Máster de Sistemas Inteligentes
Trabajo de Fin de Máster
Junio, 2025

Incorporación de técnicas de IA para asistir al preprocesamiento de datos para usuarios no expertos

Autora: Marta de Castro Leira
Tutora: Alicia García Holgado
Cotutora: Andrea Vázquez Ingelmo

Departamento de Informática y Automática
Universidad de Salamanca

A mi familia, por su paciencia constante
y por ayudarme siempre en todo lo que necesito.
A todos los amigos que he hecho por el camino en Salamanca,
que me han acompañado y apoyado para llegar hasta aquí.
Y a esta ciudad,
que me ha dado tantas experiencias valiosas
y la oportunidad de crecer personal y profesionalmente.

CERTIFICADO DE LOS TUTORES DEL TFM

La profesora Dña. Alicia García Holgado, del Departamento de Informática y Automática de la Universidad de Salamanca, en calidad de tutora, y la profesora Dña. Andrea Vázquez Ingelmo, también del Departamento de Informática y Automática de la Universidad de Salamanca, en calidad de cotutora,

HACEN CONSTAR:

Que el trabajo titulado **Incorporación de técnicas de IA para asistir al preprocesamiento de datos para usuarios no expertos** (*Incorporating AI techniques to support data preprocessing for non-expert users*), que se presenta a continuación, ha sido realizado por Dña. Marta de Castro Leira, con DNI 32718800N y correo `mardecalei@usal.es`, y constituye la memoria del Trabajo de Fin de Máster realizado para la obtención del título de Máster Universitario en Sistemas Inteligentes en esta universidad.

Dicho trabajo se presenta para su defensa en la primera convocatoria del curso académico 2024/2025.

Salamanca, 12 de junio de 2025

Fdo.:

Firmado por GARCIA HOLGADO ALICIA - ***9368** el día 13/06/2025 con un certificado emitido por AC FNMT Usuarios	Firmado por VAZQUEZ INGELMO ANDREA - ***1883** el día 13/06/2025 con un certificado emitido por AC FNMT	Firmado por DE CASTRO LEIRA MARTA - ***1880** el día 13/06/2025 con un certificado emitido
Dña. Alicia García Holgado	Dña. Andrea Vázquez Ingelmo	Dña. Marta de Castro Leira

Resumen

El preprocesamiento de datos es una etapa crítica en el análisis y aprendizaje automático, pero suele resultar compleja y poco accesible para usuarios sin experiencia técnica. Este trabajo propone un sistema interactivo que asiste el preprocesamiento mediante técnicas de Inteligencia Artificial (IA), orientado a usuarios no expertos. Para fundamentar el desarrollo, se realizó una revisión sistemática de la literatura (SLR), que permitió identificar las técnicas más relevantes y los principales vacíos de accesibilidad y explicabilidad. Con estos hallazgos se diseñó un prototipo que combina un backend en Flask y un frontend en Angular, integrando procesos automáticos de limpieza, normalización, selección de características y evaluación con AutoML. La validación práctica con usuarios no técnicos permitió identificar mejoras y refinar la herramienta para que resulte más accesible y comprensible. Los resultados sugieren que esta solución no solo automatiza el preprocesamiento, sino que también democratiza su uso, convirtiéndolo en una herramienta educativa y práctica. Como trabajo futuro, se plantean mejoras generales que fortalezcan aún más la accesibilidad y la personalización del proceso.

Abstract

Data preprocessing is a critical stage in analysis and machine learning, but it is often complex and inaccessible to users without technical experience. This work proposes an interactive system that assists preprocessing using Artificial Intelligence (AI) techniques, aimed at non-expert users. To inform the development, a systematic literature review (SLR) was conducted, which identified the most relevant techniques and the main gaps in accessibility and explainability. Based on these findings, a prototype was designed that combines a backend in Flask and a frontend in Angular, integrating automatic cleaning, normalisation, feature selection and evaluation processes with AutoML. Practical validation with non-technical users allowed us to identify improvements and refine the tool to make it more accessible and understandable. The results suggest that this solution not only automates preprocessing but also democratises its use, making it an educational and practical tool. As future work, we propose general improvements that further strengthen the accessibility and customisation of the process.

Índice

Índice de figuras IV

Índice de tablas V

1. Introducción 1
1.1. Objetivos . . . 2
1.1.1. Objetivo general . . . 2
1.1.2. Objetivos específicos . . . 2

2. Revisión sistemática de la literatura 3
2.1. Planificación de la revisión . . . 4
2.1.1. Preguntas de investigación y de mapeo . . . 4
2.1.2. Criterios de inclusión y exclusión . . . 5
2.1.3. Marco PICOC . . . 6
2.1.4. Estrategia de búsqueda . . . 7
2.1.5. Criterios de calidad . . . 9
2.2. Proceso de revisión . . . 10
2.3. Resultados . . . 11
2.4. Conclusiones de la revisión sistemática . . . 26

3. Metodología 27
3.1. Selección y análisis de algoritmos . . . 27
3.1.1. Limpieza de datos . . . 28
3.1.2. Normalización de datos . . . 29
3.1.3. Selección de características . . . 30
3.1.4. Evaluación automática . . . 31
3.2. Prototipo inicial . . . 31
3.3. Diseño e implementación del sistema web . . . 38
3.3.1. Arquitectura general del sistema . . . 39
3.3.2. Frontend: diseño e implementación con Angular . . . 40
3.3.3. Backend: implementación con Flask . . . 45
3.3.4. Comunicación frontend-backend . . . 46
3.4. Despliegue del sistema . . . 48
3.5. Evaluación del sistema . . . 49
3.5.1. Pruebas realizadas con distintos conjuntos de datos . . . 49
3.5.2. Evaluación con usuarios . . . 52
3.6. Versión final con mejoras aplicadas . . . 55

4. Discusión 64

4.1. Inspiración, alineación y aportaciones del sistema desarrollado 65
4.2. Comportamiento del flujo de preprocesamiento 66
4.3. Validación práctica del sistema 67

5. Conclusiones y líneas futuras **68**

Referencias **71**

Índice de figuras

1. Diagrama de flujo PRISMA del proceso de selección de estudios. . . . 10
2. Evolución del número de estudios publicados entre 2020 y 2025. . . . 12
3. Distribución de artículos por dominio de aplicación. 14
4. Distribución de los métodos de evaluación aplicados en los artículos revisados. 15
5. Términos más frecuentes relacionados con el diseño de herramientas de preprocesamiento automatizado con IA. 17
6. Diagrama del flujo de procesamiento del módulo inicial. 32
7. Arquitectura general del sistema. 40
8. Pantalla inicial del sistema. 41
9. Visualización de las características del sistema. 41
10. Interfaz donde el usuario puede subir su archivo CSV para iniciar el preprocesamiento automático. 42
11. Selección de la columna objetivo. La interfaz ofrece una sugerencia automática e incluye una explicación accesible sobre su importancia. . 42
12. Visualización del progreso del preprocesamiento. Se listan los pasos completados y se indica el avance del AutoML. 43
13. Resumen de resultados tras el preprocesamiento. Se muestra la puntuación obtenida, opciones de descarga y acceso a detalles explicativos. 44
14. Pantalla de detalles del preprocesamiento. 44
15. Pantalla de evaluación automática (AutoML) mostrando el mejor modelo. 45
16. Modal de advertencia cuando la columna objetivo contiene valores nulos. 56
17. Explicación accesible durante la fase de limpieza de datos. 57
18. Descripción visual del proceso de normalización con autoencoder. . . 57
19. Cómo se seleccionan automáticamente las columnas más relevantes. . 58
20. Vista del flujo de pasos con botones de “ver más” para cada etapa. . . 58
21. Vista general del nuevo modal con iconos informativos desplegables. . 59
22. Resumen de la fase de limpieza. 60
23. Vista de la fase de normalización. 60
24. Vista detallada de la selección de características. 61
25. Explicación del tipo de problema detectado por AutoML y la métrica aplicada. 62
26. Ejemplo de resultados AutoML para un problema de regresión. 63
27. Ejemplo de resultados AutoML para un problema de clasificación binaria. 63
28. La pantalla principal ya no muestra el score final. 64

Índice de tablas

1. Preguntas de investigación. 5
2. Preguntas de mapeo. 5
3. Criterios de inclusión 6
4. Criterios de exclusión 6
5. Definición del marco PICOC 7
6. Autores más prolíficos en los estudios revisados 12
7. Resumen de respuestas a la evaluación con usuarios. 53

1. Introducción

El análisis de datos y el aprendizaje automático se han convertido en herramientas fundamentales para la toma de decisiones en diversos ámbitos. Sin embargo, el éxito de cualquier proyecto de análisis de datos depende en gran medida de la calidad y preparación de los datos utilizados. Este proceso de preparación, conocido como preprocesamiento de datos, es una etapa crítica que incluye tareas como la limpieza de datos, la normalización y la selección de características. Estas tareas buscan transformar los datos en un formato adecuado para el análisis, eliminando errores, inconsistencias y valores atípicos que puedan afectar negativamente a los resultados.

A pesar de su importancia, el preprocesamiento de datos es una tarea que puede resultar compleja, especialmente para usuarios no expertos en ciencia de datos. Estos usuarios pueden incluir profesionales de diversas disciplinas, investigadores que trabajan con datos empíricos, o analistas que necesitan preparar información para tomar decisiones basadas en datos. La dificultad radica en que el preprocesamiento de datos abarca múltiples tareas técnicas, cada una con sus propios desafíos.

Por ejemplo, la limpieza de datos implica identificar y corregir valores ausentes. La normalización y transformación de datos exigen comprender cómo ajustar los valores para que sean comparables o estén en el rango adecuado. Finalmente, la selección de características requiere distinguir qué variables son realmente relevantes para un análisis determinado, lo que puede ser confuso sin conocimientos en estadística o aprendizaje automático.

Para usuarios sin formación técnica avanzada, estos procesos suelen depender de métodos manuales, que no solo son propensos a errores, sino que también consumen tiempo y están influenciados por el conocimiento subjetivo de cada persona. En consecuencia, la calidad del preprocesamiento puede variar considerablemente, afectando a los resultados de los análisis o a modelos predictivos posteriores.

La automatización del preprocesamiento de datos mediante IA ha emergido como una solución prometedora para estos desafíos. Estas técnicas reducen la intervención humana y minimizan la posibilidad de errores. Sin embargo, muchas de las herramientas actuales que utilizan IA para el preprocesamiento están diseñadas para usuarios técnicos, lo que limita su accesibilidad para usuarios no expertos.

En este contexto, el presente trabajo se centra en el desarrollo de una herramienta interactiva basada en técnicas de IA que asista al preprocesamiento de datos para usuarios sin conocimientos avanzados. La herramienta buscará simplificar las tareas clave del preprocesamiento, tales como la limpieza de datos, la normalización y la selección de características, proporcionando una interfaz amigable e intuitiva que guíe al usuario en cada paso del proceso. Esta solución no solo pretende facilitar el acceso a técnicas avanzadas de preprocesamiento, sino también mejorar la calidad de los conjuntos de datos y, en consecuencia, la precisión de los modelos predictivos que se construyan a partir de ellos.

Para fundamentar el desarrollo de esta herramienta, se realiza una revisión sis-

temática de la literatura que permita identificar las técnicas de IA más efectivas y accesibles para el preprocesamiento de datos. A partir de este análisis, se diseña un sistema que integre dichas técnicas de manera interactiva, permitiendo a los usuarios ejecutar procesos complejos de manera sencilla. Finalmente, se lleva a cabo una evaluación práctica del sistema mediante pruebas con usuarios no expertos, comparando su rendimiento con métodos tradicionales de preprocesamiento de datos.

1.1. Objetivos

Este trabajo se enmarca dentro del objetivo general de simplificar el preprocesamiento de datos mediante técnicas de inteligencia artificial orientado a usuarios no expertos. A partir de este planteamiento, se define un objetivo general y varios objetivos específicos que permiten estructurar y guiar el desarrollo del sistema.

1.1.1. Objetivo general

Desarrollar un sistema interactivo basado en técnicas de IA que asista al preprocesamiento de datos de manera accesible para usuarios no expertos, mejorando la eficiencia y la calidad del proceso en comparación con métodos tradicionales.

1.1.2. Objetivos específicos

Analizar las principales técnicas de IA aplicadas al preprocesamiento de datos, llevando a cabo una revisión sistemática de la literatura que permita identificar las técnicas más relevantes y utilizadas en tareas como: limpieza, normalización y selección de características. El objetivo es clasificar estas técnicas y extraer buenas prácticas que sirvan de base para diseñar soluciones accesibles a usuarios sin conocimientos técnicos.

Diseñar e implementar una herramienta interactiva de preprocesamiento asistido por IA, desarrollando una solución software que integre dichas técnicas para automatizar el flujo completo. La herramienta combinará un backend encargado de las tareas técnicas, con una interfaz gráfica que guíe al usuario paso a paso, proporcione explicaciones accesibles y permita procesar conjuntos de datos sin necesidad de experiencia técnica.

Evaluar la usabilidad y efectividad del sistema mediante pruebas con usuarios no expertos, realizando una evaluación práctica que permita comprobar si el sistema resulta comprensible, útil y accesible. Esta evaluación analizará aspectos como la facilidad de uso, la claridad de la interfaz, la utilidad de las explicaciones ofrecidas y la capacidad del sistema para generar datos correctamente preprocesados.

Documentar el proceso de desarrollo y los resultados de la evaluación, elaborando una documentación clara y estructurada que detalle las decisiones de diseño, las técnicas de IA utilizadas, el funcionamiento del sistema y los resultados obtenidos durante su validación.

Finalmente, el resto de este informe se estructura de la siguiente manera: la Sección 2 describe la revisión sistemática de la literatura que fundamenta el desarrollo de la herramienta, detallando las técnicas de IA más relevantes y accesibles y los vacíos detectados; la Sección 3 presenta la metodología seguida, incluyendo la selección y análisis de algoritmos, el diseño e implementación de la solución, y la validación práctica con usuarios reales; la Sección 4 discute los resultados obtenidos, explorando tanto el rendimiento técnico del flujo automático como la experiencia y percepción de los usuarios reales que participaron en las pruebas. Por último, la Sección 5 resume las principales conclusiones y propone líneas futuras de mejora, para seguir avanzando en la creación de herramientas más accesibles, comprensibles y útiles para perfiles no técnicos.

2. Revisión sistemática de la literatura

Con el objetivo de fundamentar el desarrollo del sistema propuesto, se llevó a cabo una Revisión Sistemática de la Literatura (Systematic Literature Review, SLR) centrada en técnicas de preprocesamiento de datos asistidas por IA, con especial énfasis en aquellas orientadas a usuarios sin conocimientos técnicos avanzados.

Esta revisión se desarrolló siguiendo las directrices metodológicas de Kitchenham y Charters [1], y tuvo como propósito sintetizar críticamente la evidencia científica existente sobre el uso de IA en el preprocesamiento automatizado de datos. En particular, se pretendía reconocer las técnicas más empleadas, examinar su accesibilidad y facilidad de uso, y detectar tanto buenas prácticas como limitaciones que influyen en su adopción por parte de personas sin experiencia técnica.

De forma más específica, esta revisión se propone:

- Identificar y caracterizar las técnicas de IA empleadas para automatizar tareas clave del preprocesamiento, como la limpieza de datos, la normalización y la selección de características.
- Analizar los enfoques de diseño adoptados por las herramientas existentes, prestando especial atención a su accesibilidad, explicabilidad y adecuación para usuarios sin formación técnica.
- Evaluar los métodos utilizados para medir la efectividad y la usabilidad de estas herramientas, en comparación con enfoques tradicionales.
- Detectar vacíos de investigación y oportunidades de mejora que puedan orientar el desarrollo de soluciones más inclusivas y efectivas.

Para alcanzar estos objetivos, se definieron preguntas de investigación y de mapeo, y se aplicó un protocolo sistemático de búsqueda, selección y análisis de estudios publicados entre 2020 y 2025. A continuación, se detallan los criterios metodológicos aplicados y los principales resultados obtenidos.

2.1. Planificación de la revisión

La planificación de esta SLR se llevó a cabo siguiendo un enfoque riguroso y estructurado, con el objetivo de garantizar la transparencia, reproducibilidad y validez de los resultados obtenidos. Para ello, se adoptaron las directrices PRISMA 2020 [2], ampliamente reconocidas en el ámbito de las revisiones sistemáticas, así como las recomendaciones metodológicas propuestas por Kitchenham y Charters [1] para revisiones en ingeniería del software.

Esta planificación incluyó la formulación de preguntas de investigación y mapeo específicas, la definición de criterios de inclusión y exclusión, la construcción de una estrategia de búsqueda efectiva y la selección de fuentes de información pertinentes. La búsqueda se limitó a publicaciones entre 2020 y 2025, permitiendo así conocer las contribuciones más recientes y relevantes en un campo en rápido desarrollo.

El proceso de selección de artículos se dividió en varias fases: eliminación de duplicados, revisión de títulos y resúmenes, y análisis a texto completo. Posteriormente, se aplicaron criterios de calidad para asegurar la relevancia metodológica de los estudios incluidos. A partir de los estudios seleccionados, se extrajeron datos clave que fueron organizados y analizados para responder a las preguntas de investigación y de mapeo, y ofrecer una visión clara y actualizada sobre cómo se está utilizando la IA en el preprocesamiento de datos dirigido a usuarios sin experiencia técnica.

A continuación, se detallan los distintos componentes metodológicos que conformaron esta planificación.

2.1.1. Preguntas de investigación y de mapeo

Para guiar de forma sistemática la recolección, análisis e interpretación de los datos, se formularon dos conjuntos de preguntas complementarias: por un lado, las de investigación (Research Questions, RQ), centradas en el análisis cualitativo y crítico de las contribuciones existentes; y por otro, las de mapeo (Mapping Questions, MQ), orientadas a proporcionar una visión panorámica y estructurada del campo.

Preguntas de investigación

Estas permiten explorar en profundidad las técnicas utilizadas, su aplicación a tareas específicas del preprocesamiento y su adecuación para usuarios no expertos. Las preguntas formuladas son las de la Tabla 1.

Código	Pregunta
RQ1	¿Qué técnicas de inteligencia artificial se han utilizado para automatizar el preprocesamiento de datos?
RQ2	¿Cómo estas técnicas han abordado tareas críticas como limpieza, normalización y selección de características?
RQ3	¿Qué características hacen que estas herramientas sean accesibles y efectivas para usuarios no expertos?
RQ4	¿Cuáles son las ventajas y limitaciones de las soluciones basadas en IA frente a métodos tradicionales de preprocesamiento?
RQ5	¿Qué áreas de investigación relacionadas con el preprocesamiento de datos y usuarios no expertos aún no han sido exploradas?

Tabla 1: Preguntas de investigación.

Preguntas de mapeo

Estas complementan a las RQ proporcionando una clasificación estructurada de los estudios revisados, lo que permite detectar patrones, tendencias, dominios de aplicación y vacíos existentes en la literatura. Las preguntas de mapeo utilizadas son las de la Tabla 2.

Código	Pregunta
MQ1	¿Cuántos estudios se han publicado en los últimos cinco años sobre técnicas de IA aplicadas al preprocesamiento de datos?
MQ2	¿Cuáles son los autores más activos y reconocidos en este campo de investigación?
MQ3	¿Cuáles son las revistas y conferencias más relevantes para la publicación de estos estudios?
MQ4	¿En qué dominios o aplicaciones específicas (educación, industria, salud, etc.) se han utilizado herramientas de preprocesamiento con IA?
MQ5	¿Qué métodos de evaluación se han utilizado para medir la accesibilidad y efectividad de las herramientas en usuarios no expertos?
MQ6	¿Qué definiciones y enfoques se han propuesto para diseñar herramientas de preprocesamiento accesibles?

Tabla 2: Preguntas de mapeo.

2.1.2. Criterios de inclusión y exclusión

Con el objetivo de garantizar la relevancia y calidad de los estudios incluidos en la revisión, se definieron criterios específicos de inclusión (Tabla 3) y exclusión (Tabla 4). Estos criterios permitieron filtrar de manera sistemática los trabajos recuperados durante la búsqueda bibliográfica y descartar aquellos que no se ajustaban al propósito del estudio. En concreto, se aplicaron durante la fase de selección para

asegurar que solo se analizaran publicaciones pertinentes, recientes y alineadas con los objetivos definidos.

Código	Criterio
IC1	Publicaciones que describen técnicas de IA aplicadas específicamente al preprocesamiento de datos.
IC2	Estudios que incluyen un enfoque explícito hacia la accesibilidad para usuarios no expertos.
IC3	Publicaciones en inglés o español.
IC4	Publicaciones en revistas, conferencias, libros o capítulos revisados por pares.
IC5	Publicaciones recientes y completas (2020–2025).
IC6	Acceso completo disponible mediante suscripción institucional o acceso abierto.

Tabla 3: Criterios de inclusión

Código	Criterio
EC1	Estudios que no abordan técnicas de IA aplicadas al preprocesamiento de datos.
EC2	Ausencia de enfoque hacia usuarios no expertos.
EC3	Idiomas distintos al inglés o español.
EC4	Publicaciones no revisadas por pares.
EC5	Versiones preliminares o duplicadas de otras ya incluidas.
EC6	Falta de acceso al texto completo.

Tabla 4: Criterios de exclusión

Estos criterios garantizaron que solo se analizaran estudios de alta calidad, directamente relacionados con el preprocesamiento automatizado de datos mediante IA, y con un enfoque claro hacia la accesibilidad para usuarios no expertos.

2.1.3. Marco PICOC

Para delimitar el enfoque de esta revisión y asegurar que los estudios seleccionados fueran coherentes con los objetivos del análisis, se adoptó el modelo PICOC (Población, Intervención, Comparación, Resultados y Contexto) en la Tabla 5. Este marco, ampliamente utilizado en revisiones sistemáticas y estudios de mapeo, permite definir claramente los elementos clave que orientan la búsqueda y selección de literatura relevante.

Elemento	Descripción
Población (P)	Usuarios no expertos, es decir, personas sin formación técnica avanzada en programación o ciencia de datos, que requieren realizar tareas de preprocesamiento de datos para distintos fines analíticos o predictivos.
Intervención (I)	Técnicas de IA aplicadas al preprocesamiento de datos, especialmente en tareas como limpieza, normalización y selección de características.
Comparación (C)	No se ha planteado ninguna comparación directa entre técnicas o enfoques.
Resultados (O)	Identificación de modelos, métodos y herramientas accesibles que empleen IA para facilitar el preprocesamiento de datos, con un diseño enfocado en usuarios no expertos.
Contexto (C)	Entornos donde los usuarios no disponen de formación técnica avanzada, como instituciones educativas, pymes o sectores con recursos limitados y sin acceso a expertos en ciencia de datos.

Tabla 5: Definición del marco PICOC

2.1.4. Estrategia de búsqueda

Para identificar la literatura relevante en el ámbito del preprocesamiento automatizado de datos mediante IA, se diseñó una estrategia de búsqueda basada en tres dimensiones clave: el preprocesamiento de datos, el uso de técnicas de IA, y la accesibilidad para usuarios no expertos. Esta combinación permitió enfocar la búsqueda en estudios que abordaran no solo los aspectos técnicos, sino también el diseño orientado a la usabilidad y accesibilidad.

Se utilizaron las bases de datos Scopus y Web of Science (WoS), elegidas por su amplia cobertura en el ámbito de la ingeniería, la IA y la informática, así como por la posibilidad de aplicar filtros como: campos específicos, año de publicación, idioma y tipo de documento. Ambas plataformas son ampliamente reconocidas por su calidad y por ser fuentes primarias habituales en revisiones sistemáticas.

La estrategia de búsqueda combinó términos en inglés y español, organizados en tres grandes bloques temáticos. El primero se centró en aspectos relacionados con el preprocesamiento de datos, incluyendo términos como *data preprocessing*, *data cleaning*, *feature selection*, *feature extraction*, *dimensionality reduction*, *outlier detection* y *exploratory data analysis*. El segundo bloque se enfocó en las técnicas de IA, incorporando conceptos como *artificial intelligence*, *AI*, *machine learning* y *deep learning*. Finalmente, el tercer bloque se dedicó a la accesibilidad y a los usuarios no expertos, abarcando términos como *user-friendly tools*, *interactive tools*, *non-expert*, *no expert*, *accessible data analysis* y *user interface*.

A partir de estos bloques, se diseñó una cadena de búsqueda genérica que integra los conceptos clave, combinando operadores booleanos (AND, OR) para englobar

sinónimos y conceptos relacionados. Este enfoque garantizó una cobertura amplia y exhaustiva para la identificación de estudios relevantes sobre el preprocesamiento de datos asistido por IA y orientado a usuarios sin conocimientos técnicos avanzados.

```
("data preprocessing" OR "data cleaning" OR "exploratory data analysis"
OR "EDA" OR "feature selection" OR "feature extraction" OR
"dimensionality reduction" OR "outlier detection")
AND ("artificial intelligence" OR "AI" OR "machine-learning"
OR "machine learning" OR "deep learning")
AND ("user interface" OR "user-friendly tools" OR "interactive tools"
OR "accessible data analysis" OR "non-expert" OR "no expert")
```

A partir de esta cadena genérica, se diseñaron consultas específicas adaptadas a las particularidades de cada base de datos:

- **Consulta en Scopus:**

  ```
  TITLE-ABS-KEY (
  ("data preprocessing" OR "data cleaning" OR "exploratory data
  analysis" OR "EDA" OR "feature selection" OR "feature extraction"
  OR "dimensionality reduction" OR "outlier detection")
  AND ("artificial intelligence" OR "AI" OR "machine-learning"
  OR "machine learning" OR "deep learning")
  AND ("user interface" OR "user-friendly tools" OR "interactive
  tools" OR "accessible data analysis" OR "non-expert" OR
  "no expert")
  )
  AND PUBYEAR > 2019 AND PUBYEAR < 2026
  AND (LIMIT-TO (LANGUAGE,"English") OR LIMIT-TO (LANGUAGE,"Spanish"))
  AND (LIMIT-TO (DOCTYPE , "ar") OR LIMIT-TO (DOCTYPE , "ch")
  OR LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "re")
  OR LIMIT-TO (DOCTYPE , "bk"))
  ```

- **Consulta en WoS:**

  ```
  TS=(
  ("data preprocessing" OR "data cleaning" OR "exploratory data
  analysis" OR "EDA" OR "feature selection" OR "feature extraction"
  OR "dimensionality reduction" OR "outlier detection")
  AND ("artificial intelligence" OR "AI" OR "machine-learning"
  OR "machine learning" OR "deep learning")
  AND ("user interface" OR "user-friendly tools" OR "interactive
  tools" OR "accessible data analysis" OR "non-expert" OR
  "no expert")
  )
  ```

Estas consultas devolvieron un total de 382 resultados en Scopus y 284 en Web of Science, antes de la eliminación de duplicados. La amplitud y especificidad de los términos empleados aseguran una cobertura adecuada del campo, manteniendo la relevancia temática para los objetivos de esta revisión.

2.1.5. Criterios de calidad

Para garantizar la solidez metodológica y la relevancia de los estudios incluidos en esta revisión sistemática, se definieron y aplicaron criterios de calidad específicos. Estos criterios permitieron evaluar hasta qué punto cada artículo cumplía con los aspectos clave establecidos para el análisis, asegurando que solo se consideraran estudios con una base metodológica sólida y directamente vinculados con los objetivos del estudio.

Cada publicación fue evaluada en función de los siguientes siete criterios de calidad:

- El estudio documenta claramente su metodología.
- El estudio explica las técnicas de IA utilizadas para automatizar el preprocesamiento de datos.
- El estudio se enfoca en tareas críticas de preprocesamiento, como limpieza de datos, normalización o selección de características.
- El estudio describe herramientas o técnicas específicas que facilitan el preprocesamiento.
- El estudio se orienta explícitamente a usuarios no expertos.
- El estudio aborda cuestiones de accesibilidad y usabilidad.
- El estudio incluye una evaluación empírica de las herramientas propuestas.

Cada criterio fue valorado mediante una escala trivalente: YES (cumple completamente), PARTIAL (cumple parcialmente) o NO (no cumple). Para su cuantificación, se asignaron los siguientes valores: YES = 1 punto, PARTIAL = 0.5 puntos y NO = 0 puntos. De este modo, cada estudio podía alcanzar una puntuación máxima de 7 puntos.

Solo se incluyeron en la revisión aquellos estudios que obtuvieron una puntuación igual o superior a 4.5 puntos. Este valor fue elegido para garantizar que los estudios considerados cumplieran la mayoría de los criterios, asegurando la calidad y la relevancia metodológica mínima para los objetivos de esta revisión. De este modo, se excluyeron trabajos que no aportaran información significativa o que presentaran deficiencias metodológicas relevantes, mejorando así la fiabilidad y validez de los resultados obtenidos en el análisis final.

2.2. Proceso de revisión

El proceso de revisión sistemática que sustenta este estudio se llevó a cabo siguiendo las directrices PRISMA 2020 [2], con el objetivo de garantizar una metodología rigurosa, transparente y reproducible. Esta estrategia permitió organizar y documentar cada etapa de forma clara, desde la búsqueda inicial de estudios hasta la selección final de los trabajos incluidos.

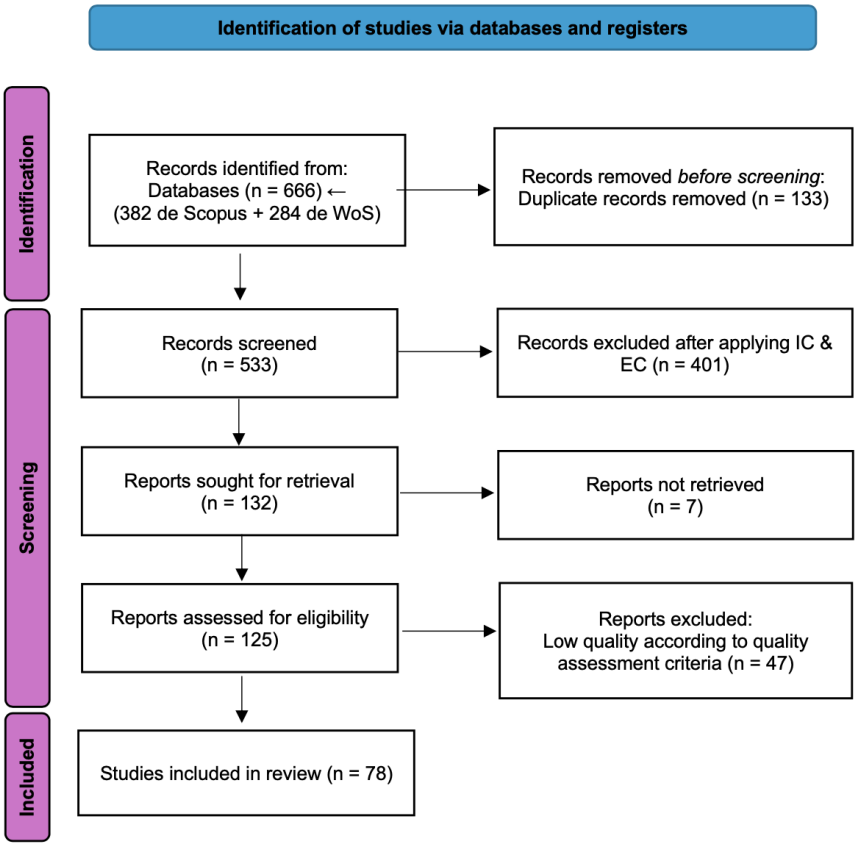


Figura 1: Diagrama de flujo PRISMA del proceso de selección de estudios.

La búsqueda bibliográfica se realizó entre el 17 y el 20 de febrero de 2025, utilizando dos bases de datos científicas ampliamente reconocidas, Scopus y Web of Science (WoS). Tras combinar los resultados de ambas bases de datos, se obtuvo un total de 666 registros. A continuación, se eliminaron 133 duplicados mediante comparación por título, DOI y autoría, lo que redujo el total a 533 registros. Estos fueron sometidos a una primera fase de cribado basada en el análisis de títulos, resúmenes y palabras clave. Durante esta etapa se aplicaron los criterios de inclusión y exclusión definidos previamente, lo que llevó al descarte de 401 artículos por no cumplir los requisitos establecidos (por ejemplo, por no centrarse en el preprocesamiento de datos o carecer de enfoque hacia usuarios no expertos).

En la segunda fase, se evaluaron en texto completo los 132 artículos restantes. Siete de ellos no pudieron ser recuperados debido a la falta de acceso al texto comple-

to. Los 125 restantes fueron leídos y analizados íntegramente, aplicando los criterios de calidad metodológica definidos en la planificación. Como resultado, 47 artículos fueron excluidos por obtener una puntuación inferior a 4.5 sobre 7. Finalmente, se seleccionaron 78 estudios que cumplían con todos los criterios de calidad y pertinencia. De cada uno de estos 78 estudios se extrajo información clave para responder tanto a las preguntas de investigación como a las de mapeo.

Todo el proceso ha sido representado gráficamente mediante el diagrama de flujo PRISMA que acompaña esta revisión, Figura 1. [1]

Este proceso garantiza que los estudios incluidos están alineados con los objetivos de la revisión y cumplen los criterios de calidad establecidos. Los datos detallados de la revisión, incluyendo la lista completa de los 78 estudios analizados (con título, autoría, año de publicación y fuente), están disponibles en un archivo Excel alojado en Zenodo, accesible en `https://zenodo.org/records/15631262`. Este archivo permite consultar el proceso completo de cribado y selección, desde los 666 registros iniciales hasta los 78 finales, así como examinar la evolución y filtrado en cada etapa mediante las pestañas correspondientes.

2.3. Resultados

En esta sección se presentan los resultados obtenidos a partir del mapeo sistemático y la revisión crítica de los estudios seleccionados. La información se organiza en torno a las preguntas de mapeo (MQ1–MQ6), que ofrecen una visión panorámica del campo, y las preguntas de investigación (RQ1–RQ5), que permiten un análisis más profundo de las técnicas y características de las herramientas revisadas. Este enfoque combinado proporciona una comprensión amplia y detallada del estado actual del preprocesamiento automatizado de datos mediante IA para usuarios no expertos.

MQ1. ¿Cuántos estudios se han publicado en los últimos cinco años sobre técnicas de IA aplicadas al preprocesamiento de datos?

La evolución anual del número de publicaciones muestra un crecimiento constante del interés en el desarrollo de herramientas de preprocesamiento automático con IA. En total, se incluyeron 78 estudios relevantes, distribuidos entre los años 2020 y 2025. El año 2024 alcanzó el pico de publicaciones. En 2025, únicamente se registraron dos artículos debido a que la búsqueda bibliográfica se cerró en febrero de ese año.

Esta tendencia se resume en la Figura 2, que ilustra visualmente la evolución del número de estudios publicados por año, y confirma que el preprocesamiento de datos con IA ha ganado relevancia en la comunidad investigadora, especialmente en contextos donde se requiere eliminar barreras técnicas para facilitar su uso por parte de personas sin formación especializada.

[1] Fuente: PRISMA 2020 flow diagram for new systematic reviews which included searches of databases and registers only. Disponible en: https://www.prisma-statement.org/PRISMAStatement/FlowDiagram

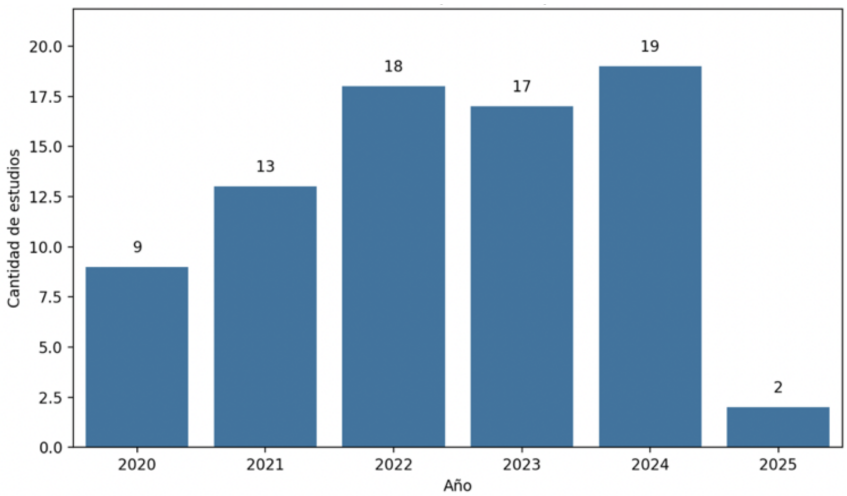


Figura 2: Evolución del número de estudios publicados entre 2020 y 2025.

MQ2. ¿Cuáles son los autores más activos y reconocidos en este campo?

Se realizó un análisis de autoría sobre los 78 estudios incluidos en esta revisión, con el propósito de identificar a los investigadores más activos en el desarrollo de técnicas de IA aplicadas al preprocesamiento de datos.

El análisis reveló una distribución amplia y dispersa de la autoría, una característica común en campos emergentes y en rápida expansión. La mayoría de los autores contribuyeron únicamente con un estudio, lo que indica un interés creciente y una exploración continua del tema por parte de diversos investigadores y equipos multidisciplinares.

Sin embargo, se identificaron 11 autores que participaron en al menos dos publicaciones diferentes, destacándose como los más prolíficos en este ámbito. Además, se observó que varios de ellos han colaborado entre sí en múltiples artículos, lo que sugiere la existencia de pequeños grupos de investigación o redes colaborativas centradas específicamente en el preprocesamiento automatizado con IA.

La Tabla 6 presenta los autores más destacados, junto con el número de artículos en los que participaron y las referencias correspondientes.

N.º de artículos	**Autores**
2	Wang D. [3, 4], Zhang J. [5, 6], Wang Y. [3, 7], McCombe N. [8, 9], Ding X. [8, 9], Prasad G. [8, 9], Finn D.P. [8, 9], Todd S. [8, 9], McClean P.L. [8, 9], Wong-Lin K. [8, 9]

Tabla 6: Autores más prolíficos en los estudios revisados

Estos nombres se consolidan como puntos de referencia dentro del conjunto de publicaciones recientes sobre IA aplicada al preprocesamiento de datos.

MQ3. ¿Cuáles son las revistas y conferencias más relevantes para la publicación de estos estudios?

Para comprender mejor las fuentes en las que se publican investigaciones sobre preprocesamiento de datos asistido por IA, se realizó un análisis detallado de las publicaciones incluidas en este mapeo.

El análisis reveló una alta dispersión en las fuentes, algo habitual en campos emergentes y multidisciplinarios. La mayoría de las revistas y conferencias solo aparecen una vez en el conjunto de estudios analizados, lo que sugiere que no existe una fuente dominante en esta área. Sin embargo, se identificaron tres publicaciones que cuentan con al menos dos artículos incluidos en la revisión:

- CEUR Workshop Proceedings: incluye investigaciones sobre clasificación automática de productos y sistemas de apoyo a la decisión en salud mental [10, 11].
- Diagnostics: publica estudios centrados en biomarcadores moleculares y en la detección facial de anomalías mediante deep learning [12, 13].
- Scientific Reports (Nature Group): recoge investigaciones sobre diagnóstico de Alzheimer con CatBoost y anotación colaborativa de vídeos para el autismo [14, 15].

Esto refleja el carácter transversal del tema y su publicación en múltiples foros académicos, lo cual es típico de áreas en rápida evolución donde aún no se han consolidado fuentes de referencia predominantes.

MQ4. ¿En qué dominios o aplicaciones específicas se han utilizado herramientas de preprocesamiento con IA?

Se han identificado varios dominios de aplicación en los que el preprocesamiento asistido por IA está teniendo un impacto significativo, tanto en contextos científicos como industriales. A continuación, se describen los principales sectores donde se ha aplicado esta tecnología, junto con los artículos relevantes en cada caso:

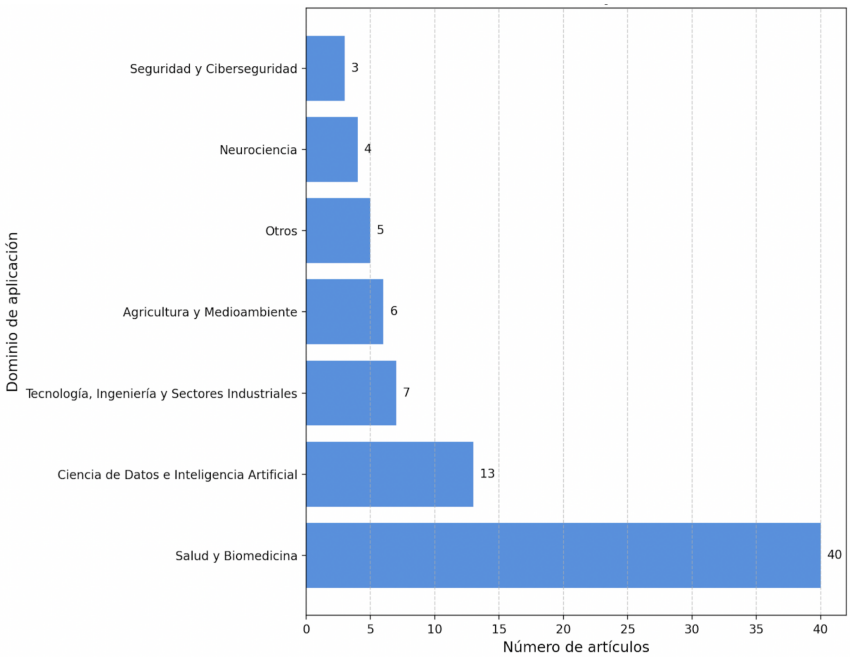


Figura 3: Distribución de artículos por dominio de aplicación.

Salud y biomedicina: este es el dominio más representado, con 40 artículos que abordan aplicaciones centradas en el diagnóstico médico, análisis de imágenes biomédicas y procesamiento automatizado de datos clínicos o genómicos. Los estudios destacan por su enfoque en la detección temprana de enfermedades, segmentación automática de imágenes médicas, y el uso de IA para apoyar la toma de decisiones clínicas [10, 3, 5, 12, 4, 11, 7, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48].

Ciencia de datos e inteligencia artificial: este grupo incluye 13 artículos que investigan técnicas de preprocesamiento aplicadas a tareas genéricas de análisis y minería de datos. Los estudios se centran en mejorar la calidad de los datos, automatizar la selección de características y optimizar modelos de aprendizaje automático [49, 50, 9, 6, 13, 51, 52, 53, 54, 55, 56, 57, 58].

Tecnología, ingeniería y sectores industriales: en este grupo se incluyen siete artículos centrados en aplicaciones industriales como la gestión de la energía, la optimización de procesos de manufactura, y la automatización en entornos de ingeniería [14, 59, 60, 61, 62, 63, 64].

Agricultura y medioambiente: seis artículos abordan el uso de IA para la clasificación de cultivos, el monitoreo ambiental y el análisis automatizado de imágenes satelitales o de sensores, con el objetivo de mejorar la eficiencia y sostenibilidad en estos sectores [65, 66, 67, 68, 69, 70].

Neurociencia: este dominio incluye estudios que aplican IA al análisis de señales cerebrales, detección de patrones neuronales y apoyo al diagnóstico de trastornos neurológicos [71, 72, 73, 74].

Seguridad y ciberseguridad: tres artículos exploran el uso de preprocesamien-

to con IA para la detección de amenazas, el análisis de comportamiento anómalo y la automatización de sistemas de protección informática [8, 15, 75].

Otros: cinco artículos no se ajustan claramente a los dominios anteriores, pero presentan aplicaciones relevantes como el análisis de datos financieros, la predicción de marketing o estudios transversales sobre IA en educación y servicios [76, 77, 78, 79, 80].

La Figura 3 resume gráficamente la distribución de los estudios por dominio principal. Aunque en el gráfico se presenta un dominio principal por artículo, cabe destacar que varios estudios abordan más de un sector. En particular, los trabajos en Salud y Biomedicina y Ciencia de Datos e Inteligencia Artificial presentan enfoques complementarios en el uso de IA para el análisis automatizado y la mejora de procesos.

MQ5. ¿Qué métodos de evaluación se han utilizado para medir la accesibilidad y efectividad de las herramientas en usuarios no expertos?

En los 78 artículos revisados, se identificaron distintos enfoques metodológicos para evaluar el grado de accesibilidad y efectividad de las herramientas de preprocesamiento cuando son utilizadas por personas sin conocimientos técnicos o en programación.

Para representar esta información, se elaboró un gráfico de barras que muestra el número de artículos que emplea cada tipo de evaluación. La Figura 4 resume visualmente estos resultados.

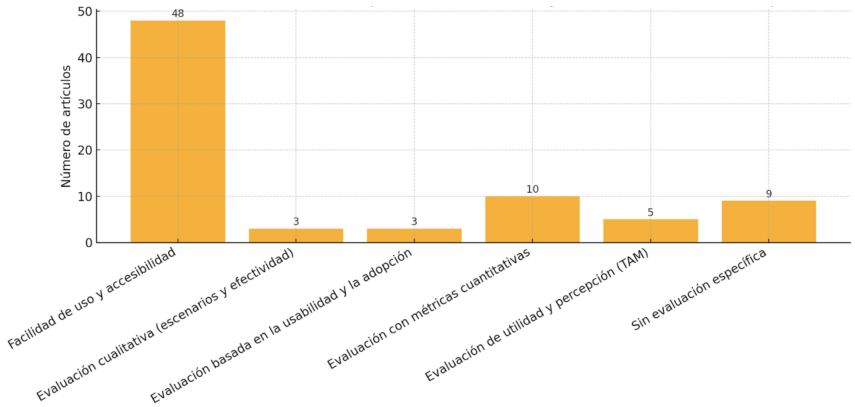


Figura 4: Distribución de los métodos de evaluación aplicados en los artículos revisados.

A continuación, se describen las seis categorías principales identificadas:

Evaluación mediante facilidad de uso y diseño accesible (48 artículos): la mayoría de los artículos afirman que las herramientas son accesibles basándose en criterios de diseño: interfaces intuitivas, menús simples, ausencia de necesidad de codificación, presencia de botones e instrucciones visuales. No se presentan datos

empíricos, pero se destaca explícitamente la intención de facilitar el uso a personas no técnicas [42, 44, 13, 71, 11, 74, 77, 17, 16, 54, 12, 47, 23, 58, 9, 76, 20, 3, 14, 21, 24, 45, 55, 30, 33, 34, 36, 61, 39, 40, 59, 66, 73, 28, 29, 67, 78, 27, 26, 25, 35, 6, 56, 63, 80, 7, 51, 46].

Evaluación cualitativa (3 artículos): este enfoque se basa en la observación directa de usuarios en entornos de prueba. Los estudios analizan el comportamiento durante el uso de la herramienta, las dificultades encontradas y las interacciones realizadas, sin apoyarse en métricas numéricas [53, 68, 60].

Evaluación de usabilidad y adopción en entornos reales (3 artículos): se consideran estudios donde la herramienta ha sido implementada exitosamente en contextos reales (hospitales, granjas, universidades), y se observa su adopción efectiva por parte de usuarios no expertos [43, 19, 65].

Evaluación con métricas cuantitativas (10 artículos): incluye estudios que miden el rendimiento de las herramientas con datos objetivos como precisión, exactitud o validación cruzada. En algunos casos se analiza si los usuarios no técnicos pueden obtener buenos resultados [50, 18, 22, 57, 4, 8, 41, 37, 72, 70].

Evaluación mediante percepción y utilidad (5 artículos): estos trabajos utilizan encuestas o modelos como el TAM (Technology Acceptance Model) para recoger opiniones subjetivas de los usuarios sobre la utilidad, facilidad de uso y satisfacción [15, 62, 5, 48, 32].

Sin evaluación específica (9 artículos): aunque los autores indican que sus herramientas están orientadas a usuarios no expertos, no se incluye ninguna evaluación empírica que sustente dicha afirmación [49, 79, 10, 75, 52, 69, 64, 31, 38].

En resumen, se observaron diversos enfoques para evaluar la accesibilidad, pero solo una minoría de artículos utilizó métodos robustos, como métricas cuantitativas, pruebas con usuarios o cuestionarios formales. La mayoría se limitó a describir la accesibilidad en función del diseño. Esta falta de evaluación rigurosa representa una debilidad metodológica en el campo, pues limita la validación real del grado de accesibilidad de las herramientas diseñadas para usuarios no técnicos.

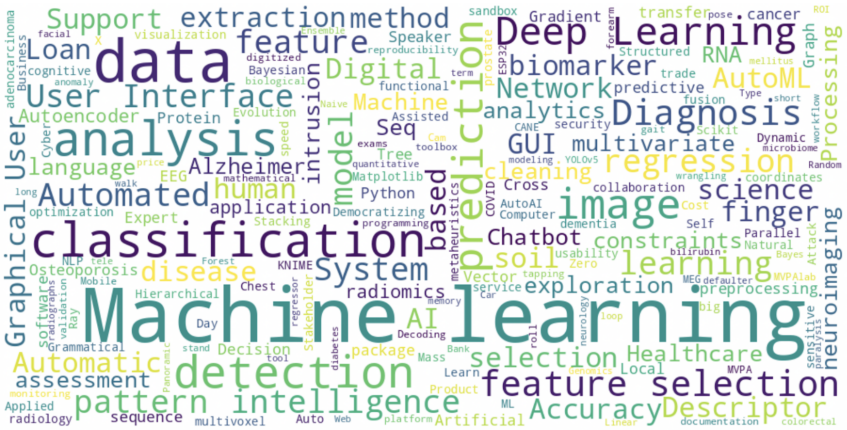
MQ6. ¿Qué definiciones y enfoques se han propuesto para diseñar herramientas de preprocesamiento?

Los estudios analizados muestran que las herramientas de preprocesamiento automatizado de datos con IA se diseñan con el objetivo común de ser accesibles para usuarios no expertos. Aunque existen diferencias en sus implementaciones, se han identificado varios enfoques recurrentes que caracterizan el diseño de estas soluciones.

Entre los enfoques más comunes identificados en las herramientas analizadas destaca la automatización, mediante la cual se automatizan tareas críticas como la selección de características, la limpieza de datos o la transformación de variables, minimizando la intervención manual. Asimismo, las herramientas priorizan la usabi-

lidad al ofrecer interfaces gráficas visuales que permiten la interacción sin necesidad de escribir código, reduciendo la barrera de entrada para personas sin experiencia técnica. Muchas de ellas incluyen ayudas contextuales, tutoriales integrados y documentación accesible que orientan al usuario paso a paso durante el proceso, brindando soporte visual y textual adicional. Por último, estas soluciones muestran una gran adaptabilidad, ya que son capaces de ajustarse a distintos tipos de datos y contextos de uso, ofreciendo configuraciones flexibles y recomendaciones automáticas en función del tipo de problema o conjunto de datos analizado.

Estos enfoques están profundamente influenciados por conceptos asociados al aprendizaje automático, la facilidad de uso y el diseño centrado en el usuario. La Figura 5 muestra una nube de palabras generada a partir de los términos más frecuentes en los estudios revisados, entre los que destacan: *Machine Learning*, *Deep Learning*, *Classification*, *Feature Selection* y *User Interface*.


Machine learning
classification
data
analysis
prediction
Deep Learning
User Interface
Support extraction method
feature
Digital
biomarker
Network
Diagnosis
AutoML
Automated
image
regression
science
finger
neuroimaging
Processing
Graphical User
Loan
Alzheimer
language
human
model
based
intrusion
Machine
Seq
analytics
GUI multivariate
cleaning
Chatbot
constraints
learning
soil
exploration
preprocessing
Healthcare
selection
feature selection
Descriptor
Accuracy
Automatic detection
assessment
pattern intelligence
sequence
Osteoporosis
disease
System
radiomics
AI
Autoencoder
Protein
Speaker
Gradient
transfer
cancer
RNA
Python
Cross
collaboration
coordinates
Decision
package
Mass
Local
Artificial
Applied radiology
multivoxel
Auto
Web
platform
documentation
colorectal
Hierarchical
Grammatical
Chest
KNIME
Forest
Mobile
NLP
Expert
application
Democratizing
programming
Stacking
optimization
Vector
service
tapping
usability
bilirubin
Natural
Parallel
Self
dementia
Cost
Zero
MEG
defaulter
MVPA
Bayes
Attack
Decoding
Learn
Product
ML
Linear
Day
Stakeholder
regression
diabetes
neurology
loop
roll
Bank
stand
Tool
Panoramic
monitoring
Bayesian
Biological
digitized
functional
Assisted
Tree
exams
toolbox
quantitative
mathematical
Matplotlib
modeling
YOLOv5
wrangling
security
CANE
gait
Scikit
microbiome
Dynamic
Random
Type
short
fusion
predictive
trade
Structured
mellitus
reproducibility
term
prostate
ESP32
forearm
sandbox
pose
ROI
Graph
workflow
Cyber
speed
Evolution
Walk
EEG
price
long
Cap
Cognitive
anomaly
facial
visualization
Business
adenocarcinoma
Computer
AutoAI
COVID
metaheuristics
Sensitive
Analysis
MVPA
documentation
Linear
Web
Auto
multivoxel
sequence
radiology
Artificial
Product
Learn
Mass
Decoding
Bank
Local
Self
Natural
Parallel
Attack
Bayes
MVPAlab
Zero
bilirubin
usability
tapping
Vector
service
memory
Car
big
Sakeholder
regression

Naive Bayes para tareas como limpieza, selección o transformación de características [8, 11, 7, 65, 18, 21, 23, 24, 25, 32, 37, 59, 67, 73, 60, 41, 57, 69, 70, 44, 64]. Algunos estudios complementan estos modelos con técnicas de optimización automática y metaheurísticas, como algoritmos de ranking supervisado [6], heurísticas de selección de características [15], métodos explicativos como SHAP (SHapley Additive exPlanations) y LIME (Local Interpretable Model-agnostic Explanations) [32], o estrategias para el ajuste automático de hiperparámetros [35].

En el análisis de datos no estructurados (imágenes, vídeo, audio, señales biomédicas), se utilizan redes neuronales convolucionales (CNN, Convolutional Neural Networks), autoencoders y arquitecturas híbridas como CNN-LSTM (Long Short-Term Memory) o CNN-RNN (Recurrent Neural Networks). Estas técnicas permiten extraer automáticamente características relevantes a partir de datos complejos [12, 33, 34, 36, 72, 78, 68, 61, 40, 74, 42, 43, 45, 58, 46].

Varios trabajos utilizan modelos preentrenados como ResNet, VGG, MobileNet, DenseNet o AlexNet para mejorar la precisión y reducir el tiempo de entrenamiento, especialmente en contextos con pocos datos [29, 30, 42, 43, 45].

Una de las líneas más relevantes en el ámbito del preprocesamiento automatizado es el desarrollo de sistemas AutoML, los cuales buscan automatizar parcial o totalmente el pipeline completo de aprendizaje automático. Este pipeline incluye tareas como la limpieza e imputación de datos, la transformación y normalización de variables, la selección de características y la construcción y evaluación de modelos [49, 50, 10, 3, 5, 9, 13, 16, 26, 38, 67, 56, 64].

Para lograr esta automatización, estos sistemas utilizan técnicas avanzadas como la optimización bayesiana, la programación genética —incluyendo enfoques como la Dynamic Structured Grammatical Evolution (DSGE)—, el ensamblado automático de modelos y la búsqueda inteligente de hiperparámetros. Algunas herramientas destacadas que implementan estas técnicas son AutoAI (Automated Artificial Intelligence) [49], AutoML-DSGE (que combina AutoML con DSGE) [50], FAE (FeAture Explorer) [10], MVPANI (Multivariate Pattern Analysis for NeuroImaging) [3] y TPOT (Tree-based Pipeline Optimization Tool) [5, 64].

En el ámbito clínico se han desarrollado herramientas visuales orientadas a profesionales no técnicos como ComPRePS (Computer-aided Preprocessing and Processing System) [47], AWESOMME (Automatic Workflow for Evaluation and Segmentation of Medical Models with Explainability) [48] y NBS-Predict (Network-Based Statistic Predict) [73]. Estas plataformas permiten realizar tareas como segmentación de imágenes médicas o evaluación de riesgo de forma automatizada. Otros sistemas aplicados en salud laboral y vigilancia usan modelos de votación combinados con CNN para detectar COVID-19 en radiografías [44], o IA explicable para evaluar riesgos ergonómicos y reconocer personas por su forma de caminar [62, 74].

En agricultura se emplean técnicas de machine learning para clasificación de semillas y análisis de imágenes de cultivos [66, 69]. En el sector industrial, herramientas como *Engineering Drawing Manager* permiten extraer datos automáticamente de planos técnicos [63]. En el contexto educativo, FASTMAN-JMP (Framework for

Automatic Segmentation, Transformation, Modeling and Analysis in Jupyter Multi-Platform) ofrece un entorno visual para aplicar modelos de IA sin escribir código [55].

También se han identificado soluciones para el tratamiento de datos complejos como secuencias genéticas, categóricas o moleculares, aplicando minería de patrones [52], codificación automática [51], y descriptores matemáticos para representar cadenas biológicas como vectores [17, 28, 39]. En procesamiento de texto, se combinan modelos de machine learning con técnicas de NLP y ontologías, a menudo con soporte humano-IA [23, 80].

En conjunto, los artículos revisados muestran una amplia gama de técnicas de IA aplicadas al preprocesamiento automatizado. Estas soluciones permiten reducir la intervención manual, mejorar la calidad de los datos y hacer el análisis accesible a usuarios no expertos, que representa un avance clave hacia la democratización del aprendizaje automático.

RQ2. ¿Cómo estas técnicas han abordado tareas críticas como limpieza, normalización y selección de características?

El preprocesamiento de datos es una fase fundamental en los sistemas de aprendizaje automático, ya que condiciona la calidad, fiabilidad y rendimiento de los modelos generados. Entre las tareas críticas más abordadas en los estudios analizados se encuentran la limpieza de datos, la normalización y la selección de características. A continuación se detallan los enfoques y técnicas utilizados en cada una de ellas.

1. Limpieza de datos.

La limpieza de datos es una etapa crítica del preprocesamiento, ya que busca llevar los datos a un estándar de calidad mediante la detección y corrección de errores, inconsistencias, valores ausentes o atípicos [9, 4]. Esta fase es especialmente relevante porque la calidad de los datos influye directamente en la efectividad de los modelos posteriores.

Las técnicas varían según el tipo de dato (estructurado, imagen, texto, señal) y el dominio de aplicación, pero suelen incluir tareas como la eliminación de redundancias (filas duplicadas, columnas con varianza cero o con más del 30 % de valores faltantes) [10, 9, 32, 55, 41], la detección y filtrado de outliers mediante técnicas estadísticas [11, 53, 26], y el tratamiento de valores nulos, ya sea mediante eliminación o imputación [52, 55, 41].

En dominios especializados, se emplean técnicas propias: en imágenes y señales, filtros como la mediana o Savitzky-Golay, transformaciones morfológicas y segmentación [20, 24, 33, 40]; en texto, tokenización, eliminación de stop words y lematización [23]; en señales EEG, se recurre a métodos como ICA para eliminar artefactos [71]; y en datos genómicos, herramientas como CD-HIT o TrimGalore eliminan secuencias redundantes o genes de baja abundancia [35, 39, 66].

Además de las técnicas clásicas, algunos estudios proponen el uso de algoritmos de aprendizaje automático como kNN, Random Forests y árboles de decisión para la imputación de valores faltantes y el manejo de outliers. Estas técnicas superan a métodos simples como la media al preservar mejor la estructura de los datos. También se plantean enfoques iterativos que permiten ajustar parámetros y validar resultados en cada etapa del preprocesamiento, lo que favorece la construcción de pipelines más flexibles y adaptados al conjunto de datos [4].

Algunos estudios utilizan asistentes inteligentes como CleverCSV [53] o integran la limpieza en pipelines AutoML [49]. En conjunto, la literatura revisada demuestra una amplia variedad de estrategias adaptadas al tipo de dato y problema abordado, con una tendencia creciente hacia su integración en sistemas AutoML que buscan ofrecer soluciones completas de preprocesamiento automatizado [49, 9].

2. Normalización de datos.

La normalización es una etapa esencial del preprocesamiento de datos, cuyo objetivo principal es ajustar la escala o distribución de las variables para garantizar una comparación equitativa entre características. Esta operación es particularmente relevante en algoritmos sensibles a las magnitudes relativas de los atributos, como los métodos basados en distancias (por ejemplo, k-NN o SVM) o los algoritmos de clustering. En los estudios revisados, se observa una amplia diversidad de técnicas de normalización, adaptadas al tipo de dato y dominio de aplicación.

Para datos numéricos continuos, se utilizan técnicas como la normalización Min-Max, la normalización Z-score o la normalización robusta (basada en la mediana y el rango intercuartílico), comúnmente implementadas a través de bibliotecas como Scikit-learn [50, 71]. Estas técnicas aparecen frecuentemente en flujos de aprendizaje automático supervisado o no supervisado, contribuyendo a mejorar la eficiencia, estabilidad y generalización de los modelos [10, 9, 32, 55, 59, 67, 73, 69].

En contextos más especializados como el análisis genómico o transcriptómico, se emplean métodos estadísticos adaptados a la naturaleza del dato. Entre ellos destacan la normalización por cuartiles, la razón mediana, y las transformaciones logarítmicas [11, 25, 26, 35]. Estas técnicas permiten corregir sesgos sistemáticos y hacer comparables las muestras, siendo fundamentales en estudios de biomarcadores o expresión génica.

También se identifican enfoques de normalización en dominios como el procesamiento de imágenes, donde las transformaciones incluyen el ajuste de tamaño, la transformación del espacio de color (por ejemplo, RGB a Lab) y la normalización de la intensidad de píxeles [42, 44, 58, 46]. Estas operaciones son cruciales para garantizar la coherencia de entrada en redes neuronales convolucionales. De forma similar, en el análisis de señales o poses humanas, se normalizan estructuras espaciales utilizando referencias anatómicas, como distancias entre articulaciones [20].

Otros estudios aplican transformaciones lineales básicas (como Min-Max) o integran herramientas automatizadas que ayudan a identificar y aplicar escalados ade-

cuados, como en [53, 77]. En el procesamiento de lenguaje natural, la normalización adopta formas distintas, como la lematización, que reduce las palabras a su forma base para facilitar su análisis semántico [23].

Por último, algunos artículos no detallan explícitamente las técnicas utilizadas, pero sí incluyen la normalización como parte de pipelines automáticos [49, 18], lo que pone de manifiesto su papel fundamental dentro del flujo de trabajo automatizado.

En conjunto, los trabajos analizados demuestran que la normalización es una práctica transversal, aplicada en contextos muy diversos, y que cumple un papel clave en la calidad del preprocesamiento y el rendimiento posterior de los modelos de aprendizaje automático.

3. Selección de características.

La selección de características es otra fase clave del preprocesamiento, ya que permite reducir la dimensionalidad, eliminar atributos irrelevantes y mejorar tanto el rendimiento como la interpretabilidad de los modelos. Los artículos revisados emplean una gran variedad de técnicas, adaptadas al tipo de datos y al dominio de aplicación.

Por un lado, se utilizan enfoques automáticos integrados en pipelines y sistemas AutoML, como TPOT o AutoAI, y modelos de aprendizaje profundo como redes neuronales convolucionales (CNN), especialmente en el procesamiento de imágenes o secuencias [49, 5, 9, 42, 44, 46, 64].

También son frecuentes las técnicas de reducción de dimensionalidad como PCA (Principal Component Analysis), t-SNE (t-distributed Stochastic Neighbor Embedding), UMAP (Uniform Manifold Approximation and Projection) o SVD (Singular Value Decomposition) [10, 9, 55, 59, 57], así como métodos estadísticos como ANOVA (Análisis de Varianza), RFE (Recursive Feature Elimination), ReliefF o LASSO (Least Absolute Shrinkage and Selection Operator) [21, 37, 57, 48].

Varios estudios adoptan enfoques wrapper y embedded utilizando modelos como Random Forest, SVM (Support Vector Machines) o regresión logística regularizada, que ofrecen métricas internas de importancia [32, 38, 73, 41]. Asimismo, se aplican métodos de filtro como la prueba Chi-cuadrado, la información mutua o VarianceThreshold [50, 11, 54, 67].

En contextos específicos se emplean herramientas y algoritmos adaptados: en genómica, métodos como DaMiRseq, Biosigner o GMDH (Group Method of Data Handling) [17, 25, 35]; en imágenes, técnicas como Gabor wavelets, mapas de saliencia o segmentación [65, 24, 40]; y en señales, descriptores como MFCC (Mel-frequency Cepstral Coefficients) o transformadas de Fourier [36, 59]. Además, algunos estudios utilizan herramientas híbridas o frameworks completos como AutoRadiomics o ComPRePS (Computer-aided Preprocessing and Processing System), que integran extracción, selección y modelado [16, 47].

En resumen, los trabajos analizados evidencian un uso extenso y variado de

técnicas de selección de características, fundamentales para optimizar los modelos, reducir la complejidad computacional y facilitar la interpretación de resultados.

Las tareas de limpieza, normalización y selección de características son abordadas mediante una diversidad de técnicas adaptadas al tipo de dato y al dominio de aplicación. Estas estrategias no solo optimizan el rendimiento de los modelos, sino que también garantizan su fiabilidad, interpretabilidad y aplicabilidad por parte de usuarios no expertos.

RQ3. ¿Qué características hacen que estas herramientas sean accesibles y efectivas para usuarios no expertos?

Una de las prioridades más destacadas en el desarrollo de herramientas basadas en IA es su accesibilidad para usuarios sin conocimientos técnicos, especialmente en ámbitos como la medicina, bioinformática, agricultura o análisis de datos general. Los artículos analizados destacan múltiples características que hacen que estas herramientas no solo sean accesibles, sino también efectivas, desde interfaces intuitivas hasta procesos totalmente automatizados.

La mayoría de las herramientas revisadas incorporan interfaces gráficas de usuario (GUIs) intuitivas, simplificadas y diseñadas para guiar al usuario paso a paso, sin requerir conocimientos de programación [76, 50, 10, 3, 5, 12, 9, 13, 11, 15, 7, 16, 17, 65, 18, 71, 20, 77, 54, 22, 23, 24, 27, 28, 29, 30, 32, 34, 36, 37, 39, 55, 59, 66, 67, 73, 61, 40, 41, 57, 62, 74, 42, 44, 45, 58, 46, 47, 48, 64]. Estas GUIs permiten realizar tareas complejas de preprocesamiento, análisis o predicción mediante menús desplegables, botones y resultados visuales (como gráficos o tablas interpretables).

Numerosas herramientas incluyen funciones de automatización en tareas clave como la selección de características, construcción de modelos o ajuste de hiperparámetros [49, 5, 14, 9, 16, 53, 26, 31, 67]. Plataformas como AutoML, AutoRadiomics o ComPRePS integran esta automatización en pipelines completos, reduciendo la necesidad de intervención humana y habilitando su uso por usuarios clínicos, docentes o técnicos sin formación específica en IA.

Una característica esencial en herramientas accesibles es la explicabilidad. Varios estudios integran mecanismos como SHAP, visualizaciones de importancia de variables o categorizaciones semánticas (por ejemplo, “riesgo alto, medio, bajo”) para que el usuario comprenda cómo se obtienen los resultados [51, 54, 32, 33, 78, 38, 41, 62, 64], aumentando así la confianza y transparencia del sistema.

Las herramientas revisadas muestran una alta compatibilidad con entornos reales. Algunas funcionan en navegadores web o dispositivos móviles [16, 65, 33, 41, 42, 45, 46], operan sin conexión [33, 42, 46] o aceptan formatos ampliamente usados como CSV y Excel [14, 77, 26, 64], lo que facilita su adopción incluso en contextos con recursos limitados.

Muchos estudios incluyen ayudas visuales, tutoriales y documentación dentro de las propias plataformas [4, 51, 71, 52, 53, 23, 28]. Además, algunas herramientas ofrecen integración con entornos interactivos como Jupyter Notebooks, Streamlit o

Shiny [5, 4, 54], lo que permite una experiencia semiguiada y personalizable.

En ámbitos como el diagnóstico médico, la bioinformática o el análisis de imágenes, las herramientas se han diseñado para permitir tareas como segmentación, clasificación o extracción de características con pocos clics, sin requerir programación [24, 27, 29, 39, 59, 66, 40, 57, 48]. Esto facilita el uso por parte de profesionales del dominio (como personal médico) sin necesidad de comprender los detalles técnicos de la IA.

Algunas plataformas permiten elegir entre modos básicos o avanzados según el perfil del usuario [5, 16, 28, 47], permitiendo un uso gradual que se adapta tanto a principiantes como a usuarios más experimentados.

Las herramientas accesibles para usuarios no expertos combinan una interfaz clara, automatización de procesos clave, explicabilidad de resultados y facilidad de integración con entornos reales. Estas características son esenciales para democratizar el uso de la IA y extender sus beneficios a contextos donde no se dispone de formación técnica avanzada.

RQ4. ¿Cuáles son las ventajas y limitaciones de las soluciones basadas en IA frente a métodos tradicionales de preprocesamiento?

Las soluciones basadas en IA han transformado significativamente el preprocesamiento de datos en distintos contextos, ofreciendo una automatización más flexible, mayor precisión y capacidad de adaptación. Frente a los métodos tradicionales —que dependen en gran medida de la intervención manual, configuraciones estáticas y reglas predefinidas— los enfoques basados en IA presentan ventajas notables, aunque también plantean desafíos importantes.

Ventajas de las soluciones basadas en IA:

Automatización y eficiencia del flujo de trabajo. Herramientas como AutoML [50, 5, 9, 16, 67, 64] y AutoAI [49] permiten automatizar todo el *pipeline* de preprocesamiento, desde la limpieza hasta la selección de modelos, reduciendo la necesidad de intervención humana y acelerando el proceso. Sistemas como FAE [10], GRASP-FS [15], MFeaST [11] o matRadiomics [38] optimizan múltiples pasos a la vez, lo que mejora la eficiencia frente a métodos manuales fragmentados.

Extracción y selección automática de características. Modelos de aprendizaje profundo como las redes convolucionales (CNN) [42, 46, 29, 65, 69] pueden extraer automáticamente características relevantes sin necesidad de ingeniería manual. En contextos clínicos o genómicos, herramientas como GeneSelectML [25], AutoRadiomics [16], TPOT [5, 64] o NBS-Predict [73] permiten una selección automatizada con alta precisión y explicabilidad [32, 59, 62].

Robustez y adaptación a múltiples contextos. Las soluciones basadas en IA se adaptan mejor a condiciones variables como ruido de fondo [33, 36, 78], imágenes de baja calidad [42, 46], texto libre o datos secuenciales [14, 51, 52, 22, 23], donde los métodos tradicionales suelen fallar.

Interfaces accesibles y democratización tecnológica. Muchas herramientas de IA incorporan interfaces gráficas intuitivas [76, 3, 12, 4, 11, 15, 18, 77, 27, 32, 34, 55, 66, 41, 57, 62, 47], lo que facilita su adopción por usuarios no técnicos, a diferencia de los métodos clásicos que suelen requerir programación o intervención experta.

Mejora del rendimiento. Varios artículos reportan que los modelos basados en IA superan a los enfoques tradicionales en términos de precisión, sensibilidad y generalización [54, 24, 29, 30, 39, 40]. Herramientas como AutoML o técnicas de *deep learning* permiten identificar relaciones no lineales y patrones complejos que los métodos clásicos no pueden captar [17, 35].

Adaptabilidad y aprendizaje continuo. Técnicas como el aprendizaje activo [58], la optimización evolutiva [37, 41] o la personalización de modelos según contexto [49, 74] permiten una adaptación más dinámica que los flujos tradicionales predefinidos.

Limitaciones de las soluciones basadas en IA frente a métodos tradicionales:

Complejidad computacional. Muchas soluciones modernas requieren elevados recursos computacionales , especialmente las basadas en redes profundas o AutoML [50, 5, 9, 78, 74, 45], lo que puede limitar su uso en contextos con restricciones técnicas.

Falta de interpretabilidad. A diferencia de los métodos tradicionales, más transparentes, muchos modelos de IA funcionan como "cajas negras", lo que genera desconfianza en áreas sensibles como la salud [32, 67, 62, 64].

Dependencia de grandes volúmenes de datos. El entrenamiento efectivo de modelos de IA suele requerir grandes cantidades de datos etiquetados y variados [42, 44, 46, 29, 35], lo que puede resultar costoso en tiempo y recursos.

Curva de aprendizaje y configuración. Aunque automatizadas, algunas herramientas necesitan configuración previa o conocimientos mínimos para integrarse adecuadamente en ciertos entornos [15, 7, 52, 53, 74].

Riesgo de sobreajuste. Cuando no se ajustan bien los parámetros o los datos están desbalanceados, los modelos pueden aprender patrones que no son representativos y que no funcionan bien con datos nuevos [9, 18, 41, 67].

Limitaciones de generalización. Algunas soluciones están demasiado enfocadas en un dominio o tarea concreta [7, 23, 26], y no se adaptan fácilmente a otros entornos o nuevas técnicas [10, 38].

Dependencia de la calidad de los datos. Herramientas como las de visión por computador o análisis de imágenes pueden verse muy afectadas por condiciones técnicas externas (iluminación, ruido, calidad del dispositivo), lo que reduce su efectividad respecto a técnicas tradicionales más estandarizadas [27, 69, 24].

En conclusión, las soluciones basadas en IA ofrecen ventajas notables en eficiencia, precisión y accesibilidad, transformando el preprocesamiento de datos en distintos contextos. No obstante, enfrentan limitaciones técnicas, interpretativas y operativas que aún deben resolverse para garantizar su aplicabilidad generalizada. Combinar la IA con las prácticas tradicionales, especialmente en entornos donde es

crucial ser cuidadoso, podría ayudar a que el análisis de datos sea más accesible, seguro y efectivo para todos.

RQ5. ¿Qué áreas de investigación relacionadas con el preprocesamiento de datos y usuarios no expertos aún no han sido exploradas?

Aunque se han desarrollado numerosas herramientas para facilitar el preprocesamiento de datos, aún existen múltiples áreas de investigación poco exploradas, especialmente en lo relativo a su accesibilidad, transparencia y personalización para usuarios sin conocimientos técnicos.

Una de las principales necesidades identificadas es la mejora de la explicabilidad y la transparencia de los pasos de preprocesamiento. Varios estudios [4, 22, 26, 59, 67, 73, 62] coinciden en que los usuarios no expertos tienen dificultades para comprender cómo afectan las decisiones de preprocesamiento a los resultados finales. Se propone, por tanto, el desarrollo de módulos de IA explicable (XAI) que acompañen cada paso del flujo y proporcionen visualizaciones interactivas, comparaciones antes-después o explicaciones accesibles.

Otra línea de investigación aún incipiente es la automatización inteligente adaptada al tipo de dato. Aunque herramientas como AutoML [50, 9, 64] han avanzado en la automatización del *pipeline*, todavía muestran limitaciones al enfrentarse a dominios específicos como imágenes médicas [10, 27, 43], señales EEG [71], microbiomas [26], datos secuenciales [52] o series temporales [14]. En estos contextos, se requiere una personalización mayor de los métodos sin necesidad de intervención técnica por parte del usuario.

También se ha identificado como prioridad el desarrollo de interfaces gráficas adaptadas a distintos perfiles de usuario. Algunos estudios [3, 5, 4, 11, 18, 59, 57, 58, 46, 64] proponen la integración de asistentes virtuales, tutoriales interactivos y sistemas de recomendación que guíen al usuario paso a paso, pero aún son pocos los trabajos que implementan de forma efectiva estas propuestas.

En paralelo, se destaca la falta de herramientas que aprendan del comportamiento del usuario y adapten dinámicamente sus recomendaciones. Algunas publicaciones [51, 53] sugieren sistemas que personalicen el flujo de trabajo a partir de interacciones previas, lo que permitiría ajustar automáticamente las opciones y simplificar el proceso para usuarios recurrentes.

En cuanto al contexto de uso, aún existe una gran brecha en el diseño de soluciones para entornos con recursos limitados. El desarrollo de herramientas que funcionen sin conexión, en dispositivos básicos o con infraestructura reducida es esencial para su aplicación en clínicas móviles, zonas rurales o países en vías de desarrollo [20, 24, 27, 46].

También se ha detectado una escasez de sistemas que orienten a los usuarios frente a datos problemáticos. Casos como desequilibrio de clases, errores de anotación, valores faltantes o regiones mal segmentadas son habituales, pero pocas herramientas incluyen mecanismos de detección automática o recomendaciones prácticas para

su tratamiento [76, 5, 8, 4, 35, 38, 67].

En sectores específicos como la odontología [29], la neuroimagen [3], el análisis de proteínas [28], la clasificación de suelos [65] o el diagnóstico mediante vídeo [20, 62], se evidencia la necesidad de soluciones más accesibles que combinen interfaces visuales, explicaciones comprensibles y procesos simplificados para el preprocesamiento y la toma de decisiones.

Finalmente, se observa una limitada integración entre técnicas de IA explicable y conocimiento experto del dominio, especialmente en entornos biomédicos e industriales. Algunos estudios [52, 59, 41] proponen como vía de futuro el diseño de sistemas híbridos que combinen análisis automático con supervisión experta, promoviendo una colaboración humano-máquina más efectiva y segura.

En conjunto, estas lagunas evidencian la necesidad de continuar investigando cómo hacer que el preprocesamiento de datos no solo sea automático y preciso, sino también comprensible, adaptable y universalmente accesible.

2.4. Conclusiones de la revisión sistemática

Los resultados obtenidos en esta SLR demuestran que se han cumplido los propósitos planteados, ofreciendo una visión crítica, estructurada y actualizada sobre el uso de técnicas de IA para el preprocesamiento automatizado de datos orientado a usuarios no expertos.

En primer lugar, se identificó y caracterizó un amplio abanico de técnicas empleadas en los últimos años, que abarcan desde algoritmos clásicos como Random Forest, SVM y KNN, hasta métodos más sofisticados como redes neuronales convolucionales (CNN), autoencoders, modelos preentrenados y sistemas AutoML. Esta diversidad confirma el avance del campo y la consolidación de un ecosistema robusto de herramientas para distintos dominios.

Además, se analizó cómo estas técnicas abordan tareas críticas del preprocesamiento, incluyendo la limpieza de datos, la normalización y la selección de características. Se constató que las herramientas ofrecen una variedad de estrategias adaptadas a distintos contextos y necesidades, mostrando así la flexibilidad y madurez técnica de la IA aplicada al preprocesamiento.

Se evaluó también la accesibilidad y efectividad de las herramientas para usuarios no expertos, identificando aspectos como interfaces gráficas intuitivas, automatización de procesos, módulos explicativos, compatibilidad multiplataforma y documentación integrada. Aunque la mayoría de las soluciones priorizan la facilidad de uso, se observó una falta de evaluaciones empíricas que validen objetivamente su accesibilidad real, que constituye una oportunidad de mejora en investigaciones futuras.

El estudio permitió comparar las ventajas de estas soluciones frente a métodos tradicionales, destacando su automatización, adaptabilidad y mayor precisión, así como sus retos, como la falta de interpretabilidad o la necesidad de grandes volúmenes de datos. Esta comparación subraya la importancia de integrar enfoques híbridos

que combinen lo mejor de la IA y los métodos clásicos.

Por último, se detectaron vacíos y oportunidades de investigación que aún no han sido completamente explorados, como la explicabilidad en el preprocesamiento, las interfaces adaptativas, el aprendizaje del comportamiento del usuario o el soporte visual frente a datos problemáticos. Estos hallazgos abren nuevas líneas de trabajo para el futuro desarrollo de herramientas más accesibles, comprensibles y eficaces.

En conjunto, esta revisión sistemática establece una base firme para crear soluciones que sean más inclusivas y fáciles de usar, ayudando así a acercar la IA a usuarios sin experiencia técnica.

3. Metodología

Tras realizar un estudio del estado del arte mediante la revisión sistemática de la literatura, se inició el proceso metodológico para desarrollar el sistema de preprocesamiento inteligente de datos. Esta sección describe cómo se abordó cada etapa del desarrollo, desde la selección y análisis de los algoritmos más adecuados hasta la construcción de una herramienta accesible y robusta.

En una primera fase, se investigaron distintas técnicas de limpieza, normalización, selección de características y evaluación automática, con el objetivo de identificar aquellas que ofrecieran el mejor equilibrio entre precisión, interpretabilidad y facilidad de automatización. Posteriormente, se desarrolló un prototipo inicial en Python que permitió implementar y validar estas técnicas de forma controlada.

A continuación, se diseñó e implementó una arquitectura cliente-servidor utilizando Angular y Flask, permitiendo a los usuarios interactuar con el sistema de forma sencilla a través de una interfaz web. Una vez construido el sistema completo, se llevó a cabo una evaluación integral que incluyó tanto una validación práctica con usuarios (no expertos y con experiencia técnica) como la realización de pruebas con diferentes conjuntos de datos reales. Esta doble evaluación permitió recoger observaciones valiosas sobre la usabilidad, comprensión y utilidad de la herramienta, así como analizar el rendimiento técnico de cada fase del flujo de preprocesamiento.

Finalmente, el sistema fue refinado incorporando las mejoras sugeridas durante la evaluación, lo que dio lugar a una versión final más robusta, clara y adaptada a las necesidades reales de su público objetivo.

3.1. Selección y análisis de algoritmos

La elección de los algoritmos empleados en el sistema de preprocesamiento inteligente de datos se fundamenta en criterios de eficacia demostrada, interpretabilidad, y facilidad de integración en un entorno automatizado accesible para usuarios no expertos. A partir de la revisión sistemática realizada (ver Sección 2), se identificó una amplia variedad de técnicas utilizadas en la literatura reciente para abordar

las principales etapas del preprocesamiento: limpieza, normalización, selección de características y evaluación automática de modelos. Sin embargo, para este trabajo se seleccionaron específicamente aquellos algoritmos que equilibran rendimiento, robustez y capacidad de automatización sin requerir una configuración técnica compleja.

Aunque los sistemas AutoML ofrecen un elevado nivel de automatización para todo el flujo de análisis de datos, este trabajo no optó por delegar completamente en ellos las tareas de preprocesamiento. La revisión sistemática reveló que muchos enfoques basados en AutoML carecen de mecanismos de transparencia, intervención del usuario o control sobre decisiones críticas como la limpieza, imputación o selección de características. Además, su integración directa puede dificultar la explicación clara del proceso a personas no expertas, que es precisamente el público objetivo de esta herramienta. Por estas razones, se decidió aplicar un enfoque híbrido: se automatizan tareas clave, pero manteniendo un flujo estructurado y comprensible. De esta forma, se ofrece una solución técnica robusta, pero también accesible y explicable.

Cabe destacar que AutoML sí se utilizó en este trabajo, pero únicamente en la etapa final de evaluación automática de modelos, donde aporta valor al comparar de forma objetiva distintos algoritmos sin requerir intervención técnica. Esta elección permite aprovechar sus ventajas sin comprometer la transparencia ni la comprensión del proceso completo.

3.1.1. Limpieza de datos

La limpieza de datos es una etapa fundamental del preprocesamiento en cualquier flujo de análisis de datos. Su objetivo principal es detectar y corregir errores, inconsistencias o valores faltantes que puedan comprometer la calidad del análisis posterior. Entre los problemas más comunes que aborda esta fase se encuentran los valores nulos, las codificaciones inconsistentes (como “N/A”, “None” o “?”), los duplicados y los datos fuera de rango. Una limpieza adecuada no solo mejora la calidad del conjunto de datos, sino que también incrementa el rendimiento y la fiabilidad de los modelos de aprendizaje automático.

En el contexto del sistema propuesto, se ha optado por centrar la limpieza en el tratamiento de valores faltantes, dado que es una de las problemáticas más frecuentes en datos reales. En lugar de eliminar filas o columnas con valores ausentes —lo que puede generar una pérdida significativa de información— se han implementado métodos de imputación automática, que estiman valores coherentes en base a la estructura interna de los datos disponibles.

Para esta tarea, se han seleccionado tres técnicas distintas: KNN, Imputación Iterativa con modelos basados en boosting, y Random Forest. Estas técnicas fueron elegidas de manera deliberada para maximizar la robustez del sistema frente a distintos tipos de datos y patrones de valores nulos.

La imputación mediante KNN consiste en estimar un valor faltante calculando la media (o moda) de los valores observados en las observaciones más cercanas al

registro incompleto, considerando su similitud en el espacio de características. Esta técnica es intuitiva, no paramétrica y suele funcionar bien cuando los datos presentan patrones locales coherentes. Se seleccionó por su simplicidad y buena capacidad para imputar columnas numéricas sin necesidad de entrenamiento complejo.

La imputación Iterativa, por otro lado, aplica un modelo predictivo (en este caso un HistGradientBoosting) para cada variable con valores faltantes, utilizando las demás como predictores. Este enfoque es más sofisticado, ya que captura relaciones multivariantes complejas entre atributos. Se escogió por su capacidad para generar imputaciones más consistentes y coherentes con la distribución general del conjunto de datos, especialmente en presencia de múltiples variables correlacionadas.

En tercer lugar, se incluyó la imputación basada en Random Forest, donde se entrena un modelo de árboles sobre las observaciones completas para predecir los valores de las observaciones con datos faltantes [4]. Su capacidad para modelar relaciones no lineales y manejar interacciones entre variables la convierte en una alternativa robusta frente a los otros métodos.

La decisión de incluir exactamente tres técnicas responde a una estrategia de cobertura y redundancia balanceada. Por un lado, incorporar sólo una técnica supondría asumir que se adapta bien a todos los contextos, lo que no es realista. Se optó por estas tres porque representan enfoques complementarios (basado en vecinos, en modelos iterativos y en ensamblados de árboles) que permiten al sistema comparar resultados y seleccionar el más adecuado de forma automática según la naturaleza y cantidad de los valores faltantes presentes en cada conjunto de datos.

3.1.2. Normalización de datos

La normalización de datos es un paso esencial en el preprocesamiento, ya que garantiza que todas las variables numéricas se encuentren en una escala comparable, evitando que las diferencias de magnitud influyan negativamente en los modelos predictivos. Para este sistema, se ha optado por una estrategia híbrida que combina métodos tradicionales con técnicas basadas en IA, concretamente un autoencoder no supervisado.

En primer lugar, se ha seleccionado la técnica clásica de escalado MinMax, que transforma cada variable numérica para que sus valores se sitúen dentro del rango $[0, 1]$. Esta normalización es fundamental para preparar los datos antes de aplicar métodos basados en redes neuronales, ya que muchas funciones de activación —como la *sigmoide*— requieren entradas acotadas en dicho intervalo. Este tipo de escalado ha sido ampliamente utilizado en entornos reales y pipelines automáticos de aprendizaje automático, como se observa en artículos como [50, 77, 30, 66, 73].

Como complemento, se incorpora una técnica más avanzada: un autoencoder no supervisado, cuya función es aprender una representación reconstruida de los datos de entrada. Aunque los autoencoders no son técnicas de normalización en sentido estricto (como MinMax o Z-score), su funcionamiento implica una codificación y decodificación que transforma los datos hacia una versión más regular, menos ruidosa

y con patrones más estables. Esta transformación cumple, en la práctica, una función de normalización funcional, tal como ha sido documentado en estudios relacionados con imágenes médicas y señales acústicas [33, 34].

Aunque estas aplicaciones se centran en dominios como la imagen o el sonido, la lógica subyacente de los autoencoders —aprender una reconstrucción eficiente y coherente de los datos— es aplicable también a conjuntos de datos estructurados. En este trabajo, se propone su uso como técnica de normalización avanzada, especialmente útil cuando se trabaja con datos ruidosos, con relaciones no lineales o con potenciales inconsistencias entre variables.

La combinación de ambos métodos —escalado MinMax como paso inicial y transformación mediante autoencoder posteriormente— permite aprovechar lo mejor de ambos enfoques: la simplicidad y compatibilidad del escalado tradicional, y la capacidad del autoencoder para modelar relaciones complejas y estabilizar las entradas.

3.1.3. Selección de características

La selección de características es una etapa crítica en todo sistema de análisis predictivo, ya que permite reducir la dimensionalidad del conjunto de datos, eliminar atributos irrelevantes o redundantes y mejorar la interpretabilidad y eficiencia de los modelos posteriores. En este trabajo se ha optado por una estrategia basada en la estimación de la importancia de características mediante modelos de aprendizaje automático supervisado.

Concretamente, se utilizan los algoritmos Random Forest y Extra Trees Classifier para calcular un ranking de importancia de las variables en función de su contribución a la predicción de la variable objetivo. Ambos métodos están basados en árboles de decisión, pero presentan ligeras diferencias en su funcionamiento interno que afectan a la forma en que determinan qué columnas son más relevantes.

El algoritmo Random Forest construye múltiples árboles de decisión entrenados sobre subconjuntos distintos del conjunto de entrenamiento mediante bootstrap sampling. En cada nodo de los árboles, se selecciona la mejor variable para dividir los datos en función de un criterio de pureza, como el índice de Gini o la entropía. La importancia de cada variable se calcula como la mejora media del criterio de división a lo largo de todos los árboles en los que aparece. En otras palabras, cuanto más contribuya una variable a mejorar la predicción en distintos árboles, mayor será su puntuación.

Por su parte, Extra Trees Classifier comparte la misma filosofía que Random Forest, es decir, el uso de conjuntos de árboles y el cálculo de importancia por mejora de división. Sin embargo, introduce una mayor aleatoriedad en su proceso de construcción: en lugar de buscar el umbral óptimo de división en cada nodo, Extra Trees selecciona aleatoriamente un valor dentro del rango de posibles divisiones. Esta aleatoriedad adicional permite explorar más combinaciones y, en muchos casos, genera un ranking de características más diverso y menos sesgado, especialmente útil cuando existen muchas variables con importancia similar.

Ambos métodos tienen ventajas complementarias: Random Forest tiende a ser más preciso y conservador en la selección, mientras que Extra Trees puede detectar relaciones útiles en variables que Random Forest subestima debido a su estructura más determinista. Estos algoritmos, son ampliamente utilizados en la literatura científica [7, 11, 30, 32, 25, 38].

Una vez obtenidas las puntuaciones de importancia, se aplica un umbral de corte de 0,02 para seleccionar únicamente aquellas características que superen dicho valor. Aunque este valor no proviene de una referencia específica, ha sido elegido de forma empírica como compromiso entre la eliminación de variables irrelevantes y la conservación de aquellas con una contribución mínima pero potencialmente significativa. Un umbral más alto podría suprimir atributos informativos con baja señal, mientras que uno más bajo mantendría demasiadas variables, reduciendo la eficacia de la selección. Este valor también garantiza que el número de variables seleccionadas sea manejable y adecuado para usuarios no expertos, evitando complejidad innecesaria en la interpretación de resultados.

3.1.4. Evaluación automática

La fase final del preprocesamiento automatizado incluye una etapa de evaluación automática de modelos predictivos mediante el uso de H2O AutoML, una herramienta de aprendizaje automático automatizado desarrollada por la plataforma de código abierto H2O.ai.

AutoML, es un enfoque que permite automatizar las etapas clave del proceso de aprendizaje automático: selección del algoritmo, ajuste de hiperparámetros, validación cruzada, entrenamiento y evaluación. Su objetivo principal es facilitar el uso de modelos avanzados a usuarios sin conocimientos técnicos en ciencia de datos.

H2O AutoML se encarga de detectar el tipo de problema (clasificación o regresión) según la variable objetivo, y lanza en paralelo distintos algoritmos como Random Forest, XGBoost, Gradient Boosting, redes neuronales profundas y modelos lineales. A cada uno se le aplican configuraciones diferentes, se evalúan sus resultados y se genera un leaderboard con sus métricas de rendimiento. Finalmente, el sistema selecciona el mejor modelo según una métrica adecuada al tipo de problema (por ejemplo, AUC para clasificación binaria o R^2 para regresión).

Este sistema permite al usuario obtener un modelo predictivo optimizado sin necesidad de configurar manualmente parámetros técnicos, lo que resulta especialmente útil para entornos donde se busca automatizar el análisis sin requerir intervención especializada [81].

3.2. Prototipo inicial

Con el fin de validar de forma incremental las técnicas seleccionadas y facilitar su posterior integración en una interfaz gráfica, se diseñó un primer módulo de preprocesamiento completamente funcional como programa en Python ejecutable desde

terminal. Este módulo permite aplicar paso a paso el flujo completo de tratamiento de un archivo CSV, desde su carga hasta la generación de un nuevo conjunto de datos listo para el modelado automático.

El programa fue concebido como una herramienta de propósito general, capaz de adaptarse a distintos conjuntos de datos estructurados, y está implementado con una lógica modular, donde cada etapa del preprocesamiento puede observarse de forma independiente, facilitando tanto su depuración como su comprensión por parte de usuarios o evaluadores técnicos.

La Figura 6 muestra un esquema general del flujo de trabajo implementado en el módulo inicial de preprocesamiento.

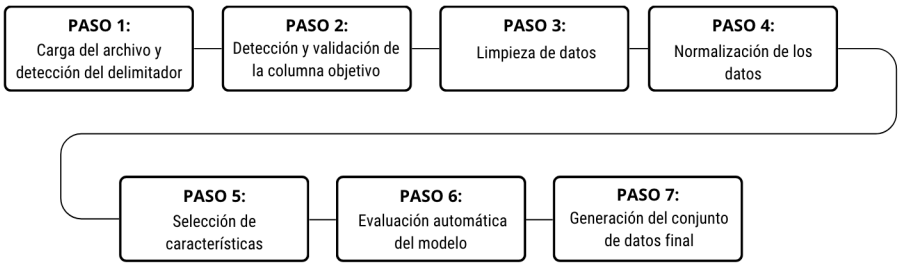


Figura 6: Diagrama del flujo de procesamiento del módulo inicial.

A continuación, se describen las principales fases implementadas en este módulo, siguiendo el flujo lógico del preprocesamiento:

Paso 1. Carga del archivo y detección del delimitador

El primer paso del módulo consiste en la carga del archivo CSV seleccionado por el usuario. Dado que los archivos CSV pueden utilizar distintos delimitadores (como comas, punto y coma, tabulaciones, entre otros), el sistema implementa una detección automática del separador utilizando la clase `csv.Sniffer` de Python. Esta técnica analiza una muestra del contenido del archivo y trata de inferir el delimitador más probable.

En caso de que no sea posible determinar el separador automáticamente, el sistema realiza una prueba secuencial con varios delimitadores comunes (`','`, `';'`, `'\t'`, `'|'`), seleccionando aquel que produzca una tabla con múltiples columnas. Si todas las pruebas fallan, se utiliza la coma (`','`) como opción por defecto.

El sistema también contempla la presencia de valores faltantes representados con diversas codificaciones como `None`, `?`, `NaN`, entre otros, los cuales son automáticamente interpretados como valores nulos durante la carga del archivo.

Este proceso inicial garantiza una lectura robusta del archivo, permitiendo la adaptación a múltiples formatos de datos sin necesidad de intervención manual.

Paso 2. Detección y validación de la columna objetivo

Una vez cargado el conjunto de datos, el sistema intenta identificar de forma automática la columna objetivo, es decir, la variable que se desea predecir. Esta detección se basa en un conjunto de nombres comunes (`income`, `target`, `label`, `objetivo`, `clase`) frecuentemente utilizados en tareas de predicción.

En caso de no encontrar ninguna coincidencia exacta, el sistema analiza las columnas disponibles y propone como sugerencia la última columna que contenga únicamente dos valores distintos es decir, una columna binaria. Este tipo de columnas son comunes en problemas de clasificación binaria, aunque sus valores no sean necesariamente 0 y 1; pueden ser cadenas como yes/no, A/B o cualquier otro par de categorías. Además, en muchos conjuntos de datos reales, la columna objetivo suele encontrarse al final del archivo, por lo que seleccionar la última columna binaria es una heurística razonable en ausencia de una designación explícita.

En todos los casos, el sistema permite al usuario confirmar o seleccionar manualmente la columna objetivo deseada, garantizando flexibilidad y control total sobre el proceso.

Paso 3. Limpieza de datos

Antes de aplicar cualquier técnica de imputación, se realiza un tratamiento especial de las variables categóricas. Estas variables, que originalmente contienen texto (como yes/no, A/B o nombres de categorías), son transformadas temporalmente en valores numéricos mediante una codificación interna. Esta codificación es necesaria para permitir que los algoritmos de imputación puedan operar sobre ellas. El proceso se aplica únicamente a las variables distintas de la columna objetivo y se guarda un diccionario de mapeo que asocia cada código numérico a su categoría original.

Este mecanismo garantiza que el flujo sea compatible tanto con variables numéricas como categóricas, evitando distorsiones semánticas durante las etapas posteriores.

La limpieza de datos es una fase crítica en todo proceso de preprocesamiento, ya que permite corregir errores, eliminar inconsistencias y garantizar la calidad del conjunto de datos antes de aplicar modelos predictivos. En este sistema, la limpieza se centra principalmente en el tratamiento de valores faltantes (nulos), una de las problemáticas más frecuentes en datos reales.

Una vez eliminadas las columnas completamente vacías y codificadas las variables categóricas (excepto la columna objetivo), se aplican tres técnicas de imputación distintas para estimar los valores ausentes en las columnas numéricas:

- K-Nearest Neighbors: estima cada valor nulo a partir del promedio de los valores de las observaciones más similares en el resto de atributos.
- Imputación Iterativa (IterativeImputer): entrena un modelo de regresión para cada columna con valores faltantes utilizando el resto de columnas como

predictores, y actualiza iterativamente las imputaciones.

- Random Forest (RF): entrena un modelo de árboles de decisión para predecir los valores nulos de cada columna a partir de las observaciones completas, modelando relaciones no lineales entre atributos.

Estas tres técnicas se aplican en paralelo sobre el mismo conjunto de datos, generando tres versiones imputadas. Posteriormente, se evalúa cuántos valores nulos quedan en cada una, y se selecciona automáticamente aquella que haya logrado eliminar completamente los valores faltantes. En caso de que varias lo consigan, se escoge la primera que haya logrado imputar todos los valores nulos. Si ninguna lo consigue, se selecciona como opción por defecto la que presente menor cantidad de nulos residuales.

Paso 4. Normalización de los datos

La normalización de los datos es un paso fundamental en el flujo de preprocesamiento, ya que garantiza que las variables numéricas se encuentren en una escala común, evitando que aquellas con valores más grandes dominen el comportamiento de los modelos. En este sistema, se ha implementado una estrategia de normalización avanzada basada en la combinación de escalado clásico y técnicas de aprendizaje profundo.

Antes de iniciar la normalización, el sistema identifica qué columnas deben ser transformadas. Para ello, excluye explícitamente tanto la columna objetivo como aquellas variables categóricas que fueron codificadas temporalmente en pasos anteriores para facilitar la imputación. Estas variables categóricas no deben tratarse como numéricas reales, ya que sus valores codificados no reflejan magnitudes o distancias válidas entre categorías.

El proceso consta de dos etapas principales:

1. Escalado MinMax: todos los valores son transformados al rango $[0, 1]$ utilizando `MinMaxScaler`. Esta transformación es importante porque la capa de salida del autoencoder utiliza una función matemática llamada *sigmoide*, que solo puede producir valores entre 0 y 1. Si los datos de entrada no están dentro de ese rango, el autoencoder no podrá aprender correctamente a reconstruirlos y su rendimiento se verá afectado.

2. Entrenamiento y reconstrucción: el autoencoder es una red neuronal que aprende a reconstruir sus propias entradas. Esto significa que intenta generar una copia lo más parecida posible de los datos originales tras pasar por una representación comprimida interna. Al hacerlo, el modelo filtra pequeñas incoherencias y patrones de ruido presentes en los datos. Esto permite obtener una versión reconstruida más estructurada.

El modelo se entrena durante un número fijo de épocas y su capacidad para reconstruir correctamente los datos se mide mediante un error. Este error calcula la diferencia promedio entre los valores originales y los reconstruidos. Aunque el objetivo principal del autoencoder no es minimizar un error, en términos estadísticos tradicionales como en un modelo predictivo, este valor sirve como indicativo de lo bien que ha aprendido a representar el conjunto de datos.

Una vez completada la normalización, las columnas reconstruidas por el autoencoder se combinan con el resto de las columnas originales (aquellas que no debían ser normalizadas). Además, todas las variables categóricas que fueron codificadas previamente son restauradas a su forma original usando los *diccionarios de mapeo* generados en la etapa de limpieza. De este modo, se preserva la semántica de los datos y se garantiza su compatibilidad con etapas posteriores.

El sistema incluye validaciones adicionales que evitan aplicar la normalización si el conjunto de datos resultante tras la limpieza no contiene suficientes columnas numéricas o filas. También contempla posibles errores durante el entrenamiento del autoencoder, en cuyo caso se omite la normalización y se conserva el conjunto imputado sin modificaciones.

Paso 5. Selección de características

La selección de características tiene como objetivo identificar qué columnas del conjunto de datos son realmente relevantes para la predicción de la variable objetivo. Esto permite reducir la complejidad del modelo, mejorar su interpretabilidad y, en muchos casos, incrementar su rendimiento.

En este paso, el sistema parte del conjunto de datos ya imputado y normalizado. Dado que en pasos anteriores las variables categóricas fueron restauradas a su forma original (por ejemplo, cadenas como `yes`, `no`, `A`, `B`), para esta etapa se aplica una codificación automática mediante one-hot encoding. Esta técnica convierte cada categoría en una columna binaria, que permite que algoritmos basados en árboles puedan interpretarlas correctamente.

La codificación se realiza únicamente para esta fase y no altera la estructura del conjunto original. Además, tras la codificación, se agrupan todas las columnas derivadas de una misma variable categórica para calcular una única medida de importancia por variable, utilizando como criterio la media de sus contribuciones individuales.

Para estimar la importancia de cada característica se entrenan dos algoritmos de aprendizaje automático ampliamente contrastados: Random Forest y Extra Trees (Extremely Randomized Trees). Random Forest se basa en un conjunto de árboles de decisión entrenados sobre subconjuntos aleatorios de datos y atributos, proporcionando una medida de importancia para cada variable. Por su parte, Extra Trees introduce mayor aleatoriedad en el proceso, lo que resulta útil para contrastar y validar la relevancia de las variables en distintos contextos.

Una vez calculadas las puntuaciones de importancia de las variables (tanto nu-

méricas como categóricas agrupadas), se aplica un umbral de corte de 0,02. Las variables cuya importancia media sea inferior a este umbral se descartan del conjunto de entrenamiento final. Este valor se ha definido empíricamente como un equilibrio entre eliminar ruido y conservar información potencialmente relevante.

Paso 6. Evaluación automática del modelo

Con las características ya seleccionadas, el sistema se prepara para lanzar el entrenamiento automático mediante H2O AutoML. Previamente, se comprueba si existe una sesión activa de H2O en ejecución; en caso afirmativo, esta se cierra de forma controlada para evitar conflictos. Si no hay ninguna instancia activa, se inicia una nueva automáticamente.

A continuación, se decide qué conjunto de características utilizar para entrenar los modelos, eligiendo entre las variables seleccionadas por Random Forest y por Extra Trees. Esta decisión depende de tres criterios: si solo un conjunto supera el umbral de importancia, se usa exclusivamente ese conjunto; si ambos conjuntos coinciden, se entrena una única vez; y si ambos conjuntos son distintos pero válidos, se entrenan ambos modelos y se elige el que tenga mejor rendimiento. Este mecanismo aprovecha las fortalezas de ambos algoritmos de selección sin requerir intervención manual. El conjunto final se pasa como entrada al módulo de AutoML, junto con la variable objetivo.

H2O AutoML automatiza la construcción de modelos mediante los siguientes pasos:

1. Preparación del conjunto de datos: se combina el conjunto de características con la variable objetivo, y se convierte en un formato compatible con H2O (`H2OFrame`).
2. Identificación del tipo de problema: el sistema detecta automáticamente si se trata de un problema de regresión, clasificación binaria o clasificación multiclase, en función de la variable objetivo.
3. Selección de métrica adecuada: se escoge una métrica principal según el tipo de problema: AUC para clasificación binaria, R^2 para regresión y error medio por clase para clasificación multiclase. Esta métrica guía tanto el entrenamiento como la evaluación de los modelos.
4. Entrenamiento de múltiples modelos: AutoML ejecuta durante un tiempo determinado un conjunto de algoritmos como Gradient Boosting Machines (GBM), XGBoost, Random Forest, redes neuronales (Deep Learning), modelos lineales (GLM) y combinaciones de ellos mediante ensamblados. Cada modelo se entrena con diferentes configuraciones internas para maximizar o minimizar la métrica seleccionada.
5. Evaluación personalizada de cada modelo: tras el entrenamiento, se evalúa manualmente cada modelo recuperado por AutoML utilizando la métrica principal.

Esta evaluación se realiza directamente con el método `model_performance()`, y los valores de métricas como `RMSE`, `MAE`, R^2, `logloss`, `AUC` o `error medio por clase` se obtienen específicamente para cada modelo, incluso si AutoML no las calcula por defecto en su *leaderboard*.

6. Selección del mejor modelo según la métrica priorizada: el sistema compara todos los modelos generados y selecciona como "ganador" aquel que obtiene el mejor resultado en la métrica principal (el más alto en el caso de AUC o R^2, o el más bajo si se trata de error). Este modelo se utiliza también como referencia para decidir entre conjuntos de características seleccionadas previamente.

7. Transformación del resultado a una escala común: para facilitar la interpretación, la puntuación obtenida por el mejor modelo se transforma a una escala de 0 a 10, independientemente de la métrica original. En métricas de error (como el error medio por clase), esta transformación invierte el valor, de forma que un mayor *score* represente siempre un mejor rendimiento.

Este enfoque permite evaluar múltiples modelos sin intervención del usuario, y seleccionar automáticamente el más adecuado para los datos y tipo de problema, lo que facilita su uso por personas sin conocimientos técnicos avanzados.

Cuando se han generado dos conjuntos de características (uno por Random Forest y otro por Extra Trees), el sistema realiza dos evaluaciones independientes. A continuación, compara los resultados obtenidos mediante una única métrica de rendimiento, seleccionada automáticamente en función de la naturaleza del problema. De este modo, se identifica el modelo que logra la mejor puntuación y se adopta como el resultado final del proceso.

La métrica empleada se determina automáticamente según la naturaleza de la variable objetivo, y se transforma posteriormente a una escala común de 0 a 10 para facilitar la interpretación. Para problemas de clasificación binaria se utiliza el AUC (Área bajo la curva ROC), una métrica robusta que evalúa la capacidad del modelo para diferenciar correctamente entre dos clases, siendo especialmente útil cuando existe desbalance de clases. En regresión, la métrica elegida es el R^2 (Coeficiente de determinación), que mide la proporción de varianza explicada y ofrece un criterio claro sobre la calidad del ajuste. Para problemas de clasificación multiclase se emplea el error medio por clase, un indicador que evalúa el desempeño de forma equilibrada en todas las clases, evitando que un buen rendimiento en las clases más representadas oculte carencias en las menos frecuentes.

La elección de una única métrica por tipo de problema permite establecer una comparación directa y coherente entre modelos. Considerar múltiples métricas simultáneamente podría generar contradicciones o dificultar la interpretación. Cada métrica seleccionada refleja el objetivo fundamental de su tarea: maximizar la discriminación, minimizar el error o equilibrar el rendimiento entre clases.

Independientemente de si se realiza una o dos evaluaciones, el sistema siempre registra la información del modelo con mejor desempeño, incluyendo su tipo, configuración, métrica utilizada y puntuación final en escala 0–10. Esta información se

presenta como parte del resultado del preprocesamiento, sin necesidad de conocimientos técnicos avanzados ni almacenamiento persistente del modelo.

Cabe destacar que el modelo entrenado no se guarda como archivo, sino que el sistema simplemente muestra sus principales características: qué tipo de modelo fue el mejor y qué rendimiento obtuvo. De este modo, el usuario puede conocer el potencial predictivo de sus datos sin necesidad de conocimientos técnicos avanzados.

Paso 7. Generación del conjunto de datos final

Una vez completado el proceso de evaluación automática, el sistema identifica cuál de los dos subconjuntos de características (el obtenido mediante *Random Forest* o *Extra Trees*) ha permitido entrenar el modelo con mejor rendimiento, de acuerdo con la métrica seleccionada automáticamente según el tipo de problema. Este subconjunto se considera el más adecuado para representar la información del conjunto de datos, y sobre él se genera el dataset final optimizado.

Con base en esta selección, se genera un nuevo conjunto de datos que contiene únicamente las columnas seleccionadas (ya imputadas y normalizadas) junto con la variable objetivo. Este conjunto representa una versión depurada y optimizada del archivo original, adecuada para ser usada directamente en tareas de modelado predictivo.

El sistema guarda automáticamente este conjunto en un archivo CSV denominado `modelo_final_preprocesado.csv`, ubicado en el directorio de usuario del sistema operativo. De esta forma, el usuario dispone de una copia local del dataset preprocesado sin necesidad de realizar configuraciones adicionales ni conocer detalles técnicos del flujo aplicado.

Este paso final asegura que todo el conocimiento adquirido durante el proceso de preprocesamiento—desde la limpieza y normalización hasta la selección de características y evaluación de modelos—quede reflejado en un archivo concreto, listo para su uso en contextos reales de análisis o predicción.

3.3. Diseño e implementación del sistema web

Tras validar las técnicas seleccionadas mediante un prototipo inicial ejecutable por terminal, el siguiente paso consistió en transformar dicha solución en un sistema accesible desde la web, orientado a usuarios finales sin conocimientos técnicos. Para ello, se diseñó e implementó una arquitectura cliente-servidor que permite ejecutar el flujo completo de preprocesamiento de forma remota, mediante una interfaz visual interactiva y una API robusta en el backend.

Este sistema permite a los usuarios cargar sus propios archivos CSV, configurar aspectos clave del preprocesamiento y visualizar los resultados del análisis sin necesidad de instalar dependencias o interactuar con código Python directamente. La interfaz gráfica simplifica la interacción y facilita el uso de técnicas complejas

mediante automatización e interacción guiada.

3.3.1. Arquitectura general del sistema

El diseño del sistema de preprocesamiento inteligente de datos se basa en una arquitectura de tipo cliente-servidor, que garantiza una separación clara de responsabilidades entre los componentes visuales (frontend) y funcionales (backend). Esta arquitectura proporciona una experiencia de usuario fluida y una gestión eficiente del procesamiento de datos, combinando simplicidad en la interacción con robustez técnica en el análisis.

El sistema está compuesto por dos módulos principales. El frontend está desarrollado con Angular y proporciona una interfaz web adaptable e intuitiva. A través de ella, los usuarios pueden cargar archivos CSV, seleccionar la columna objetivo, visualizar en tiempo real el progreso de cada etapa del flujo y acceder a los resultados del análisis. Esta interfaz ha sido diseñada para ofrecer mensajes claros y guías visuales que simplifican el uso del sistema, incluso para personas sin experiencia técnica.

Por otro lado, el backend está implementado en Flask (Python) y se encarga de ejecutar el flujo completo de preprocesamiento, que incluye la imputación de valores faltantes, la normalización con autoencoder, la selección de características y el entrenamiento automático con AutoML. Este módulo expone una API RESTful que permite al frontend iniciar, controlar y recibir los resultados del procesamiento de manera estructurada y eficiente.

La comunicación entre el frontend y el backend se realiza mediante solicitudes HTTP REST (protocolo que define cómo se intercambian datos a través de la web usando métodos como `GET` o `POST`). Además, se emplea la tecnología SSE (Server-Sent Events), que permite al servidor enviar mensajes al cliente en tiempo real, sin necesidad de que la página se actualice manualmente. Gracias a esto, el usuario puede visualizar el avance del proceso de forma continua, con información dinámica sobre el estado de cada etapa del flujo.

Para ilustrar de manera clara el diseño del sistema, en la Figura 7 se presenta un esquema de la arquitectura del sistema, que muestra la interacción entre el frontend, el backend y el flujo de datos.

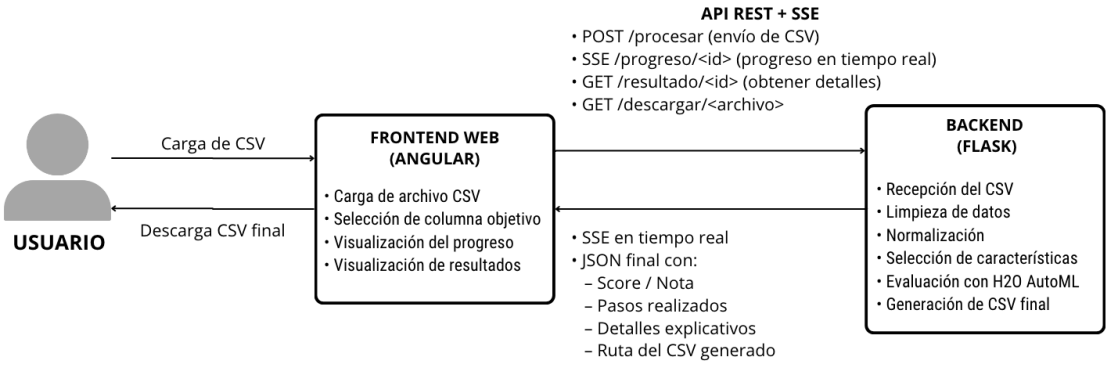


Figura 7: Arquitectura general del sistema.

La elección de tecnologías se fundamenta en su idoneidad para las necesidades del proyecto. Angular permite construir aplicaciones web dinámicas con una estructura modular y un mantenimiento claro. Su versatilidad lo convierte en una herramienta ideal para implementar formularios interactivos, realizar seguimiento en tiempo real y ofrecer validaciones personalizadas.

Por su parte, Flask ofrece una solución ligera y flexible para desarrollar APIs en Python. Su simplicidad lo convierte en una opción óptima para transformar rápidamente un script de análisis existente en un servicio web funcional, como fue el caso del prototipo inicial implementado directamente en Python. Además, Flask se integra de forma nativa con bibliotecas científicas y de aprendizaje automático como `scikit-learn`, `pandas` o `h2o`, lo que facilita la reutilización del código ya desarrollado.

Por último, la adopción de REST como estándar para la construcción de servicios web garantiza la interoperabilidad del sistema, permitiendo futuras extensiones.

En conjunto, esta arquitectura permite a usuarios sin experiencia técnica ejecutar un flujo de preprocesamiento complejo desde un navegador web, delegando todas las operaciones técnicas al backend sin necesidad de configurar entornos locales o comprender el funcionamiento interno de los algoritmos utilizados.

3.3.2. Frontend: diseño e implementación con Angular

El frontend del sistema ha sido desarrollado con Angular, un framework moderno ampliamente adoptado para la construcción de aplicaciones web dinámicas, modulares y escalables. Su arquitectura basada en componentes, el sistema de rutas y su integración con servicios asíncronos lo hacen especialmente adecuado para desarrollar interfaces interactivas y reactivas.

En este caso, la interfaz web es asíncrona porque permite que el usuario continúe interactuando con la aplicación mientras el backend realiza el procesamiento de datos. A través de la tecnología Server-Sent Events (SSE), el navegador puede recibir actualizaciones en tiempo real directamente desde el servidor, mostrando el progreso

de cada etapa (limpieza, normalización, selección de características, evaluación automática) sin necesidad de recargar la página ni esperar a que todo el flujo finalice. Esto contrasta con un enfoque síncrono, donde el usuario tendría que esperar a que todos los pasos se completen antes de recibir algún resultado.

Desde el inicio, la interfaz se diseñó para ser accesible incluso para personas sin conocimientos técnicos, combinando claridad visual, textos explicativos y un flujo guiado paso a paso. Para lograr una experiencia moderna y atractiva, se partió de la plantilla *Start Bootstrap New Age* [2], adaptándola y extendiéndola para incluir funcionalidades específicas orientadas a la interacción con un sistema de preprocesamiento de datos basado en IA.

El diseño visual se modificó para incorporar elementos que facilitaran el uso por parte de usuarios no expertos, como: mensajes orientativos, componentes explicativos, botones bien diferenciados y un flujo lógico centrado en el archivo CSV como entrada principal.

Las Figuras 8 y 9 muestra la sección de bienvenida y características del sistema, donde se presenta al usuario una descripción clara de lo que el sistema puede hacer, resaltando su orientación práctica y automatizada.

Figura 8: Pantalla inicial del sistema.

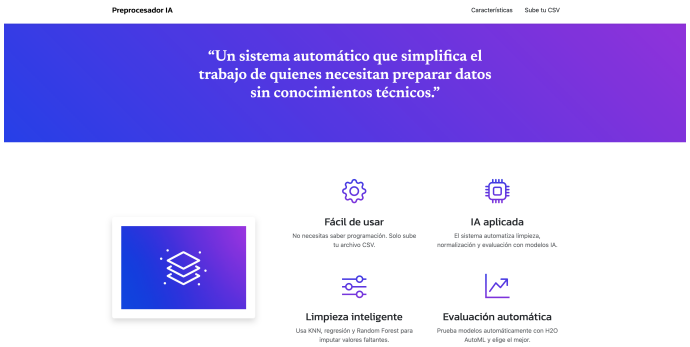


Figura 9: Visualización de las características del sistema.

[2]https://startbootstrap.com/theme/new-age

La interfaz incluye todas las funcionalidades clave necesarias para completar un flujo de preprocesamiento automático desde cero:

- Carga de archivos CSV: el usuario puede subir un archivo CSV directamente desde su dispositivo. Esta acción activa automáticamente el flujo de preprocesamiento (Figura 10).

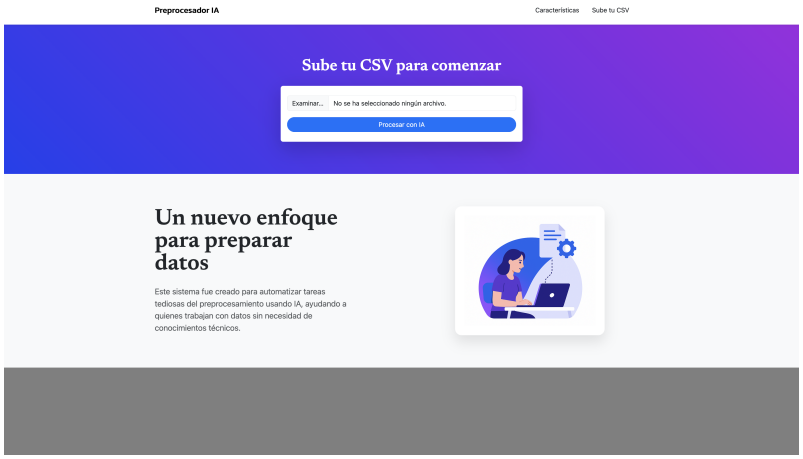


Figura 10: Interfaz donde el usuario puede subir su archivo CSV para iniciar el preprocesamiento automático.

- Selección de columna objetivo: tras cargar el archivo, el sistema analiza las columnas disponibles e intenta sugerir automáticamente una como posible variable objetivo. Independientemente de si esta sugerencia es precisa, siempre se muestra al usuario un selector que le permite confirmar o elegir libremente otra columna.

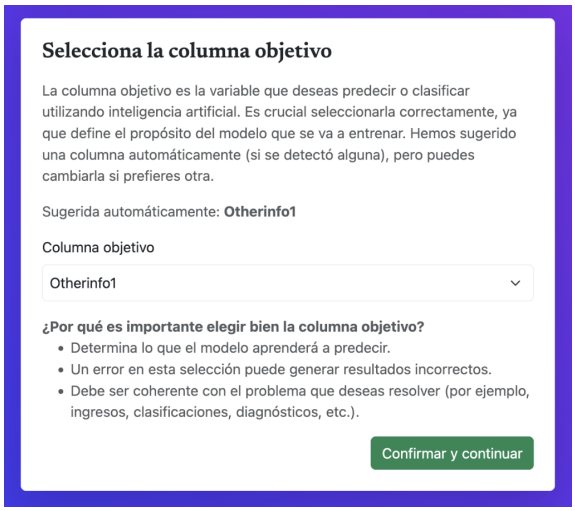


Figura 11: Selección de la columna objetivo. La interfaz ofrece una sugerencia automática e incluye una explicación accesible sobre su importancia.

La interfaz incluye además un bloque explicativo que describe claramente qué es la columna objetivo y por qué su correcta selección es crucial para el entrenamiento del modelo de IA. Esta ayuda textual está especialmente pensada para guiar a usuarios sin conocimientos técnicos, facilitando la toma de decisiones informadas (véase la Figura 11).

- Visualización del progreso en tiempo real: a medida que se avanza en el proceso de preprocesamiento, el sistema proporciona retroalimentación continua mediante una lista de pasos completados, un mensaje central de estado y una barra de progreso animada que se activa durante la etapa de evaluación automática de modelos con AutoML.

  Esta retroalimentación no solo mejora la experiencia de usuario, sino que también permite entender qué fases se están ejecutando internamente (limpieza, normalización, selección de características, evaluación). En la Figura 12 se observa un ejemplo de esta visualización, donde el sistema ha completado varias etapas y se encuentra en la evaluación automática, mostrando un porcentaje actualizado del avance.

Figura 12: Visualización del progreso del preprocesamiento. Se listan los pasos completados y se indica el avance del AutoML.

- Visualización de resultados: una vez finalizado el proceso de preprocesamiento, se presenta al usuario un resumen claro del rendimiento obtenido. Este incluye una puntuación final (score) sobre 10, calculada automáticamente en función de la métrica más adecuada para el tipo de problema (AUC, R^2, error medio por clase, etc.).

  Junto a la puntuación se ofrecen tres acciones accesibles: descargar el CSV preprocesado, procesar otro archivo o ver más detalles. La Figura 13 muestra la interfaz de resumen al finalizar el procesamiento, resaltando la nota obtenida y las opciones disponibles para el usuario.

Figura 13: Resumen de resultados tras el preprocesamiento. Se muestra la puntuación obtenida, opciones de descarga y acceso a detalles explicativos.

- Explicaciones accesibles y elementos interactivos: el sistema ha sido diseñado con un enfoque centrado en la comprensión por parte de usuarios no expertos. Por ello, al seleccionar la opción de "ver más detalles" en la interfaz (Figura 13), se accede a un primer modal que resume de forma clara y estructurada las etapas del preprocesamiento aplicadas al conjunto de datos (Figura 14).

  Este primer nivel de detalle incluye descripciones comprensibles de los algoritmos utilizados, como KNN, Random Forest y Autoencoder. También se ofrece una descripción del umbral de importancia utilizado en la selección de características (fijado en 0,02) y una gráfica que representa las cinco características más relevantes identificadas por los algoritmos.

  Adicionalmente, desde este primer modal se puede acceder opcionalmente a un segundo nivel de información, más técnico, donde se detallan los modelos probados por AutoML, sus métricas individuales (como AUC, RMSE, logloss, etc.). Esta información se presenta en formato de tabla, con el objetivo de ofrecer transparencia en la selección del modelo final (Figuras 15).

  En conjunto, estos elementos interactivos buscan facilitar la comprensión del proceso completo de análisis sin requerir conocimientos previos en estadística, programación o aprendizaje automático.

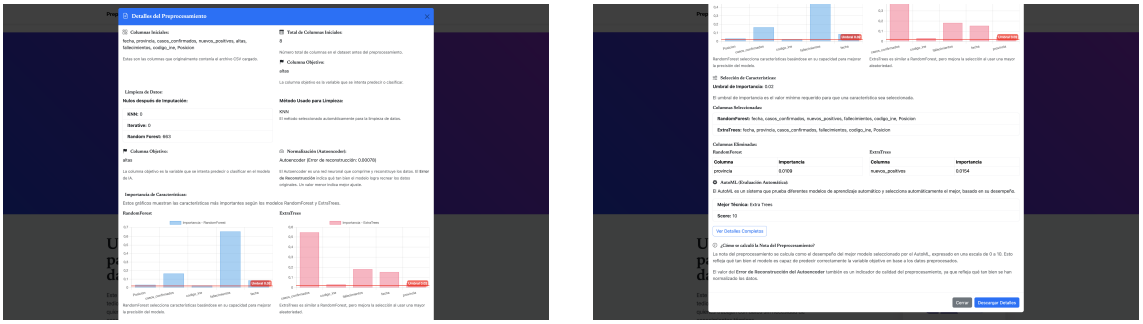


Figura 14: Pantalla de detalles del preprocesamiento.

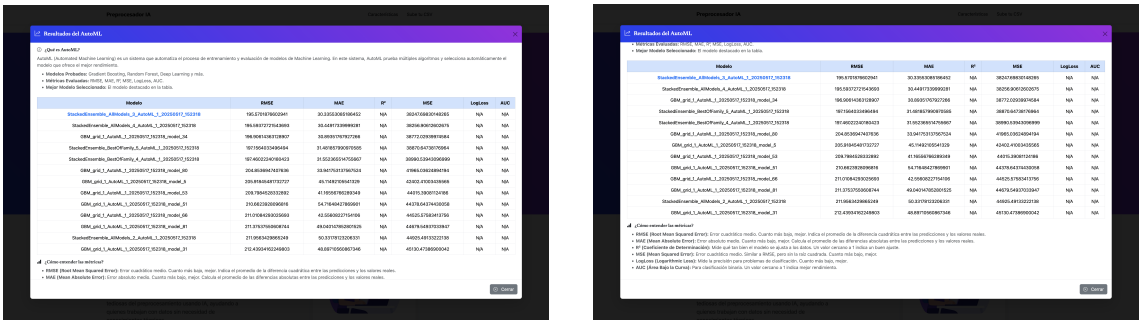


Figura 15: Pantalla de evaluación automática (AutoML) mostrando el mejor modelo.

Desde el inicio, la interfaz se diseñó para ser accesible incluso para personas sin conocimientos técnicos, combinando claridad visual, textos explicativos y un flujo guiado paso a paso. La estructura general de la aplicación se compone de un único componente principal (`AppComponent`), que centraliza el estado de la interfaz, la carga del archivo CSV y el control del flujo de procesamiento.

Para comunicarse con el backend, el sistema utiliza un servicio personalizado denominado `ApiService`, que encapsula todas las llamadas a la API REST y gestiona eventos en tiempo real mediante Server-Sent Events (SSE). Entre las principales acciones que permite destacan:

- Envío de archivos (`POST /procesar`),
- Seguimiento del progreso mediante eventos SSE (`GET /progreso/{id}`),
- Obtención de resultados al finalizar el procesamiento (`GET /resultado/{id}`),
- Descarga del archivo final procesado (`GET /descargar/{nombre}`).

Este diseño basado en un servicio centralizado permite separar claramente la lógica de comunicación del resto de la aplicación, lo que mejora la organización del código, facilita las pruebas y permite futuras extensiones de forma más sencilla.

En conjunto, el frontend fue diseñado para guiar al usuario paso a paso, brindando información contextual, retroalimentación visual y opciones claras ante eventos excepcionales o errores, sin exponer detalles técnicos innecesarios. Esta estrategia de diseño fue clave para cumplir con el objetivo principal del sistema: facilitar el preprocesamiento automatizado de datos para usuarios sin experiencia en ciencia de datos.

3.3.3. Backend: implementación con Flask

El backend del sistema se desarrolló con Flask, un microframework de Python que facilita la creación de aplicaciones web y APIs REST. Esta elección permitió integrar el prototipo inicial, ya implementado en Python, directamente en la arquitectura web sin necesidad de reescribir toda la lógica de procesamiento.

El backend hereda la estructura modular del prototipo, en el que cada etapa del preprocesamiento está organizada en funciones específicas: carga y validación del CSV, imputación de valores, normalización, selección de características y evaluación automática con H2O AutoML. Estas funciones se reutilizan dentro del servidor Flask, encapsuladas en una función central `procesar_csv_con_preprocesamiento()`, que orquesta todo el flujo según los parámetros recibidos desde el frontend.

El backend expone una API REST compuesta por distintos endpoints, que permiten controlar el sistema desde el cliente Angular.

Cada sesión de usuario se gestiona de forma aislada mediante un `session_id`, que permite asociar un archivo, una cola de eventos y una estructura de resultados a cada proceso de preprocesamiento iniciado. Esto facilita que el sistema pueda pausar su ejecución en puntos clave (como la selección de columna objetivo) y reanudarla posteriormente sin perder contexto.

El backend guarda de forma temporal tanto el CSV subido como el estado intermedio de procesamiento. Además, cada resultado se estructura en un diccionario con detalles organizados que incluyen puntuaciones, técnicas utilizadas y columnas seleccionadas, facilitando su recuperación posterior para el frontend.

Durante la ejecución del flujo, se han implementado múltiples mecanismos de control de errores:

- Si no se seleccionan columnas relevantes en la etapa de selección de características, se informa al cliente y se continua con todas las columnas.
- Si el modelo AutoML no puede entrenarse por falta de clases distintas, se captura el error y se devuelve una explicación accesible al usuario.

Además, toda la comunicación sobre el estado del proceso se realiza a través de eventos SSE enviados al frontend, permitiendo una visualización del progreso en tiempo real.

En resumen, el backend no solo ejecuta el preprocesamiento técnico, sino que actúa como coordinador de la experiencia completa del usuario, garantizando continuidad entre pasos, controlando excepciones y ofreciendo una interfaz de programación clara y robusta. Además, utiliza Server-Sent Events (SSE) para enviar actualizaciones de estado al frontend en tiempo real, mejorando la experiencia de usuario con retroalimentación continua y sin bloqueos.

3.3.4. Comunicación frontend-backend

El sistema implementa una arquitectura de tipo cliente-servidor, en la que el frontend, se comunica con el backend, a través de una combinación de tecnologías: API RESTful y Server-Sent Events (SSE). Esta estructura permite separar claramente la presentación visual del procesamiento funcional, y asegura una experiencia de usuario fluida, en tiempo real y escalable.

La API RESTful define un conjunto de convenciones para el intercambio de información entre aplicaciones mediante solicitudes HTTP. En este sistema, el frontend realiza llamadas a rutas específicas del backend utilizando métodos como `POST` (para enviar archivos y datos) y `GET` (para consultar resultados), y recibe respuestas en formato estructurado `JSON`. Esta interfaz es stateless, lo que significa que cada solicitud es independiente, simplificando la gestión de sesiones y mejorando la escalabilidad del sistema.

Por su parte, la tecnología SSE permite establecer una conexión unidireccional persistente desde el servidor al cliente, que se mantiene abierta durante toda la sesión. A diferencia de un enfoque síncrono (en el que el cliente espera la respuesta completa antes de continuar), los SSE permiten que el servidor emita actualizaciones de forma continua en cuanto hay nueva información. Esto reduce la latencia y mejora la experiencia de usuario, ofreciendo una interfaz interactiva y reactiva.

A continuación, se resumen los endpoints más relevantes que ofrece el backend:

- `POST /procesar`: el usuario envía el archivo CSV y el backend inicia el flujo de análisis asignando un identificador único de sesión (`session_id`) y lanza el procesamiento en un hilo independiente. Durante este proceso, se generan mensajes (como "Limpieza completada" o "AutoML progress: 75 %") que se almacenan en una cola de eventos vinculada a la sesión.
- `GET /progreso/{session_id}`: mantiene una conexión persistente mediante SSE y transmite los mensajes de la cola de eventos en tiempo real al navegador. En el frontend, estos eventos se reciben mediante un canal EventSource que permanece activo hasta la finalización o error del procesamiento.
- `GET /resultado/{session_id}`: recupera el resultado final del preprocesamiento, incluyendo la puntuación obtenida, el nombre del archivo generado y los detalles técnicos del flujo.
- `GET /descargar/{nombre_archivo}`: permite al usuario descargar el archivo CSV procesado que contiene las columnas seleccionadas y los datos normalizados.
- `POST /cerrar_sesion`: elimina los archivos temporales y los resultados asociados a la sesión, liberando recursos.

En el **frontend**, esta comunicación se gestiona mediante un servicio dedicado (`ApiService`) que envía las peticiones `POST` y `GET` y mantiene abierto un canal `EventSource` para recibir actualizaciones en tiempo real mediante SSE, evitando la necesidad de recargar la página y ofreciendo una experiencia fluida e interactiva.

Aunque Flask gestiona las solicitudes de forma síncrona, la ejecución de cada flujo de análisis se lanza en hilos independientes, lo que permite gestionar múltiples usuarios y sesiones simultáneamente. Esta arquitectura modular y reactiva asegura que el usuario pueda seguir el avance del preprocesamiento en tiempo real, recibir

retroalimentación inmediata y, en caso de problemas, entender rápidamente qué ocurre sin necesidad de intervenciones técnicas.

Para facilitar la comunicación entre ambos componentes, especialmente en el entorno de desarrollo donde pueden ejecutarse en puertos distintos, se habilitó la política de CORS (Cross-Origin Resource Sharing). Esta política permite que un cliente web en un dominio (o puerto) pueda acceder a recursos ofrecidos por un servidor ubicado en otro dominio, eliminando restricciones de seguridad por defecto en los navegadores. Así, se garantiza un flujo de trabajo transparente, consistente y sin bloqueos innecesarios.

En conjunto, este diseño modular y reactivo ofrece a los usuarios una experiencia de preprocesamiento clara, accesible y técnicamente robusta, eliminando barreras para perfiles no expertos y permitiendo un uso confiable y seguro de la herramienta.

3.4. Despliegue del sistema

Durante las fases iniciales del desarrollo, tanto el frontend como el backend fueron ejecutados en entornos locales separados, lo que permitió una validación modular y un control detallado del comportamiento del sistema.

Entorno de desarrollo local

El sistema se compone de dos servidores que operan de forma independiente:

- Frontend (Angular): ejecutado en el puerto `4200` mediante el comando `ng serve`. Esta interfaz permite interactuar en tiempo real con el usuario, mostrando el flujo de carga, selección de columna objetivo, progreso del procesamiento y resultados finales.
- Backend (Flask): ejecutado en el puerto `5000` a través del comando `python app.py`. Expone todos los endpoints REST necesarios y mantiene las conexiones SSE con el cliente.

Este entorno local fue fundamental para probar la lógica de preprocesamiento, detectar errores, y validar el comportamiento de eventos en tiempo real, así como para comprobar que la gestión del estado por sesión funcionaba correctamente.

Despliegue en producción

Una vez verificada la funcionalidad en el entorno local, el sistema fue desplegado en un servidor de producción accesible desde Internet bajo el dominio:

`https://smartcsv.grial.eu`

Para este despliegue en producción, se implementó un esquema robusto y escalable compuesto por:

- Frontend (Angular): El cliente web fue compilado utilizando el comando `ng build -prod`, generando los archivos estáticos necesarios. Estos se sirven directamente mediante el servidor web Nginx, que actúa como proxy inverso, mejorando el rendimiento y manejando las solicitudes concurrentes de manera eficiente.
- Backend (Flask): En el entorno de producción, la aplicación Flask no se ejecuta directamente, sino que se gestiona a través del servidor WSGI Gunicorn. Este componente permite servir la aplicación en múltiples procesos simultáneamente, optimizando la capacidad de respuesta y garantizando la estabilidad. Gunicorn, a su vez, se encuentra detrás de Nginx, que gestiona el enrutamiento de las peticiones y la distribución de la carga.

Este esquema de despliegue asegura la disponibilidad y la escalabilidad del sistema, facilitando el acceso para usuarios finales sin requerir entornos locales y garantizando una experiencia fluida y consistente, incluso bajo carga concurrente.

3.5. Evaluación del sistema

Para validar la utilidad, comprensión y robustez del sistema propuesto, se realizaron dos tipos de evaluaciones complementarias. Primero, se realizaron pruebas técnicas con distintos conjuntos de datos, para confirmar la versatilidad y estabilidad del flujo de preprocesamiento automático en escenarios variados. Después, se llevó a cabo una evaluación con usuarios reales para comprobar la usabilidad, accesibilidad y confianza generada

3.5.1. Pruebas realizadas con distintos conjuntos de datos

Además de la validación con usuarios, se realizaron pruebas técnicas con diversos conjuntos de datos de diferentes dominios, para evaluar la consistencia y versatilidad del sistema. A continuación se presentan algunos de los casos de prueba utilizados:

1. Consultas atendidas en Oficinas de Información Turística de la Red del Gobierno de Navarra[3]

- Tamaño: 1.7 MB.
- Columna objetivo: `PROVINCIA`.

[3]https://datos.gob.es/es/catalogo/a15002917-consultas-atendidas-en-oficinas-de-informacion-turistica-de-la-red-del-gobierno-de-navarra1: Este dataset contiene registros de consultas realizadas en oficinas turísticas de Navarra, incluyendo país, provincia y demandas.

- Sin valores nulos iniciales.
- Columnas iniciales (9): `_id`, `MES`, `NOMBRE_OIT`, `PAIS`, `CCAA`, `PROVINCIA`, `ZONIFICACION_UGET`, `N_DEMANDAS`, `N_SOLICITUDES`.
- Columnas seleccionadas tras evaluación:
  - Random Forest: `_id`, `N_DEMANDAS`, `N_SOLICITUDES`, `CCAA`.
  - Extra Trees: `_id`, `N_DEMANDAS`, `N_SOLICITUDES`, `CCAA`.
- Tipo de problema: clasificación multiclase.
- Mejor modelo tras evaluación: Random Forest y Extra Trees (idénticos).
- Métrica: error medio por clase.
- Score final obtenido: 7/10.

2. Titanic Dataset (Kaggle)[4]

- Tamaño: 61 KB.
- Columna objetivo: `Survived`.
- Valores nulos iniciales: 866.
- Nulos después de imputación:
  - KNN: 0
  - Regresión Iterativa: 0
  - Random Forest: 866
- Columnas iniciales: `PassengerId`, `Survived`, `Pclass`, `Name`, `Sex`, `Age`, `SibSp`, `Parch`, `Ticket`, `Fare`, `Cabin`, `Embarked`.
- Columnas seleccionadas:
  - Random Forest: `PassengerId`, `Pclass`, `Age`, `SibSp`, `Fare`, `Sex`.
  - Extra Trees: `Pclass`, `Age`, `Fare`, `Sex`.
- Tipo de problema: clasificación binaria.
- Mejor modelo tras evaluación: Random Forest.
- Métrica: AUC.
- Score final obtenido: 10/10.

[4]https://www.kaggle.com/competitions/titanic/data: Datos de pasajeros del Titanic para predecir la supervivencia, con variables como edad, clase o tarifa.

3. Early Stage Diabetes Risk Prediction Dataset[5]

- Tamaño: 34 KB.
- Columna objetivo: `class`.
- Sin valores nulos iniciales.
- Columnas iniciales: `Age`, `Gender`, `Polyuria`, `Polydipsia`,`sudden weight loss` , `weakness`, `Polyphagia`, `Genital thrush`, `visual blurring`, `Itching`, `Irritability`, `delayed healing`, `partial paresis`, `muscle stiffness`, `Alopecia`, `Obesity`,`class`.
- Columnas seleccionadas:
  - Random Forest: Age, Gender, Polyuria, Polydipsia, Genital thrush, Irritability, delayed healing, muscle stiffness, Alopecia.
  - Extra Trees: Age, Gender, Polyuria, Polydipsia, Genital thrush, Itching, Irritability, delayed healing, muscle stiffness, Alopecia.
- Tipo de problema: clasificación binaria.
- Mejor modelo tras evaluación: Extra Trees.
- Métrica: AUC.
- Score final obtenido: 10/10.

4. Situación epidemiológica COVID-19 en Castilla y León [6]

- Tamaño: 697 KB.
- Columna objetivo: `altas`.
- Valores nulos iniciales: 663.
- Nulos después de imputación:
  - KNN: 0
  - Regresión Iterativa: 0
  - Random Forest: 663
- Columnas iniciales: `fecha`, `provincia`, `casos_confirmados`, `nuevos_positivos`, `altas`, `fallecimientos`, `codigo_ine`, `Posicion`.

[5]https://archive.ics.uci.edu/dataset/529/early+stage+diabetes+risk+prediction+dataset: Datos de síntomas y factores de riesgo para predecir la diabetes en etapas tempranas.

[6]https://datosabiertos.jcyl.es/web/jcyl/set/es/salud/situacion-epidemiologica-coronavirus/1284940407131. Contiene datos diarios de casos, altas y fallecimientos por provincia en Castilla y León.

- Columnas seleccionadas tras la evaluación: `casos_confirmados`, `fallecimientos`.
- Tipo de problema: regresión.
- Métrica: R^2.
- Score final obtenido: 10/10.

Estas pruebas confirman la capacidad del sistema para adaptarse a diferentes tamaños y tipos de datos, validando su robustez y consistencia en flujos de preprocesamiento automatizado. Además, ilustran cómo la naturaleza de la variable objetivo determina el tipo de problema:

- Clasificación binaria cuando la variable objetivo tiene sólo dos valores posibles (por ejemplo, `Survived` o `class` en los casos de Titanic y Diabetes).
- Clasificación multiclase cuando la variable objetivo tiene más de dos categorías discretas (por ejemplo, `PROVINCIA` en consultas turísticas), son etiquetas categóricas.
- Regresión cuando la variable objetivo representa valores continuos y numéricos, como en el caso de las `altas` diarias en los datos de COVID-19.

La correcta distinción de estos tipos de problemas es clave para elegir la métrica y el enfoque adecuados durante la fase de AutoML. En conjunto, estas pruebas muestran que el sistema es capaz de adaptarse a distintas características de los datos, ofreciendo resultados confiables y adaptados a las necesidades de cada análisis.

3.5.2. Evaluación con usuarios

Con el objetivo de validar la usabilidad, claridad y utilidad del sistema, se realizó una evaluación cualitativa mediante dos grupos focales, en la que participaron cinco usuarios en total. El primer grupo focal se realizó con dos residentes clínicos sin experiencia técnica en ciencia de datos, pero con conocimiento del dominio sanitario. El segundo grupo focal se realizó con usuarios expertos en análisis de datos en el contexto sanitario.

Si bien el sistema está orientado a usuarios no expertos en análisis de datos, la evaluación con usuarios expertos permite validar el flujo de preprocesamiento implementando de forma interna. Mientras que, la evaluación con usuarios no expertos permite validar la usabilidad y la satisfacción con el sistema.

La primera parte del grupo focal se centró en que utilizaran el sistema completo, desde la carga de un archivo CSV hasta la obtención del resultado final, y que comentaran su experiencia, nivel de comprensión y sugerencias de mejora.

Durante la evaluación, todos los participantes respondieron a las siguientes preguntas, orientadas a recoger impresiones clave:

1. ¿Te genera confianza que el sistema tome decisiones automáticamente por ti durante el preprocesamiento de datos?
2. ¿Preferirías tener la opción de tomar tú mismo las decisiones en cada paso del preprocesamiento?
3. ¿Comprendes claramente qué ha hecho el sistema con tus datos durante el preprocesamiento?
4. ¿Sientes que falta información importante en el informe de resultados que muestra el sistema?
5. ¿Echas en falta alguna funcionalidad o característica en el sistema?

A continuación se presenta un resumen con las principales respuestas agrupados por usuario en la Tabla 7.

Usuario	**Confianza en decisiones automáticas**	**Preferencia por decisiones manuales**	**Comprensión del preprocesamiento**	**Falta información en informe**	**Faltan funcionalidades o mejoras**
Residente 1	Valora positivamente que el sistema automatice decisiones, especialmente por su falta de conocimientos técnicos.	Le gustaría poder seleccionar algunas variables que si o si deberían estar.	Comprende bien la selección de características, pero no entiende términos como KNN o autoencoder.	La información parece completa, aunque echa en falta más información básica para poder entender los algoritmos.	Sugerencia de mostrar de forma más visual las columnas eliminadas y explicar mejor el funcionamiento de los algoritmos.
Residente 2	Agradece que el sistema lo haga todo por él, ya que no entiende los procesos internos.	Preferiría poder ampliar secciones para saber un resumen de lo que ocurre en cada etapa.	Identifica qué variables se han seleccionado, pero no entiende qué significa el umbral ni cómo se decide.	No comprende bien qué significa el score final ni cómo se interpreta.	Propone añadir explicaciones más claras sobre los modelos utilizados y permitir ver la evolución del proceso mientras se espera.
Experto 1	Necesita ver qué ha hecho el sistema para confiar.	Preferiría ver las decisiones y modificarlas si es necesario.	No comprende bien qué decisiones se han tomado con los datos, no las considera ilustradas.	No sabe muy bien como ha funcionado la normalización ni entre que rangos esta normalizando.	Avisar si la variable objetivo tiene nulos, explicar normalización.
Experto 2	Le parece correcto si se explican bien las fases.	Le da igual ver los pasos mientras entienda el resultado.	No entiende cómo se calcula el score ni por qué en la grafica solo se muestran las top 5 características.	No se explica bien el score ni el error de reconstrucción.	Explicar mejor que ocurrió en los pasos de preprocesamiento en el informe.
Experto 3	Confía si puede revisar lo que se ha hecho al final, por si lo necesita repetir manualmente.	Sí, mejor si puede avanzar paso a paso.	Quiere ver menos detalle de columnas eliminadas y más información sobre las seleccionadas y sus importancias.	Considera confusa la explicación del score final.	Quiere entender mejor cómo se calcula el score final y cómo se gestionan los valores faltantes (N/A) en los modelos evaluados.

Tabla 7: Resumen de respuestas a la evaluación con usuarios.

Este análisis permitió identificar tanto puntos fuertes del sistema —como su capacidad para automatizar tareas tediosas del preprocesamiento y su facilidad de uso— como áreas de mejora, principalmente en la comprensión del flujo interno, la transparencia de las decisiones automáticas y la claridad de los resultados presentados. Los principales hallazgos obtenidos pueden agruparse en varias categorías.

En primer lugar, destaca la confianza en las decisiones automáticas tomadas por el sistema. Aunque los usuarios valoraron positivamente esta automatización, especialmente los residentes sin formación técnica, todos coincidieron en que la confianza aumentaría si se explicaran mejor las decisiones adoptadas en cada fase del flujo. Los expertos, en particular, reclamaron mayor transparencia y control sobre el proceso. Además, sugirieron que el sistema debería comprobar explícitamente si la columna objetivo contiene valores nulos, en lugar de continuar el flujo sin advertir al usuario. Consideraban esencial detectar esta situación al inicio, dado que la variable objetivo es clave para el aprendizaje automático y el que no este completa esta columna puede comprometer la validez del modelo resultante.

Otro hallazgo relevante fue la visualización del flujo en tiempo real. Los participantes valoraron favorablemente que el sistema mostrara el progreso, pero propusieron mejorar esta funcionalidad para que, mientras se ejecutan procesos como el entrenamiento con AutoML, se pueda revisar de forma detallada qué pasos se han completado y cómo se han llevado a cabo. En particular, sugirieron que el sistema muestre explicaciones claras y visuales sobre el funcionamiento de cada fase —como la limpieza, la normalización, la selección de características y la construcción de modelos— indicando qué se hizo en cada paso, qué algoritmos o técnicas se aplicaron (por ejemplo, cómo funciona KNN o qué significa un umbral de selección), y cómo influyen estos en la preparación de los datos y el rendimiento final del modelo. Esta mejora permitiría aprovechar mejor el tiempo de espera y dar a los usuarios una comprensión más profunda y estructurada del flujo completo de preprocesamiento.

La comprensión del score final fue un aspecto que generó dudas. Los usuarios sugirieron añadir una explicación detallada del criterio de cálculo (según la métrica usada en clasificación o regresión) y contextualizarla con ejemplos, para entender por qué un resultado es, por ejemplo, 6.3 en lugar de 8.0.

Por último, demandan la necesidad de explicaciones más accesibles sobre algoritmos y transformaciones. Los usuarios no comprendieron qué significaban ciertos términos como KNN, Autoencoder o RandomForest y solicitaron conocer cómo se seleccionan las variables (criterio de umbral), qué significa el error de reconstrucción y en qué rango se realiza la normalización. Se propuso incluir explicaciones simplificadas con ejemplos visuales e iconos interactivos junto a cada paso, para facilitar la comprensión incluso a personas sin conocimientos técnicos.

Estos resultados fueron clave para priorizar las mejoras implementadas en la versión final del sistema, centradas en: incorporar iconos informativos junto a cada sección del modal de resultados, mostrar las columnas eliminadas con su puntuación de importancia y representar gráficamente las seleccionadas, añadir modales explicativos con lenguaje accesible sobre cada técnica usada (imputación, normalización, selección, evaluación) y permitir una mayor revisión del flujo incluso durante

el procesamiento.

Estas adaptaciones reflejan el compromiso del sistema con su objetivo principal: hacer comprensible y confiable el preprocesamiento automatizado para usuarios sin conocimientos técnicos.

3.6. Versión final con mejoras aplicadas

Tras la evaluación con usuarios, se llevó a cabo una revisión profunda del sistema con el objetivo de mejorar tanto la experiencia de uso como la claridad de los resultados presentados. Aunque se recogieron sugerencias interesantes de expertos y usuarios no técnicos, las decisiones de rediseño se tomaron principalmente atendiendo a estos últimos, dado que el sistema está específicamente orientado a facilitar el preprocesamiento automático de datos a personas sin conocimientos previos en ciencia de datos o programación.

Aplicar de forma indiscriminada todas las sugerencias de los usuarios expertos hubiera conllevado una sobrecarga de información y complejidad que habría afectado negativamente a la experiencia de los usuarios no expertos. Por ello, se buscaron soluciones que mejoraran la transparencia y comprensión del sistema sin comprometer su accesibilidad.

Las principales mejoras implementadas fueron las siguientes:

- Gestión avanzada de columnas objetivo con nulos: esta mejora surgió directamente de las observaciones realizadas por los usuarios expertos durante la evaluación. Varios participantes expresaron dudas sobre qué ocurría si la columna objetivo —la variable que el sistema debe predecir— contenía valores nulos, y señalaron la falta de información o control sobre esa situación.

  Este punto es especialmente crítico, ya que la columna objetivo es la base sobre la que se entrenan los modelos de aprendizaje automático. Si esta contiene valores faltantes, no solo se ve afectada la calidad del entrenamiento, sino que también pueden generarse errores o resultados engañosos en la evaluación del modelo. En consecuencia, entrenar con una columna objetivo incompleta puede hacer que las métricas de rendimiento pierdan validez y que el sistema extraiga conclusiones poco fiables.

  Para evitar este problema, se incorporó una verificación automática que analiza la columna objetivo justo antes de iniciar el procesamiento. Si contiene valores nulos, el sistema detiene el flujo y presenta al usuario un modal interactivo que explica la situación y ofrece tres opciones claras: seleccionar otra columna sin valores nulos, continuar con la columna actual (bajo su responsabilidad) o cargar un nuevo archivo.

  En la Figura 16 se muestra el modal implementado para gestionar este caso:

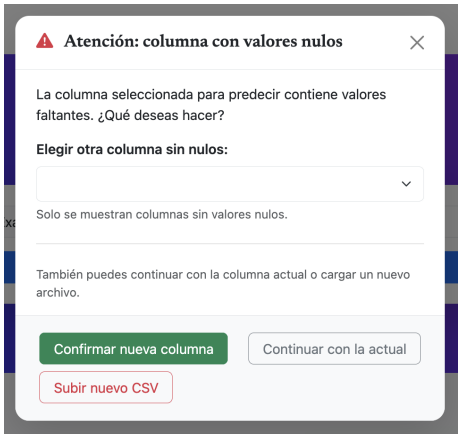


Figura 16: Modal de advertencia cuando la columna objetivo contiene valores nulos.

Además, esta mejora requirió la incorporación de dos nuevos endpoints en la API del backend: `POST /reanudar_columna` (cuando el usuario selecciona una nueva columna) y `POST /continuar_columna_actual` (cuando decide seguir con la columna original). Ambos endpoints permiten retomar el flujo desde el punto exacto en el que se había interrumpido, garantizando así una experiencia continua sin reiniciar el análisis completo.

Esta mejora no solo evita errores técnicos, sino que también contribuye a educar al usuario no experto, ayudándole a entender por qué ciertos requisitos son importantes en un análisis de datos con IA. Al incorporar esta decisión de forma guiada y comprensible, se refuerza la transparencia del sistema y se promueve la toma de decisiones informadas.

- Explicación progresiva del flujo: una de las principales mejoras incorporadas tras la evaluación con usuarios fue la introducción de explicaciones paso a paso que acompañan al progreso del sistema. Durante el procesamiento automático, cada etapa se muestra como un paso en pantalla, ahora incluye un botón *"Ver más"* que permite desplegar una descripción accesible de lo que está haciendo el sistema en ese momento.

  Esta retroalimentación textual cumple una doble función: por un lado, mantiene al usuario informado y reduce la percepción de espera pasiva; por otro, actúa como un mecanismo educativo que explica conceptos complejos de forma sencilla. Las explicaciones se diseñaron específicamente para ser comprensibles por personas sin formación técnica, evitando terminología matemática innecesaria y recurriendo a ejemplos.

  Por ejemplo, en la etapa de limpieza de datos, se explica qué son los valores faltantes, por qué pueden causar errores y cómo los algoritmos seleccionados (como KNN o Random Forest) intentan completarlos automáticamente (véase Figura 17).

  En la fase de normalización, se aclara por qué es importante llevar todas las variables a la misma escala, y se describe de forma visual el funcionamiento de un autoencoder no supervisado (Figura 18).

En la selección de características, se presenta cómo el sistema decide qué columnas son útiles para predecir, y por qué se eliminan otras, con ejemplos prácticos comprensibles (Figura 19).

Finalmente, la interfaz permite acceder a estas explicaciones a través de botones desplegables junto a cada fase del flujo, haciendo la información accesible solo cuando el usuario la necesita (véase Figura 20).

Limpieza de datos completada

Ocultar

¿Qué hace esta fase?
Aquí el sistema busca valores que faltan en tus datos, como celdas vacías o datos marcados como "N/A" o "None". Estos vacíos pueden hacer que los modelos fallen o se confundan, por eso es importante completarlos bien.

¿Cómo lo resuelve?
En lugar de borrar datos, el sistema los rellena automáticamente usando tres métodos diferentes. Cada uno intenta adivinar los valores que faltan basándose en la información que sí está disponible:

- **KNN:** Busca otras filas parecidas y rellena los huecos copiando lo más probable.
- **Regresión iterativa:** Usa las demás columnas para hacer una predicción del dato que falta.
- **Random Forest:** Usa un conjunto de reglas basadas en árboles para predecir los valores perdidos.

¿Qué hace el sistema con esto?
Aplica los tres métodos y se queda con el que mejor haya rellenado todos los huecos. Así te aseguras de que los datos estén listos para el análisis, sin que tengas que decidir qué técnica usar.

Ejemplo:
Si falta la "edad" de una persona, el sistema puede estimarla usando su "ocupación", "ingresos" o "nivel educativo".

¿Y si hay columnas con texto?
Primero se convierten en números para que el sistema las entienda, y luego se restauran para que tú veas los nombres originales. Todo esto se hace automáticamente.

Figura 17: Explicación accesible durante la fase de limpieza de datos.

Normalización aplicada

Ocultar

¿Qué hace esta fase?
En esta etapa, el sistema ajusta todos los valores numéricos para que estén en una escala similar, del 0 al 1. Esto es importante porque si una columna tiene valores muy grandes (por ejemplo, salarios) y otra valores pequeños (por ejemplo, edad), los modelos pueden confundirse o dar más peso a unas que a otras, por lo que esto evita que algunas columnas tengan más peso solo por tener números más grandes.

¿Qué técnicas usa?
La normalización en este sistema se hace en dos pasos:

- **MinMaxScaler:** Primero, cada número se transforma para que esté entre 0 y 1. Así, todas las columnas tienen una escala parecida.
- **Autoencoder:** Luego, se usa una pequeña red neuronal que aprende a reconstruir los datos. Esto sirve para hacer los datos más estables, eliminar pequeñas incoherencias y prepararlos mejor para los modelos de IA.

¿Qué decide el sistema?
Solo normaliza las columnas que realmente lo necesitan: las columnas numéricas. Evita normalizar las columnas que fueron codificadas desde texto (por ejemplo: "sí", "no") porque sus valores numéricos no tienen sentido como escala.

¿Qué pasa si algo falla?
Si no hay suficientes columnas numéricas o si algo va mal durante este paso, el sistema lo detecta y sigue adelante con los datos originales, sin que tú tengas que hacer nada.

Ejemplo:
Si tienes una columna "edad" con valores de 0 a 100, y otra "ingresos" con valores de 0 a 100.000, el modelo puede confundirse. Con este paso, ambas se transforman a una escala común (por ejemplo, entre 0 y 1) para que el análisis sea más justo.

Figura 18: Descripción visual del proceso de normalización con autoencoder.

Selección de características completada Ocultar

¿Qué hace esta fase?
En este paso, el sistema analiza todas las columnas para decidir cuáles son realmente útiles para predecir la variable objetivo. Las columnas menos relevantes se eliminan automáticamente, lo que mejora el rendimiento del modelo y lo hace más fácil de interpretar.

¿Qué técnicas usa?
Se utilizan dos modelos de inteligencia artificial para calcular qué tan importante es cada columna:

- **Random Forest:** Este modelo crea muchos árboles de decisión y analiza qué columnas ayudan más a predecir correctamente. Si una columna aparece muchas veces en las decisiones importantes, se considera valiosa.
- **Extra Trees:** Es parecido a Random Forest, pero introduce más aleatoriedad. Esto permite confirmar si una variable es realmente importante o solo lo parecía por casualidad.

¿Cómo decide qué columnas eliminar?
Cada modelo da una puntuación a cada columna según su importancia. Si esa puntuación es menor a **0.02**, la columna se considera poco útil y se elimina del conjunto de entrenamiento.

¿Qué pasa con las columnas de texto?
Las columnas con valores de texto (por ejemplo, "sí", "no", "tipo A", "tipo B") se transforman en varias columnas con valores binarios (1 o 0). Luego, el sistema agrupa esas columnas para calcular una única importancia para cada variable original.

Ejemplo sencillo:
Imagina que estás tratando de predecir si una persona tiene riesgo de diabetes. Tienes columnas como:

- Edad
- Altura
- Color de ojos
- Historial médico

El sistema puede detectar que **"color de ojos"** no tiene relación con la diabetes, así que la elimina. En cambio, **"edad"** y **"historial médico"** sí ayudan, por lo que se conservan.

Figura 19: Cómo se seleccionan automáticamente las columnas más relevantes.

Limpieza de datos completada Ver más
Normalización aplicada Ver más
Selección de características completada Ver más
Iniciando evaluación del CSV preprocesado...
39% completado de la evaluación del CSV preprocesado...

Figura 20: Vista del flujo de pasos con botones de "ver más" para cada etapa.

Gracias a esta mejora, incluso los participantes menos familiarizados con conceptos de ciencia de datos tendrían una mayor comprensión del proceso general, y podrían poder consultar lo que ocurre en cada fase en tiempo real.

- Reorganización del informe final: el modal de resultados fue reestructurado por completo para presentar la información de forma más clara, accesible y secuencial. Cada sección del flujo (limpieza, normalización, selección de características y evaluación automática) incluye ahora textos explicativos más desarrollados e iconos informativos interactivos (`i`) que el usuario puede desplegar para obtener detalles adicionales según su nivel de conocimiento o interés.

  Antes de analizar cada fase por separado, la Figura 21 muestra la estructura general del informe generado automáticamente.

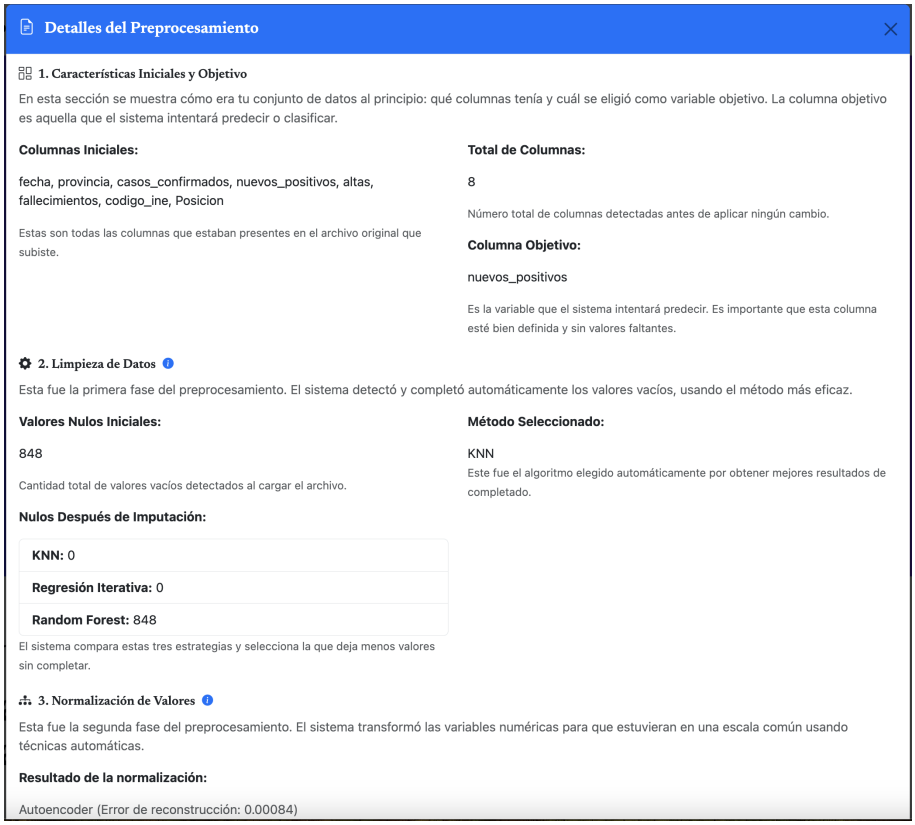


Figura 21: Vista general del nuevo modal con iconos informativos desplegables.

En cuanto a la limpieza de datos, se añadió una comparativa clara entre la cantidad de valores nulos que existían inicialmente y los que permanecen tras aplicar cada técnica de imputación (KNN, Regresión Iterativa, Random Forest). El sistema también indica cuál de las tres fue seleccionada automáticamente por ser la más eficaz, como se puede apreciar en la Figura 22.

2. Limpieza de Datos

Antes de aplicar técnicas de IA, es necesario asegurarse de que no haya datos vacíos. En esta etapa, el sistema detecta automáticamente los valores nulos (faltantes) y los completa utilizando distintos métodos, en lugar de eliminar información valiosa.

Las variables de texto como "sí", "no" o nombres de categorías se convierten temporalmente en números para que los algoritmos puedan trabajar correctamente. Esta conversión es interna y se revierte más adelante.

El sistema prueba tres formas distintas de rellenar los huecos:

- **KNN:** busca casos similares y estima el valor que falta.
- **Regresión Iterativa:** entrena un pequeño modelo para predecir cada columna con huecos.
- **Random Forest:** utiliza árboles de decisión para estimar los valores ausentes con mayor precisión.

Tras probar las tres, el sistema escoge automáticamente la que logra rellenar más huecos.

Esta fue la primera fase del preprocesamiento. El sistema detectó y completó automáticamente los valores vacíos, usando el método más eficaz.

Valores Nulos Iniciales:

848

Cantidad total de valores vacíos detectados al cargar el archivo.

Método Seleccionado:

KNN

Este fue el algoritmo elegido automáticamente por obtener mejores resultados de completado.

Nulos Después de Imputación:

KNN: 0

Regresión Iterativa: 0

Random Forest: 848

El sistema compara estas tres estrategias y selecciona la que deja menos valores sin completar.

Figura 22: Resumen de la fase de limpieza.

Respecto a la normalización, se explicó de forma accesible el concepto de *error de reconstrucción*, una métrica derivada del uso del autoencoder que algunos usuarios no comprendían completamente. Ahora se aclara que no se trata de un fallo, sino de una medida interna de calidad de la transformación, como se muestra en la Figura 23.

3. Normalización de Valores

La normalización sirve para poner todas las variables numéricas en una escala común, normalmente entre 0 y 1. Esto es clave para evitar que algunas columnas (como ingresos o edad) influyan más que otras solo por tener números más grandes.

El sistema identifica qué columnas pueden normalizarse (excluyendo la columna objetivo y las categóricas codificadas) y aplica dos pasos:

1. **Escalado MinMax:** transforma todos los valores al rango 0–1. Esto es necesario porque el modelo que se usa después (autoencoder) solo puede trabajar con valores dentro de ese rango.
2. **Autoencoder:** una red neuronal pequeña que aprende a reconstruir los datos desde una versión comprimida. Este proceso ayuda a eliminar ruido y obtener una versión más coherente de las variables.

Durante la reconstrucción, el sistema calcula un **error de reconstrucción**, que mide cuánto se parecen los datos originales a los reconstruidos. Si ese valor es muy alto, puede deberse a ruido, incoherencias o pocas columnas numéricas válidas.

En esos casos, el sistema detecta el problema y omite automáticamente la normalización para evitar distorsionar los datos.

Esta fue la segunda fase del preprocesamiento. El sistema transformó las variables numéricas para que estuvieran en una escala común usando técnicas automáticas.

Resultado de la normalización:

Autoencoder (Error de reconstrucción: 0.00084)

¿Qué significa el "error de reconstrucción"?
Es una medida que indica qué tan parecidos son los datos originales y los reconstruidos por el autoencoder. Cuanto menor sea este valor, mejor ha aprendido el sistema a representar los datos.

A diferencia de técnicas clásicas como MinMax o Z-score, aquí usamos un autoencoder para refinar los datos. El error refleja la calidad de esa reconstrucción.

Un valor alto no indica necesariamente un fallo, sino que los datos eran complejos o ruidosos. El sistema lo detecta y, si fuera necesario, evita aplicar la normalización para preservar la integridad del conjunto.

Figura 23: Vista de la fase de normalización.

En la fase de selección de características, ahora en lugar de mostrar únicamente

las cinco características principales, los gráficos exhiben todas las variables que superaron el umbral de importancia, acompañadas de una lista con las columnas eliminadas, si las hubiera. Asimismo, se ofrece al usuario la opción de desplegar una tabla para visualizar las puntuaciones completas de importancia por columna, lo que mejora la transparencia del proceso y permite comprender por qué se descartaron ciertas variables, como se observa en la Figura 24.

4. Selección de Características

Esta fase busca reducir las columnas del dataset a solo aquellas que realmente ayudan a predecir la variable objetivo. Evalúa la importancia de cada columna (característica) según su capacidad de ayudar al modelo a predecir correctamente.

- **Random Forest:** Analiza qué columnas ayudan más a tomar decisiones correctas con árboles de decisión.
- **Extra Trees:** Variante más aleatoria que explora más combinaciones posibles para encontrar patrones útiles.

Cada columna recibe una **puntuación de importancia**. Solo se conservan las que superan un umbral mínimo. En este sistema, ese **umbral está fijado en 0.02**, lo que significa que solo se mantienen las columnas cuya contribución al modelo sea superior al 2 % del total. Este valor fue elegido como equilibrio entre conservar información útil y evitar ruido o variables irrelevantes. Las columnas que no alcanzan este umbral se consideran poco útiles para predecir y se eliminan automáticamente.

En esta tercera fase, el sistema analiza todas las columnas y se queda solo con las más útiles para hacer predicciones. Esto mejora la precisión del modelo, evita el uso de columnas irrelevantes y acelera el procesamiento.

Umbral de Importancia: 0.02

Es el valor mínimo que una columna debe alcanzar para ser considerada relevante. Las que están por debajo se eliminan.

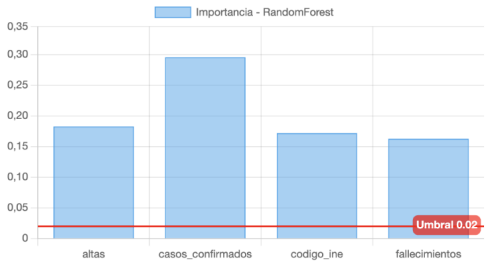
Columnas Seleccionadas:

Random Forest: casos_confirmados, altas, fallecimientos, codigo_ine

Extra Trees: casos_confirmados, altas, fallecimientos, codigo_ine

Estos gráficos muestran las características seleccionadas según los modelos RandomForest y ExtraTrees.

RandomForest



RandomForest selecciona características basándose en su capacidad para mejorar la precisión del modelo.

ExtraTrees

ExtraTrees es similar a RandomForest, pero mejora la selección al usar una mayor aleatoriedad.

Columnas Eliminadas por Bajo Peso

Lista de columnas eliminadas:

RandomForest: fecha, provincia, Posicion

ExtraTrees: fecha, provincia, Posicion

Ocultar detalles de importancia

Importancia (RandomForest)

Columna	Importancia
Posicion	0.0056
fecha	0.0001
provincia	0.0043

Importancia (ExtraTrees)

Columna	Importancia
Posicion	0.0072
fecha	0.0002
provincia	0.0051

Figura 24: Vista detallada de la selección de características.

En lo relativo a la evaluación automática mediante AutoML, se eliminó la antigua nota sobre 10 y se reemplazó por una explicación clara del *score* como resultado directo de la métrica más adecuada según el tipo de problema (clasificación binaria, multiclase o regresión). Ahora se explica el tipo de problema

detectado, la métrica aplicada y cómo interpretar el *score* resultante, como se puede ver en la Figura 25.

5. Evaluación con AutoML

¿Qué es AutoML?
AutoML (Aprendizaje Automático Automatizado) es una herramienta que prueba automáticamente muchos algoritmos de inteligencia artificial y elige el que mejor se adapta a tus datos. Así no necesitas saber programación ni estadísticas para obtener un buen modelo.

¿Qué tipo de problema tengo?
El sistema detecta automáticamente si tu archivo representa uno de estos tres tipos:

- **Regresión:** cuando la columna objetivo contiene números continuos, como sueldos, temperaturas o edades. El modelo intenta predecir un número exacto.
- **Clasificación binaria:** cuando solo hay dos posibles respuestas, por ejemplo: "Sí/No", "Enfermo/Sano", "1/0".
- **Clasificación multiclase:** cuando hay más de dos categorías, como "Bajo", "Medio" y "Alto" o diferentes tipos de enfermedades.

¿Cómo se mide el resultado?
Para cada tipo de problema, AutoML usa una métrica específica para evaluar la calidad del modelo:

- **Regresión:** se usa el coeficiente de determinación **R^2**. Cuanto más cerca de 1, mejor.
- **Clasificación binaria:** se usa **AUC** (Área Bajo la Curva). Cuanto más cerca de 1, mejor.
- **Clasificación multiclase:** se usa el **Error medio por clase**. Cuanto más bajo, mejor.

El sistema convierte automáticamente cada métrica a una escala de 0 a 10 para que sea más fácil de entender. Sin importar la métrica original, un score más alto siempre indica mejor rendimiento general del modelo.

El sistema utiliza AutoML (Machine Learning Automatizado) para probar diferentes modelos de inteligencia artificial y elegir automáticamente el mejor, sin que el usuario tenga que configurarlos manualmente.

Tipo de problema: Regresión

Mejor modelo de selección de caracteristicas tras la evaluación: Random Forest y Extra Trees (idénticas)

Score obtenido:

Para este problema de **regresión** , el sistema usó la métrica **R2**.

El score final obtenido fue: **10** sobre 10.

Aunque algunas métricas como el **Error medio por clase** se interpretan "mejor cuanto más bajo", el sistema las transforma internamente para que todos los scores estén en la misma escala: **más alto = mejor rendimiento**.

Ver Detalles Completos

Figura 25: Explicación del tipo de problema detectado por AutoML y la métrica aplicada.

Además, se reorganizó la sección de AutoML para que los usuarios no expertos comprendan mejor qué ha hecho el sistema y por qué. Aunque ya existía una tabla con los modelos probados y sus métricas, ahora se complementa con explicaciones detalladas sobre el significado de cada métrica (RMSE, MAE, R^2, MSE, LogLoss, AUC, Error medio por clase) y cuándo se aplican. También se aclara que si una celda aparece como `N/A`, no indica un fallo, sino que ciertos modelos o métricas no son compatibles con el tipo de problema. Por ejemplo, AUC solo se calcula en clasificación binaria.

Esto se ilustra con dos ejemplos reales: en la Figura 26, el sistema detectó un problema de regresión y utilizó como métrica principal R^2; mientras que en la Figura 27, se trataba de clasificación binaria y AutoML aplicó AUC como métrica principal.

Resultados del AutoML

AutoML (Aprendizaje Automático Automatizado) es una herramienta que entrena y compara múltiples modelos de IA por ti. Este sistema utiliza la librería **H2O AutoML**, una tecnología de código abierto ampliamente usada en la industria.
Prueba de forma automática diferentes tipos de algoritmos como **Gradient Boosting (GBM)**, **XGBoost**, **Deep Learning**, **Random Forest**, **GLM** y **modelos ensamblados** (combinaciones de varios modelos).
Luego, selecciona el modelo que haya obtenido el mejor rendimiento según el tipo de problema detectado (regresión, clasificación binaria o multiclase), usando una métrica adecuada para cada caso.

Modelo	Score (métrica principal)	RMSE	MAE	R^2	MSE	L
DeepLearning_grid_3_AutoML_1_20250526_141204_model_4	0.4926	30.844245962651776	24.934387310365587	0.49258144326171294	951.3675090045604	
DeepLearning_grid_3_AutoML_1_20250526_141204_model_11	0.3227	35.636233556612304	28.3824195280851	0.3226679538994157	1269.9411421014208	
XGBoost_lr_search_selection_AutoML_1_20250526_141204_select_grid_model_1	0.9957	2.8341108767149827	2.0363110371430717	0.9957159778862096	8.032184461514168	
DeepLearning_grid_1_AutoML_1_20250526_141204_model_18	0.4956	30.753786903994175	25.31551546606491	0.4955533620500734	945.7954089362836	
DeepLearning_grid_3_AutoML_1_20250526_141204_model_3	0.4405	32.388886974068335	27.045227580458178	0.4404871330713508	1049.0399994189736	
XGBoost_grid_1_AutoML_1_20250526_141204_model_80	0.9971	2.347929052327165	1.7464026129245758	0.9970597248758992	5.512770834761939	
XGBoost_grid_1_AutoML_1_20250526_141204_model_55	0.8871	14.551253291100453	11.832666161855062	0.8870675288634543	211.7389723417617 8	
XGBoost_grid_1_AutoML_1_20250526_141204_model_8	0.9876	4.821212019049162	3.606904757817586	0.9876026034938669	23.244085332624103	
XGBoost_grid_1_AutoML_1_20250526_141204_model_74	0.9993	1.1358417198128745	0.8233506212631861	0.9993118966643135	1.2901364124674686	

Mejor modelo y score obtenido

El modelo **XGBoost_grid_1_AutoML_1_20250526_141204_model_1** fue el mejor según AutoML.
Obtuvo un **score final de 10 / 10**, basado en la métrica **R2**.

Figura 26: Ejemplo de resultados AutoML para un problema de regresión.

Resultados del AutoML

AutoML (Aprendizaje Automático Automatizado) es una herramienta que entrena y compara múltiples modelos de IA por ti. Este sistema utiliza la librería **H2O AutoML**, una tecnología de código abierto ampliamente usada en la industria.
Prueba de forma automática diferentes tipos de algoritmos como **Gradient Boosting (GBM)**, **XGBoost**, **Deep Learning**, **Random Forest**, **GLM** y **modelos ensamblados** (combinaciones de varios modelos).
Luego, selecciona el modelo que haya obtenido el mejor rendimiento según el tipo de problema detectado (regresión, clasificación binaria o multiclase), usando una métrica adecuada para cada caso.

Score (métrica principal)	RMSE	MAE	R^2	MSE	LogLoss	AUC	Error Medio por Clase
0.0897	0.7720384187487098	N/A	-1.4640071104754369	0.5960433200240082	1.7446881000752028	0.08970649028524184	0.03632071242842093,0.5
0.0690	0.785378605571257	N/A	-1.549894808139943	0.6168195540890522	1.8172213785713	0.06903679206283589	0.0186387770292404,0.5
0.7350	0.4523016485191979 6	N/A	0.1542919336371058	0.2045767812531841	0.593405254202212	0.7350144687887558	0.46778050821730155,0.31376601901612244
0.6718	0.4801405329307192	N/A	0.04698250780076474	0.23053493136299502	0.663110016110297	0.6717651922281934	0.5484481787251633,0.32224059528730886
0.7350	0.47321686896336124	N/A	0.07426951189959907	0.22393420507148698	0.6398206643451032	0.7350144687887558	0.4854656906795865,0.31376601901612244
0.7718	0.4352921264662906	N/A	0.21670427712465845	0.18947923536354513	0.5537608949556674	0.7718065316246383	0.31814234806218955,0.27015295576684584
0.7350	0.47321686896336124	N/A	0.07426951189959907	0.22393420507148698	0.6398206643451032	0.7350144687887558	0.4854656906795865,0.31376601901612244
0.7160	0.45901720933210916	N/A	0.12899215186838642	0.21069679846303732	0.6066442669207622	0.7159983464241422	0.43720776121726 67,0.31583298883836297
0.5845	0.5977578320010363	N/A	-0.4771162700230638	0.35731442571857913	1.2034664015518581	0.5845390657296403	0.10663276450326072,0.412980570483671

Mejor modelo y score obtenido

El modelo **GBM_grid_1_AutoML_1_20250526_144332_model_3** fue el mejor según AutoML.
Obtuvo un **score final de 10 / 10**, basado en la métrica **AUC**.

Figura 27: Ejemplo de resultados AutoML para un problema de clasificación binaria.

Estas mejoras responden directamente al feedback de usuarios no expertos durante la fase de evaluación, y están orientadas a facilitar la comprensión sin renunciar al valor técnico del análisis.

- Explicación del score final: varios usuarios confundieron el score mostrado como si fuera una nota global sobre la calidad del archivo. Para evitar malentendidos, se eliminó de la pantalla principal y se mantuvo únicamente dentro del informe técnico detallado. Allí se acompaña ahora de una explicación más clara sobre qué representa: el rendimiento del mejor modelo generado con AutoML (según AUC, R^2 o error medio por clase), utilizado internamente para comparar variantes del preprocesamiento, pero no como valoración general del dataset.

Como se muestra en la Figura 28, la pantalla principal del sistema ya no incluye el score final de AutoML para evitar interpretaciones erróneas por parte del usuario.

Figura 28: La pantalla principal ya no muestra el score final.

En resumen, estas mejoras se fundamentan directamente en las observaciones recogidas durante la fase de evaluación con usuarios. Refuerzan la coherencia del sistema con su objetivo central: permitir el uso de técnicas de preprocesamiento automatizado a usuarios sin experiencia en ciencia de datos, manteniendo la calidad técnica y la transparencia del flujo. Así, la versión final logra ofrecer un equilibrio entre la potencia del procesamiento automático y la claridad necesaria para un perfil no técnico.

4. Discusión

La discusión se organiza en tres apartados principales que, en conjunto, ofrecen una visión completa del impacto y la relevancia del sistema propuesto. Primero, se contextualiza la inspiración y las aportaciones del sistema en relación con los vacíos identificados durante la Revisión Sistemática de la Literatura. A continuación, se analiza en detalle el comportamiento del flujo técnico, explorando cómo se integran y funcionan sus principales etapas. Por último, se retoman los resultados de la evaluación práctica presentada en la metodología, destacando cómo la experiencia de usuarios reales —con y sin conocimientos técnicos— sirvió para validar la utilidad, la comprensión y la confianza generada por el sistema. Esta triple perspectiva permite no solo valorar la eficacia técnica, sino también su adecuación a las necesidades reales de los usuarios, así como las oportunidades de mejora y refinamiento para futuras versiones.

4.1. Inspiración, alineación y aportaciones del sistema desarrollado

El diseño del sistema interactivo de preprocesamiento automático propuesto en este trabajo no solo se fundamenta en principios de diseño accesible y automatización inteligente, sino que se alinea con los vacíos y necesidades detectados a lo largo de la Revisión Sistemática de la Literatura presentada previamente (ver Sección 2).

A partir del análisis de las preguntas de mapeo y de investigación, se detectaron varias limitaciones comunes en los estudios revisados: poca claridad en los procesos de preprocesamiento [4, 32, 59], ausencia de herramientas accesibles que expliquen y muestren estos pasos de forma visual [22, 73], y falta de sistemas iterativos que permitan a usuarios no expertos intervenir y entender las decisiones tomadas por el sistema [11, 53, 62].

Para solventar estas limitaciones, el sistema desarrollado en este trabajo incorpora varias estrategias basadas en los hallazgos de la SLR. En primer lugar, se explica paso a paso cada fase del flujo, mediante iconos informativos que despliegan definiciones comprensibles de cada técnica y visualizaciones intuitivas de los cambios aplicados. Esto aborda directamente la necesidad identificada en estudios previos de hacer más comprensibles las decisiones automáticas [4, 32, 62]. En segundo lugar, se diseñó una interfaz gráfica clara y guiada, con barra de progreso, avisos y mensajes comprensibles, alineada con las recomendaciones de los estudios sobre diseño centrado en usuarios sin formación técnica [76, 9, 11, 77].

Además, para ofrecer mayor control sobre decisiones críticas, se implementó una pausa activa si se detectan valores nulos en la variable objetivo, ofreciendo al usuario tres alternativas antes de continuar. Esta intervención directa, inspirada por propuestas de flujo iterativo adaptado [4, 53], evita errores comunes y mejora la confianza en el sistema. También se adoptó una evaluación adaptada al tipo de problema, utilizando métricas distintas según si se trata de regresión, clasificación binaria o multiclase (como AUC, R^2, MAE o error medio por clase), y transformando el mejor resultado a una escala estandarizada de 0 a 10 para facilitar su interpretación, de acuerdo con las mejores prácticas observadas en sistemas AutoML [50, 5, 9, 64].

Por último, para abordar la falta de orientación ante datos problemáticos [76, 5, 35], se desarrolló un sistema de clasificación de errores con mensajes diferenciados en la interfaz de usuario, permitiendo una gestión más eficaz y comprensible de los fallos.

En conjunto, el sistema final no solo automatiza el preprocesamiento de forma eficaz, sino que lo hace de manera transparente, controlada y accesible. Las funcionalidades desarrolladas están alineadas con las recomendaciones extraídas de la literatura y representan una respuesta práctica a carencias observadas en múltiples estudios revisados, contribuyendo así a la democratización del uso de IA para el análisis de datos en contextos sin experiencia técnica.

4.2. Comportamiento del flujo de preprocesamiento

El sistema implementa una secuencia robusta de pasos: imputación, normalización, selección de características y evaluación automática mediante AutoML. En las distintas pruebas realizadas, cada fase del proceso se comportó de manera consistente, adaptándose a los datos.

En la limpieza de valores nulos se aplicaron tres técnicas distintas: KNN, Imputación Iterativa y una variante basada en Random Forest. En todas las pruebas, KNN e Imputación Iterativa lograron eliminar completamente los valores nulos del dataset, mientras que la imputación con Random Forest no siempre lo consiguió. Esta diferencia se explica porque KNN y la Imputación Iterativa están diseñadas específicamente para rellenar valores faltantes de manera precisa: KNN encuentra los registros más similares y promedia sus valores para completar los nulos, mientras que la Imputación Iterativa utiliza un modelo de regresión inteligente que predice los valores faltantes en función del resto de columnas, repitiendo el proceso para mejorar la estimación. En cambio, la imputación con Random Forest, tal y como se implementa en este sistema, se limita a rellenar muchos valores con la moda y solo usa un modelo de predicción para la columna objetivo, dejando algunas columnas sin completar en ciertos casos. Por ello, se comprobó que KNN e Imputación Iterativa siempre dejan el conjunto de datos completamente limpio, mientras que la opción basada en Random Forest no. Una posible mejora del sistema sería prescindir de la técnica basada en Random Forest para la limpieza y optar únicamente por las otras dos. Esta simplificación aumentaría la robustez del proceso, reduciría el código innecesario y evitaría resultados inconsistentes o impredecibles cuando se trabaja con conjuntos de datos con alta proporción de valores faltantes.

En cuanto a la normalización de datos, esta etapa combinó un escalado MinMax con un autoencoder no supervisado. Se comprobó que la normalización se aplicaba correctamente en todos los conjuntos, incluso en aquellos con muchas columnas, garantizando que solo se normalizan variables numéricas y evita distorsionar variables categóricas codificadas.

Durante la selección de características, se aplicaron Random Forest y Extra Trees de forma independiente. Ambos métodos ofrecieron resultados estables y eficaces. Cuando ambos coincidían, se procedía directamente al entrenamiento; si eran distintos, se comparaban mediante AutoML. Sin embargo, en algunos conjuntos pequeños o con muchas variables irrelevantes, al aplicar el umbral de importancia predefinido de 0,02, se detectó que a veces solo una característica superaba el umbral o incluso ninguna. En esos casos, el sistema detiene el flujo automáticamente, avisando al usuario con un mensaje claro. Este mecanismo evita entrenar modelos sobre datos no informativos, pero también revela una posible limitación del enfoque basado en umbral fijo. Una mejora futura consistiría en establecer un número mínimo deseable de características (por ejemplo, cinco), y reducir dinámicamente el umbral hasta alcanzarlo. Esta estrategia permitiría mantener una selección de variables consistente, sin comprometer la calidad del modelo por descartar columnas que, aunque con menor puntuación, podrían ser relevantes.

Por último, la evaluación automática (AutoML) demostró ser altamente efectiva. En la mayoría de las pruebas, los modelos alcanzaron puntuaciones transformadas entre 7 y 9 sobre 10, adaptadas a la métrica correspondiente. Esta puntuación elevada no solo refleja un buen rendimiento del modelo, sino que también indica que el flujo completo de preprocesamiento generó datos consistentes y bien preparados para el entrenamiento. La detección automática del tipo de problema permitió emplear las métricas más adecuadas. Estas métricas permitieron comparar los resultados de forma clara y coherente.

En conjunto, los resultados mostraron un flujo de preprocesamiento consistente y confiable, adaptándose bien a diferentes tipos de datos. Estas pruebas confirmaron que el sistema logra mantener un equilibrio entre la automatización técnica y la transparencia necesaria para usuarios no expertos.

4.3. Validación práctica del sistema

La validación práctica del sistema complementó y reforzó las observaciones técnicas previas. Tras la fase de pruebas con datasets, se realizó una evaluación cualitativa con cinco usuarios reales: tres expertos con perfil técnico y dos residentes clínicos sin experiencia en programación o ciencia de datos. Esta fase buscó identificar no solo la utilidad técnica de la herramienta, sino también su grado de comprensibilidad y la confianza que generaba entre personas con distintos niveles de conocimiento.

Los participantes valoraron positivamente la facilidad de uso general y la reducción del esfuerzo técnico que implica el preprocesamiento automático. Los residentes clínicos, en particular, destacaron la claridad de la interfaz, que les permitió completar el flujo completo sin ayuda externa, comprendiendo los pasos realizados gracias a los textos explicativos y la presentación visual. Los expertos, por su parte, enfatizaron la importancia de contar con mayor control y transparencia en el proceso, solicitando información más detallada sobre qué técnicas se estaban usando y cómo se estaban tomando las decisiones.

Por otro lado, la retroalimentación también permitió identificar carencias y oportunidades de mejora, que fueron abordadas en la versión final. Varios usuarios, especialmente los expertos, señalaron la falta de advertencia cuando la columna objetivo contenía valores nulos, una situación que podría comprometer la validez del modelo. En respuesta, se implementó una verificación automática que pausa el flujo si se detectan nulos y ofrece al usuario tres opciones: continuar, seleccionar otra columna o subir otro archivo. Además, los usuarios no técnicos manifestaron dificultades para comprender algunos términos técnicos como KNN, Autoencoder o score final. Para abordar esta carencia, se añadieron iconos informativos desplegables con explicaciones claras, visuales y accesibles sobre cada técnica empleada.

Otro aspecto destacado fue el de poder revisar las fases ya completadas mientras AutoML se ejecuta, facilitando que el usuario pueda aprovechar la espera para entender qué se hizo en cada paso, cómo se aplicaron los algoritmos y por qué. Finalmente, se rediseñó la presentación del score final, trasladándolo a un apartado

más técnico y detallado para evitar que se interpretara como una nota general del archivo en lugar de un indicador de rendimiento del modelo.

Estas mejoras consolidaron el sistema como una herramienta no solo funcional y técnicamente robusta, sino también accesible y transparente para usuarios sin conocimientos en ciencia de datos. La retroalimentación de esta validación práctica resultó esencial para transformar la herramienta en una solución centrada en las necesidades reales de los usuarios, mostrando la importancia de integrar evaluaciones con perfiles diversos como parte fundamental de un diseño humanocéntrico.

En conjunto, los resultados técnicos y la validación práctica confirman que el sistema ofrece no solo automatización confiable, sino también una experiencia comprensible y transparente para usuarios sin conocimientos técnicos. Las lecciones aprendidas y las mejoras implementadas consolidan un enfoque centrado en las personas, demostrando que la colaboración con usuarios reales es clave para democratizar la IA en contextos sensibles y de alto impacto. Este aprendizaje servirá de base para futuras versiones más adaptativas, accesibles y alineadas con las necesidades reales de cada perfil de usuario.

5. Conclusiones y líneas futuras

Este trabajo ha abordado el diseño, desarrollo y validación de un sistema interactivo de preprocesamiento de datos asistido por IA, con el objetivo de hacerlo accesible y comprensible para usuarios sin conocimientos técnicos avanzados. A lo largo del informe se han planteado y cumplido los objetivos generales y específicos definidos en la Sección 1, siguiendo un enfoque sólido y bien justificado.

En primer lugar, se ha llevado a cabo una revisión sistemática de la literatura que permitió identificar, clasificar y analizar las principales técnicas de IA aplicadas al preprocesamiento de datos. Esta revisión ofreció una visión panorámica del estado del arte y sirvió como base sólida para orientar las decisiones de diseño del sistema propuesto. A través de la SLR, se recopilaron buenas prácticas, enfoques accesibles y vacíos metodológicos, cumpliendo así el primer objetivo específico (Sección 2).

En segundo lugar, se ha diseñado e implementado una herramienta funcional que combina un backend inteligente con un frontend accesible. La solución desarrollada integra técnicas de limpieza, normalización, selección de características y evaluación automática mediante AutoML, todo ello guiado por una interfaz comprensible, explicativa y centrada en la experiencia del usuario. Esta implementación responde al segundo objetivo específico y se apoya directamente en los hallazgos teóricos obtenidos (Sección 3).

El tercer objetivo específico se ha abordado mediante una validación práctica con usuarios no expertos, incluyendo perfiles técnicos y no técnicos. Los resultados obtenidos reflejan una alta tasa de comprensión, usabilidad y utilidad percibida. A partir de esta evaluación se identificaron mejoras clave, que posteriormente se incorporaron en la versión final del sistema, como explicaciones visuales, gestión

de errores y control interactivo del flujo. Estas funcionalidades no fueron evaluadas directamente, pero están alineadas con las observaciones recogidas y buscan reforzar la comprensión y confianza de usuarios no expertos en futuras aplicaciones (Sección 3).

Además, el trabajo ha prestado especial atención a la documentación del proceso completo de desarrollo. Se ha elaborado una descripción clara y estructurada de las decisiones de diseño adoptadas, las técnicas de IA utilizadas en cada etapa del flujo, el funcionamiento general del sistema y los resultados obtenidos durante su validación. Esta documentación busca no solo facilitar la comprensión del sistema, sino también servir como referencia para su reproducción, mejora o ampliación futura.

Por lo tanto, todos los objetivos del trabajo han sido alcanzados, y el sistema desarrollado constituye una solución viable, accesible y replicable para el preprocesamiento automatizado de datos.

Finalmente, a partir del análisis realizado y de la experiencia acumulada durante la validación del sistema, se identifican varias oportunidades de mejora que podrían orientar futuras versiones de la herramienta:

- Modificación manual de decisiones automáticas: actualmente, el sistema toma decisiones autónomas en varias fases del preprocesamiento. Una mejora importante consistiría en permitir que el usuario revise y modifique algunas de estas decisiones tras visualizar el informe final, especialmente en lo relativo a las columnas seleccionadas.

- Tratamiento configurable para columnas con alta proporción de nulos: actualmente, el sistema solo elimina columnas que contienen exclusivamente valores nulos. Sin embargo, en muchos casos existen columnas con un porcentaje elevado de valores faltantes (por ejemplo, superiores al 60 %) que pueden comprometer la fiabilidad del modelo. Una posible mejora futura consistiría en introducir un umbral, podrá ser configurable por el usuario o fijo, que determine cuándo una columna debe eliminarse automáticamente o ser marcada para revisión.

- Personalización del umbral de selección de características: actualmente, el sistema emplea un umbral fijo (0.02) para determinar qué variables tienen la relevancia suficiente como para ser seleccionadas en el análisis. Una futura ampliación podría permitir al usuario ajustar este umbral manualmente, o bien establecer un número mínimo deseado de características (por ejemplo, cinco). Si el número de variables seleccionadas es insuficiente, el sistema podría reducir dinámicamente el umbral o incluir automáticamente aquellas características con importancia moderada que cumplan ciertos criterios. Este enfoque evitaría la exclusión de variables útiles y aumentaría la flexibilidad del sistema sin comprometer la calidad del modelo.

- Explicaciones adaptadas al perfil del usuario: aunque la interfaz actual ya proporciona explicaciones accesibles, sería beneficioso incluir un modo dual (experto vs. no experto), donde los usuarios puedan seleccionar el nivel de

detalle que desean recibir. En el modo experto podrían mostrarse métricas avanzadas, configuraciones técnicas o detalles del algoritmo, mientras que en el modo básico se priorizarían explicaciones simplificadas, visuales y orientadas a la comprensión conceptual.

- Simplificación del módulo de limpieza de datos: dado que las técnicas de imputación basadas en KNN e Iterative Imputer han demostrado ser más robustas, coherentes y efectivas en todos los conjuntos de datos evaluados, se plantea eliminar el método basado en Random Forest para la limpieza. Esta simplificación no solo reduciría la complejidad del código y el riesgo de inconsistencias, sino que también mejoraría la fiabilidad del sistema al evitar resultados impredecibles en escenarios con alta proporción de valores faltantes.

- Despliegue en contextos reales: una vez consolidada la validación técnica y la evaluación con usuarios, una línea futura de gran interés sería implementar el sistema en entornos reales de trabajo, como laboratorios de investigación o departamentos clínicos, para medir su rendimiento en situaciones reales y recopilar retroalimentación directa de los usuarios finales.

- Ampliación de formatos de entrada: actualmente, el sistema acepta únicamente archivos CSV como entrada. Sin embargo, durante la búsqueda de datasets para las pruebas se encontraron fuentes en otros formatos comunes (como .data o .xlsx). Una mejora relevante sería ampliar la compatibilidad del sistema a estos formatos, aumentando su utilidad y flexibilidad para distintos casos de uso.

Estas líneas de mejora permitirían mejorar aún más el sistema y responder de forma más precisa a las necesidades reales de los usuarios, manteniendo el equilibrio entre automatización, explicabilidad y control por parte del usuario. Además, refuerzan el enfoque humanocéntrico del desarrollo, asegurando que las decisiones técnicas estén alineadas con las expectativas y capacidades del público objetivo.

El propósito inicial de este trabajo era incorporar técnicas de IA para asistir de forma interactiva al preprocesamiento de datos, facilitando su uso a personas sin conocimientos avanzados. Los resultados obtenidos demuestran que la herramienta desarrollada ha cumplido este objetivo, no solo automatizando tareas complejas como la limpieza, normalización y selección de características, sino también ofreciendo explicaciones y guías visuales que hacen comprensible y transparente cada decisión del proceso. Esta combinación de automatización e interacción humana evidencia que la IA no debe concebirse como un sustituto del conocimiento experto, sino como un facilitador que empodera al usuario final para explorar y comprender sus propios datos. De este modo, el sistema desarrollado ofrece una solución práctica y concreta para que cualquier usuario pueda aprovechar el potencial de la IA en el análisis de datos, haciendo más accesible esta tecnología y abriendo nuevas posibilidades en entornos donde antes era inaccesible.

Referencias

[1] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. Technical report, EBSE Technical Report, 2007. [*Citado en págs. 3 y 4.*]

[2] Matthew J. Page, Joanne E. McKenzie, Patrick M. Bossuyt, David Moher, Iain A. Boutron, Terri D. Turner, Larissa Shamseer, Douglas G. Altman, Paul T. Higgins, Jeremy M. G. Westwood, et al. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372:n71, 2021. [*Citado en págs. 4 y 10.*]

[3] Y. Peng, X. Zhang, Y. Li, Q. Su, S. Wang, F. Liu, C. Yu, and M. Liang. Mvpani: A toolkit with friendly graphical user interface for multivariate pattern analysis of neuroimaging data. *Frontiers in Neuroscience*, 14:576015, 2020. [*Citado en págs. 12, 14, 16, 18, 22, 24, 25 y 26.*]

[4] S. Malkusch, L. Hahnefeld, R. Gurke, and J. Lötsch. Visually guided preprocessing of bioanalytical laboratory data using an interactive r notebook (pguimp). *CPT: Pharmacometrics and Systems Pharmacology*, 2021. [*Citado en págs. 12, 14, 16, 19, 20, 22, 23, 24, 25, 26, 29 y 65.*]

[5] R. Ooms and M. Spruit. Self-service data science in healthcare with automated machine learning. *Applied Sciences (Switzerland)*, 2020. [*Citado en págs. 12, 14, 16, 18, 21, 22, 23, 24, 25, 26 y 65.*]

[6] N. M. Patil, T. P. Rane, and A. A. Panjwani. Ml suite: An auto machine learning tool. In *Lecture Notes in Networks and Systems*. 2021. [*Citado en págs. 12, 14, 16 y 18.*]

[7] N. McCombe, X. Ding, G. Prasad, P. Gillespie, D. P. Finn, S. Todd, P. L. McClean, and K. Wong-Lin. Alzheimer's disease assessments optimized for diagnostic accuracy and administration time. *IEEE Journal of Translational Engineering in Health and Medicine*, 2022. [*Citado en págs. 12, 14, 16, 18, 22, 24 y 31.*]

[8] P. Washington, Q. Tariq, E. Leblanc, B. Chrisman, K. Dunlap, A. Kline, H. Kalantarian, Y. Penev, K. Paskov, C. Voss, N. Stockham, M. Varma, A. Husic, J. Kent, N. Haber, T. Winograd, and D. P. Wall. Crowdsourced privacy-preserved feature tagging of short home videos for machine learning asd detection. *Scientific Reports*, 2021. [*Citado en págs. 12, 15, 16, 18 y 26.*]

[9] M.-B. Bîndilă and M. Negru. End-to-end automated machine learning system for supervised learning problems. In *Proceedings of the 2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP 2021)*, 2021. [*Citado en págs. 12, 14, 16, 18, 19, 20, 21, 22, 23, 24, 25 y 65.*]

[10] Y. Song, J. Zhang, Y.-D. Zhang, Y. Hou, X. Yan, Y. Wang, M. Zhou, Y.-F. Yao, and G. Yang. Feature explorer (fae): A tool for developing and comparing radiomics models. *PLoS ONE*, 15(8):e0237587, 2020.
[*Citado en págs. 13, 14, 16, 18, 19, 20, 21, 22, 23, 24 y 25.*]

[11] J. Gerolami, J. J. M. Wong, R. Zhang, T. Chen, T. Imtiaz, M. Smith, T. Jamaspishvili, M. Koti, J. I. Glasgow, P. Mousavi, N. Renwick, and K. Tyryshkin. A computational approach to identification of candidate biomarkers in high-dimensional molecular data. *Diagnostics*, 2022.
[*Citado en págs. 13, 14, 16, 18, 19, 20, 21, 22, 23, 24, 25, 31 y 65.*]

[12] J. Y. An, H. Seo, Y.-G. Kim, K. E. Lee, S. Kim, and H.-J. Kong. Codeless deep learning of covid-19 chest x-ray image dataset with knime analytics platform. *Healthcare Informatics Research*, 2021. [*Citado en págs. 13, 14, 16, 18, 22 y 24.*]

[13] K.-H. Lai, D. Zha, G. Wang, J. Xu, Y. Zhao, D. Kumar, Y. Chen, P. Zumkhawaka, M. Wan, D. Martinez, and X. Hu. Tods: An automated time series outlier detection system. In *35th AAAI Conference on Artificial Intelligence (AAAI 2021)*, 2021. [*Citado en págs. 13, 14, 16, 18 y 22.*]

[14] M. Karatzoglidi, P. Kerasiotis, and V. Kantere. Automated energy consumption forecasting with enforce. In *Proceedings of the VLDB Endowment*, 2021.
[*Citado en págs. 13, 14, 16, 22, 23 y 25.*]

[15] H. Mendes, S. E. Quincozes, and V. E. Quincozes. A web user interface tool for metaheuristics-based feature selection assessment for idss. In *2022 6th Cyber Security in Networking Conference (CSNet 2022)*, 2022.
[*Citado en págs. 13, 15, 16, 18, 22, 23 y 24.*]

[16] P. Woznicki, F. Laqua, T. Bley, and B. Baeßler. Autoradiomics: A framework for reproducible radiomics research. *Frontiers in Radiology*, 2022.
[*Citado en págs. 14, 16, 18, 21, 22 y 23.*]

[17] R. P. Bonidia, D. S. Domingues, D. S. Sanches, and A. C. P. L. F. De Carvalho. Mathfeature: Feature extraction package for dna, rna and protein sequences based on mathematical descriptors. *Briefings in Bioinformatics*, 2022.
[*Citado en págs. 14, 16, 19, 21, 22 y 24.*]

[18] N. McCombe, X. Ding, G. Prasad, D. P. Finn, S. Todd, P. L. McClean, and K. Wong-Lin. Multiple cost optimisation for alzheimer's disease diagnosis. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2022.
[*Citado en págs. 14, 16, 18, 21, 22, 24 y 25.*]

[19] E. Afgan, A. Nekrutenko, B. A. Grüning, D. Blankenberg, J. Goecks, M. C. Schatz, A. E. Ostrovsky, A. Mahmoud, A. J. Lonie, A. Syme, ..., and P. J. Briggs. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update. *Nucleic Acids Research*, 2022.
[*Citado en págs. 14 y 16.*]

[20] T.-H. Hoang, M. Zehni, H. Xu, G. Heintz, C. Zallek, and M. N. Do. Towards a comprehensive solution for a vision-based digitized neurological examination. *IEEE Journal of Biomedical and Health Informatics*, 2022.
[*Citado en págs. 14, 16, 19, 20, 22, 25 y 26.*]

[21] X. Wu, F. Zhai, A. Chang, J. Wei, Y. Guo, and J. Zhang. Application of machine learning algorithms to predict osteoporosis in postmenopausal women with type 2 diabetes mellitus. *Journal of Endocrinological Investigation*, 2023.
[*Citado en págs. 14, 16, 18 y 21.*]

[22] E. A. Rodríguez Martínez, O. Polezhaeva, F. Marcellin, E. Colin, L. Boyaval, F.-R. Sarhan, and S. Dakpé. Deepsmile: Anomaly detection software for facial movement assessment. *Diagnostics*, 2023.
[*Citado en págs. 14, 16, 22, 23, 25 y 65.*]

[23] P. Hema, A. Darbha, N. Sunny, and R. V. Naganjani. Disease prediction using symptoms based on machine learning algorithms and natural language processing. In *Proceedings of the International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering, ICECONF 2023*, 2023.
[*Citado en págs. 14, 16, 18, 19, 21, 22, 23 y 24.*]

[24] M. Ü. Öziç, M. Tassoker, and F. Yuce. Fully automated detection of osteoporosis stage on panoramic radiographs using yolov5 deep learning model and designing a graphical user interface. *Journal of Medical and Biological Engineering*, 2023.
[*Citado en págs. 14, 16, 18, 19, 21, 22, 23, 24 y 25.*]

[25] O. Dag, M. Kasikci, O. Ilk, and M. Yesiltepe. Geneselectml: A comprehensive way of gene selection for rna-seq data via machine learning algorithms. *Medical and Biological Engineering and Computing*, 2023.
[*Citado en págs. 14, 16, 18, 20, 21, 23 y 31.*]

[26] G. Papoutsoglou, S. Tarazona, M. B. Lopes, T. Klammsteiner, E. Ibrahimi, J. Eckenberger, P. Novielli, A. Tonda, A. Simeon, R. Shigdel, et al. Machine learning approaches in microbiome research: challenges and best practices. *Frontiers in Microbiology*, 2023. [*Citado en págs. 14, 16, 18, 19, 20, 22, 24 y 25.*]

[27] F.-T.-Z. Khanam, A. Al-Naji, A. G. Perera, D. Wang, and J. Chahl. Non-invasive and non-contact automatic jaundice detection of infants based on random forest. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization*, 2023. [*Citado en págs. 14, 16, 22, 23, 24 y 25.*]

[28] D. Guevara-Barrientos and R. Kaundal. Profeatx: A parallelized protein feature extraction suite for machine learning. *Computational and Structural Biotechnology Journal*, 2023. [*Citado en págs. 14, 16, 19, 22, 23 y 26.*]

[29] I. Shafi, M. Sajad, A. Fatima, D. G. Aray, V. Lipari, I. D. L. T. Diez, and I. Ashraf. Teeth lesion detection using deep learning and the internet of things post-covid-19. *Sensors*, 2023. [*Citado en págs. 14, 16, 18, 22, 23, 24 y 26.*]

[30] K. Kamasani, D. Venna, H. Badavathula, and N. Ranga. A deep learning approach to ringworm and vitiligo identification in skin imagery. In *Proceedings of the 5th International Conference on Smart Electronics and Communication, ICOSEC 2024*, 2024. [*Citado en págs. 14, 16, 18, 22, 24, 29 y 31.*]

[31] J. F. Paoli, P. Chatterjee, and M. F. Pollo-Cattaneo. A machine learning-based clinical decision support system for mental health risk profiling. In *CEUR Workshop Proceedings*, 2024. [*Citado en págs. 14, 16 y 22.*]

[32] M. Ma, C. Chen, D. Chen, H. Zhang, X. Du, Q. Sun, L. Fan, H. Kong, X. Chen, C. Cao, and X. Wan. A machine learning–based prediction model for acute kidney injury in patients with community-acquired pneumonia: Multicenter validation study. *Journal of Medical Internet Research*, 2024. [*Citado en págs. 14, 16, 18, 19, 20, 21, 22, 23, 24, 31 y 65.*]

[33] N. Nanthini, D. Aishwarya, A. Simon, N. Baby Vishnupriya, and K. Jeyalakshmi. A novel approach for prediction of the lung disease using deep learning. In *Proceedings - 2024 8th International Conference on Inventive Systems and Control, ICISC 2024*, 2024. [*Citado en págs. 14, 16, 18, 19, 22, 23 y 30.*]

[34] A. Monika, R. Saranya, M. Revathi, and J. G. B. Patturose. Advanced deep learning framework for cancer cell detection using mammography images. In *Proceedings of the 5th International Conference on Data Intelligence and Cognitive Informatics, ICDICI 2024*, 2024. [*Citado en págs. 14, 16, 18, 22, 24 y 30.*]

[35] R. Shukla and T. R. Singh. Alzgenpred - catboost-based gene classifier for predicting alzheimer's disease using high-throughput sequencing data. *Scientific Reports*, 2024. [*Citado en págs. 14, 16, 18, 19, 20, 21, 24, 26 y 65.*]

[36] C. Rajan, A. G. B. Suruthika, B. Sudharshini, and K. Gowtham. Artificial intelligence enabled hybrid machine learning application for dyslexia detection using optimized multiclass support vector machine and personalized interactive and assistive tools using adaptive reinforcement. In *4th International Conference on Innovative Practices in Technology and Management 2024, ICIPTM 2024*, 2024. [*Citado en págs. 14, 16, 18, 21, 22 y 23.*]

[37] R. Subha, B. R. Nayana, R. Radhakrishnan, and P. Sumalatha. Computational intelligence for early detection of infertility in women. *Engineering Applications of Artificial Intelligence*, 2024. [*Citado en págs. 14, 16, 18, 21, 22 y 24.*]

[38] C. Camastra, G. Pasini, A. Stefano, G. Russo, B. Vescio, F. Bini, F. Marinozzi, and A. Augimeri. Development and implementation of an innovative framework for automated radiomics analysis in neuroimaging. *Journal of Imaging*, 2024. [*Citado en págs. 14, 16, 18, 21, 22, 23, 24, 26 y 31.*]

[39] J. Sun, H. Yin, C. Ju, Y. Wang, and Z. Yang. Dtvf: A user-friendly tool for virulence factor prediction based on prott5 and deep transfer learning models. *Genes*, 2024. [*Citado en págs. 14, 16, 19, 22, 23 y 24.*]

[40] M. S. Mabrouk, A. Y. Sayed, H. M. Afifi, M. A. Sheha, and A. Sharwy. Fully automated approach for early detection of pigmented skin lesion diagnosis using abcd. *Journal of Healthcare Informatics Research*, 2020.
[*Citado en págs. 14, 16, 18, 19, 21, 22, 23 y 24.*]

[41] S. Simsek, A. Dag, T. Tiahrt, and A. Oztekin. A bayesian belief network-based probabilistic mechanism to determine patient no-show risk categories. *OMEGA - International Journal of Management Science*, 2021.
[*Citado en págs. 14, 16, 18, 19, 21, 22, 24 y 26.*]

[42] E. Yuzer, V. Dogan, V. Kilic, and M. Sen. Smartphone embedded deep learning approach for highly accurate and automated colorimetric lactate analysis in sweat. *Sensors and Actuators B: Chemical*, 2022.
[*Citado en págs. 14, 16, 18, 20, 21, 22, 23 y 24.*]

[43] M. N. Ibrahim, R. Blazquez-Garcia, A. Lightstone, F. Meng, M. Bhat, A. El Kaffas, and E. Ukwatta. Automated fatty liver disease detection in point-of-care ultrasound b-mode images. *Journal of Medical Imaging*, 2023.
[*Citado en págs. 14, 16, 18 y 25.*]

[44] K. M. Sunnetci and A. Alkan. Biphasic majority voting-based comparative covid-19 diagnosis using chest x-ray images. *Expert Systems with Applications*, 2023. [*Citado en págs. 14, 16, 18, 20, 21, 22 y 24.*]

[45] P. V. S. P. Raghavendra, C. Charitha, K. G. Begum, and V. B. S. Prasath. Deep learning-based skin lesion multi-class classification with global average pooling improvement. *Journal of Digital Imaging*, 2023.
[*Citado en págs. 14, 16, 18, 22 y 24.*]

[46] Y. Zhong, Y. Dan, Y. Cai, J. Lin, X. Huang, O. Mahmoud, E. S. Hald, A. Kumar, Q. Fang, and S. S. Mahmoud. Efficient malaria parasite detection from diverse images of thick blood smears for cross-regional model accuracy. *IEEE Open Journal of Engineering in Medicine and Biology*, 2023.
[*Citado en págs. 14, 16, 18, 20, 21, 22, 23, 24 y 25.*]

[47] S. Mimar, A. S. Paul, N. Lucarelli, S. Border, B. A. Santo, A. Naglah, L. Barisoni, J. Hodgin, A. Z. Rosenberg, W. Clapp, and P. Sarder. Compreps: An automated cloud-based image analysis tool to democratize ai in digital pathology. *bioRxiv: the preprint server for biology*, 2024.
[*Citado en págs. 14, 16, 18, 21, 22, 23 y 24.*]

[48] T. Diot-Dejonghe, B. Leporq, A. Bouhamama, H. Ratiney, F. Pilleul, O. Beuf, and F. Cervenansky. Development of a secure web-based medical imaging analysis platform: The awesomme project. *Journal of Imaging Informatics in Medicine*, 2024. [*Citado en págs. 14, 16, 18, 21, 22 y 23.*]

[49] D. Wang, P. Ram, D. K. I. Weidele, S. Liu, M. Muller, J. D. Weisz, A. Valente, A. Chaudhary, D. Torres, H. Samulowitz, and L. Amini. Autoai: Automating

the end-to-end ai lifecycle with humans-in-the-loop. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, 2020.
[*Citado en págs. 14, 16, 18, 20, 21, 22, 23 y 24.*]

[50] F. Assunção, N. Lourenço, B. Ribeiro, and P. Machado. Evolution of scikit-learn pipelines with dynamic structured grammatical evolution. In *Lecture Notes in Computer Science*. Springer, 2020.
[*Citado en págs. 14, 16, 18, 20, 21, 22, 23, 24, 25, 29 y 65.*]

[51] L. M. Matos, J. Azevedo, A. Matta, A. Pilastri, P. Cortez, and R. Mendes. Categorical attribute transformation environment (cane): A python module for categorical to numeric data preprocessing. *Software Impacts*, 2022.
[*Citado en págs. 14, 16, 19, 22, 23 y 25.*]

[52] X. Wang, A. Hosseininasab, P. Colunga, S. Kadıoğlu, and W.-J. van Hoeve. Seq2pat: Sequence-to-pattern generation for constraint-based sequential pattern mining. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI 2022*, 2022. [*Citado en págs. 14, 16, 19, 22, 23, 24, 25 y 26.*]

[53] T. Petricek, G. J. J. V. D. Burg, A. Nazabal, T. Ceritli, E. Jimenez-Ruiz, and C. K. I. Williams. Ai assistants: A framework for semi-automated data wrangling. *IEEE Transactions on Knowledge and Data Engineering*, 2023.
[*Citado en págs. 14, 16, 19, 20, 21, 22, 24, 25 y 65.*]

[54] U. Nishitha, R. Kandimalla, M. Mourya Vardhan Reddy, and C. Jyotsna. Automobile price prediction using machine learning with data visualization. In *2023 14th International Conference on Computing Communication and Networking Technologies, ICCCNT 2023*, 2023. [*Citado en págs. 14, 16, 21, 22, 23 y 24.*]

[55] K. Territo and J. Romagnoli. Fastman-jmp: All-in-one tool for data mining and model building. *Computer Aided Chemical Engineering*, 2024.
[*Citado en págs. 14, 16, 19, 20, 21, 22 y 24.*]

[56] J. A. Camara Filho, J. M. Monteiro, C. L. Mattos, and J. S. Nobre. A practical guide to support predictive tasks in data science. In *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021), Vol 1*, 2021. [*Citado en págs. 14, 16 y 18.*]

[57] P. Zhang, J. Wu, H. Zhai, and S. Li. Abcmodeller: An automatic data mining tool based on a consistent voting method with a user-friendly graphical interface. *Briefings in Bioinformatics*, 2021.
[*Citado en págs. 14, 16, 18, 21, 22, 23, 24 y 25.*]

[58] A. L. S. Meirelles, T. Kurc, J. Kong, R. Ferreira, J. Saltz, and G. Teodoro. Effective and efficient active learning for deep learning-based tissue image analysis. *Bioinformatics*, 2023. [*Citado en págs. 14, 16, 18, 20, 22, 24 y 25.*]

[59] J. K. Dewangan, H. Patel, A. Sinha, and D. Das. Graphical user interface aided intelligent diagnosis of stator faults in induction motors. In *INDISCON 2024 - 5th IEEE India Council International Subsections Conference: Science,*

Technology and Society, 2024.
[*Citado en págs. 14, 16, 18, 20, 21, 22, 23, 25, 26 y 65.*]

[60] G. L. Scoccia. Automated feature identification for android apps. In *SOFTWARE ENGINEERING AND FORMAL METHODS, SEFM 2019*, 2020.
[*Citado en págs. 14, 16 y 18.*]

[61] T. Xiao, T. Tabashum, B. Metwally, M. V. Albert, A. Du, R. Jebamalaidass, M. Leal, and E. Oliveira. Conversation moderator: A mobile app for tracking individual speaking in group conversations. In *2020 IEEE 14th International Conference on Semantic Computing (ICSC 2020)*, 2020.
[*Citado en págs. 14, 16, 18 y 22.*]

[62] G. Zhou, V. Aggarwal, M. Yin, and D. Yu. Video-based ai decision support system for lifting risk assessment. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2021.
[*Citado en págs. 14, 16, 18, 22, 23, 24, 25, 26 y 65.*]

[63] H. Yang, Y. Du, J. Guo, S. Wei, and H. Ma. Engineering drawing manager: A smart data extractor and management software. In *Intelligent Robotics and Applications (ICIRA 2022), Pt IV*, 2022. [*Citado en págs. 14, 16 y 18.*]

[64] S. Liang, Z. Fei, J. Wu, and X. Lin. Tree-based pipeline optimization-based automated-machine learning model for performance prediction of materials and structures: Case studies and ui design. *Structural Control & Health Monitoring*, 2024. [*Citado en págs. 14, 16, 18, 21, 22, 23, 24, 25 y 65.*]

[65] T. Veerraju, P. K. Mani, U. Jhanani Shree, V. G. Pratheep, P. Valarmathi, and B. U. Maheswari. Ml and iot-based soil type analysis and prediction system. In *3rd International Conference on Smart Electronics and Communication, ICOSEC 2022 - Proceedings*, 2022. [*Citado en págs. 14, 16, 18, 21, 22, 23 y 26.*]

[66] V. Sagan, R. Coral, S. Bhadra, H. Alifu, O. Al Akkad, A. Giri, and F. Esposito. Hyperfidelis: A software toolkit to empower precision agriculture with geoai. *Remote Sensing*, 2024. [*Citado en págs. 14, 16, 18, 19, 22, 23, 24 y 29.*]

[67] P. Liang, C.-Z. Qin, and A.-X. Zhu. Using automated machine learning for spatial prediction—the heshan soil subgroups case study. *Land*, 2024.
[*Citado en págs. 14, 16, 18, 20, 21, 22, 23, 24, 25 y 26.*]

[68] H. F. Robinson and J. N. A. Vink. Rapid and robust monitoring of phytophthora infectivity through detached leaf assays with automated image analysis. In *Methods in Molecular Biology*. Humana Press, 2025.
[*Citado en págs. 14, 16 y 18.*]

[69] P. Xu, R. Yang, T. Zeng, J. Zhang, Y. Zhang, and Q. Tan. Varietal classification of maize seeds using computer vision and machine learning techniques. *Journal of Food Process Engineering*, 2021. [*Citado en págs. 14, 16, 18, 20, 23 y 24.*]

[70] M. Aghababaei, A. Ebrahimi, A. A. Naghipour, E. Asadi, A. Perez-Suay, M. Morata, J. L. Garcia, J. P. Rivera Caicedo, and J. Verrelst. Introducing artmo's machine-learning classification algorithms toolbox: Application to plant-type detection in a semi-steppe iranian landscape. *Remote Sensing*, 2022. [*Citado en págs. 14, 16 y 18.*]

[71] D. López-García, J. M. G. Peñalver, J. M. Górriz, and M. Ruz. Mvpalab: A machine learning decoding toolbox for multidimensional electroencephalography data. *Computer Methods and Programs in Biomedicine*, 2022. [*Citado en págs. 14, 16, 19, 20, 22 y 25.*]

[72] K. Kaur, P. Khandnor, and A. Khosla. Classification of meditation expertise from eeg signals using shallow neural networks. In *Communications in Computer and Information Science*. 2024. [*Citado en págs. 14, 16 y 18.*]

[73] E. Serin, N. Vaidya, H. Walter, and J. D. Kruschwitz. Nbs-predict: An easy-to-use toolbox for connectome-based machine learning. In *Neuromethods*. Humana Press, 2025. [*Citado en págs. 14, 16, 18, 20, 21, 22, 23, 25, 29 y 65.*]

[74] A. Parashar, A. Parashar, and R. S. Shekhawat. A robust covariate-invariant gait recognition based on pose features. *IET Biometrics*, 2022. [*Citado en págs. 14, 16, 18, 22 y 24.*]

[75] A. Das, Pramod, and S. Praveen Kumar. An enhanced optimization model with ensemble autoencoder for zero-day attack detection. *Journal of Theoretical and Applied Information Technology*, 2022. [*Citado en págs. 15 y 16.*]

[76] M. Harth, C. Schorr, and R. Krieger. A hierarchical multi-level product classification workbench for retail. In *CEUR Workshop Proceedings*, 2020. [*Citado en págs. 15, 16, 22, 24, 26 y 65.*]

[77] N. Uddin, M. K. Uddin Ahamed, M. A. Uddin, M. M. Islam, M. A. Talukder, and S. Aryal. An ensemble machine learning based bank loan approval predictions system with a smart application. *International Journal of Cognitive Computing in Engineering*, 2023. [*Citado en págs. 15, 16, 21, 22, 24, 29 y 65.*]

[78] R. Damdoo, S. Choulwar, A. Nagbhidkar, A. Yadav, and P. Vyawahare. Developing a speech recognition system for educational settings. In *15th International Conference on Advances in Computing, Control, and Telecommunication Technologies, ACT 2024*, 2024. [*Citado en págs. 15, 16, 18, 22, 23 y 24.*]

[79] A. A. Ahmed and S. Ahmed. A real-time car towing management system using ml-powered automatic number plate recognition. *Algorithms*, 2021. [*Citado en págs. 15 y 16.*]

[80] K. Opasjumruskit, S. Boening, S. Schindler, and D. Peters. Ontohuman: Ontology-based information extraction tools with human-in-the-loop interaction. In *Cooperative Design, Visualization, and Engineering (CDVE 2022)*, 2022. [*Citado en págs. 15, 16 y 19.*]

[81] Erin LeDell and Sebastien Poirier. H2O AutoML: Scalable Automatic Machine Learning. In *Proceedings of the 7th ICML Workshop on Automated Machine Learning (AutoML 2020)*. PMLR, 2020. H2O.ai, USA. [*Citado en pág. 31.*]