Vectores Similares (Coseno)

Esta técnica calcula el coseno entre dos vectores de atributos de los objetos como medida de correlación, el resultado es un valor que va desde 0 hasta 1, donde 1 implica que los vectores son paralelos y por lo tanto presentan una correlación fuerte entre ellos, mientras que un valor 0 significa que los vectores son perpendiculares entre sí y no presentan una buena correlación. Su fórmula es:

$$\cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{uv} \tag{1}$$

II-B. Coeficiente de Correlación de Pearson

Esta métrica fue propuesta por Karl Pearson como medida de correlación lineal entre dos vectores (muestras), el resultado es un valor que va desde -1 a 1, donde -1 indica una fuerte correlación lineal negativa, 0 una nula correlación lineal y 1 una fuerte correlación lineal positiva. Su fórmula es:

$$\rho(\vec{u}, \vec{v}) = \frac{\sum(u_i - \bar{u}) \cdot (v_i - \bar{v})}{\sqrt{\sum(u_i - \bar{u})^2}\sqrt{\sum(v_i - \bar{v})^2}} \tag{2}$$

II-C. *Distancia Euclidiana*

La métrica calcula la distancia entre dos puntos n dimensionales (donde n = número de atributos) en un espacio Euclidiano, si la distancia entre los puntos es pequeña, indica que ambos objetos presentan características muy similares, mientras que, si aumenta la diferencia entre éstos, la distancia aumenta. La fórmula es:

$$d(\vec{u}, \vec{v}) = \sqrt{\sum (u_i - v_i)^2} \tag{3}$$

II-D. *Distancia Euclidiana NaN*

Esta métrica, es implementada cuando los datos presentan valores nulos, de modo que a la distancia se le asigna un peso proporcional cuando éstas están presentes. La fórmula es:

$$d(\vec{u}, \vec{v}) = \sqrt{w \sum (u_i - v_i)^2} \tag{4}$$

donde w es el peso y se define como $w = \frac{\text{Dimensión del vector}}{\text{Coordenadas presentes}}$

II-E. *Distancia Manhattan*

Es también conocida como cuadras de ciudad (city blocks), la métrica evalúa la suma de las diferencias absolutas, es decir, siempre sumará distancias positivas, lo cual se representa como una trayectoria que se desplaza solamente de manera horizontal y vertical (de ahí su nombre). Su fórmula es:

$$d(\vec{u}, \vec{v}) = \sum |(u_i - v_i)| \tag{5}$$

III. Conjunto de Datos

III-A. *Descripción del conjunto de Datos*

Los datos a utilizar en este proyecto fueron extraídos del sitio web The Movie Data DB a través de su API. Para la obtención de una llave API puede consultarse el archivo readme.md anexo en la carpeta del proyecto.

El conjunto de datos a utilizar en este proyecto consiste en 174,399 películas todas lanzadas después del año 1900 y de las cuales cada una contiene 128 atributos incluyendo el título de la película, la cantidad de atributos se debe a que el género y palabras clave de cada película han sido binarizadas. El resultado es una matriz de datos de dimensiones (174399, 128).

III-B. *Atributos a utilizar dentro del modelo*

Para los propósitos de este proyecto, solo utilizaremos una porción de los atributos mostrados, concretamente éstos son:

- **id**: identificador de la película.
- **original title**: título original de la película.
- **budget**: presupuesto de la película.
- **genres**: lista de géneros a los que pertenecer.
- **popularity**: popularidad de la película (métrica generada por el sitio).
- **revenue**: ingresos recaudados.
- **runtime**: duración de la película.
- **vote average**: calificación media otorgada por usuarios.
- **vote count**: cantidad de usuarios que calificaron la película.
- **collection**: atributo binario que muestra si una película pertenece a una colección.
- **keywords**: las palabras clave utilizadas por película (top 100 keywords).

III-C. *Recopilación de datos por película*

Todos los atributos a excepción del atributo keywords, pueden extraer a partir de la siguiente sintaxis de consulta:

https://api.themoviedb.org/3/movie/[movie_id]?api_key=[api_key]

Para ilustrar la obtención de datos se utilizará la siguiente película como ejemplo:

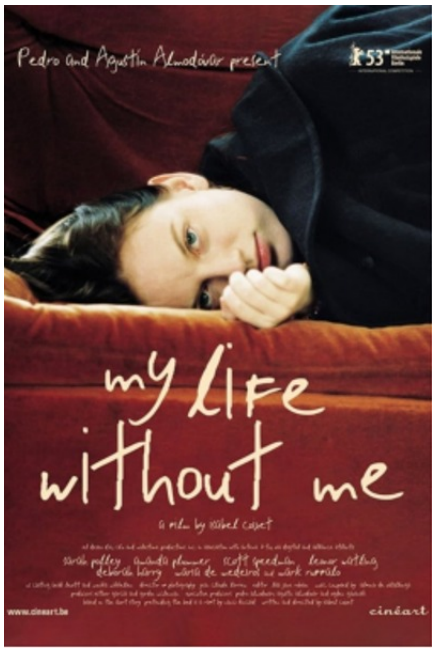


Figura 2. Ejemplo de película.

Id = 20, Nombre: My Life Without Me

Cuya sintaxis de consulta es:

https://api.themoviedb.org/3/movie/20?api_key=fhk6739847

donde movie id=20 es el identificador de la película y api key = “secreta” es la llave generada para el uso de la herramienta.

Para la obtención de la información hacemos uso de la biblioteca `requests` de Python. El resultado es un archivo en formato json, el cual podemos cargar mediante la biblioteca `json`. El resultado de la consulta se muestra en la figura 3.

```
{'adult': False,
 'backdrop_path': '/kZyurQjTMLHalUxs7sHgH5XeiwO.jpg',
 'belongs_to_collection': None,
 'budget': 2500000,
 'genres': [{'id': 18, 'name': 'Drama'}, {'id': 10749, 'name': 'Romance'}],
 'homepage':
'http://www.clubcultura.com/clubcine/clubcineastas/isabelcoixet/mividasinmi/index.htm',
 'id': 20,
 'imdb_id': 'tt0314412',
 'original_language': 'en',
 'original_title': 'My Life Without Me',
 'overview': 'A fatally ill mother with only two months to live creates a list of things
she wants to do before she dies without telling her family of her illness.',
 'popularity': 13.863,
 'poster_path': '/9Fa7tCEKIha11lGH7E41mxSpaF6.jpg',
 'production_companies': [{'id': 49,
   'logo_path': '/xpf5iHdvvBtsH8jBM1gIJHAET0c.png',
   'name': 'El Deseo',
   'origin_country': 'ES'},
  {'id': 77,
   'logo_path': None,
   'name': 'Milestone Productions',
   'origin_country': ''}],
 'production_countries': [{'iso_3166_1': 'CA', 'name': 'Canada'},
  {'iso_3166_1': 'ES', 'name': 'Spain'}],
 'release_date': '2003-03-07',
 'revenue': 12300000,
 'runtime': 106,
 'spoken_languages': [{'english_name': 'English',
   'iso_639_1': 'en',
   'name': 'English'}],
 'status': 'Released',
 'tagline': '',
 'title': 'My Life Without Me',
 'video': False,
 'vote_average': 5.8,
 'vote_count': 364}
```


Figura 3. Resultado de recopilación de datos por película.

III-D. Binarización del atributo género

Dado que el atributo género es una lista, es necesario binarizarla para poder evaluarla en el modelo de FBC. Para ello necesitamos conocer los diferentes géneros que utiliza esta base de datos, esto lo logramos a través de su API mediante la siguiente url:

https://api.themoviedb.org/3/genre/movie/list?api_key=[api_key]&language=en-US

Este enlace nos retorna los géneros oficiales utilizados, los cuales son:


```
{28: 'Action', 12: 'Adventure', 16: 'Animation', 35: 'Comedy', 80: 'Crime', 99:
'Documentary', 18: 'Drama', 10751: 'Family', 14: 'Fantasy', 36: 'History', 27: 'Horror',
10402: 'Music', 9648: 'Mystery', 10749: 'Romance', 878: 'Science Fiction', 10770: 'TV
Movie', 53: 'Thriller', 10752: 'War', 37: 'Western'}
```


Figura 4. Géneros oficiales utilizados en fuente de datos.

Para el ejemplo antes mencionado donde el id = 20, la binarización realiza la siguiente transformación:

Atributo antes de procesarse	Atributo después de procesarse
`[{'id': 18, 'name': 'Drama'}, {'id': 10749, 'name': 'Romance'}]`	`{'Action': 0, 'Adventure': 0, 'Animation': 0, 'Comedy': 0, 'Crime': 0, 'Documentary': 0, 'Drama': 1, 'Family': 0, 'Fantasy': 0, 'History': 0, 'Horror': 0, 'Music': 0, 'Mystery': 0, 'Romance': 1, 'Science Fiction': 0, 'TV Movie': 0, 'Thriller': 0, 'War': 0, 'Western': 0}`

Figura 5. Película ejemplo con géneros binarizados.

III-E. Atributo keywords

El atributo keywords consiste en las palabras clave que son utilizadas por películas, las palabras clave que maneja toda la base de datos puede extraerse a través del siguiente enlace:

http://files.tmdb.org/p/exports/keywords_ids_03_25_2022.json.gz

La extensión de este archivo es de alrededor de 400,00 palabras, lo cual dificulta la implementación del modelo dado nuestra capacidad de hardware. Para este modelo se utilizarán solo 100 palabras clave y el criterio de selección de éstas es realizar un muestreo de 100,000 películas, de las cuales se seleccionarán las 100 palabras con mayor frecuencia, el resultado se muestra a continuación:


```
[short film, woman director, based on novel or book, murder, musical, concert, silent
film, biography, sports, stand-up comedy, lgbt, christmas, world war ii, revenge, family,
love, anime, philippines, based on true story, martial arts, friendship, romance, coming
of age, softcore, black and white, kidnapping, wrestling, opera, new york city, ghost,
based on play or musical, sequel, police, politics, serial killer, horror, pre-code,
found footage, prison, holiday, erotic movie, drugs, vampire, zombie, rape, parent child
relationship, death, gay interest, dance, remake, high school, art, dark comedy, monster,
slasher, gay, religion, stop motion, dog, gore, time travel, supernatural, lost film,
marriage, alien, football (soccer), gangster, detective, superhero, nazi, africa,
suicide, japan, "rock n roll", nature, mockumentary, racism, cartoon, fairy tale, road
trip, satire, pregnancy, spy, small town, mystery, infidelity, investigation, thriller,
noir, los angeles,  california, avant-garde, school, robbery, dutch cabaret, london,
england, 1970s, sibling relationship, surrealism, anthology, france]
```


Figura 6. 100 palabras más frecuente (keywords).

Al igual que el atributo género, este atributo debe binarizarse para ser utilizado dentro del modelo.

III-F. Atributos por película

Los atributos para la película antes mencionada (id = 20) serían los siguientes:

budget	popularity	release_date	revenue	runtime	vote_average	vote_count	Action	Adventure	Animation	Comedy	Crime	Documentary	Drama	Family	Fantasy	History	Horror	Music	Mystery	Romance	Science Fiction	TV Movie	Thriller	War	Western
2500000	13.604	3/7/2003	12300000	106	5.8	364	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0

collection	short film	woman director	based on novel or book	murder	musical	concert	silent film	biography	sports	stand-up comedy	lgbt	christmas	world war ii	revenge	family	love	anime	philippines	based on true story	martial arts	friendship	romance	coming of age	softcore	black and white
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

kidnapping	wrestling	opera	new york city	ghost	based on play or musical	sequel	police	politics	serial killer	horror	pre-code	found footage	prison	holiday	erotic movie	drugs	vampire	zombie	rape	parent child relationship	death	gay interest	dance	remake	high school
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

art	dark comedy	monster	slasher	gay	religion	stop motion	dog	gore	time travel	supernatural	lost film	marriage	alien	football (soccer)	gangster	detective	superhero	nazi	africa	suicide	japan	rock 'n' roll	nature	mockumentary	racism
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

cartoon	fairy tale	road trip	satire	pregnancy	spy	small town	mystery	infidelity	investigation	thriller	noir	los angeles, california	avant-garde	school	robbery	dutch cabaret	london, england	1970s	sibling relationship	surrealism	anthology	france
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 7. Atributos para película ejemplo.

III-G. Directorio de películas de la base de datos

Para extraer la información antes mostrada es necesario conocer el listado de películas con su identificador del ID, la API nos brinda acceso a un directorio de películas, durante este proyecto se procesarán todas las películas del directorio actualizado al día 24 de marzo de 2022, el directorio se encuentra en el siguiente enlace:

http://files.tmdb.org/p/exports/movie_ids_03_24_2022.json.gz

La cantidad de películas a procesar en la base de datos es de 688,316.

III-H. *Obtención de datos por lotes*

Dado que la cantidad de películas es larga, esta se procedió a realizarla en lotes, en total son 4,000 lotes que contienen 172 películas cada uno, esto se hizo así para evitar errores por problemas de conexión o tamaño de memoria, ya que, si existe un error, solo es necesario reiniciar el contador al número de lote en el que se quedó.

El resultado es un archivo csv llamado `dataset_movies.csv` al cual se le va añadiendo los datos procesados por lote, de modo que entre más lotes se procesen, más películas contendrá el conjunto.

III-I. *Manejo de valores nulos*

En muchas películas se desconoce ciertos datos como el presupuesto, fecha de lanzamiento, ingreso recaudado, duración, etc. Hasta este punto se manejarán estos valores nulos cambiándolos por valor 0, además de realizar un chequeo para saber si existen duplicados.

III-J. *Exportación de la base de datos*

Al realizar la iteración por lotes, se obtiene un set de datos en formato csv. El set de datos final es el siguiente:

id	budget	original_title	popularity	release_date	revenue	runtime	vote_average	vote_count	Action	Adventure	...
2	0	Ariel	8.165	10/21/1988	0	73	6.8	157	0	0	...
3	0	Varjoja paratiisissa	8.509	10/17/1986	0	74	7.2	158	0	0	...
5	4000000	Four Rooms	14.441	12/9/1995	4257354	98	5.7	2127	0	0	...
6	21000000	Judgment Night	13.336	10/15/1993	12136938	110	6.5	230	1	0	...
8	42000	Life in Loops (A Megacities RMX)	2.352	1/1/2006	0	80	7.5	18	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
285854	0	Reunited	0.600	1/26/2010	0	90	4.3	2	0	0	...
285855	0	Der Schrecken der Garnison	0.877	4/23/1931	0	86	0.0	0	0	0	...
285856	0	Big Muddy	2.304	9/4/2014	0	104	4.9	6	0	0	...
285857	0	ダロス	2.340	12/21/1983	0	83	6.3	9	0	0	...
285858	0	Corbo	3.649	4/17/2014	0	119	6.0	16	0	0	...

175188 rows × 128 columns

Figura 8. Set de datos resultado de la exportación.

Todo el código y procedimientos abarcados hasta punto pueden encontrarse en el archivo `etl_process.ipynb` y `etl_process_2.ipynb` del proyecto.

IV. Preprocesamiento de Datos

Para usar el conjunto de datos en el modelo, se hará una serie de transformaciones las cuales se describirán a continuación:

- **Eliminación de películas con fecha antes de 1900**:estas películas fueron eliminadas ya que presentan errores de lectura y son películas de relevancia nula para este modelo.
- **Eliminación del título de la película**: dado que el atributo es de tipo texto, no será utilizado para el modelo.
- **Transformación de la fecha a formato ordinal**: para realizar la transformación utilizamos la función `datetime` del paquete `datetime` y utilizando el método `toordinal()`.
- **Reemplazo de los valores 0 a formato NaN**: el motivo de esto es que durante la implementación del modelo usaremos la métrica "distancia euclidiana NaN" la cual ponderiza la distancia de acuerdo a la cantidad de datos faltantes.

El resultado de estas transformaciones se muestra a continuación:

id	budget	popularity	release_date	revenue	runtime	vote_average	vote_count	Action	Adventure	Animation	...
2	NaN	8.165	726031.0	NaN	73.0	6.8	157.0	0	0	0	...
3	NaN	8.509	725296.0	NaN	74.0	7.2	158.0	0	0	0	...
5	4000000.0	14.441	728636.0	4257354.0	98.0	5.7	2127.0	0	0	0	...
6	21000000.0	13.336	727851.0	12136938.0	110.0	6.5	230.0	1	0	0	...
8	42000.0	2.352	732312.0	NaN	80.0	7.5	18.0	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
285854	NaN	0.600	733798.0	NaN	90.0	4.3	2.0	0	0	0	...
285855	NaN	0.877	705030.0	NaN	86.0	NaN	NaN	0	0	0	...
285856	NaN	2.304	735480.0	NaN	104.0	4.9	6.0	0	0	0	...
285857	NaN	2.340	724265.0	NaN	83.0	6.3	9.0	0	0	1	...
285858	NaN	3.649	735340.0	NaN	119.0	6.0	16.0	0	0	0	...

174399 rows × 127 columns

Figura 9. Resultado de la primera parte del preprocesamiento del set de datos.

V. Análisis Descriptivo

V-A. *Tabla descriptiva*

La descripción de los datos excluyendo el género y collection se muestra a través de sus respectivos histogramas. Como primer paso para conocer nuestros datos, podemos observar la siguiente tabla descriptiva, misma que puede obtenerse a partir del método `describe()` de la biblioteca `pandas`.

	budget	popularity	release_date	revenue	runtime	vote_average	vote_count
count	14436	173886	170108	10493	149947	133415	133458.00
mean	14202620	3.079512	725869	50842450	85.02974	5.779741	101.88
std	29028330	8.155153	9566	127524200	132.0611	1.553085	725.56
min	1	0.6	693596	1	1	0.5	1.00
25%	368250	0.6	720259	1071255	70	5	2.00
50%	3000000	1.279	729811	8251071	90	6	6.00
75%	14625000	2.646	733400	40050880	101	6.7	17.00
max	380000000	403.432	739966	2847246000	43200	10	31166.00

A primera vista podemos observar que cada característica presenta diferentes escalas, además la característica "budget" y "runtime" son las que más valores nulos presentan.

V-B. *Histogramas*

A partir de los gráficos podemos observar que la distribución no es uniforme salvo para "vote average" (debido al teorema de límite central), además se presenta un sesgo pronunciado en estos datos y outliers bastante grandes, de modo que para implementar nuestro modelo realizaremos un proceso de normalización.

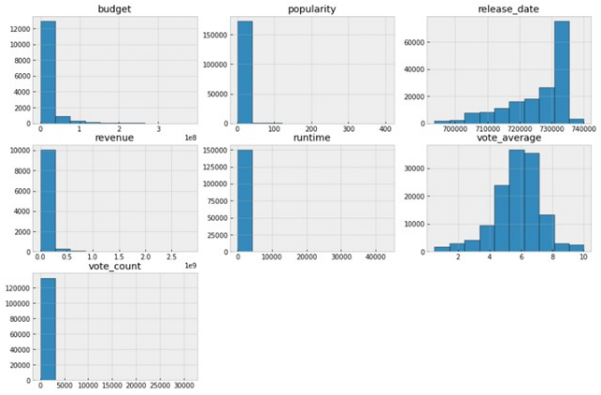


Figura 10. Histogramas de variables numéricas no binarias.

VI. Normalización de Datos

Para normalizar los datos y que presenten un buen desempeño en el modelo, se hicieron las siguientes transformaciones:

- **Reasignación de outliers**: en primera instancia se optó por utilizar la función `MinMaxScaler` de la biblioteca `scikitlearn`, sin embargo, dado que el conjunto de datos presenta datos mucho mayores a los de la media o el rango intercuantílico, los valores asignados a la mayoría de los datos son bastante bajos y, por lo tanto, no contribuyen mucho a la métrica de distancia utilizada en el modelo. El método propuesto para lidiar con este problema fue el de reasignar los valores que están por encima del percentil 90 y asignarles el valor correspondiente a dicho percentil, de esta manera aquellos valores excesivamente grandes serán reevaluados.
- **Escalamiento de datos**: para escalar los datos se utilizó la función `StandardScaler()` de la biblioteca `scikitlearn`, la cual realiza la siguiente operación [5]:

$$z_i = \frac{x_i - u}{\sigma} \tag{6}$$

VI-A. Boxplot

El resultado de estas transformaciones son datos que presentan una escala muy similar y con outliers bastante aceptables, mismos que podemos apreciar en el siguiente diagrama de caja:

VII. Prueba del Modelo

La métrica utilizada en el modelo será la Distancia Euclidiana NaN, la cual se encuentra en la biblioteca `scikitlearn`. Durante el proceso de experimentación se utilizaron otras métricas como la distancia de cosenos y la distancia euclidiana clásica, obteniendo resultados bastante similares, sin embargo, por la naturaleza del conjunto de datos se optó por la distancia euclidiana NaN para este proyecto.

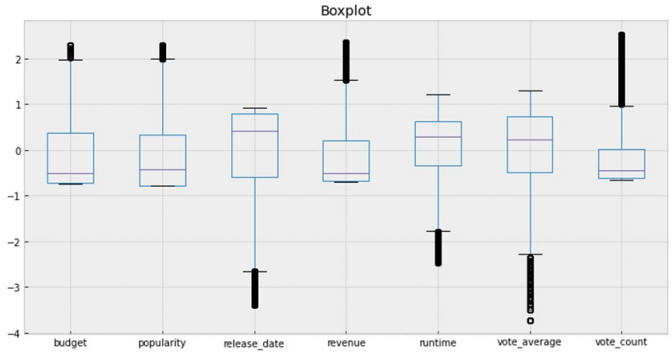


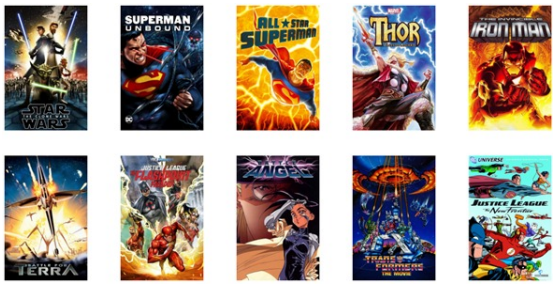
Figura 11. Boxplot de datos normalizados.

Para evaluar el modelo, utilizaremos dos ejemplos de géneros distintos. La primera película pertenece al género de animación, mientras que la segunda película pertenece al género romántico.

Pelicula de animación:

"Batman: Gotham Knight (2/2/2014)"	Lista de recomendaciones
	Star Wars: The Clone Wars Superman: Unbound All Star Superman Thor: Tales of Asgard The Invincible Iron Man Battle for Terra Justice League: The Flashpoint Paradox 銃夢 -GUNNM The Transformers: The Movie Justice League: The New Frontier

Portadas de las películas recomendadas:



El resultado es una recomendación de películas bastante ligada al género seleccionado, sin embargo, algunas de las películas son "anime" y no pertenecen estrictamente al mismo género siendo esto irrelevante para algunos.

Pelicula romántica:

"The Notebook"	Lista de recomendaciones
	오아시스 君に届け 파이란 백만장자의 첫사랑 甜蜜蜜 山楂树之恋 ラブレター Stellet Licht Tabu Welcome

Portadas de las películas recomendadas:

El resultado son una serie de películas pertenecientes al mismo género, donde puede notarse que varias de las películas son originalmente en un idioma diferente al de la película en la que se basa la recomendación.

VIII. Parámetros para mejorar el modelo

Establecer qué modelo es mejor es una tarea bastante subjetiva, además depende del propósito que se tenga y los recursos con los que se cuente en una implementación dada [4]. Para la implementación mostrada no utilizaremos ninguna, sin embargo, describiremos en que consiste cada una:

- **Asignación de pesos a las características del conjunto**: determinar un nivel de importancia para cada característica y realizar el escalamiento acorde al peso asignado.
- **Eliminación/Incorporación de características al conjunto de datos**:
  - Eliminar fecha, algunas palabras clave, géneros, etc.
  - Añadir al conjunto idioma, país, etc.

El proceso de mejoramiento del modelo es una tarea ardua y que debe de llevarse a cabo mediante el criterio humano, sin embargo, puede adoptarse la práctica del caviar que consiste en la realización de muchos modelos con diferentes parámetros y determinar cual puede ser el mejor.

IX. Conclusiones

La extracción de datos se ha llevado a cabo de manera exitosa mediante el uso de la API de "The Movie DB", siendo este un proceso arduo y tardado ya que el sistema no permite obtener los datos mediante una sola consulta, de modo que éste se convierte en un proceso iterativo.

El resultado del modelo basado en las pruebas realizadas parece ser bastante bueno, es decir, proporciona películas con características muy similares, sin embargo, para algunos usuarios quizás este resultado no sea satisfactorio debido a la falta de configuración de región. Empresas como Netflix utilizan filtros geográficos para recomendar películas más acordes al país en el que se encuentra el usuario, ya que en muchos casos el gusto por una película se relaciona a la similitud cultural que se tiene con el usuario, esto puede ser una desventaja o no, según los propósitos del modelo.

El modelo puede mejorarse incorporando atributos geográficos como los países donde se filmó la película o el idioma en la que fue filmada. Otra forma en la que puede manipular la selectividad es mediante la asignación de pesos por atributo, esto hace que un atributo particular tenga una mayor importancia para el modelo.

Se puede concluir que el modelo puede hacerse más complejo, añadiendo restricciones y haciéndolo más selectivo teniendo en cuenta algunos criterios como región o país, idioma, actores, compañías productoras, etc.

X. Contenido del Proyecto

El proyecto completo puede consultarse en el enlace:

https://github.com/juanagsolano/content_based_filtering

El cual contiene los siguientes archivos:

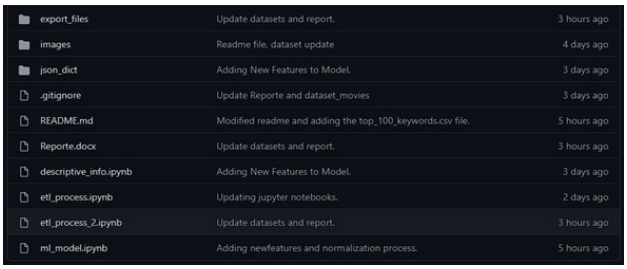


Figura 12. Esquema del proyecto en Github.

Referencias

[1] De 2013 a 2014: INEGI. Módulo sobre Disponibilidad y Uso de Tecnologías de la Información en los Hogares. [Online]. Disponible en: https://www.inegi.org.mx/temas/ticshogares/

[2] Fkih, F. (2021). Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. Journal of King Saud University - Computer and Information Sciences. [Online]. Disponible en: https://doi.org/10.1016/j.jksuci.2021.09.014

[3] Paialunga, P. (2022, 22 enero). Hands-on Content Based Recommender System using Python. Medium. [Online]. Disponible en: https://towardsdatascience.com/hands-on-content-based-recommender-system-using-python-1d643bf314e4

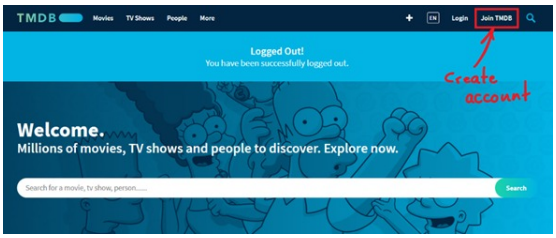
[4] Shankhar, B. S. (2021, 14 diciembre). Hyperparameters tuning in practice: Pandas vs. Caviar. Medium.[Online]. Disponible en: https://medium.com/optimizing-hyperparameters/hyperparameters-tuning-in-practice-pandas-vs-caviar-82ab9763d8af

[5] Buitinck et al. (2013). API design for machine learning software: experiences from the scikit-learn project.

ANEXO. OBTENCIÓN DE API

1. Crear cuenta en el sitio web

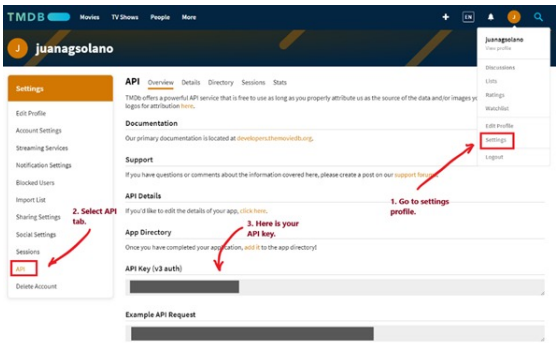
Para extraer los datos del sitio web es necesario obtener la llave API. Ir al sitio web The Movie DB y crear una cuenta como se muestra a continuación:



Llenar los datos solicitados y dar click en “Sign up”.

2. Obtener la llave API

Una vez creada la cuenta, dirigirse a la configuración de la cuenta y en el panel izquierdo seleccionar la pestaña API, donde se encontrará la llave en el apartado API key (v3 auth),



La documentación para el uso de la API puede encontrarse en el siguiente enlace: Documentación de la API.