



Universidad Europea

UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

GRADO EN INGENIERÍA INFORMÁTICA

SISTEMAS INTELIGENTES

Proyecto Final: MovieLens

GRUPO 1:

MovieLens

IGNACIO GIL GARZÓN

ÁLVARO FARRENY BOIXADER

CARLOS GONZÁLEZ VAN LIEMPT

Dirigido por

BORJA MONSALVE PIQUERAS

CURSO 2022-2023

Índice

Capítulo 1. INTRODUCCIÓN	3
Capítulo 2. DESCRIPCIÓN Y ORGANIZACIÓN DEL PROCESO	4
Organización del código	4
Proceso de Web Scraping	4
Proceso de recuperación de texto	5
Proceso de modelado	5
Proceso de recomendación	6
Dada una película	6
Dado un usuario	6
Capítulo 3. PRUEBAS	7
Capítulo 4. RESULTADOS	8
A partir de los géneros de una película	8
A partir de un usuario dado	9
Información sobre una película seleccionada	10
Capítulo 5. MANUAL DE USUARIO E INSTALACIÓN	11
Manual de instalación	11
Manual de usuario	11
Capítulo 6. CONCLUSIONES	15
Conclusiones del trabajo	15
Futuras líneas de trabajo	15
Capítulo 7. REFERENCIAS	16

Capítulo 1. INTRODUCCIÓN

El proyecto planteado por nuestro grupo consiste en crear un sistema recomendador de películas a partir del dataset de películas y de usuarios de MovieLens.

MovieLens es una plataforma online de usuarios que opinan sobre películas y valoran cada una de las películas del cero al cinco.

Con estos datos, nos hemos propuesto crear 3 tipos diferentes de predicciones:

- Recomendaciones de películas en base a un set de géneros
- Recomendaciones de películas en base a un usuario dado
- Rating de mejores valoraciones de películas dado un usuario

Estas predicciones se realizan utilizando un modelo espacio-vectorial [1] (asigna un valor a la frecuencia de aparición de cierto término en un documento) y un método TF-IDF [2] (mide la relevancia de un texto en relación con los términos de una consulta de búsqueda).

Por último, para obtener valores numéricos de predicción, usaremos el algoritmo de similitud del coseno [3], que consiste en calcular la similitud existente entre dos vectores en un espacio y evaluar el valor del coseno del ángulo comprendido entre ellos.

Además, extraemos datos públicos de tmdb como la sinopsis utilizando la técnica de Web Scraping [4], software que sirve para extraer información de sitios web.

Capítulo 2. DESCRIPCIÓN Y ORGANIZACIÓN DEL PROCESO

Organización del código

Para trabajar de manera asíncrona, se ha decidido crear un archivo de Jupyterlab Notebook (.ipynb) para desarrollar cada una de las predicciones y, una vez obtenidos los resultados buscados, añadir todas a una sola interfaz gráfica realizada en Python con la librería Tkinter.

Estas son las librerías que utilizamos en el proyecto:

- Interfaz gráfica: **tkinter**.
- Creación, edición y visualización de dataframes: **pandas** y **pandastable**.
- Generación de recomendaciones usando Machine Learning: **sklearn** [6] e **intertools**.
- Web Scraping: **Selenium** y **BeautifulSoup**.

La principal función que utilizamos en nuestro proyecto es *generarRecomendaciones()*.

A esta función le pasamos los siguientes parámetros: *i* (nombre de la película), *M* (dataframe de similitud), *items* (dataframe de títulos y géneros) y *k* (el número de recomendaciones que queremos obtener).

Las tres predicciones que componen nuestro proyecto parten de esta función.

Proceso de Web Scraping

Partiremos de las fuentes de datos proporcionadas para el proyecto: *ml-latest-small* (*movies.csv*, *ratings.csv*, *tags.csv*, *links.csv*). A estas, añadiremos otro archivo generado mediante Web Scraping en el que almacenaremos la sinopsis de cada película (*sinopsisDB.csv*) con su id de película correspondiente.

Para realizar el web scraping [4] de una forma correcta y en formato inglés, se ha utilizado la librería *selenium* junto con *BeautifulSoup* [8] para scrapear y obtener la sinopsis de la película deseada. Una vez se tenga dicha sinopsis, se obtendrá el rating de la película y estos datos se almacenarán en un dataframe con el *movieId* correspondiente para extraerlo en formato csv y poder trabajar posteriormente con ellos. Además se mostrará la película elegida por la interfaz gracias a la pestaña final de información de la película.

Proceso de recuperación de texto

Para poder aplicar el modelo de recomendación se debe realizar un procesamiento de películas previo para poder sacar la matriz con el tf idf [2] y obtener valores más precisos en función de la recomendación que el usuario prefiera.

Para ello se va a realizar un tf idf en función de los géneros de cada película usando la librería *TfidfVectorizer* de Sklearn. Gracias a esta librería podremos obtener dataframe similar al de la Figura 1. que nos proporciona el valor de semejanza entre películas basado en el género de cada una de ellas.

title	Toy Story	Jumanji	Grumpier Old Men	Waiting to Exhale	Father of the Bride Part II
Toy Story	1.000000	0.474735	0.033432	0.019663	0.082550
Jumanji	0.474735	1.000000	0.000000	0.000000	0.000000

Figura 1 - Similitud_df

Para el almacenamiento de las películas se ha decidido usar el archivo de movies.csv y el de synopsisDB.csv gracias al scraper realizado con anterioridad para poder obtener los distintos valores necesarios para el uso del usuario. Al seleccionar una película, el usuario puede visitar la sección de información en función de dicha película y obtendrá el nombre de la misma, la sinopsis y el rating basado en la página web tmdb.

Proceso de modelado

En cuanto al proceso de modelado de nuestro proyecto, partimos del archivo original ml-latest-small, el cual contiene los archivos movies, tags, ratings y links.

Para obtener unos datos útiles, realizamos una limpieza inicial en la que eliminamos toda fila que contenga un valor nulo.

En el archivo de movies, los títulos de las películas vienen con el año de estreno, por lo que otro proceso que hacemos es el de extraer este año y posicionarlo en su propia columna.

Hemos juntado los dataframes de movies con el dataframe que hemos realizado el web scraping de las sinopsis de dichas movies para tenerlo todo junto en un mismo csv y que nos fuera más fácil y rápido el proceso de información.

Proceso de recomendación

Se han realizado 3 recomendaciones:

- en base a una película dada
- en base a un usuario dado
- mejores valoraciones de películas de un usuario dado

Dada una película

A la hora de recomendar una película en base a su género, nuestro sistema utiliza el algoritmo TF-IDF para generar un dataframe de similitudes para todas sus películas.

Una vez tenemos este dataframe con datos entre 1 y 0 (le llamaremos “similitudG”), buscamos las 11 mejores similitudes para una película dada y eliminamos el primer valor porque será esa misma película.

Dado un usuario

Para poder realizar la recomendación en base a un usuario utilizamos un modelo llamado Recommender Net.

Nos apoyamos de los archivos csv de “ratings.csv” y “películas.csv” que respectivamente tienen todos los ratings de todos los usuarios de todas las películas. Por lo que primeramente preparamos el csv de ratings. Necesitamos obtener un mapa con los ID 's de los usuarios y las películas para que el modelo pueda acceder a ellas de una manera más sencilla. Posteriormente guardamos en una lista aquellas películas que el usuario ya ha calificado para más tarde excluirse de la predicción y así evitar que al usuario le recomiende películas ya vistas.

Cuando el usuario introduce el ID de un usuario, nuestro sistema realiza un cálculo de similitudes entre este usuario y otros usuarios para saber que usuarios se asemejan al dado. Finalmente mostramos aquellas películas que el usuario haya calificado con mejor nota y aquellas películas que nuestro modelo de predicción considera interesantes para el usuario dado.

Capítulo 3. PRUEBAS

Código	Test	Descripción
PG1	Recomendaciones por género	Se selecciona una película de la pestaña de recomendación de películas y recomienda películas en función de su género.
PG2	Nueva recomendación	Una vez realizada la prueba PG1, intentaremos seleccionar múltiples películas a la vez para ver si nuestra aplicación produce errores o bugs en este apartado.
PU1	Recomendaciones por usuario	Se introduce en el campo correspondiente el número de usuario que queremos ver sus recomendaciones de películas y ver cuales son las mejores películas valoradas por dicho usuario.
PW1	Web Scraping	Probaremos este apartado seleccionando películas al azar en la ventana uno y esperando una descripción correcta de dicha selección en la ventana de Web Scraping.
PW2	Múltiple selección Web Scraping	Intentaremos romper la ventana de Web Scraping intentando una múltiple selección de películas.

Nuestra aplicación ha pasado todas las pruebas planteadas con resultados que explicaremos en el siguiente capítulo.

Capítulo 4. RESULTADOS

Finalmente, se ha llegado al producto final de MovieLens, una aplicación ejecutable de código Python [7]. Hemos realizado una aplicación que recomienda películas en base a otras películas, en base a un usuario y que realiza web scraping para hallar las sinopsis y los ratings de la página tmdb.

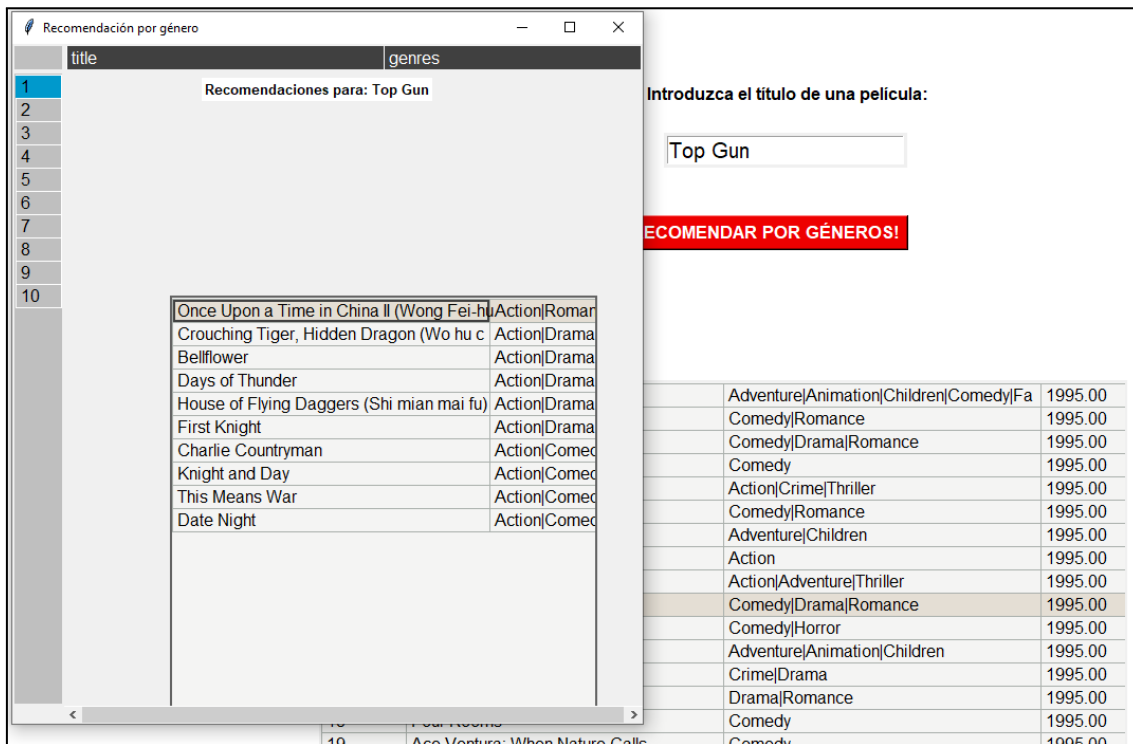
Ahora se va a proceder a mostrar cada uno de los resultados que se pueden obtener al ejecutar la aplicación.

Enlace al GitHub: <https://github.com/ignaciogg/SSII-Movielens>

A partir de los géneros de una película

Como podemos observar se ha creado un algoritmo que analiza todas las películas que se han recopilado en la aplicación y en función de sus géneros recomienda películas similares a ella.

En este caso se ha buscado la película “Top Gun” y obtenemos un total de 10 películas similares a dicha película. En este caso, películas similares por género que puede ser “Acción, Drama, Romance...”



Recomendación por género

Introduzca el título de una película:

Top Gun

RECOMENDAR POR GÉNEROS!

Recomendaciones para: Top Gun	title	genres
1	Once Upon a Time in China II (Wong Fei-hung)	Action Romance
2	Crouching Tiger, Hidden Dragon (Wo hu cang long)	Action Drama
3	Bellflower	Action Drama
4	Days of Thunder	Action Drama
5	House of Flying Daggers (Shi mian mai fu)	Action Drama
6	First Knight	Action Drama
7	Charlie Countryman	Action Comedy
8	Knight and Day	Action Comedy
9	This Means War	Action Comedy
10	Date Night	Action Comedy

Adventure Animation Children Comedy Fantasy	1995.00
Comedy Romance	1995.00
Comedy Drama Romance	1995.00
Comedy	1995.00
Action Crime Thriller	1995.00
Comedy Romance	1995.00
Adventure Children	1995.00
Action	1995.00
Action Adventure Thriller	1995.00
Comedy Drama Romance	1995.00
Comedy Horror	1995.00
Adventure Animation Children	1995.00
Crime Drama	1995.00
Drama Romance	1995.00
Comedy	1995.00
Comedy	1995.00

A partir de un usuario dado

Se ha dado la posibilidad de buscar por usuario por lo que si buscamos un usuario por ejemplo el usuario “432” nos va a mostrar dos factores, las películas mejor valoradas por dicho usuario y las 5 películas que no ha visto y que le pueden gustar.

RECOMENDAR PELÍCULAS EN BASE A UN USUARIO

Introduzca el ID de un usuario:

RECOMENDAR POR USUARIOS!

-> Mostrando las recomendaciones para el usuario: 432

Estas son tus películas mejor valoradas:

- Lion King, The
- Misérables, Les
- Exorcist, The

Estas son las películas que te aconsejamos:

- Red Lights (Feux rouges)
- Jacket, The
- Infernal Affairs 2 (Mou gaan dou II)
- Sea Inside, The (Mar adentro)
- Interpreter, The

Información sobre una película seleccionada

Finalmente hemos implementado la funcionalidad de seleccionar una película en la pantalla de recomendar por géneros. Una vez seleccionada dicha película podremos obtener toda la información proveniente de tmdb como puede ser: título, sinopsis y ranking.

Para que nuestra aplicación no tarde tanto al ejecutarse y realizar dicha query se han juntado dos csv, el de movies y el csv que hemos realizado de scrapper con todas las sinopsis y ratings en función de las movies del movies.csv.

WEB SCRAPING

Película seleccionada: Sabrina

Sinopsis: An ugly duckling having undergone a remarkable change, still harbors feelings for her crush: a carefree playboy, but not before his business-focused brother has something to say about it.

Rating: 62

Capítulo 5. MANUAL DE USUARIO E INSTALACIÓN

Manual de instalación

Para utilizar nuestra solución frente a MovieLens, es necesario primero [instalar Python](#) en nuestra máquina y contar con un IDE o un editor de texto para correr nuestro código, nosotros recomendamos [Visual Studio Code](#).

En el apartado de extensiones de Visual Studio Code, buscaremos e instalaremos la extensión Python.

Lo siguiente será cambiar al explorador de archivos de Visual Studio Code y seleccionar la carpeta del proyecto.

Para instalar las librerías que utilizamos en nuestra aplicación, basta con abrir una ventana de terminal (segundo icono arriba a la izquierda) y ejecutar estos comandos uno a uno:

```
py -m pip
```

```
pip install -r requirements.txt
```

Con esto terminaríamos la instalación de nuestra aplicación y podremos ejecutar main.py.

Manual de usuario

Para empezar a usar la aplicación se debe ejecutar el main.py. Una vez lo ejecutemos podremos ver una ventana similar a la que se muestra en la siguiente figura.





Como podemos observar, se pueden ver en la barra de arriba a la izquierda un total de 4 iconos que llevan a diferentes ventanas

La estructura del menú es la siguiente:

- Home: vuelta al menú de inicio
- Ranking: Ranking de películas en base a otras y los tags
- User: Ranking de películas en base a usuarios
- Movie: Pestaña donde aparece la película seleccionada gracias a un web scraping

- Apartado Recomendador por genero

Para poder recomendar en base a una película por tags, debemos primero introducir en la caja superior el nombre de la película de la que queremos obtener una recomendación. Una vez hayamos seleccionado por ejemplo la película “Toy Story”.

Bienvenido a nuestro Proyecto Final de SSII

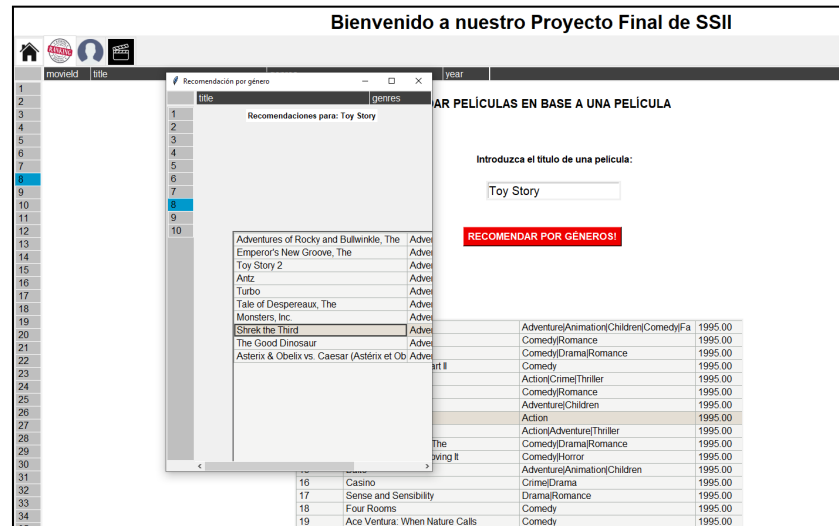
Introduzca el título de una película:

Toy Story

RECOMENDAR POR GÉNERO!

1	Toy Story	Adventure Animation Children Comedy Fa	1995.00
3	Grumpier Old Men	Comedy Romance	1995.00
4	Waiting to Exhale	Comedy Drama Romance	1995.00
5	Father of the Bride Part II	Comedy	1995.00
6	Heat	Action Crime Thriller	1995.00
7	Sabrina	Comedy Romance	1995.00
8	Tom and Huck	Adventure Children	1995.00
9	Sudden Death	Action	1995.00
10	GoldenEye	Action Adventure Thriller	1995.00
11	American President, The	Comedy Drama Romance	1995.00
12	Dracula: Dead and Loving It	Comedy Horror	1995.00
13	Balto	Adventure Animation Children	1995.00
16	Casino	Crime Drama	1995.00
17	Sense and Sensibility	Drama Romance	1995.00
18	Four Rooms	Comedy	1995.00
19	Ace Ventura: When Nature Calls	Comedy	1995.00
20	Money Train	Action Comedy Crime Drama Thriller	1995.00
21	Get Shorty	Comedy Crime Thriller	1995.00
22	Copycat	Crime Drama Horror Mystery Thriller	1995.00

Procederemos a buscar en función del género y nos saltara un pop up que nos va a recomendar las mejores películas en base a “Toy Story”.



- Apartado Recomendador por usuario

Para poder utilizar este recomendador únicamente debemos introducir el ID de usuario al cual queremos recomendar películas nuevas de interés. Esto deberá ser un número limpio, sin símbolos ni letras.

RECOMENDAR PELÍCULAS EN BASE A UN USUARIO

Introduzca el ID de un usuario:

RECOMENDAR POR USUARIOS!

-> Mostrando las recomendaciones para el usuario: 354

Estas son tus películas mejor valoradas:

- Usual Suspects, The
- Pulp Fiction
- Shawshank Redemption, The

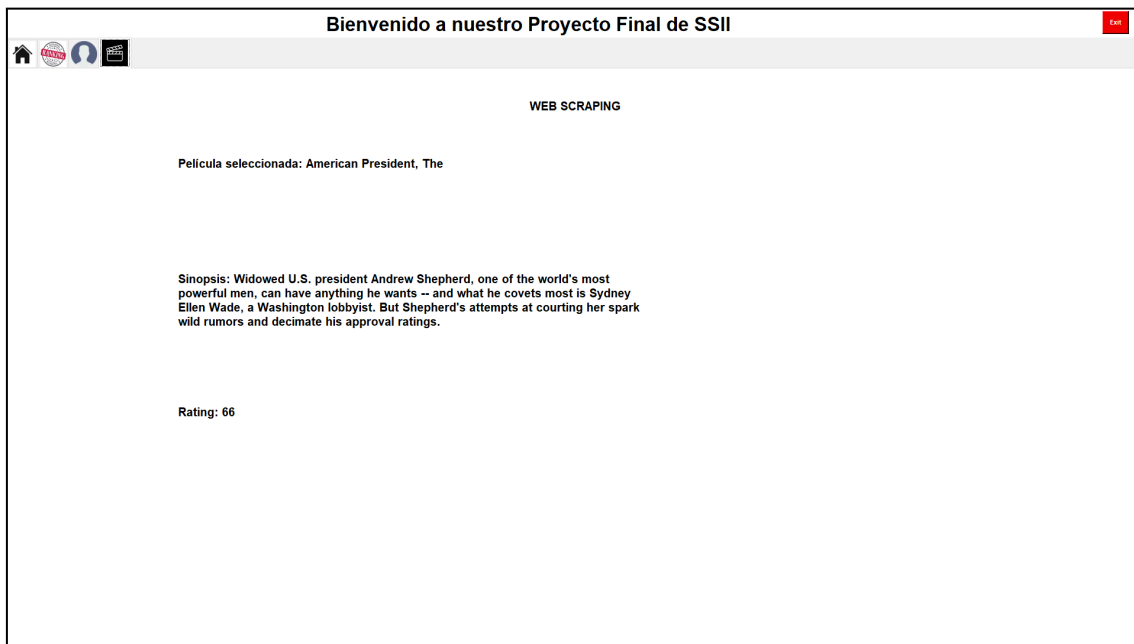
Estas son las películas que te aconsejamos:

- Hogfather (Terry Pratchett's Hogfather)
- Zone, The (La Zona)
- Frozen River
- Drive Angry
- Confessions (Kokuhaku)

Si todo ha funcionado correctamente podremos visualizar en primer lugar las películas que el usuario ha valorado muy positivamente y en segundo lugar las películas que se han recomendado al usuario teniendo en cuenta sus valoraciones y las valoraciones de otros usuarios que se asemejan a él.

- Apartado Web Scraping

En el apartado final de web scraping podemos visualizar el título de la película seleccionado en la pantalla de rankings. Además, podemos leer de una forma rápida y sencilla la sinopsis de dicha película y contrastarla con el ranking obtenido de la página de tmdb.



Capítulo 6. CONCLUSIONES

Conclusiones del trabajo

Tras el desarrollo de este trabajo, hemos conseguido una aplicación que consideramos que cumple con todos los objetivos propuestos:

- Recomendar en base a una película dada
- Recomendar en base a un usuario dado
- Obtener información sobre la película elegida (título, sinopsis, rating)

Además se ha realizado una app sencilla e intuitiva para que el usuario pueda disfrutar de nuevas películas que han sido seleccionadas para él teniendo en cuenta sus gustos anteriores. El usuario también puede visualizar información relevante acerca de cualquier película que desee consultar.

Consideramos que este producto es un producto muy beneficioso para cualquier usuario ya que permite crear “tendencias” personalizadas y esto aporta mucho valor al mercado permitiendo a todos los usuarios tener más interés en visualizar aquellas películas que se relacionen con los gustos que él tiene.

Futuras líneas de trabajo

Nuestra futura línea de trabajo se basa en obtener recomendaciones de películas utilizando la sinopsis de las mismas.

Durante el proyecto, intentamos múltiples veces obtener predicciones en base a una película pero en vez de utilizar los géneros de las películas, usando su sinopsis.

El principal problema que encontramos fue la larga duración del proceso destinado a crear las similitudes por sinopsis.

Para solucionar este inconveniente, intentamos sin éxito dividir en 5 el archivo de similitudes.

Otra línea de trabajo secundaria consiste en mejorar la interfaz gráfica, que consistiría en migrar la aplicación Python a una interfaz web.

Capítulo 7. REFERENCIAS

[1] **MODELO ESPACIO VECTORIAL. 2022.** *Modelo Vectorial - Modelos Clásicos.*

[En línea] 2022. [Citado el 10 de enero de 2023.]

<https://sites.google.com/site/modelosclasicosri/vectorial>

[2] **TF-IDF. 2022.** *term Frequency Inverse Document Frequency (TF-IDF).*

[En línea] 2022. [Citado el 10 de enero de 2023.]

<https://www.visitor-analytics.io/es/glosario/t/term-frequency-inverse-document-frequency-tf-idf/>

[3] **ALGORITMO DE SIMILITUD DE COSENO. 2022.** *Algoritmo de similitud de coseno.*

[En línea] 2022. [Citado el 10 de enero de 2023.]

<https://www.grapheverywhere.com/algoritmo-de-similitud-de-coseno/>

[4] **WEB SCRAPING. 2022.** *¿Qué es el Web Scraping?.*

[En línea] 2022. [Citado el 10 de enero de 2023.]

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-el-web-scraping/>

[5] **INTELIGENCIA ARTIFICIAL. 2022.** *¿Qué es la Inteligencia Artificial?.*

[En línea] 2022. [Citado el 10 de enero de 2023.]

<https://www.oracle.com/es/artificial-intelligence/what-is-ai/>

[6] **SKLEARN. 2022.** *Guía de usuario.*

[En línea] 2022. [Citado el 15 de enero de 2023.]

https://scikit-learn.org/stable/user_guide.html

[7] **PYTHON. 2022.** *¿Qué es Python?.*

[En línea] 2022. [Citado el 15 de enero de 2023.]

<https://www.becas-santander.com/es/blog/python-que-es.html>

[8] **BEAUTIFULSOUP. 2023.** *Beautiful Soup is a Python library for pulling data out of HTML.*

[En línea] 2023. [Citado el 15 de enero de 2023.]

<https://beautiful-soup-4.readthedocs.io/en/latest/#>