# FILTRADO BASADO EN CONTENIDO

PORTADA

# Tabla de contenido

# Introducción

La tecnología digital cada vez se integra más a nuestra vida cotidiana, en los últimos años ha habido un

Las transacciones vía internet se han incrementado de manera considerable en los últimos años, tan solo de 2015-2020 el porcentaje de usuarios en México que realizan transacciones se incrementó en un 20%, al día de hoy la cifra debe ser mucho mayor debido a la pandemia surgida durante el 2019.

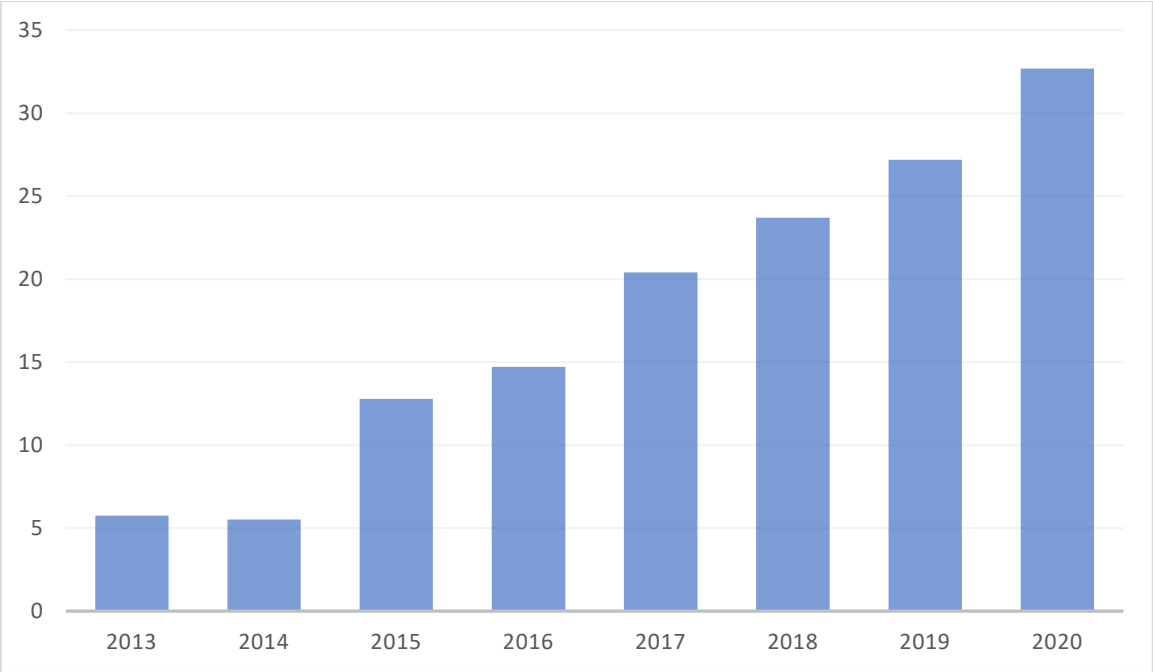



*Ilustración 1. Proporción de usuarios que realizan transacciones vs usuarios de internet.*

Los

## 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 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 y algunas de las más usuales son:

### 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:

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

### 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:

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

### 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:

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

### 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:

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

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

### 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:

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

# Conjunto de datos

## Extracción de datos

Los datos ha utilizar en este proyecto fueron extraídos del sitioweb 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.

## Atributos a utilizar dentro del modelo

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

- budget: presupuesto de la película.
- genres: lista de géneros a los que pertenece.
- id: identificador de la película.
- original_title: título original de la película.
- popularity: popularidad de la película (métrica generada por el sitio).
- release_date: fecha de lanzamiento.
- 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).

## Recopilación de datos por película

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

Los detalles de la película se pueden obtener a través de la siguiente url:

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

Para ilustrar la obtención de datos utilizaremos la siguiente película como ejemplo:

Id = 20 Nombre: My Life Without Me

Cuya sintaxis de consulta es:

*https://api.themoviedb.org/3/movie/20?api_key=fhk6739847hsjdf*

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 del proceso es el siguiente:


```
{'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': '/9Fa7tCEKIha1llGH7E41mxSpaF6.jpg',
 'production_companies': [{'id': 49,
   'logo_path': '/xpf5iHdvvBtsH8jBMlgIJHAET0c.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}
```

### 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'}
```

Esta binarización realizará 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}` |

### 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 40000 palabra, lo cual hace 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 100000 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
```

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

## Atributos por película

Los atributos para la película antes mencionada serían los siguientes:

| | |
|---|---|
| budget | 2500000 |
| popularity | 13.604 |
| release_date | 731281 |
| revenue | 12300000 |
| runtime | 106 |
| vote_average | 5.8 |
| vote_count | 364 |
| 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 |
| collection | 0 |
| short film | 0 |
| woman director | 1 |
| based on novel or book | 0 |
| murder | 0 |
| musical | 0 |
| concert | 0 |
| silent film | 0 |
| biography | 0 |
| sports | 0 |
| stand-up comedy | 0 |
| lgbt | 0 |
| christmas | 0 |
| world war ii | 0 |
| revenge | 0 |
| family | 0 |
| love | 0 |
| anime | 0 |
| philippines | 0 |
| based on true story | 0 |
| martial arts | 0 |
| friendship | 0 |
| romance | 0 |
| coming of age | 0 |

| | |
|---|---|
| softcore | 0 |
| black and white | 0 |
| kidnapping | 0 |
| wrestling | 0 |
| opera | 0 |
| new york city | 0 |
| ghost | 0 |
| based on play or musical | 0 |
| sequel | 0 |
| police | 0 |
| politics | 0 |
| serial killer | 0 |
| horror | 0 |
| pre-code | 0 |
| found footage | 0 |
| prison | 0 |
| holiday | 0 |
| erotic movie | 0 |
| drugs | 0 |
| vampire | 0 |
| zombie | 0 |
| rape | 0 |
| parent child relationship | 1 |
| death | 0 |
| gay interest | 0 |
| dance | 0 |
| remake | 0 |
| high school | 0 |
| art | 0 |
| dark comedy | 0 |
| monster | 0 |
| slasher | 0 |
| gay | 0 |
| religion | 0 |
| stop motion | 0 |
| dog | 0 |
| gore | 0 |
| time travel | 0 |
| supernatural | 0 |
| lost film | 0 |
| marriage | 0 |
| alien | 0 |
| football (soccer) | 0 |
| gangster | 0 |
| detective | 0 |
| superhero | 0 |
| nazi | 0 |
| africa | 0 |
| suicide | 0 |
| japan | 0 |
| rock 'n' roll | 0 |
| nature | 0 |
| mockumentary | 0 |

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

### 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 688316.

### Obtención de datos por lotes

Dado que la cantidad de películas es larga, esta se procedió a realizarla en lotes, en total son 4000 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.

### 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.

### Exportación de la base de datos

Al realiza la iteración por lotes, se obtiene un set de datos en formato csv, sin embargo, contiene valores nulos, por lo que en este punto solo se realiza una actualización de dicho set de datos para lidiar momentáneamente con estos valores nulos. 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 | ... |

174675 rows × 128 columns

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

## Preprocesado de datos para el modelo FBS

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 utilizada para el modelo y, por lo tanto, no lo utilizaremos.
- *Transformar la fecha en formato ordinal*: para realizar la transformación utilizamos la función datetime del paquete datetime y utilizando el método toordinal().
- *Reemplazando 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 | ... |

173886 rows × 127 columns

## Primer análisis exploratorio

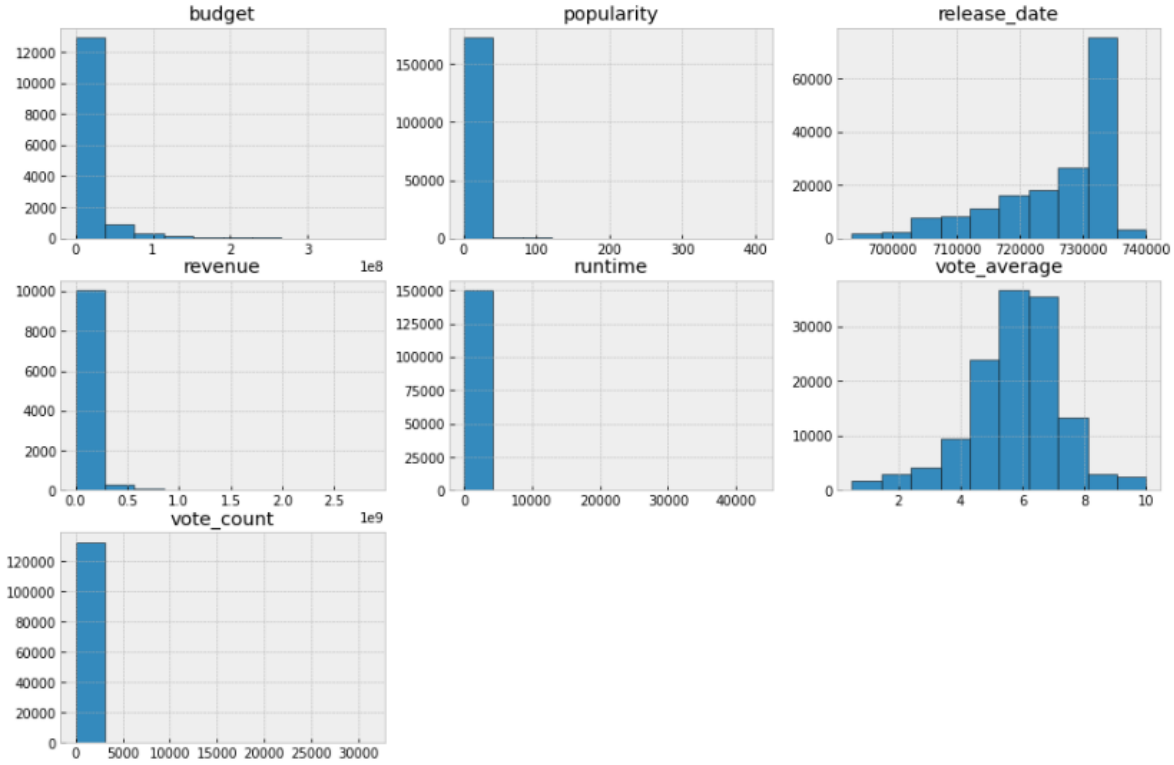
### Tabla descriptiva

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.

**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.

## Normalización de datos

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

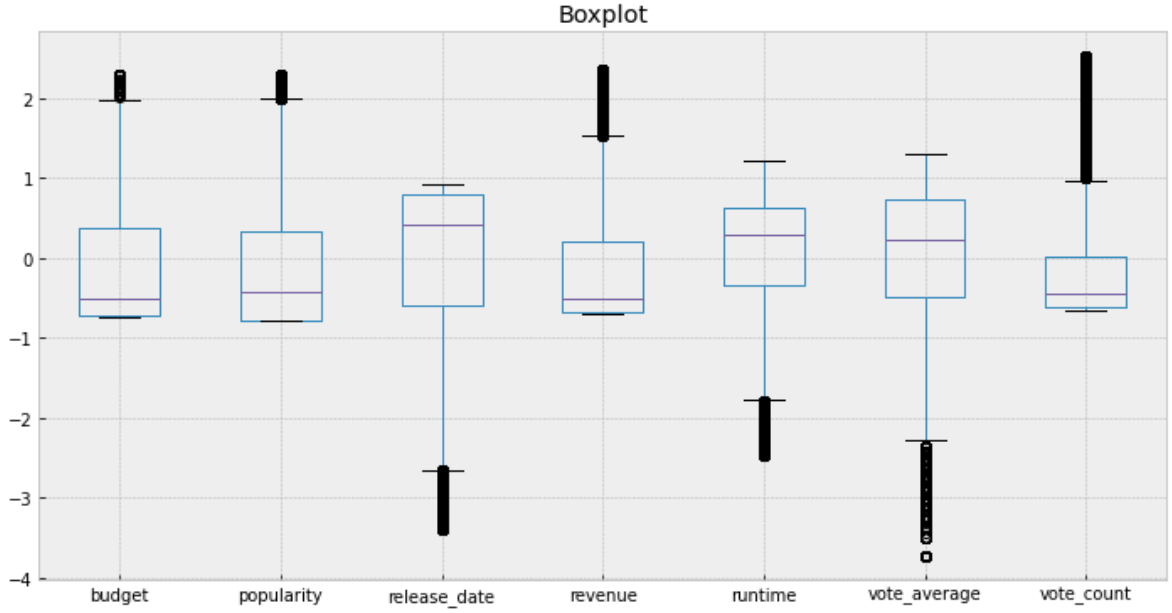
Reasignación de outliers: en primera instancia se optó por utilizar la función MinMaxScaler de la biblioteca scikitlearn, sin embargo, dado que el conjunto de datos presenta datos mucho mayores a los de la media o el rango intercuantílico, los valores asignados a la mayoría de los datos son bastante 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.

Escalado de datos: para escalar los datos se utilizó la función StandardScaler() de la biblioteca scikitlearn, la cual realiza la siguiente operación:

$$z_i = \frac{x_i - \mu}{\sigma}$$

**BoxPlot**

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




## 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. Para la implementación mostrada no utilizaremos ninguna, sin embargo, describiremos en que consiste cada una:

- Pesos a las características del conjunto: aquí podemos definir la importancia de cada característica, por ejemplo, si queremos darle una mayor o menor importancia a la fecha, lo que podríamos hacer es multiplicar la columna “date_release” por el peso asignado, un valor por encima de 1 si queremos darle mayor importancia o un valore menor a 1 si queremos darle menor peso.
- Eliminar características del conjunto: otra de las formas en las que podemos manipular el desempeño del modelo es eliminando características como la fecha, algunas palabras clave, géneros, 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.

## Métrica utilizada en el modelo

La métrica utilizada en el modelo será la distancia euclidiana nan, la cual ya describió anteriormente. 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, se optó por la distancia euclidiana nan para este proyecto.

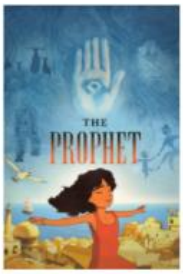
# Prueba del modelo

## Ejemplo 1

Para probar el modelo se utilizó la película “Liga de la justicia: War (2/2/2014)”


Las 10 películas sugeridas según el modelo fueron:

| ID | Title | Release Date |
|---|---|---|
| 14711 | Afro Samurai: Resurrection | 10/16/2009 |
| 35837 | Boogie, el Aceitoso | 10/22/2009 |
| 76589 | Justice League: Doom | 2/28/2012 |
| 283161 | Kahlil Gibran's The Prophet | 9/6/2014 |
| 14092 | 攻殻機動隊 2.0 | 7/12/2008 |
| 30675 | Planet Hulk | 2/2/2010 |
| 251519 | Son of Batman | 5/13/2014 |
| 51859 | トライガン バッドランド ランブル | 4/2/2010 |
| 103269 | Superman vs. The Elite | 6/12/2012 |
| 268092 | Transformers: Prime Beast Hunters | 10/4/2013 |



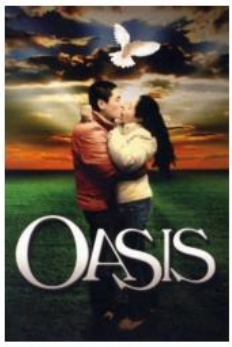

El resultado es una recomendación de películas bastante ligada al género seleccionado, sin embargo, algunas de las películas no que son “anime” no estrictamente del género y quizás para algunos usuarios esto sea irrelevante, esto podría mejorarse si se añade un atributo de la región y asignarle importancia a éste.

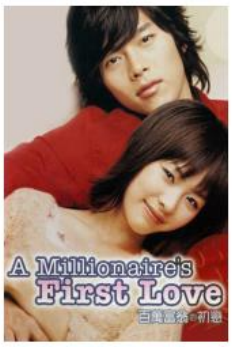
### Ejemplo 2

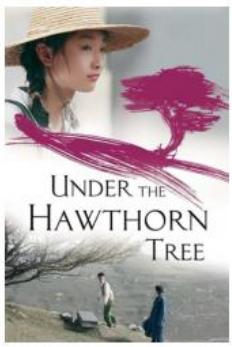
El segundo ejemplo consiste en una película del género romántico “The Notebook”:

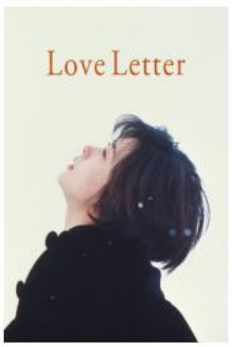
La lista de recomendaciones es la siguiente:

| ID | Title | Release Date |
|---|---|---|
| 26955 | 오아시스 | 8/9/2002 |
| 59944 | 君に届け | 9/25/2010 |
| 18704 | 파이란 | 4/28/2001 |
| 40211 | 백만장자의 첫사랑 | 2/9/2006 |
| 37185 | 甜蜜蜜 | 11/2/1996 |
| 50542 | 山楂树之恋 | 9/15/2010 |
| 47002 | ラブレター | 3/25/1995 |
| 2012 | Stellet Licht | 10/2/2007 |
| 93858 | Tabu | 4/5/2012 |
| 15712 | Welcome | 3/11/2009 |







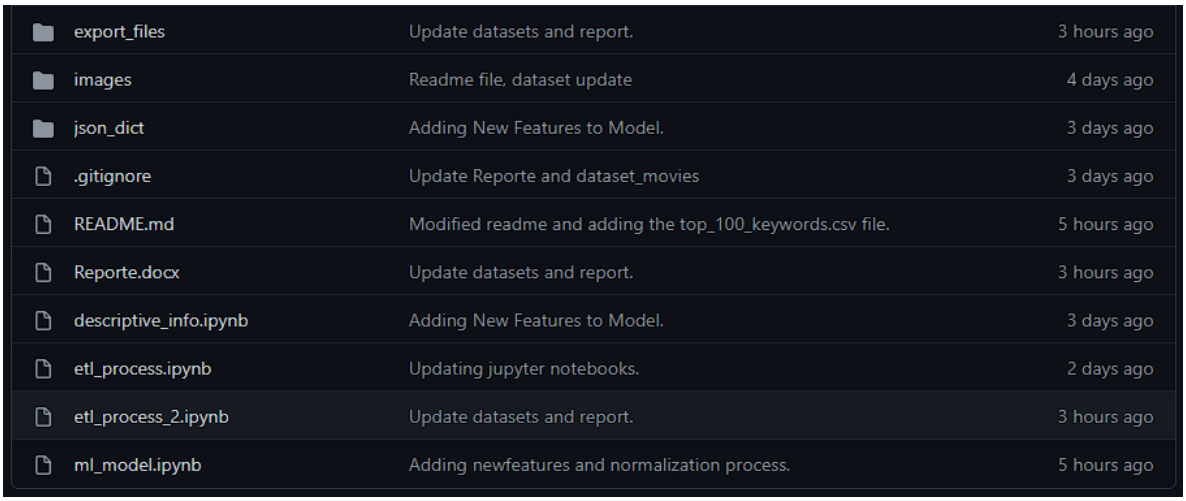



## Contenido del proyecto

El proyecto completo puede consultarse en el enlace:

*https://github.com/juanagsolano/content_based_filtering*

El cual contiene los siguientes archivos.




## Bibliografía

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

https://medium.com/optimizing-hyperparameters/hyperparameters-tuning-in-practice-pandas-vs-caviar-82ab9763d8af