UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Impact Factor Oracle

Presentado por Gadea Lucas Pérez
en Universidad de Burgos
a 8 de junio de 2023
Tutores: Virginia Ahedo y Álvar Arnaiz

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

Dña. Virginia Ahedo, profesora del departamento de Ingeniería de Organización, área de Organización de Empresas y D. Álvar Arnaiz, del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Exponen:

Que el alumno Dña. Gadea Lucas Pérez, con DNI 71483074V, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado «Impact Factor Oracle».

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 8 de junio de 2023

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
Dña. Virginia Ahedo	D. Álvar Arnaiz

Resumen

El presente proyecto se centra en la extracción de datos bibliográficos con el fin de calcular y predecir el Factor de Impacto. Esta métrica se utiliza para evaluar la importancia de una revista en un campo científico determinado. Se mide a través de la frecuencia con la que los artículos de la misma han sido citados en un año específico. Se trata de un criterio importante en la evaluación de la calidad del trabajo científico y puede ser de gran ayuda en la selección de la revista adecuada para publicar un nuevo trabajo.

El proyecto extraerá los datos históricos disponibles en la web (tales como Google Scholar, Crossref, Web of Science, Scopus...) empleando técnicas de *web scrapping* sobre las distintas fuentes. Estos datos se utilizarán como entradas para los algoritmos de aprendizaje automático, que serán supervisados y se utilizarán para estimar el valor del Índice de Impacto de las revistas indexadas en el JCR (Journal Citation Reports).

El objetivo final es desarrollar una aplicación web accesible y de fácil uso para la comunidad científica, que permita predecir la importancia de las revistas científicas en tiempo real. Esta aplicación web será de acceso abierto y se encontrará en un repositorio público para garantizar su disponibilidad y uso por parte de la comunidad científica.

Descriptores

Bibliometría, publicación de artículos, revistas científicas, Índice de Impacto, aplicación web, aprendizaje automático.

Abstract

The present project focuses on extracting bibliographic data in order to calculate and predict the Impact Factor. This metric is used to evaluate the importance of a journal in a specific scientific field. It is measured by the frequency with which articles from the same journal have been cited in a specific year. It is an important criterion in evaluating the quality of scientific work and can be of great help in selecting the appropriate journal to publish new work.

The project will extract historical data available on the web (such as Google Scholar, Crossref, Web of Science, Scopus...) using web scraping techniques on various sources. This data will be used as inputs for machine learning algorithms, which will be supervised and used to estimate the value of the Impact Factor of the journals indexed in the JCR (Journal Citation Reports).

The ultimate goal is to develop an accessible and easy-to-use web application for the scientific community, which will allow the prediction of the importance of scientific journals in real-time. This web application will be open access and will be located in a public repository to ensure its availability and use by the scientific community.

Keywords

Bibliometrics, articles publication, scientific journals, impact index, web application, machine learning.

Índice general

Índice de figuras

Índice de tablas

Introducción

El Factor de Impacto de una publicación científica mide la frecuencia con la cual ha sido citado el artículo promedio de una publicación en un año en particular. Específicamente, sirve para evaluar la importancia de una revista dentro de un determinado campo científico. Existen múltiples metodologías de cálculo y sus correspondientes métricas, siendo el JCR (Journal Citation Reports) y el SJR (Scimago Journal Rank) los dos índices de impacto más utilizados. Además, cabe destacar que el Factor de Impacto es uno de los principales criterios empleados en los procesos de acreditación y promoción interna para evaluar la calidad del trabajo científico de millones de académicos en todo el mundo.

Por su propia naturaleza, el Factor de Impacto se calcula con carácter retrospectivo, i.e., sobre datos de años anteriores. Así pues, a la hora de seleccionar la revista a la que mandarán un nuevo trabajo, los académicos quieren tener en cuenta el posible Factor de Impacto que tendrá la revista en el año de publicación del artículo. Sin embargo, lo único que pueden hacer es fijarse en las métricas de los años anteriores y hacer sus propias hipótesis y predicciones de futuro.

Con esta problemática en mente, creemos que puede ser muy útil para la comunidad científica utilizar los datos bibliográficos que hay disponibles en la web como *inputs* de modelos de aprendizaje automático para estimar el Índice de Impacto que tendrán las distintas revistas científicas en el año en curso (o en años futuros).

A día de hoy la herramienta Google Scholar recoge información extremadamente actualizada sobre la publicación y citación de artículos científicos (podría decirse que se actualiza prácticamente en tiempo real). Por ello, al inicio del proyecto, consideramos Google Scholar como primera opción

de donde extraer los datos. Sin embargo, debido a la gran cantidad de limitaciones de esta herramienta (que fueron descubriéndose a lo largo de la creación de prototipos), se terminó descartando esta opción. Posteriormente, se probaron otras fuentes como Scopus o WoS (Web of Science) pero, finalmente, nos terminamos decantando por Crossref. Esta elección se justificará de forma detallada más adelante.

En resumen, el *output* esperado del proyecto será una aplicación web de tipo open-access, la cual implementará algoritmos de aprendizaje supervisado que utilizarán los datos históricos extraídos para predecir el valor del Índice de Impacto de todas las revistas científicas indexadas en el JCR. Dicha aplicación se dejará en un repositorio público, para así garantizar que pueda ser utilizada por toda la comunidad científica.

1.1. Enlaces relevantes

A continuación, se facilitan los hiperenlaces para acceder al resultado final del proyecto:

- **Repositorio GitHub**: https://github.com/glp1002/JCR_Impact_Factor
- ***Workspace* de ZenHub**: https://app.zenhub.com/workspaces/tfg-workspace-632982cfea90eb5c79154db7?invite=LSbRoaSSdht7DStyN3ryjsTn
- **Aplicación web**: https://paperrank.herokuapp.com/

Objetivos del proyecto

El objetivo principal de este proyecto consiste en llevar a cabo una investigación que permita sobre cómo estimar el Índice de Impacto de una revista. Para lograrlo, se desarrollará un modelo de inteligencia artificial capaz de realizar predicciones precisas al respecto. Como último paso, se diseñará y programará una aplicación web que facilite el acceso a los resultados obtenidos.

A continuación, se enumeran los principales objetivos de este proyecto:

1. Lectura de literatura científica sobre bibliometría para comprender bien el marco conceptual en el que se encuadra este proyecto.
2. Estudio de la metodología de cálculo de los distintos índices de impacto en general, y del JCR (Journal Citation Report) en particular. Asimismo, se estudiarán las sucesivas modificaciones/excepciones que se han ido introduciendo en el cálculo del JCR a lo largo de los años.
3. Estudio de la API de Google Scholar y otras APIs de datos bibliográficos.
4. Diseño y creación de una base de datos en la que se almacenará la información bibliográfica descargada.
5. Implementación de las funciones de cálculo del índice JCR para aplicarlas sobre los datos extraídos.
6. Experimentación y evaluación de distintos modelos de regresión (aprendizaje supervisado) para predecir el índice de impacto JCR a partir de las series temporales históricas disponibles. Selección del mejor algoritmo.

7. Diseño y creación de una aplicación web donde se incluirá el mejor modelo de regresión para ponerlo a disposición de la comunidad científica.

Conceptos teóricos

En términos generales, se podría decir que cuando un investigador termina una etapa de su investigación, genera un artículo científico en el que plasma el resultado de su trabajo. Esta documentación sirve de precedente para aquellos que posteriormente investiguen sobre temáticas relacionadas. Así pues, el artículo científico es el elemento principal en torno al que giran los estudios bibliométricos.

Antes de adentrarnos en más detalles, se ilustrarán algunos conceptos esenciales sobre este proyecto, para dotarlo de mayor comprensión y claridad. En esta sección, por tanto, se abarcarán los conceptos de bibliometría, cienciometría e índice de impacto, como conceptos relativos al avance de la ciencia y la producción de conocimientos a partir de la actividad de la investigación.

3.1. Bibliometría

Palabra que proviene del griego *biblio* (libro) y *-metría* (medición)

Ciencia que aplica métodos matemáticos para encontrar comportamientos estadísticos en la literatura científica. Estos estudios y análisis pretenden cuantificar toda la actividad científica escrita con el objetivo final de orientar sobre el impacto de una investigación [26].

Antiguamente, se evaluaba la producción científica por pares, que es un proceso en el que expertos en un campo revisan y evalúan la calidad y el mérito de una investigación antes de su publicación. Al final de la década de 1950, en un momento en que la producción científica aumentaba progresivamente a un ritmo sostenido, nació la idea de una evaluación basada

en cantidades, ciertamente menos costosa que la evaluación de pares: desde ese momento, comenzó a hablarse de bibliometría [29]. Así, los estudios bibliométricos tienen su origen en la década de 1960 con la creación del Science Citation Index por Eugene Garfield y el análisis de redes de citas realizado por Derek John de Solla Price. Estos trabajos establecieron las bases sólidas de la bibliometría como disciplina.

El objetivo de los estudios bibliométricos pueden limitase solamente al análisis de la envergadura, el crecimiento y la distribución de la literatura científica, pero también son útiles a la hora de encontrar las revistas donde es más conveniente publicar un artículo o descubrir a los autores más importantes en cada ámbito, así como las nuevas tendencias.

Sin embargo, la información organizada de esta forma también tiene sus limitaciones. Ejemplo de ello es el patrón de citas usado en cada área de investigación; es mucho más común incluir citas en investigaciones tecnológicas y científicas. Es decir, cada temática deberá ser tratada de forma distinta. Para ello se normalizará el índice de impacto. Por otro lado, también influye la base de datos utilizada, ya que cada una tiene un método de indexación distinta. El idioma es otro ejemplo de limitación. Ya que el inglés es el idioma predominante, será más complicado encontrar citas de documentos escritos en otros idiomas. También es preciso tener en cuenta problemas de dispersión debidos a perfiles duplicados o nombres similares. Para poder solucionar este tipo de confusiones, es recomendable crear identificadores (por ejemplo, el ORCID) para cada autor. De esta forma, todos los comportamientos irregulares (como las citas a uno mismo) pueden tenerse en consideración. Finalmente, el momento en que se realizan los estudios es también fundamental, ya que el nivel de conocimientos en torno a una materia determinada varía con el tiempo.

Ejemplo

Un ejemplo de análisis bibliométrico (tomado del libro «Documentación científica y nuevas tecnologías de la información» [26]) podría involucrar el determinar los autores que abordan un tema específico en una revista en particular. Para ello, deberían registrarse los siguientes datos:

1. Lista de identificadores de los autores (ordenados alfabéticamente) aparecidos en la revista en cuestión.
2. Cantidad de artículos de cada uno de estos autores en la revista.
3. Revistas en las que aparecen los artículos citados.

4. Los artículos (cantidad) que aparecen en cada una de las revistas listadas.

3.2. Cienciometría

Palabra que proviene del latín *scientia* (ciencia) y del griego *-metría* (medición)

Se trata del estudio de los metadatos e indicadores sobre la bibliografía científica con el fin de medir y analizar toda la producción científica [26].

La cienciometría tiene como objetivo clasificar y organizar el conocimiento científico a través de la creación de sistemas de conceptos y facilitar su transferencia a través de la educación y formación. Además, promueve la comunicación de conocimientos de un idioma a otro mediante la utilización de símbolos lingüísticos, permite la síntesis y resumen de la información científica, y proporciona un medio para recuperar y almacenar información a través de la indexación y lenguajes documentales.

La cienciometría, puesta en práctica, trabaja relacionada con otras ciencias y disciplinas como las que se muestran en la siguiente imagen:

Figura 3.1: Diagrama de Venn visualizando la relación entre disciplinas

Es decir, se trata de un concepto más amplio que engloba al anterior (bibliometría). Así, se podría decir que la bibliometría surge como resultado del contacto interdisciplinar de entre el conjunto de disciplinas que integran lo que se conoce como «ciencia de la ciencia», cuya fuente es la propia cienciometría [26].

Sin embargo, hoy en día la frontera entre cienciometría y bibliometría ha desaparecido casi por completo y ambos términos se usan prácticamente de forma sinónima [30].

3.3. Webmetría

Palabra que proviene del inglés web y del griego -metría (medición)

Aparte de las citaciones tradicionales, existen también referencias generadas por los lectores en la web. Se trata de la «métrica de la web» (*webmetrics* o *cybermetrics*). Se define como el estudio de los aspectos cuantitativos de la construcción y uso de recursos de información, estructuras y tecnologías en la web, basándose en enfoques bibliométricos [16].

El objetivo es obtener información sobre el número y el tipo de hiperenlaces, la estructura del World Wide Web y los modelos de uso de los recursos. Así pues, se contabiliza el número de veces que un sitio web o un documento publicado en Internet es accedido y se divide entre el número de páginas del mismo sitio [29]. Esto nos permite calcular la frecuencia con la que una página web promedio ha sido enlazada en un momento dado; un alto factor de impacto de la web indica la popularidad y, probablemente, el prestigio de una página web. Debido al creciente número de documentos publicados y disponibles en la red, en especial los *e-journals* de acceso abierto, se han desarrollado nuevas herramientas de medición, lo que ha llevado a algunos investigadores a considerar un posible paralelismo entre las citaciones tradicionales y las referencias en la web [29].

3.4. Índice de impacto

Existe un gran abanico de índices distintos englobados dentro de la bibliometría. Sin embargo, en este caso, nos centraremos especialmente en los índices de citas o, en inglés, «Citation Index».

Se trata de índices de autor con características especiales, pues no solamente citan junto a cada autor la lista de los documentos por él publicados, sino que añade, en cada referencia, la lista de los documentos que ha citado esta referencia en su propia bibliografía. Permiten localizar otros autores que han tratado las mismas materias y buscar también documentos más recientes que éste ya conocido [26]. Contienen literatura científica de medicina, psicología, agricultura, tecnología y documentación científica en general.

A continuación, se enumeran algunos de los índices de impacto más comunes.

SJR

SJR (SCImago Journal Rank) es un indicador de calidad de revistas científicas basado en la idea de que *cuanto más citada es una revista, mayor importancia tiene*.

Además, se entiende que *no todas las citas son iguales*[15]. Es decir, el indicador se calcula utilizando un algoritmo que también tiene en cuenta la importancia de la revista que hace la cita, de tal manera que una cita hecha por una revista de alto impacto tendrá más peso que una cita de una revista de bajo impacto [15].

Así pues, se calculan las citas **ponderadas** de los últimos tres años para determinar el SJR de cada revista.

H-index

El h-index es un indicador de productividad y visibilidad de un investigador. El h-index se calcula a partir de la cantidad de artículos publicados por un autor y la cantidad de veces que estos artículos han sido citados.

Esta métrica se determina buscando el número de publicaciones con al menos ese número de citas. Por ejemplo, un investigador con un h-index de 10 tiene al menos 10 publicaciones que han sido citadas al menos 10 veces cada una [15].

Por lo tanto, se podría decir que este indicador es una métrica a nivel de autor se utiliza para medir la importancia, la productividad y el impacto de un investigador en la comunidad científica [15]. Este índice también puede ser aplicado a un grupo de investigadores (por ejemplo, un departamento, universidad o país).

Es importante mencionar que Google Scholar utiliza métricas basadas en el h-index como, por ejemplo, el g-index[1] o el h-median[2].

[1] g-index: métrica se basa en el número de citas recibidas por los artículos de un autor y en el número de artículos que han recibido al menos «g» citas. El valor de «g» se calcula a partir de la curva de distribución de citas del autor y se utiliza para determinar el número mínimo de artículos necesarios para alcanzar un nivel de impacto determinado.

[2] h-median: esta métrica representa la mediana de citas de los artículos incluidos en el h-index del autor.

CiteScore

Es un indicador desarrollado por Scopus que tiene en cuenta el número de citas recibidas por una revista en un período de tres años, el número de documentos en la base de datos de Scopus y el número de documentos publicados en esa revista en ese período.

De esta forma, el CiteScore se calcula dividiendo el número de citas que recibe una revista en un año por los documentos publicados en los tres años previos, y dividiendo este número por el número de documentos indexados en Scopus publicados en los mismos tres años [15].

JCR

El JCR (*Journal Citation Reports*) es una base de datos desarrollada por Clarivate Analytics que proporciona información sobre revistas científicas.

Consiste en el análisis biométrico de las revistas del banco de datos ISI (*Intellect Scientific Information*), con número de citas al año de cada una, revistas que citan a otras revistas, listado de abreviaturas de títulos y su desarrollo, etc. JCR proporciona información estadística sobre la frecuencia de citas de las revistas, incluyendo el *Impact Factor* (IF) y el *Impact Factor of a Year* (IFY) entre otros indicadores [27].

De entre los datos estadísticos que se obtienen, nos importa especialmente el ya mencionado **Factor de Impacto**, que permiten determinar de una manera sistemática y objetiva la importancia relativa de las principales revistas de investigación internacionales dentro de sus categorías temáticas.

El Factor de Impacto se calcula para cada revista en cada categoría. Para ello, se suma el número de citas que, durante un año, han "referenciado" a los artículos publicados durante los dos años anteriores en dicha revista. El resultado de esa suma se divide entre el número total de artículos publicados en esos últimos dos años. Por ejemplo, si quisiéramos calcular el Factor de Impacto de la revista Educación en 2016, se haría de la siguiente manera:

$$\text{FI(Educación 2016)} = \frac{\sum \text{Citas a publicaciones de Educación en 2015 y 2014}}{\sum \text{Publicaciones de Educación en 2015 y 2014}}$$

El resultado de esta operación se interpretará en función de si es mayor o menor que 1:

- Mayor que 1: La revista ha tenido un mejor impacto del esperado en esa categoría.
- Menor que 1: La revista ha tenido un peor impacto del esperado en esa categoría.

Es decir, para poder realizar los cálculos, necesitaremos extraer el número de citas de cada artículo, la revista a la que pertenece, el año de publicación y la categoría a la que se refiere el artículo.

JCR cubre más de 12 000 revistas de más de 80 disciplinas diferentes, lo que permite comparar el impacto y la calidad de las revistas en diferentes campos. Los datos de JCR se utilizan a menudo como un indicador de la calidad y el impacto de una revista en la comunidad científica [27].

La última actualización del JCR ofrece los datos del Factor de Impacto del 2022. Este índice se calcula con un cierto retraso respecto al final del año: suele aparecer alrededor de los meses de mayo o junio del año siguiente. Esto limita el acceso a la información actualizada y justifica la importancia del trabajo que está siendo realizado, de forma que se pueda brindar información actualizada y accesible sobre el impacto de los artículos y revistas a la comunidad científica, lo cual es esencial para la toma de decisiones y la evaluación del desempeño científico.

Técnicas y herramientas

En esta sección se presentarán las distintas herramientas y recursos que se han utilizado para la realización del proyecto.

4.1. Técnicas

En el presente proyecto, se han utilizado diversas técnicas para llevar a cabo el análisis y cálculo del índice de impacto de publicaciones científicas. Estas técnicas han sido seleccionadas con el objetivo de obtener resultados precisos y confiables, y de cubrir las necesidades específicas de este estudio.

Web-Scraping

Es la práctica de recopilar datos a través de un programa que interactúe con una API [24]. Más concretamente, un programa automatizado compuesto por *queries* que realizan solicitudes HTTP para adquirir recursos de un sitio web específico. Esta solicitud se puede formatear en una URL que contenga una consulta GET o en un mensaje HTTP que contenga una consulta POST [32]. Una vez que la petición es exitosamente recibida y manejada por el sitio web seleccionado, el recurso requerido será extraído y luego devuelto al programa de *web scraping* específico.

El uso *web-scraping* resulta ser una técnica efectiva en este proyecto debido a que permite obtener datos de las distintas fuentes web de manera automatizada y eficiente. La información obtenida a través de esta técnica es esencial para calcular el índice de impacto (JCR). Además, el *web-scraping* permite obtener grandes cantidades de información en un corto período de

tiempo, lo que resulta muy útil en proyectos en el que se requiere una gran cantidad de datos.

4.2. Lenguaje de programación

En primer lugar, nos planteamos la cuestión del lenguaje o lenguajes de programación más adecuados para nuestro objetivo. Las listas de popularidad actuales nos muestran dos entornos ganadores para proyectos de *web-scraping*: Python y Javascript.

Aunque ambos lenguajes son altamente capaces para nuestro proyecto, la enorme base de conocimientos y la diversidad de herramientas creadas en el universo Python decanta la balanza hacia ese lado. Quizás JavaScript permita mejores resultados usando la gestión de memoria en *requests* simultáneas, pero a costa de un código más oscuro y difícil de mantener. Aunque JavaScript cuenta con un gran repertorio de paquetes Node.JS como utilidades de *web-scraping*, en el entorno Python es difícil imaginar una tarea para la que no se haya escrito una (o más) herramientas que resuelvan eficazmente nuestro problema.

Por otro lado, la comunidad de programadores de Python es inmensa y su creciente popularidad facilita el hallazgo de soluciones rápidamente, tanto en los foros como en la extensa documentación con la que cuenta. Aunque encontramos un rendimiento ligeramente inferior a otros lenguajes en ciertas búsquedas, es el precio a pagar por el tipado dinámico.

Como valor añadido, Python es fácil de mantener cuando necesitamos adaptar nuestro código a las cambiantes estructuras de las páginas web. Además, sus reconocidas herramientas de análisis de datos nos permiten continuar en el mismo entorno, sin necesidad de buscar alternativas para afrontar tareas relacionadas con la *data science*.

4.3. Bibliotecas y *Frameworks*

A lo largo del proyecto se ha recurrido a diversas bibliotecas de Python. A continuación, se presenta brevemente cada una de ellas.

Scholarly

Scholarly [6] se trata de una biblioteca de Python que permite acceder a los datos de Google Scholar de manera fácil y rápida. La biblioteca

proporciona una interfaz sencilla para buscar y recuperar información sobre artículos, autores y revistas en Google Scholar, incluyendo metadatos, citas y otra información relacionada.

Sin embargo, esta biblioteca ha terminado siendo descartada para este proyecto. A diferencia de otras técnicas de *web-scraping*, Scholarly no está diseñada para extraer grandes cantidades de datos de Google Scholar. La biblioteca tiene una serie de limitaciones en cuanto a la cantidad de datos que se pueden recolectar, ya que está diseñada para ser utilizada en investigaciones científicas y no para la extracción masiva de datos. Además, Scholarly está diseñada para respetar los términos de servicio de Google Scholar y no violar la política de uso de la plataforma, por lo que no se recomienda su uso para recolectar grandes cantidades de datos.

En conclusión, Scholarly es una herramienta útil para acceder a información científica y académica de manera rápida y sencilla, pero no esta diseñada para extraer grandes cantidades de datos.

Beautiful Soup

Beautiful Soup [2] es una biblioteca de Python extremadamente útil para la extracción de datos de páginas HTML o XML. Actualmente se encuentra en su versión 4.8.1.

Esta biblioteca nos proporciona numerosos módulos para navegar a través de las páginas web y para extraer fácilmente su contenido. Puesto que gran parte del proyecto se basa en el uso de *web-scraping*, se ha hecho uso intensivo de la misma para extraer los principales datos de los artículos científicos que posteriormente conformarán la BBDD del proyecto.

De Beautiful Soup hay que destacar su facilidad de uso y la amplia documentación que aporta. En su página web nos aconseja utilizar el analizador *lxml*, que proporciona al entorno Python la disponibilidad de las bibliotecas *libxml2* y *libxslt*. Allí mismo, se anima incluso a utilizar aisladamente este parser cuando el tiempo de respuesta sea una cuestión crítica. En nuestro caso, las facilidades que proporciona Beautiful Soup justifican ampliamente su uso, aunque la rapidez de resultados no iguale la de la utilización aislada de los analizadores sobre los que trabaja.

Habanero

Para poder hacer uso de la API de Crossref, se han explorado diversas alternativas. De entre todas las bibliotecas de Python se ha seleccionado

Habanero [4], ya que es una biblioteca muy fácil de usar y está en constante actualización.

Esta biblioteca está diseñada para facilitar el acceso a las bases de datos de revistas científicas y a otras relacionadas con el ámbito académico. Ofrece una interfaz simple para recuperar información utilizando los protocolos y las API de diferentes bases de datos, incluyendo JSTOR, Unpaywall, Crossref, DataCite, etc. Además, Habanero es compatible con las normas de Open Access, lo que permite a los usuarios acceder a contenido científico gratuito y de libre acceso.

Selenium

Selenium [9] es una biblioteca de Python de automatización de navegadores web que se utiliza para controlar el comportamiento de un navegador en tiempo real. Es una herramienta muy poderosa para realizar *web scraping* de sitios web, especialmente en aquellos casos en que se necesita interactuar con la página web como si lo hiciera un usuario humano.

A diferencia de otras bibliotecas de *web scraping*, Selenium permite simular la navegación humana en la web, permitiendo realizar búsquedas, hacer clic en botones, llenar formularios, entre otras acciones. Esto es muy útil cuando se trata de sitios web que tienen medidas de seguridad para evitar el *web scraping* y requieren de interacción humana.

Scikit-learn

Scikit-learn [8] es una biblioteca de aprendizaje automático en Python que ofrece una amplia variedad de algoritmos de clasificación, regresión y clustering, entre otros. Esta biblioteca destaca por su facilidad de uso y la implementación eficiente de los algoritmos, lo que permite su uso en grandes conjuntos de datos. Cuenta también con herramientas de preprocesamiento de datos y evaluación de modelos. Sus ventajas incluyen una documentación completa y una comunidad activa que ofrece soporte y mejora continuamente la biblioteca, así como la capacidad de integrarse fácilmente con otras bibliotecas y herramientas de Python para el análisis de datos.

Unittest

Para los proyectos de pruebas, hemos utilizado la biblioteca de pruebas unitarias de Python llamada Unittest [7]. Una de las principales ventajas de unittest es que se integra fácilmente con el flujo de trabajo de Test Data

Driven (metodología de pruebas empleada), permitiendo que las pruebas se ejecuten automáticamente y de manera aislada del código principal. Además, unittest proporciona una gran cantidad de funcionalidades y herramientas para facilitar la escritura y ejecución de pruebas, lo que nos ha permitido realizar pruebas complejas de manera eficiente y eficaz.

Psycopg2

Psycopg2 [5] es un adaptador de base de datos que permite a los desarrolladores interactuar de manera eficiente con PostgreSQL desde sus aplicaciones Python. Esta biblioteca es conocida por su robustez, rendimiento y amplia gama de funcionalidades. Psycopg2 proporciona una interfaz intuitiva y fácil de usar para realizar consultas, transacciones y manipulación de datos en PostgreSQL, lo que la convierte en una opción popular para aquellos que buscan una solución sólida y confiable en el desarrollo de aplicaciones Python basadas en bases de datos.

Flask

Flask [10] es un popular *framework* web de Python que ofrece una amplia gama de herramientas y funcionalidades para crear aplicaciones web de manera eficiente. Además, Flask tiene un sistema de enrutamiento flexible, plantillas Jinja2[3] y una amplia variedad de extensiones para crear aplicaciones web personalizadas y escalables.

Una de las principales ventajas de Flask es su simplicidad y facilidad de uso. Este *framework* se centra en la simplicidad y en seguir el principio de «hacer una cosa y hacerlo bien». Flask proporciona una estructura básica para crear aplicaciones web, pero también permite a los desarrolladores personalizar y ampliar sus funcionalidades según sus necesidades.

Finalmente, se ha elegido trabajar con Flask ya que es una poderosa herramienta para el desarrollo de aplicaciones web en Python. Su simplicidad, flexibilidad y capacidad de integración lo convierten en una elección popular tanto para proyectos pequeños como para aplicaciones web más grandes y complejas.

[3] Jinja2 es un motor de plantillas en Python ampliamente utilizado y altamente flexible. Proporciona a los desarrolladores una forma eficiente de generar contenido dinámico y estructurar la presentación de datos en aplicaciones web [3].

Babel

Babel [1] es una herramienta esencial en el desarrollo de aplicaciones web con Flask, especialmente cuando se trata de la internacionalización y localización de contenido. Babel proporciona una solución eficiente y sencilla para adaptar una aplicación Flask a diferentes idiomas y regiones, permitiendo así llegar a un público global.

Con Babel, los desarrolladores pueden generar y gestionar archivos de traducción que contienen cadenas de texto en diferentes idiomas. Estos archivos de traducción se denominan catálogos de mensajes y se utilizan para proporcionar versiones localizadas de la interfaz de usuario y otros textos en la aplicación. Babel ofrece una variedad de herramientas y funciones para extraer, traducir y compilar estas cadenas de texto en diferentes idiomas.

La integración de Babel en una aplicación Flask es sencilla, ya que Flask ofrece soporte nativo para esta herramienta a través de una extensión llamada Flask-Babel.

4.4. Bases de datos

Antes de su compra por Oracle Corporation (2010), MySQL era la aplicación de base de datos más popular de código abierto para la programación web. En el momento actual, el software libre nos ofrece dos soluciones ampliamente contrastadas en gestores de bases de datos relacionales: MariaDB y PostgreSQL (ver comparativa en Tabla 4.1). Necesitaremos una herramienta de este tipo para organizar los datos recolectados. La estructura tabular de la información nos permite aplicar sobre ella el lenguaje de interrogación SQL.

En realidad, MariaDB es una bifurcación de MySQL nacida para garantizar la supervivencia del proyecto como código abierto. Hoy en día, MariaDB es altamente compatible con MySQL e incluso superior en sus últimas versiones (10.1.1), ya que la comunidad ha ido añadiendo nuevas características al proyecto original. Por otro lado, aunque no está tan extendido como MySQL, PostgreSQL es posiblemente el gestor de bases de datos de código abierto más sólido y potente a día de hoy.

PostgreSQL

Finalmente se opta por PostgreSQL (también denominado Postgres). Pese a que ambas opciones son muy similares, se ha elegido esta última

MariaDB	PostgreSQL
No totalmente compatible con SQL	Compatible con SQL estándar
Soporte para tipos de datos estándar SQL	Soporta además tipos avanzados
Tipos de datos flexibles	Tipos de datos estrictos
Tamaño pequeño de base de datos	Tamaño grande de base de datos
Sin soporte directo para JSON	Soporte directo de JSON
No índices parciales	Índices parciales
No soporte para web dinámica	Soporta sitios web dinámicos

Tabla 4.1: Comparativa entre MariaDB y PostgreSQL

debido a que ya se ha trabajado con ella anteriormente. Además, PostgreSQL posee una sólida reputación por su arquitectura comprobada, confiabilidad, integridad de datos, conjunto sólido de características, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para ofrecer soluciones innovadoras y de rendimiento constante [18].

Al igual que MariaDB, es un sistema de gestión de bases de datos relacional de código abierto. Así pues, está dirigido y desarrollado por una comunidad altruista de desarrolladores (PostgreSQL Global Development Group). PostgreSQL utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas [18]. Su última versión y la usada para el proyecto es la versión 15.1 lanzada el 10 de noviembre de este año.

4.5. APIs externas

A lo largo del proyecto se ha recurrido a varias APIs diferentes para la fase de extracción de datos. A continuación, se enumeran aquellas de acceso gratuito con las que se ha podido trabajar (a lo largo de esta memoria se nombrarán también otras APIs pero que, finalmente, terminan siendo descartadas por ser de pago y con grandes limitaciones legales).

Google Scholar

Se trata de una de las principales herramientas que ofrece Google a los investigadores. Google Scholar es, fundamentalmente, un buscador de contenido y bibliografía científica que permite localizar artículos de revistas especializadas ordenados por relevancia en función de las palabras clave

introducidas en el buscador. También se puede filtrar la información en función de su fecha de publicación, idioma o número de citas.

Crossref

Crossref es una organización sin ánimo de lucro que provee de una herramienta que facilita el acceso a la información de los artículos científicos a partir de su DOI[4].

Como ya se ha mencionado previamente, en base a la información de los DOIs la agencia de registro CrossRef es capaz de proporcionarnos aplicaciones útiles para hacer nuestro flujo de investigación más sencillo. Se trata de una asociación sin ánimo de lucro de editoriales científicas que no solo facilita el registro de DOIs a las editoriales, sino que también ofrece servicios y aplicaciones para el personal investigador que tienen como base estos códigos.

Los DOIs que CrossRef almacena van acompañados de información que refleja las cualidades básicas de una publicación científica. Me estoy refiriendo a datos como títulos, abstracts, palabras clave, autores...

CrossRef Metadata Search hace posible obtener toda esta información al instante con tan solo proporcionar el DOI asociado a una publicación, o al contrario, obtener el DOI de la publicación con tan solo introducir algunos de estos datos en su buscador.

4.6. ZenHub

ZenHub [11] es una herramienta de gestión de proyectos que se utiliza en muchos entornos empresariales para mejorar la eficiencia en el seguimiento y administración de tareas. Con ZenHub, es posible planificar y visualizar fácilmente las actividades de un proyecto, monitorizar el progreso de cada tarea y hacer un seguimiento de los plazos de entrega. También ofrece funciones útiles como integración con GitHub, herramientas de informes y análisis de datos, y seguimiento de errores y problemas.

Además, ZenHub utiliza la secuencia de Fibonacci para medir el peso de las tareas en *story points* (que es una escala comúnmente utilizada en el contexto de la metodología ágil). La secuencia de Fibonacci es una serie

[4]El DOI (*Digital Object Identifier*) es el acrónimo con el que se conoce al identificador inequívoco y persistente de un objeto digital (como un artículo de revista, un libro electrónico, una imagen, etc...). Se trata de un enlace permanente al contenido electrónico de dicho objeto. Por lo general, un DOI tiene forma de código alfanumérico.

matemática en la que cada número es la suma de los dos anteriores: 1, 2, 3, 5, 8, 13, 21, etc. La razón del uso de la secuencia de Fibonacci para los *story points* es evitar la tendencia a dar estimaciones precisas y lineales. De esta forma, se busca que los equipos se enfoquen más en la relativa complejidad y esfuerzo de una tarea en lugar de estimaciones exactas de tiempo.

Aspectos relevantes del desarrollo del proyecto

En esta sección se presenta un resumen de los hallazgos más relevantes obtenidos a lo largo del proyecto, así como una descripción detallada de las distintas fases por las que se ha atravesado hasta lograr alcanzar una solución satisfactoria al problema planteado. A saber, la fase de extracción de datos, la fase de aprendizaje automático y la fase de creación de la aplicación web.

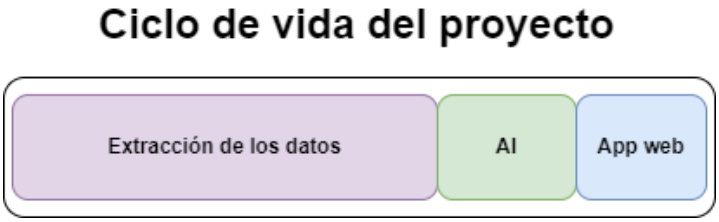

Figura 5.1: Fases del proyecto

5.1. Extracción de la información

La primera fase de este proyecto es la más extensa y consiste en la extracción de datos para alimentar la base de datos con la que entrenar los modelos. Para esto, es necesario tener en cuenta varias bases de datos bibliográficas que ofrecen diversos recursos para los investigadores. Entre las más importantes se encuentran Google Scholar, Scopus, WoS y Crossref.

Prototipos para Google Scholar

De todas las bases de datos bibliográficas mencionadas, la más destacada es Google Scholar, puesto que crece más rápido que cualquier base de datos tradicional en todos los campos científicos [20, 23].

Así pues, el primer prototipo diseñado consiste en la extracción de datos de Google Scholar. Sin embargo, si bien es cierto que Google Scholar es un motor de búsqueda gratuito, universal y rápido con una amplia cobertura, tiene muchas limitaciones [23].

Por ejemplo, no hay funcionalidades de exportación de datos o API disponibles debido a restricciones comerciales. Además, solo se pueden mostrar los primeros 1 000 resultados de cada consulta. Aunque algunas técnicas como los retrasos temporales o el uso de *proxies* pueden ayudar, no resuelven completamente estas limitaciones. Además, la extracción automatizada de estos datos va en contra de las políticas del archivo `robots.txt` de Google Scholar, lo que hace que los usuarios que realizan demasiadas consultas automatizadas sean bloqueados cada 200 solicitudes detectadas [23].

Durante el avance del proyecto, nos dimos cuenta de muchas de las limitaciones mencionadas.

Prototipo inicial

Tras comprender que las complicaciones eran numerosas, se decidió crear un prototipo sencillo, consistente en un *script* en Python, que trataría de lanzar mil peticiones de búsqueda. Esto nos permitiría establecer los límites de realizar *web scrapping* sobre Google Scholar.

Así pues, manos a la obra, se desarrolló un *script* sencillo, que solicita acceso a la página principal de Google Scholar y, mediante métodos HTTP, realiza una búsqueda (a partir de parámetros solicitados por pantalla). Finalmente, extrae el título de la página resultante tras hacer la búsqueda. Todo esto se logra haciendo uso de la biblioteca de Python BeautifulSoup.

El primer obstáculo encontrado es que Google Scholar no permite hacer búsquedas en función de la revista. Si se trata de escribir el nombre de una revista en el buscador, aparecerán artículos relacionados que mencionan esa revista, pero no siempre artículos de la revista en cuestión. En cambio, permite hacer búsquedas a partir de palabras clave. Sin embargo, aunque no se puede buscar directamente por revista, sí se puede extraer de los resultados encontrados tras buscar una temática concreta.

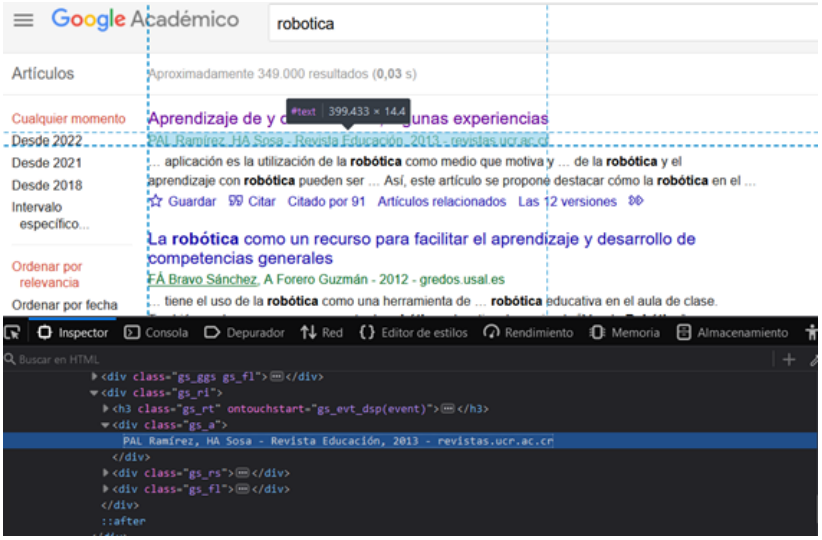

Figura 5.2: Ejemplo de búsqueda por palabra clave

Sin embargo, existe una problemática: en cada artículo se hace referencia a la revista que lo publica de forma distinta. Por ejemplo, como se puede apreciar en la figura 5.3, la revista se encuentra ubicada en una «etiqueta» HTML diferente.

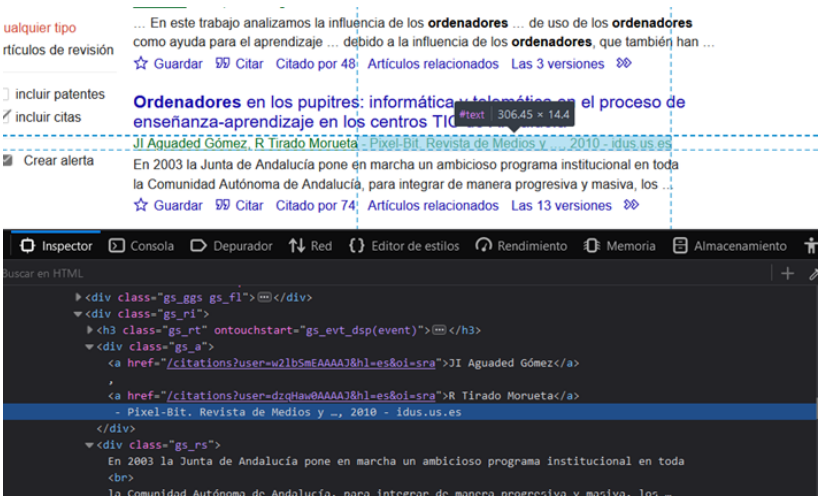

Figura 5.3: Localización de la revista en Google Scholar

Por ello se buscó una solución alternativa. Existe una etiqueta (`<b> ... </b>`) que incluye aquellas palabras resaltadas en negrita en la página web. Google Scholar resalta en una búsqueda aquellas palabras que coinciden

con alguno de los parámetros buscados. Así pues, si se realiza una búsqueda en Google Scholar pasando como parámetro el nombre de la revista que nos interesa, bastará con buscar en la clase `gs_a` (donde se almacenan los detalles del artículo) alguna etiqueta `<b>`. Se ha escrito un pequeño código de prueba y, en principio, parece que funciona sin problemas. Se adjunta dicho código (`extraerRevistas.py`) en el repositorio de GitHub. Este código hace dos búsquedas: una a la revista «Alas Peruanas» y otra a la revista «The Lancet» (elegidas de forma aleatoria).

La salida que se mostrará por pantalla sigue la siguiente estructura: [`'nombre artículo', 'nombre de la revista', número_citas, fecha, id`]

De igual forma se recogen los resultados en los CSVs (codificados en UTF-8) con nombre: `BBDD1.csv` y `BBDD2.csv`.

Otra de las mejoras que se implementa en el prototipo en la capacidad de navegar a través de las distintas páginas de resultados de Google Scholar. Para ello, se incluye un nuevo campo en la consulta de búsqueda que irá aumentando de 10 en 10 por cada página de resultados nueva que se consulte. Ahora ya se obtiene un número de resultados considerable (aproximadamente 100). Estos resultados se guardarán también en los CSVs (codificados en UTF-8) con los siguientes nombres: `BBDD3.csv` y `BBDD4.csv`.

Tras diseñar y programar el código mencionado, se procede a su prueba. La primera ejecución del *script* resultó desaletadora, ya que, a partir de la solicitud número 726, Google Scholar detecta que un *bot* está realizando búsquedas. A partir de ese momento, nuestra dirección IP queda bloqueada y las solicitudes fallan sin excepción. Google Schoolar nos redirecciona a una página (figura 5.4) donde se solicita al usuario resolver un *captcha*.

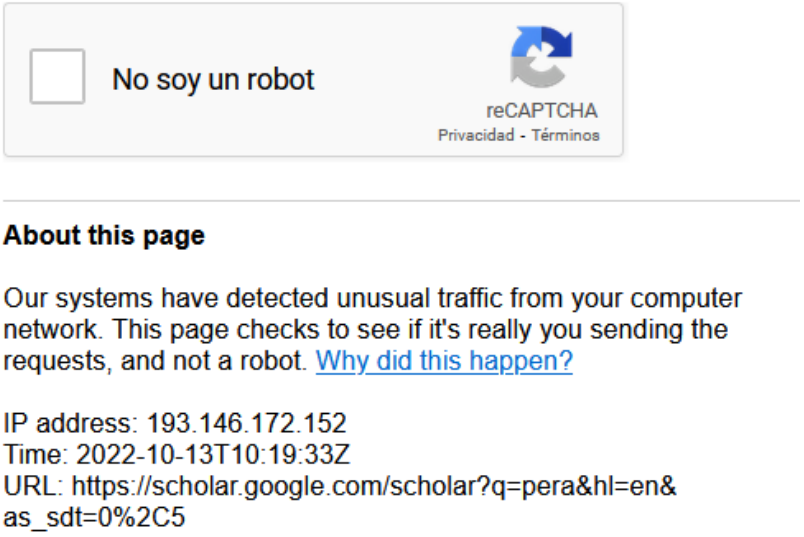

Figura 5.4: Captura de reCAPTCHA de Google

El número de solicitudes exitosas es demasiado bajo para cumplir su función en nuestro proyecto, por lo que se procede a buscar una solución alternativa.

Tras distintas pruebas, se encontró una forma de superar la barrera del *captcha*. A saber: añadiendo a la *url* una sección de texto extra que permite suprimir esta excepción durante un periodo concreto de tiempo. Así pues, logramos ejecutar con éxito el *script* tantas veces como fuese necesario. Sin embargo, esta solución tampoco es válida a largo plazo. Si se intenta ejecutar de nuevo el programa, en otra sesión, la dirección vuelve a ser inválida. Además, es un remedio poco práctico, ya que se debe concatenar distintos parámetros y cadenas de texto que, dependiendo del momento y el contexto en que se ejecute, pueden no servir.

Como la propuesta anterior no fue satisfactoria, se siguieron buscando opciones. La siguiente propuesta consistía en usar un agente de usuario distinto para cada solicitud[5]. De esta forma «enmascaramos» nuestra dirección IP. La biblioteca Request de Python ofrece métodos para lograrlo.

[5]Un agente de usuario es cualquier software, que actúa en nombre de un usuario, que «recupera, presenta y facilita la interacción del usuario final con el contenido web». Algunos ejemplos destacados de agentes de usuario son los navegadores web. La cadena User-Agent es uno de los criterios por los cuales los rastreadores web pueden ser excluidos

Dicho esto, se implementó un método que extrae *proxies* de listas públicas y gratuitas de Internet (e.g., proxyscrape.com). Se prueba la ejecución del prototipo una vez más y, finalmente, funciona sin inconvenientes. Ahora ya se puede decir que el proyecto es **viable**.

Prototipo con extracción del DOI

La siguiente meta de nuestro prototipo es la obtención del DOI, que es un campo fundamental que funcionará a modo de clave primaria en la base de datos. De esta forma se busca evitar futuros errores a la hora de identificar un artículo o a la hora de comparar artículos para eliminar duplicados. Sin embargo, se plantea una problemática al respecto: Google Scholar no comparte el DOI de los artículos en ninguna división de su página web (si bien es cierto que algunos artículos lo incluyen al final de su URL, pero no siempre ocurre esto). Puesto que por el momento no existe una forma de obtenerlo directamente, se han propuesto varias soluciones, a saber:

- Comprobar el porcentaje de éxito de extraer el DOI de la URL de los artículos (en aquellos casos en los que aparezca).
- Acceder a la página del artículo en cuestión y extraerlo de dicha página.
- Introducir el nombre del artículo en la página de Crossref o similares, que te permiten obtener el DOI del artículo cuyo nombre pases como parámetro.

Como conclusión, se descarta la primera opción tras hacer una breve prueba, ya que falla en seguida. Se descarta también la segunda opción, ya que cada página sitúa el DOI en un sitio haciendo muy difícil la búsqueda del mismo a través de un algoritmo. Por lo tanto, la opción más adecuada en este caso es la tercera. Aunque supone una búsqueda extra en la complejidad algorítmica del prototipo, es la única forma segura de calcular el DOI sin equivocaciones. Por lo tanto, se incluye en el bucle del prototipo un nivel extra de profundidad de búsqueda.

Para hacer la búsqueda en Crossref es necesario indicar tanto el título como el año de publicación, a fin de asegurarnos de que se obtiene exactamente el artículo que deseamos y no otros similares. Además, es preciso tener en cuenta que Crossref no permite a los programas hacer búsquedas desde la misma interfaz que los usuarios (para evitar que los programas bloqueen las

del acceso a ciertas partes de un sitio web utilizando el Estándar de exclusión de robots (`archivo robots.txt`).

búsquedas de los usuarios). Por lo tanto, se realizarán las búsquedas en la dirección donde se ubica la API creada por Crossref especialmente para la extracción de datos por parte de programas.

Prototipo multihilo

Con el prototipo listo, se prueba a extraer la información de los artículos de las primeras 20 páginas de resultados de Google Scholar de dos revistas. Los resultados no son muy esperanzadores: se obtienen 180 artículos de cada revista en 15 minutos. Estos resultados no son eficientes, por lo que se tratará de optimizar el prototipo siguiendo dos pautas.

- La primera pauta es tratar de extraer las llamadas a Crossref de forma que solo se tenga que realizar una única llamada una vez que se han extraído el resto de detalles sobre los artículos.
- La segunda pauta es emplear programación concurrente para ejecutar varios hilos al mismo tiempo.

Así, cada revista será procesada por un hilo distinto. Para ello, será necesario recurrir a la librería de Python *multiprocessing*. Tras actualizar estos cambios, el prototipo obtiene resultados mucho más rápidos: obtiene los 180 artículos de ambas revistas en tan solo 8 minutos. El próximo objetivo es determinar el número máximo de hilos que se pueden lanzar sin impactar negativamente el rendimiento del prototipo.

Para ello, se solicitó permiso para acceder a los servidores de la universidad. A través de SSH, nos conectamos a una de las máquinas y, utilizando un entorno virtualizado de Miniconda, comenzamos a lanzar hilos. Sin embargo, el resultado fue desastroso, ya que al realizar 200 llamadas a Google Scholar, nos bloqueaban y teníamos que resolver un CAPTCHA. Intentamos insertar esperas de tiempo aleatorias y cambiar el user-agent en cada una de las llamadas. Incluso se probó a cambiar el término de búsqueda. Sin embargo, nada de esto funcionó. Matemáticamente, después de aproximadamente 200-250 llamadas, aparecía el CAPTCHA.

Prototipo con Selenium

La única solución posible frente a las restricciones de Google Scholar es tratar de «humanizar» las búsquedas para que parezca que las realiza un ser humano. Por lo tanto, se decidió emplear Selenium, que nos permite emular los pasos que realizaría un usuario normal al hacer una búsqueda.

El nuevo prototipo entra directamente en el navegador con la ruta de la búsqueda, lo que ahorra la necesidad de realizar llamadas adicionales. Luego, hace clic en la sección citar de cada uno de los resultados de la búsqueda y aparece un cuadro emergente del cual se puede extraer la información completa de la cita. Esto resuelve otro de los problemas que teníamos con el prototipo anterior: Google Scholar acorta los nombres largos añadiendo puntos suspensivos, mientras que en la sección citar aparecen los nombres completos.

Después de extraer esta información, se cierra el cuadro y se procede a realizar la misma acción en el siguiente artículo. Cuando se llega al décimo resultado (ya que solo hay 10 artículos por página de resultados), se emula un clic en el botón «Siguiente» y se procede a realizar lo mismo en la siguiente página de resultados, todo con pausas aleatorias de tiempo entre cada acción. Posteriormente, se procesan los datos de las citas usando expresiones regulares y transformando los diccionarios de datos en un CSV.

Como nota adicional, cabe mencionar que, si se incluye la etiqueta `source: <nombre de la revista>` en la búsqueda, es posible hacer una búsqueda solo por el nombre de la revista, algo que desconocíamos hasta ahora.

Aunque este método ha probado ser mejor que el anterior, ya que se obtuvieron casi 300 resultados, Google Scholar detecta que son consultas automatizadas y salta el CAPTCHA al final.

Para intentar resolver esta situación, se ha intentado superar el CAPTCHA emulando un clic sobre el botón correspondiente. No obstante, para lograrlo es necesario superar varias capas ocultas que se encuentran sobre dicho botón, lo cual resulta complejo. Una vez se logra acceder al botón, aparecen las imágenes que se deben reconocer y esto no se puede automatizar de manera sencilla. Se llega a la conclusión de que no es posible forzar más llamadas de las permitidas en Google Scholar, tal y como se ha mencionado en otros trabajos, como en el caso de *Publish or Perish* [20].

Web of Science y Scopus

Se consideraron también estas dos potentes fuentes bibliográficas. Sin embargo, para acceder a los datos a través de sus APIs correspondientes, se requiere una licencia que permita su uso. A pesar de haber solicitado dichas licencias, las limitaciones y restricciones de ambas APIs (tanto en el ámbito legal como en el económico) han impedido la realización de prototipos que

cumplan con los objetivos requeridos para este proyecto. Finalmente, tras varios intentos, se terminan descartando ambas opciones.

Prototipos para Crossref

Vista la imposibilidad de obtener una cantidad aceptable de resultados de Google Scholar, Scopus y WoS, se decidió cambiar la fuente de datos. En este caso, Crossref (aunque no es una base de datos tan potente, tiene una API gratuita y sin tantas restricciones).

Este modelo nos permite la extracción de una gran cantidad de datos en un tiempo reducido. Para ilustrar la eficiencia de este modelo, se ha generado una gráfica donde se muestra el tiempo que se ha tardado en extraer los datos (de los últimos 20 años) de cada revista de la categoría de *Computer Science*.

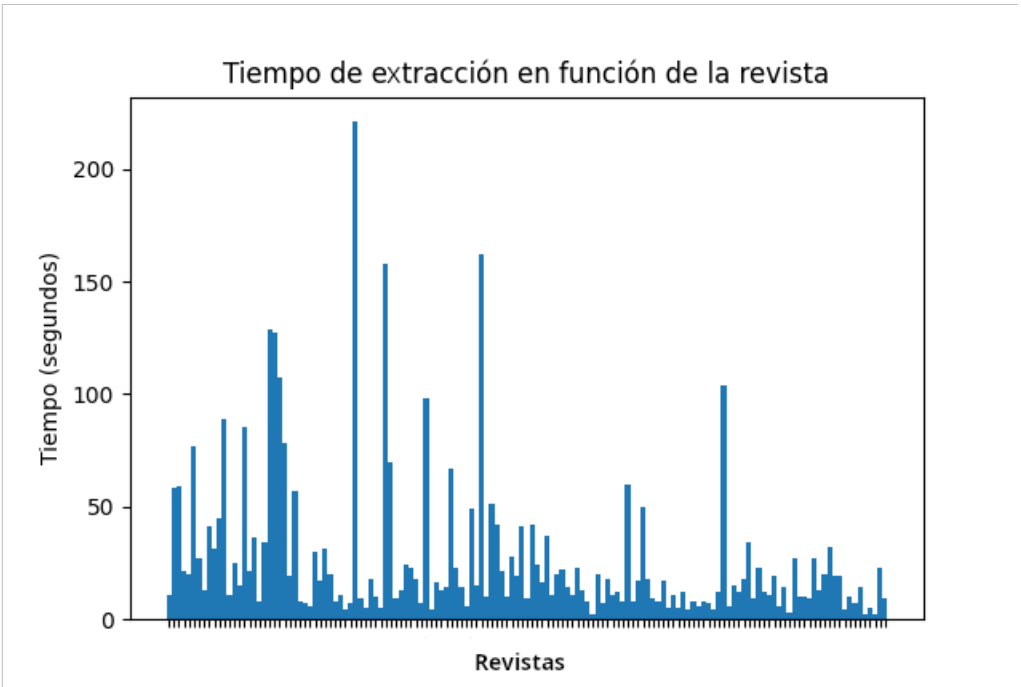

Figura 5.5: Tiempo de extracción de datos de Crossref

Sin embargo, puesto que los datos que se extraen son menos completos y exactos, posteriormente se deberá tratar el margen de error al calcular el Índice de Impacto. Además, es preciso mencionar que, según retrocedemos en el tiempo, la escasez de datos disponibles en Crossref aumenta. Así, resulta

evidente que la fiabilidad del cálculo del Índice de Impacto disminuye como consecuencia.

5.2. Aprendizaje automático

La segunda fase del proyecto consiste en predecir y estimar, a partir de los datos obtenidos, el valor del Índice de Impacto de cada revista seleccionada.

Cálculo del JCR

Antes de comenzar a predecir el JCR, se ha desarrollado un algoritmo para calcular el Índice de Impacto de las revistas científicas a partir de los datos extraídos en la fase anterior. Para comprobar la exactitud de los resultados obtenidos, se ha generado un gráfico de cajas en el que se contrastan los valores obtenidos con los datos reales del JCR.

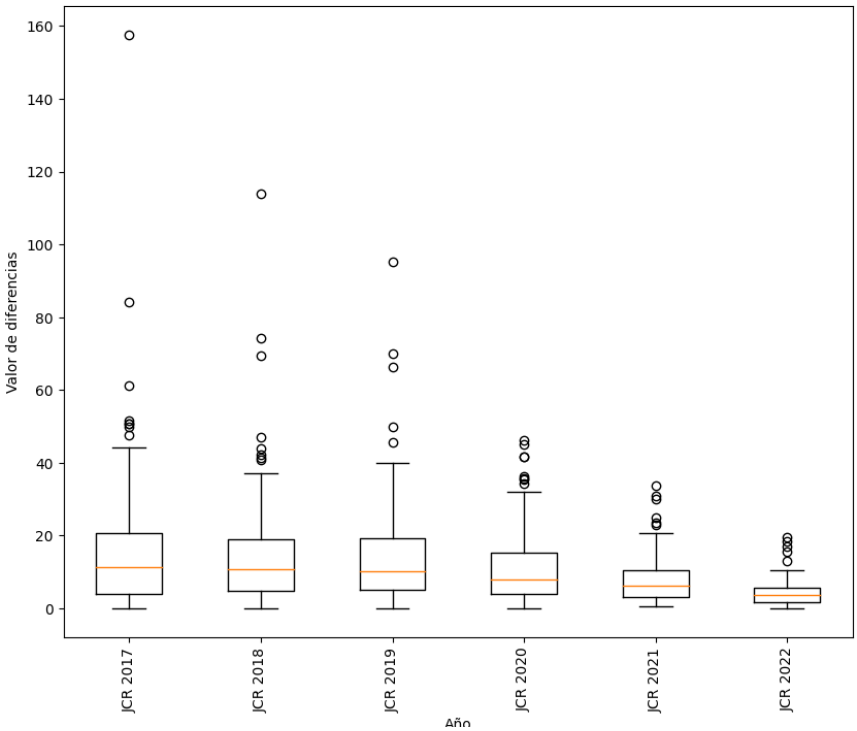

Figura 5.6: Dispersión entre las diferencias del JCR calculado y el real en función del año

Al analizar el gráfico, se observa claramente que los años más antiguos presentan un margen de error mucho más elevado debido a la falta de datos de Crossref en esos años. De esta manera, se puede asegurar que el módulo desarrollado es preciso y confiable, aunque se debe tener en cuenta la limitación de la falta de datos para los años más antiguos.

Dadas las circunstancias, se ha tomado la decisión de utilizar solamente los datos de los años más recientes para entrenar los modelos de aprendizaje automático. Esta medida se ha tomado debido a la limitación de datos disponibles para los años más antiguos, que resultaría en un margen de error demasiado elevado si se incluyeran en el entrenamiento de los modelos. Aunque esto pueda suponer una pérdida de información valiosa, se considera que es preferible tener resultados más precisos y confiables al utilizar datos más actualizados. Con esta estrategia, se espera obtener resultados más precisos y útiles.

Modelos de predicción

Se han probado varios modelos regresores utilizando la librería Scikit-learn, con el objetivo de encontrar el que mejor se ajuste a los datos disponibles y pueda hacer predicciones más exactas. Además, estos modelos se han evaluado mediante el error cuadrático medio[6]. A continuación, se enumeran los distintos modelos probados:

- Linear Regression
- Random Forest Regressor
- AdaBoost Regressor
- XGB Regressor
- Support Vector Machine Regressor (SVM)
- Multi Layer Perceptron (MLP)
- Stacking Regressor

[6]RMSE (Root Mean Square Error) es una métrica ampliamente utilizada para evaluar la precisión o el rendimiento de un modelo de regresión. Representa la raíz cuadrada de la media de los errores cuadrados entre los valores predichos por el modelo y los valores reales del conjunto de datos.

Se han seleccionado estos modelos debido a su popularidad y a su variedad, que abarca una amplia gama de métodos disponibles.

Además se ha tenido en cuenta la relevancia y los hallazgos presentados en el artículo titulado «Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?» [17]. Este estudio evaluó 179 clasificadores provenientes de 17 familias diferentes, abarcando una amplia variedad de métodos, incluyendo análisis discriminante, Bayesianos, redes neuronales, máquinas de soporte vectorial, árboles de decisión, clasificadores basados en reglas, técnicas de *boosting*, *bagging*, *stacking*, modelos lineales generalizados, vecinos más cercanos, regresión de múltiples splines de adaptación y muchos otros métodos.

Los resultados de este estudio destacaron que los clasificadores de Random Forest fueron los más prometedores, alcanzando una precisión máxima del 94,1 % en los conjuntos de datos utilizados, superando el 90 % en el 84,3 % de los casos. Además, se encontró que los clasificadores SVM con núcleo gaussiano, también obtuvieron una alta precisión del 92,3 %. Estos hallazgos respaldan la decisión de considerar estos modelos en este estudio, ya que se ha demostrado que ofrecen un rendimiento destacado en comparación con otros clasificadores.

Por otro lado, hay otros artículos que sugieren la utilización de los modelos XGBoost y Stacking en proyectos de aprendizaje automático. El artículo titulado «Getting Started with XGBoost in scikit-learn» [31] destaca que XGBoost es un algoritmo de aprendizaje automático que ha ganado popularidad debido a su desempeño sobresaliente en competiciones de Kaggle y en la predicción de datos tabulares. XGBoost es un modelo de *ensemble* que combina varios modelos de aprendizaje en uno solo, ofreciendo resultados superiores a los modelos individuales. Además, se destaca su capacidad de regularización y velocidad de procesamiento, lo que lo convierte en una opción atractiva para aplicaciones prácticas.

Por su parte, el artículo «Stacked generalization: an introduction to super learning» [25] presenta el método de *Stacked Generalization*, también conocido como *Super Learner*. Este enfoque permite combinar varios algoritmos de predicción en un único modelo. Utilizando la validación cruzada, se busca obtener una combinación óptima de las predicciones de una biblioteca de algoritmos candidatos. La optimización se realiza mediante una función objetivo especificada por el usuario, como minimizar el error cuadrático medio o maximizar el área bajo la curva característica de operación del receptor. Aunque la implementación de *Super Learner* puede tener ciertas complejidades conceptuales y técnicas, su uso ha demostrado ser beneficioso

en diversas aplicaciones y puede ofrecer mejoras significativas en la precisión de las predicciones.

En nuestro caso, tanto XGBoost como Stacking han demostrado ser enfoques efectivos en la predicción del JCR, y su inclusión se justifica por su rendimiento sobresaliente y su potencial para mejorar la precisión de las predicciones en este proyecto.

Conjunto de datos

Para el conjunto inicial de datos, se ha realizado una selección cuidadosa de entre toda la información extraída, eligiendo los siguientes atributos: el número de citas, el factor de impacto JCR y la diferencia entre los datos extraídos y los valores reales. Estos atributos han sido tomados exclusivamente de los años comprendidos entre el 2018 y el 2020. Además, se ha incluido el número de citas correspondiente al año 2021. La elección de estos años se basa en la observación de que presentan menor error en los datos extraídos, lo cual contribuye a mejorar la calidad de los atributos seleccionados. Finalmente, se realizará la predicción del Factor de Impacto JCR para el año 2021.

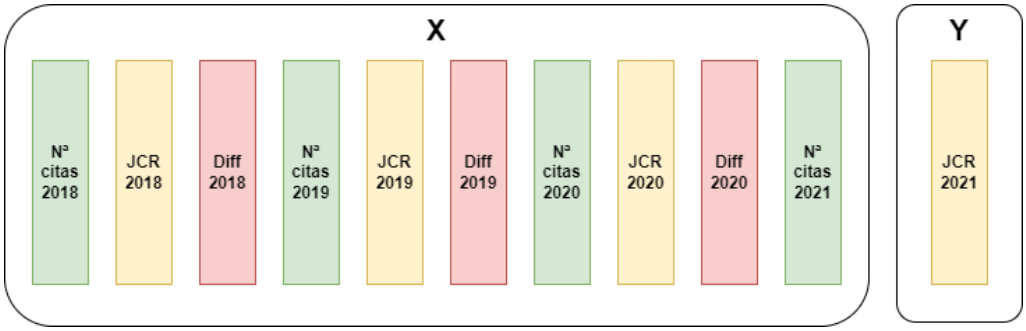

Figura 5.7: Representación esquemática de los datos de entrenamiento (X,y)

En la figura anterior (Figura 5.7) X representa las variables de entrada al conjunto de entrenamiento e y la clase a predecir.

Por otro lado, cuando se trata de valores nulos o vacíos (i.e.,*missing values*), se ha utilizado la mediana como método imputación de valores.

Entrenamiento de los modelos

Para el entrenamiento, se ha llevado a cabo una técnica de validación cruzada anidada (*nested cross-validation*) con el fin de evaluar y seleccionar

el mejor modelo posible para predecir el Índice de Impacto de las revistas científicas. Esta técnica implica el uso de dos niveles de validación cruzada: en el nivel externo se evalúa el desempeño general del modelo, mientras que en el nivel interno se ajustan los parámetros del modelo mediante una búsqueda en rejilla (`grid search`) para encontrar la mejor combinación de hiperparámetros. La ventaja de esta técnica es que permite evaluar la capacidad de generalización del modelo de manera más realista, evitando la selección de modelos sobreajustados (*overfitting*). En comparación con una validación cruzada al uso, la técnica de validación cruzada anidada puede resultar más costosa en términos de cómputo, pero proporciona resultados más precisos y fiables.

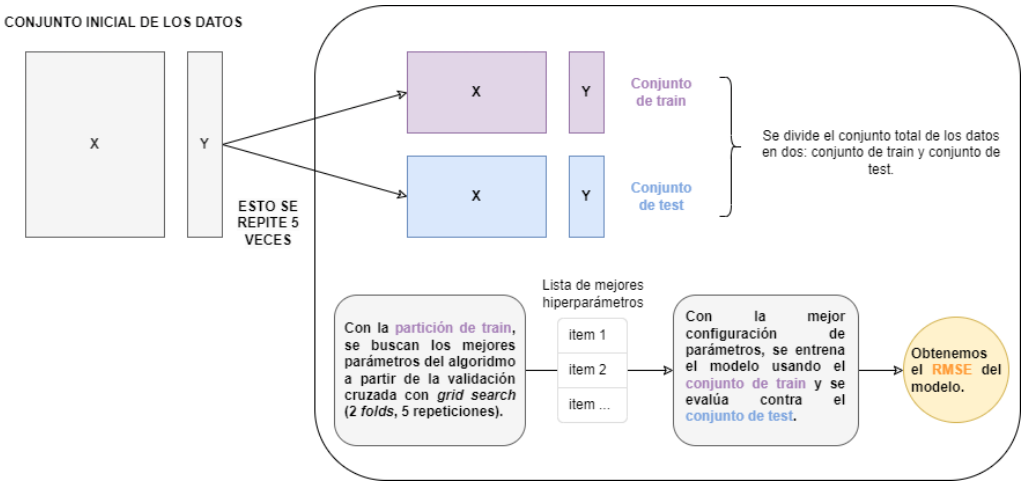

Figura 5.8: Esquema de la validación cruzada anidada (5 repeticiones, 2 grupos)

Evaluación de los modelos

Finalmente, tras la evaluación de todos los modelos, se seleccionan aquellos que han obtenido mejores resultados. Como se puede observar en los siguientes gráficos, los modelos con mayor precisión son AdaBoost Regressor, Random Forest Regressor y XGB Regressor.

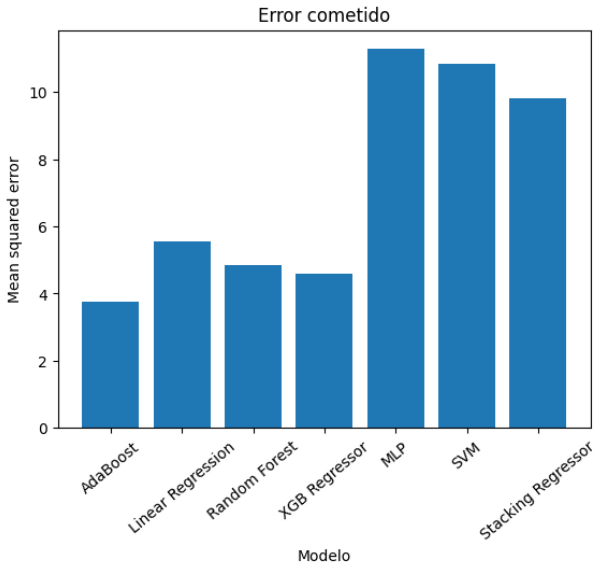

Figura 5.9: Estimación del RMSE de los modelos evaluados en la experimentación

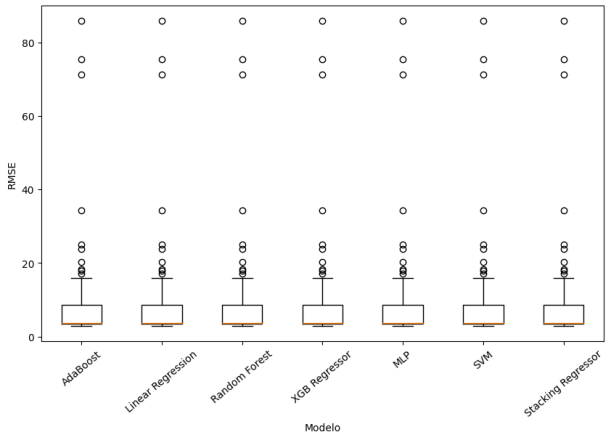

Figura 5.10: Desviación de las estimaciones

Los motivos por los cuales se han obtenido estos resultados pueden ser diversos. Por un lado, podemos ver que los modelos de *ensemble* como AdaBoost, Random Forest y XGBoost obtienen, en general, mejores resultados. Esto puede deberse a su capacidad para capturar relaciones no lineales en los datos. Así, pueden modelar relaciones complejas entre las variables de entrada y la variable de salida, lo cual es especialmente útil cuando existen patrones no lineales en los datos.

Por otro lado, los algoritmos de *ensemble* tienden a reducir el sobreajuste en comparación con modelos individuales como MLP y SVM (los algoritmos de *ensemble* combinan múltiples modelos más simples, lo que ayuda a mitigar el sesgo y la variabilidad inherente a un solo modelo).

Estos algoritmos suelen ser, también, más robustos ante la presencia de ruido o valores atípicos en los datos (como ocurre en nuestro caso). Utilizan técnicas como el muestreo *bootstrap* y la combinación de múltiples árboles de decisión, lo que les permite ser menos sensibles a observaciones atípicas y errores de medición.

En cambio, MLP y SVM a menudo requieren una cuidadosa normalización y escala de los datos de entrada para un rendimiento óptimo. Además, MLP puede ser más sensible a la selección de hiperparámetros y puede requerir una búsqueda más exhaustiva de la configuración adecuada, aumentando así el tiempo de entrenamiento.

Resultados

Tras analizar los resultados de esta etapa, para poder hacer uso de ellos más adelante, se guardan en los siguientes ficheros:

- Fichero CSV con los resultados de la validación cruzada. Por cada iteración incluye el modelo que se estima, el valor de los parámetros con mejores resultados y el RMSE.
- Ficheros binarios *pickle*, donde se almacenan cada uno de los modelos entrenados. Posteriormente, se almacenarán en la base de datos para poder realizar predicciones desde la aplicación web.

5.3. Creación de la aplicación web

Se pueden destacar algunos aspectos del proceso de creación de la aplicación web, especialmente los relacionados con la base de datos y el servidor de aplicaciones en la nube Heroku.

Uno de estos aspectos es la imposibilidad de realizar con Heroku Postgres[7] operaciones para optimizar la base de datos (i.e.: sentencias como `ANALIZE`), probablemente por una cuestión de permisos de usuario.

[7]Heroku Postgres es una solución de alojamiento y gestión de bases de datos PostgreSQL proporcionada por Heroku.

Otro aspecto relevante en cuanto a Heroku es el comportamiento inconstante en la carga de las variables de sesión. En este caso, se están considerando variables de sesión tanto el idioma como el nombre del usuario que ha iniciado sesión. En algunas ocasiones, estas variables se cargan correctamente, mientras que en otras ocasiones no se cargan como se espera. Esto es particularmente desconcertante, ya que se espera que el comportamiento de las variables de sesión sea determinista y consistente. Se sospecha que este problema está relacionado con la velocidad de respuesta de la aplicación alojada en Heroku. Es posible que, debido a la naturaleza distribuida de la plataforma y a las variaciones en la carga del servidor, la respuesta del sistema para cargar las variables de sesión no siempre sea uniforme. Esto puede dar lugar a una experiencia inconsistente para los usuarios, ya que las variables de sesión pueden o no estar disponibles en diferentes momentos.

Por otro lado, en relación con la librería *psycopg2* (que se ha usado para realizar la conexión a la base de datos), se ha descubierto (utilizando la función `copy_from`) que no admite nombres de columnas en mayúsculas, lo cual es incompatible con la notación de PostgreSQL. En la documentación de psicopg2 se asegura que `copy_from` funciona con la misma sintaxis de PostgreSQL, mientras que, en la documentación de PostgreSQL encontramos que «*Key words and unquoted identifiers are case insensitive*» [14]. Se ha notificado este error en la *issue* #1581.

Trabajos relacionados

En el campo de la evaluación de revistas científicas, existen varios trabajos y proyectos previos que han tratado de extraer datos y calcular el factor de impacto. A continuación se presenta una breve descripción de uno de los trabajos relacionados más relevante.

6.1. Publish or Perish

Publish or Perish (PoP) es una aplicación de escritorio gratuita que se utiliza para evaluar la producción académica de un investigador o institución en base a una serie de métricas bibliométricas. Fue desarrollada por la profesora Anne-Wil Harzing en 2006 y, desde entonces, ha sido una herramienta muy exitosa entre los académicos.

Como se afirma en el artículo *Google Scholar: the 'big data' bibliographic tool*, *«If any third party tool deserves a place in Google Scholar's Hall of Fame, this would undoubtedly be Publish or Perish [...]»* [23].

La aplicación ha sido programada en lenguaje de programación Delphi y está disponible para su descarga gratuita en la página web de Harzing. La interfaz de usuario es sencilla y fácil de usar, lo que la hace accesible para cualquier investigador, independientemente de su nivel de experiencia en informática.

La aplicación utiliza datos de las bases de datos bibliográficas Scopus, Google Scholar, Web of Science, Crossref, OpenAlex, Semantic Scholar y PubMed para obtener información sobre la producción académica de un investigador. A partir de esta información, la aplicación calcula una serie de métricas bibliométricas, como el h-index, el g-index, el número de

citas y el número de artículos publicados, entre otros [20]. Además, toda esta información puede ser copiada y exportada en numerosos formatos fácilmente.

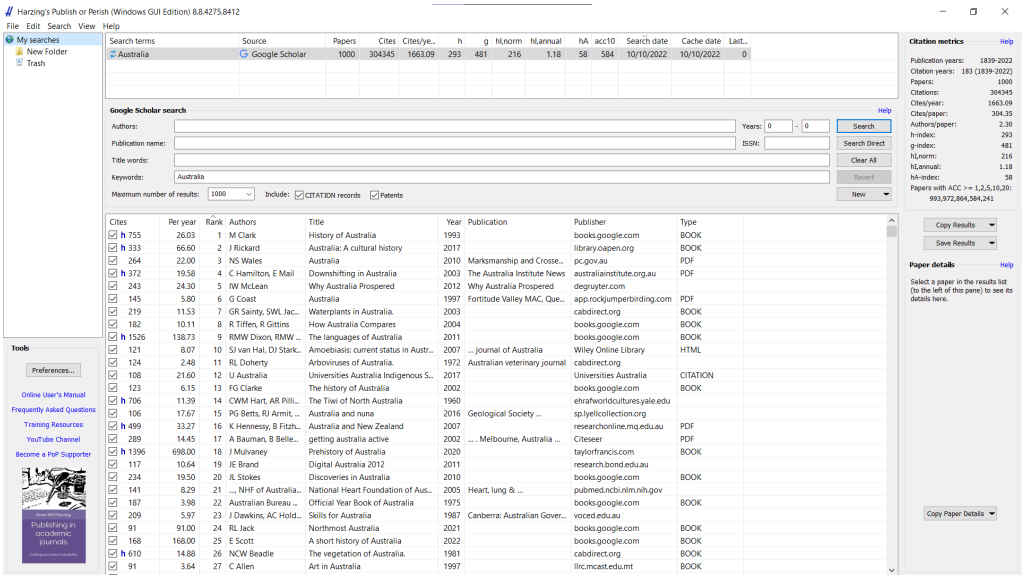

Figura 6.1: Ejemplo de búsqueda con PoP usando Google Scholar

Cabe destacar que las limitaciones en la cantidad de solicitudes que se pueden hacer a las bases de datos bibliográficas (como Google Scholar) están relacionadas con la política de cada sitio y con las restricciones de las APIs que utilizan para acceder a sus datos [20]. En el caso de Publish or Perish, su autora ha seguido las políticas de cada sitio y ha utilizado las APIs proporcionadas, lo que significa que las limitaciones de búsqueda están dictadas por las fuentes de los datos, y no por la interfaz de PoP en sí misma. Es importante recordar que PoP es simplemente una interfaz para acceder a estas bases de datos y no genera los datos ni limita las búsquedas por sí misma [19].

En cuanto a la seguridad, la autora asegura que la aplicación es segura y que no recopila ni almacena datos personales de los usuarios [20]. Además, la aplicación se actualiza periódicamente para corregir posibles errores y vulnerabilidades de seguridad [19].

6.2. Academic Accelerator

Academic Accelerator [13] es una plataforma en línea que ofrece una variedad de herramientas y recursos para investigadores académicos. Esta plataforma se ha convertido en una herramienta valiosa para la comunidad científica, ya que proporciona una amplia gama de servicios y funcionalidades para mejorar la experiencia de investigación.

Una de las características principales de Academic Accelerator es su capacidad para buscar y acceder a artículos científicos. La plataforma recopila información de diversas fuentes, incluyendo Google Scholar y Microsoft Academic, lo que permite a los investigadores encontrar fácilmente estudios relevantes para sus áreas de interés. Además, Academic Accelerator ofrece análisis de citas, lo que permite a los investigadores evaluar el impacto de sus trabajos y comparar su visibilidad con la de otros investigadores.

Otra funcionalidad destacada de Academic Accelerator es la generación de perfiles de investigadores. La plataforma permite a los investigadores crear perfiles personalizados que incluyen información sobre sus publicaciones, citas y colaboraciones. Estos perfiles ayudan a los investigadores a destacar su trabajo y establecer conexiones con otros profesionales de su campo.

Además, Academic Accelerator proporciona métricas de impacto y estadísticas detalladas sobre revistas académicas, lo que permite a los investigadores evaluar la relevancia y calidad de diferentes publicaciones. Esto ayuda a los investigadores a tomar decisiones informadas sobre dónde enviar sus trabajos para su publicación.

6.3. Otros trabajos

Existen otros trabajos relacionados con esta temática que se enumeran a continuación:

1. **Scholarometer** [12]: Esta es una herramienta menos conocida pero poderosa desarrollada por la Escuela de Informática y Computación de la Universidad de Indiana-Bloomington, lanzada en 2009 [22]. Es una herramienta social que pretende no solo facilitar el análisis de citas, sino también facilitar el etiquetado social de recursos académicos [23].

   En su interfaz principal, muestra una red en la que cada nodo representa una discplina. Si alguno de los nodos es seleccionado, se despliega un panel con el ranking de autores según el h-index.

Figura 6.2: Interfaz principal de Schoolarmeter

Scholarometer se instala como extensión en el navegador web y su principal función es extraer datos de autores de Google Scholar Citations (GSC). Los usuarios pueden realizar búsquedas de autores a través de una barra de búsqueda o ingresando el ID del académico. Si se utiliza esta última opción, el sistema extraerá y mostrará el perfil GSC del autor, con funcionalidades y métricas adicionales como la clasificación de artículos. [23].

2. **An index to quantify an individual's scientific research output** [21]: Este estudio realizado por Jorge E. Hirsch en el año 2005, presenta el h-index como una nueva métrica para evaluar el impacto de la investigación científica. El estudio argumenta que el h-index es más preciso que el Factor de Impacto y está menos sujeto a manipulación.

3. **Modeling and Prediction of the Impact Factor of Journals Using Open-Access Databases** [28]: Este estudio de 2020 se enfoca al uso de bases de datos de acceso abierto para modelar y predecir el factor de impacto de las revistas científicas. El estudio sugiere que es posible utilizar bases de datos como Google Scholar, ResearchGate y Scopus para estimar el factor de impacto de revistas nuevas, pequeñas o independientes que no están incluidas en el Science Citation Index

(SCI) y que no reciben un factor de impacto de la base de datos Web of Science (WoS). El estudio también señala que los resultados obtenidos con el modelo desarrollado sugieren que es posible predecir el factor de impacto WoS utilizando bases de datos de acceso abierto alternativas.

Conclusiones y Líneas de trabajo futuras

7.1. Conclusiones

A lo largo de este proyecto, se ha llevado a cabo el desarrollo de una aplicación web innovadora que muestra predicciones del Índice de Impacto (JCR) de las revistas científicas. Para lograr este objetivo, se ha enfrentado el desafío de extraer datos relevantes a través de técnicas de *web scraping*, que se ha revelado como una de las partes más arduas y tediosas del proyecto. La exploración de diversas técnicas de *web scraping*, la investigación de diferentes páginas y APIs, y el procesamiento de grandes volúmenes de datos han sido actividades fundamentales para el éxito de este proyecto.

El principal logro obtenido ha sido alcanzar una estimación del JCR bastante buena.

Por otro lado, se ha creado una aplicación web de acceso abierto, cuya funcionalidad se basa en algoritmos de aprendizaje supervisado. Estos algoritmos utilizan los datos históricos extraídos para predecir el valor del Índice de Impacto de todas las revistas científicas indexadas en el JCR. Esta aplicación representa una contribución significativa para la comunidad científica, ya que proporciona una herramienta accesible para estimar la relevancia y repercusión de las revistas académicas.

Durante el desarrollo de este proyecto, se ha adquirido un profundo conocimiento sobre las complejidades y desafíos asociados con el *web scraping*. La tarea de extraer datos de manera automatizada desde diferentes fuentes web requiere un enfoque meticuloso, teniendo en cuenta las variaciones en la estructura y el formato de las páginas. Además, se han explorado diversas

técnicas y bibliotecas, así como APIs específicas, para obtener los datos necesarios. Esta experiencia ha demostrado la importancia de contar con habilidades sólidas en el ámbito del *web scraping* y la capacidad de adaptarse a diferentes entornos y requisitos.

Otro aspecto destacable del proyecto ha sido la investigación de modelos de inteligencia artificial para predecir el Índice de Impacto. Mediante un análisis exhaustivo de distintos enfoques de aprendizaje supervisado, se ha buscado identificar los modelos más eficientes y precisos para esta tarea específica. Además, se ha obtenido información valiosa sobre las características y variables que influyen en el Índice de Impacto, lo cual puede ser de utilidad para futuras investigaciones en el campo de la bibliometría.

La conclusión más relevante de este proyecto es que la combinación de *web scraping* y modelos de inteligencia artificial puede ofrecer soluciones prometedoras para el análisis y la predicción de indicadores científicos. La aplicación desarrollada no solo ha demostrado la viabilidad de estas técnicas, sino que también ha puesto de manifiesto su potencial para facilitar la toma de decisiones informadas en el ámbito académico.

En resumen, este proyecto ha representado un desafío integral que ha involucrado desde la extracción de datos mediante *web scraping* hasta la implementación de algoritmos de IA y la creación de una aplicación web funcional y de código abierto. A través de esta experiencia, se han aprendido valiosas lecciones sobre la importancia de la extracción y minería de los datos como parte fundamental de los proyectos, así como sobre el potencial de los modelos de inteligencia artificial para predecir indicadores científicos. Se espera que este trabajo pueda facilitar futuras investigaciones en el ámbito de la bibliometría.

7.2. Líneas de trabajo futuras

En esta sección se proponen posibles direcciones para continuar avanzando en la comprensión del tema en cuestión y en su aplicación práctica.

Extracción de los datos

En un futuro, si se quiere aumentar el nivel de precisión en el cálculo del JCR, se recomienda mejorar la calidad de los datos extraídos. Para ello, se podría considerar el pago de una licencia para cualquiera de las APIs de pago propuestas a lo largo de la memoria (i.e., GS, WoS, Scopus, etc.). Siguiendo la línea de trabajos como Publish or Perish (ver sección 6.1),

se deberá advertir al usuario de las limitaciones en cuanto a número de *requests* que se pueden realizar y el tiempo de espera que supondría obtener el número de resultados deseado. Además, sería necesario mantener los datos extraídos en privado de forma que no se violen las políticas de privacidad y términos de uso de las distintas APIs.

Aplicación web

Por un lado, si en el futuro se desea utilizar la aplicación para calcular el Índice de Impacto en un campo distinto al de *computer science*, se puede desarrollar el área del usuario administrador de forma que solo se requiera subministrar un CSV con la lista de revistas del nuevo campo deseado. La red se volverá a entrenar y se reiniciará todo el proceso, permitiendo así la adaptación de la aplicación a diferentes áreas de investigación. Con esta flexibilidad, la aplicación puede ser empleada para el cálculo del índice de impacto en una amplia variedad de campos, ampliando así su alcance y aplicabilidad.

Otra funcionalidad interesante podría ser la ampliación de la sección de los usuarios de forma que se pueda visualizar un historial de búsquedas recientes o que se pueda acceder de forma más rápida a la información de las revistas que dicho usuario suele consultar habitualmente. Finalmente, se podría habilitar una opción para la descarga de los resultados obtenidos en distintos formatos (CSV, HTML, etc.).

Bibliografía

[1] Babel documentation. https://babel.pocoo.org/en/latest/. [Internet; acceso 07-junio-2023].

[2] Beautiful soup documentation - beautiful soup 4.4.0 documentation. https://beautiful-soup-4.readthedocs.io/en/latest/. [Internet; acceso 07-junio-2023].

[3] Full stack python: Jinja2. https://www.fullstackpython.com/jinja2.html. [Internet; acceso 07-junio-2023].

[4] habanero 1.2.3 documentation - habanero 1.2.3 documentation. https://habanero.readthedocs.io/en/latest/. [Internet; acceso 07-junio-2023].

[5] Psycopg2 documentation. https://www.psycopg.org/. [Internet; acceso 07-junio-2023].

[6] Pypi, scholarly. https://pypi.org/project/scholarly/. [Internet; acceso 07-junio-2023].

[7] Python documentation. https://docs.python.org/3/library/unittest.html. [Internet; acceso 07-junio-2023].

[8] Scikit-learn documentation. https://scikit-learn.org/stable/index.html. [Internet; acceso 07-junio-2023].

[9] Selenium documentation - selenium 4.9 documentation. https://www.selenium.dev/selenium/docs/api/py/api.html. [Internet; acceso 07-junio-2023].

[10] Welcome to flask - flask documentation (2.3.x). https://flask.palletsprojects.com/en/2.3.x/. [Internet; acceso 07-junio-2023].

[11] Zhenhub. https://www.zenhub.com/. [Internet; acceso 03-junio-2023].

[12] Crowdsource field annotations and measure universal citation impact, 2018. https://scholarometer.indiana.edu/.

[13] Latest journal's impact if - trend · prediction - academic accelerator, 2023. https://academic-accelerator.com/Impact-of-Journal/System. [Internet; acceso 03-junio-2023].

[14] Postgres documentation: Syntex identifiers, May 2023. https://www.postgresql.org/docs/current/sql-syntax-lexical.html#SQL-SYNTAX-IDENTIFIERS. [Internet; acceso 07-junio-2023].

[15] Svetla Baykoucheva. *Driving science information discovery in the digital age*. Chandos information professional series. Chandos Publishing, Cambridge, Massachusetts, 2022.

[16] Lennart Björneborn et al. *Small-world link structures across an academic web space: a library and information science approach*. Citeseer, 2004.

[17] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1):3133–3181, 2014.

[18] The PostgreSQL Global Development Group. Postgresql: The world's most advanced open source database, 2019. https://www.postgresql.org/. [Internet; acceso 05-diciembre-2022].

[19] Anne-Wil Harzing. Publish or perish, 2007. https://harzing.com/resources/publish-or-perish/. Last update on sun 6 nov 2022 14:58.

[20] Anne-Wil Harzing. *The publish or perish book*. Tarma Software Research Pty Limited Melbourne, 2010.

[21] Jorge E Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences*, 102(46):16569–16572, 2005.

[22] Jasleen Kaur, Diep Thi Hoang, Xiaoling Sun, Lino Possamai, Mohsen JafariAsbagh, Snehal Patil, and Filippo Menczer. Scholarometer: A social framework for analyzing impact across disciplines. 2012.

[23] Emilio Delgado López-Cózar, Enrique Orduña-Malea, Alberto Martín-Martín, and Juan M Ayllón. Google scholar: the big data bibliographic tool. In *Research Analytics*, pages 59–80. Auerbach Publications, 2017.

[24] Ryan Mitchell. *Web scraping with Python: Collecting more data from the modern web*. O'Reilly Media, Inc., 2018.

[25] Ashley I Naimi and Laura B Balzer. Stacked generalization: an introduction to super learning. *European journal of epidemiology*, 33:459–464, 2018.

[26] Nuria Amat Noguera. *Documentación Científica y Nuevas Tecnologías de la Información*. EDICIONES PIRÁMIDE S.A., Josefa Valcárcel, 27. 28027 Madrid, 1989.

[27] Web of Science Group. Journal citation reports, 2019. https://clarivate.com/webofsciencegroup/solutions/journal-citation-reports/. [Internet; acceso 02-enero-2023].

[28] Matthias Templ. Modeling and prediction of the impact factor of journals using open-access databases: With an application to the austrian journal of statistics. *Austrian Journal of Statistics*, 49(5):35–58, 2020.

[29] Simona Turbanti. *Bibliometria e scienze del libro : internazionalizzazione e vitalità degli studi Italiani*. Studi e saggi ; 170. Firenze University Press, Italy, 2017.

[30] Nikolay K. Vitanov. *Science Dynamics and Research Production Indicators, Indexes, Statistical Laws and Mathematical Models*. Qualitative and Quantitative Analysis of Scientific and Scholarly Communication. Springer International Publishing, Cham, 1st ed. 2016. edition, 2016.

[31] Corey Wade. Getting started with xgboost in scikit-learn, Nov 2020. https://towardsdatascience.com/getting-started-with-xgboost-in-scikit-learn-f69f5f470a97.

[32] Bo Zhao. Web scraping. *Encyclopedia of big data*, pages 1–3, 2017.