

Diagnóstico de enfermedad hepática mediante técnicas de aprendizaje automático y su implementación en una aplicación web.

Santiago González Berruga

Máster Universitario de Bioinformática y Bioestadística UOC - UB
Inteligencia artificial y desarrollo web

Consultora: Romina Astrid Rebrij

Profesor responsable de la asignatura: Antoni Pérez Navarro

Diciembre 2021



Esta obra está sujeta a una licencia de
Reconocimiento-NoComercial-CompartirIgual
[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Diagnóstico de enfermedad hepática mediante técnicas de aprendizaje automático y su implementación en una aplicación web.</i>
Nombre del autor:	<i>Santiago González Berruga</i>
Nombre del consultor/a:	<i>Romina Astrid Rebrij</i>
Nombre del PRA:	<i>Antoni Pérez Navarro</i>
Fecha de entrega:	12/2021
Titulación:	<i>Máster Universitario de Bioinformática y Bioestadística (interuniversitario: UOC, UB)</i>
Área del Trabajo Final:	<i>Inteligencia artificial y desarrollo web</i>
Idioma del trabajo:	Castellano
Número de créditos:	15
Palabras clave	Aprendizaje automático; Redes neuronales artificiales; Enfermedad hepática
Resumen del Trabajo:	
<p>Las enfermedades hepáticas se han incrementado notablemente en los últimos años debido a los cambios en los hábitos de vida, siendo una de las principales causas de mortalidad a nivel mundial. Sin embargo, el diagnóstico de las enfermedades hepáticas sigue siendo complejo, costoso y la mayoría de las veces tardío.</p> <p>Este trabajo busca un modelo de clasificación automático que permita realizar un diagnóstico precoz y sencillo de pacientes hepáticos. Para ello, se generan modelos a partir del conjunto de datos de pacientes hepáticos ILPD y los algoritmos de aprendizaje automático K-nearest neighbor (KNN), Naive Bayes (NB), Decision tree (DT), Random Forest (RF), Support Vector Machines (SVM), Artificial Neural Network (ANN) y Logistic Regression (LR).</p> <p>Para determinar el mejor modelo se emplearon las métricas tasa de éxito, falsos negativos, falsos positivos, tasa de error, estadístico kappa, sensibilidad, especificidad, precisión, recall y F1-score. En base a esto, los modelos ANN y RF muestran mejores resultados que el resto de modelos para la predicción de pacientes hepáticos, con una tasa de éxito del 75,1% y 74,6% y una precisión del 76,8% y 75,3%.</p> <p>Por lo tanto, este trabajo demuestra que es posible diagnosticar pacientes hepáticos empleando modelos de clasificación automáticos entrenados con</p>	

variables clínicas sencillas de obtener, sin tener que recurrir a métodos invasivos para el paciente.

Además, se ha implementado el modelo ANN en una aplicación web, generando una herramienta única y con gran potencial para apoyar a profesionales sanitarios durante el diagnóstico de enfermedades hepáticas, permitiendo un diagnóstico precoz sin necesidad de técnicas intrusivas.

Abstract:

Liver diseases have increased considerably in recent years due to changes in lifestyle habits and are one of the leading causes of mortality worldwide. However, the diagnosis of liver diseases remains complex, expensive and most of the times late.

This work seeks an automatic classification model that allows for an early and simple diagnosis of liver patients. To this end, models are generated using the ILPD liver patient dataset and the machine learning algorithms K-nearest neighbour (KNN), Naive Bayes (NB), Decision tree (DT), Random Forest (RF), Support Vector Machines (SVM), Artificial Neural Network (ANN) and Logistic Regression (LR).

To determine the best model, the metrics accuracy, false negatives, false positives, error rate, kappa statistic, sensitivity, specificity, precision, recall and F1-score were used. Based on this, the ANN and RF models showed better results than the other models for the prediction of liver patients, with an accuracy of 75.1% and 74.6% and precision of 76.8% and 75.3%.

Therefore, this work has demonstrated that it is possible to diagnose liver patients using automatic classification models trained with simple clinical variables, without having to use invasive methods on the patients.

Furthermore, the ANN model has been implemented in a web application, generating a unique tool with great potential to support healthcare professionals during the diagnosis of liver diseases, allowing an early diagnosis without the need of intrusive techniques.

Índice

1. Resumen	1
2. Introducción	1
2.1. Contexto y justificación del Trabajo	1
2.2. Objetivos del Trabajo	4
2.3. Enfoque y método seguido	4
2.4. Planificación del Trabajo	8
2.5. Breve resumen de contribuciones y productos obtenidos	11
2.6. Breve descripción de los otros capítulos de la memoria	11
3. Estado del arte	12
3.1. Algoritmos de aprendizaje automático	14
4. Metodología	17
4.1. Software	17
4.2. Modelos de aprendizaje automático	17
4.2.1. Datos	17
4.2.2. Métodos	18
4.3. Aplicación web	26
4.3.1. Funcionalidades de la aplicación web	26
4.3.2. Estructura de la aplicación web	26
4.3.3. Bocetos	27
4.3.4. Implementación en Shiny	27
4.3.5. Publicar la aplicación web	29
5. Resultados	29
5.1. Rendimiento de los modelos	29
5.2. Aplicación web	30
6. Discusión	35
7. Conclusiones	36
7.1. Conclusiones	36
7.2. Líneas de futuro	37
7.3. Seguimiento de la planificación	37
Glosario	38
Bibliografía	40
Anexo: Gráficos complementarios	45

Lista de figuras

Figura 1. Principales funciones del hígado en el cuerpo humano.....	2
Figura 2. Metodología.	6
Figura 3. Matriz de confusión.....	7
Figura 4. Diagrama de Gantt.....	10
Figura 5. Gráfico de tarta de la variable Class.	19
Figura 6. Boxplots de las variables numéricas del conjunto ILPD.	19
Figura 7. Gráficos de dispersión de las variables numéricas del conjunto ILPD.	20
Figura 8. Correlograma de las variables numéricas del conjunto ILPD.	21
Figura 9. Histogramas de las variables numéricas del conjunto ILPD.	21
Figura 10. Importancia de las variables del conjunto ILPD.	22
Figura 11. Bocetos de la aplicación web.	27
Figura 12. Aspecto inicial al abrir la aplicación web.....	30
Figura 13. Configuración de la aplicación web para realizar una predicción única e introducir manualmente los datos mediante un formulario.	31
Figura 14. Pop-up de ayuda de la aplicación web.	31
Figura 15. Ejemplo de pop-up indicador de error de la aplicación web.	31
Figura 16. Pop-up de la aplicación web para indicar una predicción en progreso.	32
Figura 17. Mensajes informativos del formulario manual.	32
Figura 18. Resultados de una predicción única.....	33
Figura 19. Resultados de una predicción múltiple.	33
Figura 20. Informe en pdf generado automáticamente en la aplicación web.	34
Figura 21. Aplicación web abierta desde un dispositivo móvil.	35
Figura 22. Gráficos de densidad y boxplots por clase de la edad (Age).	45
Figura 23. Gráficos de densidad y boxplots por clase del logaritmo de la bilirrubina total (TB).....	46
Figura 24. Gráficos de densidad y boxplots por clase del logaritmo de la bilirrubina directa (DB).....	46
Figura 25. Gráficos de densidad y boxplots por clase del logaritmo de la fosfatasa alcalina (Alkphos).....	47
Figura 26. Gráficos de densidad y boxplots por clase del logaritmo de la alanina aminotransferasa (Sgpt).	47

Figura 27. Gráficos de densidad y boxplots por clase del logaritmo de la aspartato aminotransferasa (Sgot).	48
Figura 28. Gráficos de densidad y boxplots por clase de las proteínas totales (TP).	48
Figura 29. Gráficos de densidad y boxplots por clase de la albúmina (ALB).	49
Figura 30. Gráficos de densidad y boxplots por clase de la proporción entre albúmina y globulina (AG).	49

Lista de tablas

Tabla 1. Planificación de las fechas de inicio/final y horas estimadas para cada tarea...	8
Tabla 2. Análisis de riesgos del Trabajo Final de Máster.....	9
Tabla 3. Fortalezas y debilidades de los grupos de algoritmos empleados.....	16
Tabla 4. Variables del conjunto de datos ILPD.....	18
Tabla 5. Análisis estadístico de las variables numéricas del conjunto ILPD.....	18
Tabla 6. Algoritmos de aprendizaje automático empleados.....	24
Tabla 7. Variables necesarias para realizar predicciones en la aplicación web.....	28
Tabla 8. Rendimiento de los modelos de aprendizaje automático.....	30

1. Resumen

Las enfermedades hepáticas se han incrementado notablemente en los últimos años debido a los cambios en los hábitos de vida, convirtiéndose en una de las principales causas de mortalidad a nivel mundial. Sin embargo, el diagnóstico de las enfermedades hepáticas sigue siendo complejo, costoso y la mayoría de las veces tardío.

Este trabajo busca un modelo de clasificación automático que permita realizar un diagnóstico precoz y sencillo de pacientes hepáticos. Para ello, se generan modelos a partir del conjunto de datos de pacientes hepáticos ILPD y los algoritmos de aprendizaje automático K-nearest neighbor (KNN), Naive Bayes (NB), Decision tree (DT), Random Forest (RF), Support Vector Machines (SVM), Artificial Neural Network (ANN) y Logistic Regression (LR). Para determinar el mejor modelo se emplean las métricas tasa de éxito, falsos negativos, falsos positivos, tasa de error, estadístico kappa, sensibilidad, especificidad, precisión, recall y F1-score. En base a esto, los modelos ANN y RF muestran mejores resultados que el resto de modelos para la predicción de pacientes hepáticos, con una tasa de éxito del 75,1% y 74,6% y una precisión del 76,8% y 75,3%.

Por lo tanto, este trabajo demuestra que es posible diagnosticar pacientes hepáticos empleando modelos de clasificación automáticos entrenados con variables clínicas sencillas de obtener, sin tener que recurrir a métodos invasivos para el paciente. Además, se ha implementado el modelo ANN en una aplicación web, generando una herramienta única y con gran potencial para apoyar a profesionales sanitarios durante el diagnóstico de enfermedades hepáticas, permitiendo un diagnóstico precoz sin necesidad de técnicas intrusivas.

2. Introducción

2.1. Contexto y justificación del Trabajo

El hígado es un órgano indispensable para el funcionamiento del cuerpo humano. Las funciones que desempeña son muy variadas e importantes (Figura 1): controla el aporte alimenticio, la concentración de muchas sustancias químicas y de desecho en sangre, la fabricación de bilis, la síntesis de proteínas, la regulación del metabolismo de hormonas y la inmunidad^[1,2]. Por lo tanto, es muy importante mantener el hígado en buen estado. Sin embargo, uno de los principales problemas de salud mundial es el número creciente de pacientes con enfermedades hepáticas^[3] y que en gran medida está desatendido en los países menos desarrollados^[4]. Las enfermedades hepáticas representan una causa importante de mortalidad en el mundo occidental, más aún en los países de Asia y África, donde constituye una de las tres principales causas de muerte^[1,5]. El continuo aumento de pacientes con enfermedad hepática se debe al consumo excesivo de alcohol, la inhalación de gases nocivos, la ingesta de alimentos contaminados, encurtidos y drogas. Todo estos factores aumentan en paralelo con la prevalencia de diabetes, síndrome metabólico y obesidad^[4,6,7].

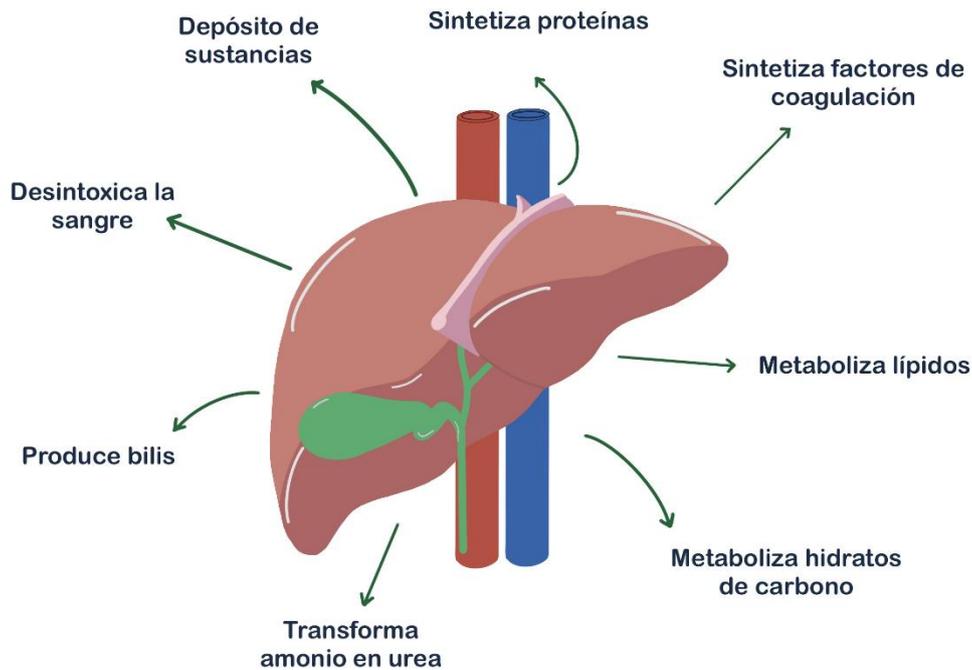


Figura 1. Principales funciones del hígado en el cuerpo humano.

Las principales enfermedades hepáticas son: hepatitis víricas A, B, C, D y E, cirrosis, enfermedad de hígado asociada al alcohol (ALD por sus siglas en inglés), enfermedad de hígado graso no alcohólico (NAFLD por sus siglas en inglés), Hepatitis autoinmune y carcinoma hepatocelular [1,4,8-12]. En 2010, las enfermedades hepáticas representaron aproximadamente 2 millones de muertes por año en todo el mundo, que se pueden dividir principalmente en 1 millón debido a complicaciones de la cirrosis y 1 millón debido a hepatitis viral y carcinoma hepatocelular. La cirrosis fue la undécima causa de muerte más común a nivel mundial y el cáncer de hígado la decimosexta causa de muerte, por lo que juntos representaron el 3,5% de las muertes en todo el mundo [8].

Las enfermedades hepáticas se manifiestan de forma variable y pueden ser agudas o crónicas. Se denominan agudas cuando comienzan de forma súbita y crónicas cuando causan daño a lo largo del tiempo, incluso tras desaparecer los síntomas. Además, aproximadamente la mitad de los pacientes con enfermedad hepática son asintomáticos.

Sin embargo, a pesar de los avances médicos, el diagnóstico de las enfermedades hepáticas sigue siendo complejo, lento, costoso, en ocasiones peligroso y la mayoría de las veces tardío, ya que no se detectan síntomas hasta que la enfermedad está en fases muy avanzadas [1,13]. Su diagnóstico en una etapa inicial es muy difícil porque el hígado funciona correctamente incluso si está parcialmente dañado y, en general, los pacientes no presentan síntomas a menos que la enfermedad haya progresado considerablemente, aumentando el riesgo y dificultando el tratamiento, prevención y control de la enfermedad [6,14-16]. Por ejemplo, algunas de las pruebas más usadas para el diagnóstico de NAFLD son la biopsia hepática, la resonancia magnética, la ecografía y las pruebas de enzimas hepáticas. Sin embargo, la naturaleza invasiva de las biopsias, el

alto coste de las resonancias magnéticas, la naturaleza no cuantitativa y la baja sensibilidad de los ultrasonidos convencionales y la baja precisión de las pruebas de enzimas hepáticas son grandes limitaciones [1,13,17-19].

Frecuentemente se realizan pruebas de función hepática, tanto a pacientes sintomáticos como asintomáticos, como los test bioquímicos, niveles de bilirrubina sérica, albúmina o pruebas de coagulación como la actividad de protrombina [13]. Sin embargo, se necesitan especialistas para la correcta evaluación de estas pruebas, ocasionando grandes problemas para dar un diagnóstico rápido y correcto. Para abordar esta limitación, se han desarrollado varios índices de predicción de la grasa hepática, pero ninguno de ellos tiene una capacidad predictiva lo suficientemente alta como para ser considerado un estándar [17]. Por lo tanto, facilitar la identificación de pacientes en etapas iniciales de enfermedades hepáticas sigue siendo una tarea pendiente para mejorar la prevención, el tratamiento adecuado del paciente y aliviar la carga de trabajo en los hospitales [2].

Para solucionar este problema, se han desarrollado numerosos modelos predictivos mediante métodos estadísticos tradicionales con el objetivo de diagnosticar enfermedades hepáticas: índice de hígado graso (FLI), índice de esteatosis hepática (HSI), índice Child-Pugh, modelo para enfermedades hepáticas en etapas terminales (MELD), escalas ajustadas de MELD, índice albúmina-bilirrubina, índice de descompensación aguda del Consorcio de insuficiencia hepática crónica (CLIF) e índice CLIF de insuficiencia orgánica secuencial [20,21]. Sin embargo, hoy en día las investigaciones se centran en buscar y crear modelos predictivos y de clasificación haciendo uso del aprendizaje automático, llegando a demostrar un gran potencial de estas técnicas en el diagnóstico de enfermedades [2,3,22,23]. Estos resultados están haciendo que la popularidad de estas técnicas de clasificación automáticas aumente rápidamente como ayuda para el diagnóstico médico [6,20].

El aprendizaje automático (*Machine learning* en inglés) es un campo de la inteligencia artificial que permite generar modelos predictivos de manera eficiente y efectiva a través de la detección de patrones ocultos dentro de grandes conjuntos de datos. La clasificación de datos en el aprendizaje automático es un proceso de dos fases: la primera es entrenar un modelo con un conjunto de datos de entrenamiento y la segunda es la evaluación de su rendimiento con un conjunto de datos de prueba [2,6,24]. Con esto en mente, se pueden emplear dichas técnicas para realizar diagnósticos más tempranos de las enfermedades hepáticas utilizando los datos obtenidos mediante pruebas de función hepática, reduciendo así los costes, evitando los métodos más intrusivos como la biopsia y reduciendo la carga de trabajo de los especialistas.

Se han realizado numerosos estudios buscando el mejor algoritmo de aprendizaje automático para el diagnóstico de enfermedades hepáticas. Los algoritmos más empleados son: K-nearest neighbor (KNN), Decision tree (DT), Naïve Bayes (NB), Random Forest (RF), Support Vector Machines (SVM), Artificial Neural Network (ANN) [2,3,6,7,14-16,20,23,25-38]. Estos estudios obtuvieron resultados positivos clasificando correctamente a los pacientes con enfermedad hepática, presentando un rendimiento mejor o similar al de los enfoques tradicionales de modelado estadístico [20,21]. Sin embargo, no se ha encontrado un algoritmo que mantenga un rendimiento constante con diferentes conjuntos de datos ya que, en función de las características del conjunto de entrenamiento, el mejor rendimiento será de un algoritmo u otro [16,21].

En conclusión, los modelos de aprendizaje automático para el diagnóstico de enfermedades hepáticas pueden ahorrar tiempo y dinero, mejorando la detección de la enfermedad en etapas tempranas [15,27,33]. La propuesta de este trabajo sigue la línea de los estudios previos, que es buscar y crear el mejor modelo posible para el diagnóstico automático de enfermedades hepáticas. Además, a diferencia del resto de estudios, el modelo se implementa en una aplicación web sencilla e intuitiva para que todos los usuarios puedan hacer uso del modelo predictivo, tengan o no conocimientos técnicos sobre aprendizaje automático.

2.2. Objetivos del Trabajo

Los objetivos generales y específicos de este trabajo son:

- **Objetivos Generales:**
 - 1) Crear un modelo de aprendizaje automático para clasificar pacientes de enfermedad hepática a partir de distintos parámetros bioquímicos.
 - 2) Desarrollar una aplicación web que implemente el modelo creado.
- **Objetivos Específicos:**
 - 1) Analizar el conjunto de datos para seleccionar los mejores predictores.
 - 2) Entrenar, evaluar y optimizar distintos algoritmos de aprendizaje automático.
 - 3) Determinar el mejor modelo de predicción con una precisión mínima del 80%.
 - 4) Crear una aplicación web sencilla e intuitiva para el usuario que permita realizar predicciones y visualizar los resultados.

2.3. Enfoque y método seguido

Para realizar este trabajo, se emplea el aprendizaje automático. Los algoritmos de aprendizaje automático pueden agruparse en tres grandes bloques [24,39,40]:

- *Aprendizaje supervisado*: Son métodos en los que se proporciona un conjunto de entrenamiento etiquetado con instrucciones claras sobre lo que se necesita aprender y cómo hacerlo. Se utiliza para generar modelos predictivos como clasificación y predicción numérica.
- *Aprendizaje no supervisado*: En este caso, no hay un conjunto de datos de entrenamiento ni ninguna etiqueta y los datos de salida no son conocidos. Se utiliza para generar modelos descriptivos, como descubrimiento de patrones y agrupación (*clustering* en inglés).
- *Aprendizaje por refuerzo*: El algoritmo no tiene información de ninguna categoría deseada y sólo es informado cuando la respuesta es incorrecta, pero en caso de fallo no se le indica cómo corregirlo ni cual es la respuesta correcta. Por lo tanto, el algoritmo tiene que explorar y probar varias posibilidades hasta encontrar la respuesta correcta. Sin embargo, si trabajamos solo con dos categorías y con igual peso de los errores, entonces el aprendizaje por refuerzo es equivalente al aprendizaje supervisado.

Uno de los objetivos principales del trabajo es la obtención de un modelo predictivo para el diagnóstico de enfermedades hepáticas mediante técnicas de aprendizaje automático. Por lo tanto, el modelo predictivo en este caso realizaría una tarea de clasificación, ya que queremos predecir si el paciente sufre enfermedad hepática o no. La clasificación hace uso principalmente de algoritmos de aprendizaje supervisado y por refuerzo, ya que los algoritmos de aprendizaje no supervisado se centran en tareas de agrupación ^[24]. En nuestro caso, como trabajaremos solo con dos categorías (enfermedad hepática o no) el aprendizaje por refuerzo es equivalente al aprendizaje supervisado. Por lo tanto, usamos algunos de los algoritmos de clasificación supervisados más populares y con mejores resultados en estudios previos clasificando enfermedades hepáticas ^[16,21,22]. Los principales grupos de algoritmos que vamos a emplear durante el trabajo son *k-Nearest Neighbor* (kNN), *Naïve Bayes* (NB), *Artificial Neural Network* (ANN), *Support Vector Machine* (SVM), *Decision tree* (DT), *Random Forest* (RF) y *Logistic Regression* (LR).

Actualmente, la mayoría de los trabajos de aprendizaje automático siguen una metodología basada en el proceso estándar multisectorial para la minería de datos (CRISP-DM por sus siglas en inglés). CRISP-DM divide el proceso de minería de datos en seis pasos: entender el mercado, entender los datos, preparación de los datos, modelado, evaluación y despliegue ^[41]. Normalmente, en el aprendizaje automático no se aplican como tal los pasos de CRISP-DM, si no que se modifican en un proceso de 5 pasos: Obtención de los datos, exploración y preparación de los datos, entrenamiento del modelo, evaluación del modelo y mejora del modelo ^[24]. Este trabajo subdivide aún más los 5 pasos anteriores obteniendo un proceso de 7 pasos (Figura 2) que indicamos a continuación:

- 1) **Obtener los datos:** Consiste en obtener los datos con los que entrenaremos y evaluaremos los diferentes modelos de aprendizaje automático. En nuestro caso, trabajaremos con el conjunto de datos ILPD (Indian Liver Patient Dataset) ^[6] obtenido en el repositorio de aprendizaje automático de la Universidad de California en Irvine (UCI) ^[42]. Este conjunto de datos contiene 416 registros de pacientes hepáticos y 167 registros de pacientes no hepáticos. Los datos se obtuvieron en el noreste de Andhra Pradesh, India. Está formado por diez variables comúnmente empleadas en el diagnóstico de enfermedades hepáticas ^[1,5,13]: edad, género y las pruebas de función hepática Bilirrubina total, Bilirrubina directa, Proteínas totales, Albúmina, Ratio albúmina/globulina, Alanina aminotransferasa, Aspartato aminotransferasa y Fosfatasa alcalina.
- 2) **Explorar los datos:** La calidad de los datos es un factor determinante de los resultados de un proyecto de aprendizaje automático. Por lo tanto, es importante saber qué tipo de datos son y qué características presentan.
- 3) **Preprocesar los datos:** En muchas ocasiones tras explorar los datos es necesario realizar un preprocesamiento de estos: como corregir y/o limpiar los datos faltantes, eliminar los datos innecesarios, volver a codificar los datos para que se ajusten a los requerimientos de los algoritmos, etc. Normalmente, no todos los algoritmos de clasificación tienen los mismos requerimientos sobre el conjunto de datos, por lo que hay que preparar correctamente los datos para cada uno de ellos.

En este paso, también se divide el conjunto de datos en dos subconjuntos, uno de entrenamiento (70%) y otro de prueba (30%), intentando que ambos contengan la misma proporción de clases que el conjunto original.

- 4) **Selección de predictores:** Los conjuntos de datos de grandes dimensiones presentan un gran desafío para los métodos de aprendizaje. Una gran cantidad de predictores o variables puede tender a sobreajustarse y provocar una disminución del rendimiento del modelo ^[43]. La selección de predictores para el modelo de clasificación intenta seleccionar un subconjunto de tamaño mínimo siempre que la precisión de la clasificación aumente.
- 5) **Entrenar el modelo:** Tras preparar y seleccionar los datos, se emplea el subconjunto de entrenamiento para entrenar los diferentes algoritmos de clasificación, de modo que representen los datos en forma de modelos.
- 6) **Evaluar el modelo:** Debido a que cada modelo de aprendizaje automático da como resultado una solución sesgada al problema de aprendizaje, es importante evaluar el rendimiento de los modelos generados. Para ello, se pide al modelo que prediga las clases del conjunto de datos de prueba y se compara el resultado con las clases originales, generando una matriz de confusión donde se indican los verdaderos positivos y negativos, así como los falsos positivos y negativos (Figura 3). A partir de esta matriz de confusión se pueden calcular diferentes métricas que evalúan el rendimiento del modelo como: tasa de éxito (accuracy en inglés), sensibilidad, especificidad, precisión, recall, estadístico kappa y F1-score.
- 7) **Mejorar y evaluar el modelo:** Para mejorar el rendimiento de los modelos se modifican los parámetros de los algoritmos de modo que aumentemos el rendimiento de los modelos generados.



Figura 2. Metodología seguida para la creación de modelos de aprendizaje automático en este trabajo.

Para trabajar con aprendizaje automático existen dos lenguajes de programación predominantes en el mercado, que son R y Python ^[24,39]. Ambos son softwares libres, lo que ha permitido su gran expansión y que numerosas investigaciones hagan uso de ellos. En nuestro caso se ha empleado R por familiaridad con este lenguaje y para profundizar más en el uso de algoritmos de aprendizaje automático, ya que proporciona un gran soporte para el aprendizaje automático en forma de numerosos paquetes creados por la comunidad.

		Valores reales	
		A	B
Valores predichos	A	Verdaderos Positivos (TP)	Falsos Positivos (FP)
	B	Falsos Negativos (FN)	Verdaderos Negativos (TN)

Figura 3. Matriz de confusión.

Otro de los objetivos principales es desarrollar una aplicación web que implemente el mejor modelo de clasificación obtenido. En este caso, como empleamos R para desarrollar los modelos de clasificación, la mejor opción para desarrollar la aplicación web es usar Shiny ya que permite integrar en la aplicación web código y resultados de R ^[44]. Además, también permite el uso de HTML y CSS, lenguajes de programación que emplearemos para maquetar la aplicación de forma que sea lo más sencilla e intuitiva para el usuario.

En base a esta metodología y teniendo en cuenta la redacción de la memoria y presentación del trabajo, las principales herramientas empleadas son:

- **RStudio:** Es un entorno gráfico que permite trabajar con el lenguaje de programación R. Se empleó para generar modelos de aprendizaje automático.
- **Shiny:** Es un paquete de R que facilita la creación de aplicaciones web interactivas directamente desde R. Se empleó para el desarrollo de la aplicación web junto con los lenguajes de programación HTML y CSS para maquetar la aplicación.
- **Github:** Es una plataforma que aloja repositorios de Git que se empleó para mantener el código y llevar un control de las distintas versiones.
- **Microsoft Word:** Es un editor de texto que se empleó para redactar los diferentes informes de este trabajo.
- **Microsoft PowerPoint:** Es un programa de presentación que se empleó para elaborar la presentación final del trabajo.
- **Google Drive:** Es un servicio de almacenamiento de datos que se empleó para mantener los informes y controlar las versiones de los informes del trabajo. También permite una fácil revisión de los trabajos.
- **Mendeley:** Es un gestor bibliográfico que se empleó para organizar y almacenar la bibliografía consultada.

2.4. Planificación del Trabajo

A continuación, se muestra la Tabla 1 con las tareas realizadas durante el proyecto junto a su estimación en horas, su fecha de inicio y su fecha de fin, seguidas de un diagrama de Gantt que muestra la organización temporal de las tareas (Figura 4). Esta planificación se basa en un tiempo total de unas 375 horas (15 créditos ECTS) que es lo estimado para el trabajo, divididas entre las 16 semanas disponibles.

Tabla 1. Planificación de las fechas de inicio/final y horas estimadas para cada tarea.

Tarea	Inicio	Fin	Horas estimadas
Elegir temática del proyecto	15/09/21	18/09/21	10
Buscar bibliografía	15/09/21	24/12/21	40
Redactar contexto y justificación del trabajo	18/09/21	20/09/21	10
Definir objetivos	19/09/21	20/09/21	1
Preparar la PEC0	20/09/21	22/09/21	4
Definir el enfoque, etapas y métodos a seguir	23/09/21	27/09/21	15
Planificar el trabajo	27/09/21	29/09/21	6
Evaluar los riesgos	29/09/21	29/09/21	1
Definir los resultados esperados	30/09/21	30/09/21	2
Definir la estructura del proyecto	30/09/21	30/09/21	1
Preparar la PEC1	30/09/21	04/10/21	4
Explorar paquetes de Aprendizaje automático en R	03/10/21	05/10/21	6
Aprender a usar Github	05/10/21	06/10/21	2
Explorar los datos	07/10/21	10/10/21	10
Preprocesar los datos	10/10/21	14/10/21	15
Seleccionar las variables	15/10/21	18/10/21	15
Construir los modelos de aprendizaje automático	19/10/21	22/10/21	15
Entrenar los modelos	23/10/21	24/10/21	8
Evaluar los modelos	25/10/21	27/10/21	7
Mejorar los modelos	27/10/21	31/11/21	15
Evaluar los modelos	01/11/21	03/11/21	7
Seleccionar el mejor modelo	04/11/21	05/11/21	4
Preparar la PEC2	05/11/21	08/11/21	4
Aprender a desarrollar aplicaciones web con Shiny	07/11/21	14/11/21	30
Prototipado de la aplicación web	15/11/21	16/11/21	8
Desarrollar la aplicación web	17/11/21	03/12/21	40
Subir la aplicación al servidor Shiny	04/12/21	06/12/21	10
Preparar la PEC3	07/12/21	09/12/21	4
Elaborar la memoria	18/09/21	24/12/21	50
Elaborar las diapositivas	25/12/21	29/12/21	20
Grabar y editar el video de presentación	30/12/21	03/01/22	10
Defender el trabajo	13/01/22	21/01/22	1

Siguiendo esta planificación, tras finalizar el proyecto se ha cumplido con cada uno de los objetivos y plazos propuestos, estos eran:

- **PEC 0 (22/09/21):** Finalizar con el tema escogido, su contexto, justificación y la definición de los objetivos.
- **PEC 1 (04/10/21):** Finalizar con el enfoque, metodología a seguir durante el proyecto, una planificación clara, análisis de riesgos y resultados esperados.
- **PEC 2 (08/11/21):** Finalizar el estudio de los modelos creados y seleccionar el que presente mejor rendimiento.
- **PEC 3 (09/12/21):** Finalizar el desarrollo de la aplicación web que implemente el modelo seleccionado.
- **PEC 4 (24/12/21):** Finalizar la memoria.
- **PEC 5a (03/01/22):** Finalizar el video de presentación.
- **PEC 5b (21/01/22):** Realizar la defensa del Trabajo Final de Máster (TFM).

También se realizó un análisis de riesgos (Tabla 2) evaluando los peligros potenciales que pudieran surgir durante la elaboración del trabajo y las posibles actuaciones para mitigarlos en caso necesario.

Tabla 2. Análisis de riesgos del Trabajo Final de Máster.

Descripción del riesgo	Severidad	Probabilidad	Mitigación
Fallo de las librerías de R	Alta	Baja	Buscar si alguna actualización resuelve el conflicto y, en caso de no ser así, buscar una librería alternativa.
Conflicto entre librerías	Moderada	Moderada	Ver las librerías afectadas y encontrar solución entre ellas. Buscar si alguna actualización resuelve el conflicto. Buscar librerías alternativas.
No alcanzar la precisión mínima del 80% en el modelo	Alta	Moderada	Uso de diferentes algoritmos y encontrar parámetros que los mejoren. En caso de que esto no funcione reduciremos la precisión mínima exigida.
Pérdida del trabajo desarrollado	Muy alta	Muy baja	Crear copias de seguridad diarias en repositorios online como GitHub y Google Drive para recuperar la última versión.
No finalizar la aplicación web	Alta	Baja	Elaborar un informe dinámico con R-Markdown como alternativa.
Baja capacidad de procesamiento del ordenador	Moderada	Moderada	Ordenador alternativo con el que realizar el trabajo o utilizar las herramientas online de Kaggle o Google Colab.
Enfermedad o lesión leve	Moderada	Moderada	Reorganizar la planificación para intentar completar todas las tareas
Enfermedad o lesión grave	Alta	Baja	Reorganizar la planificación para intentar completar la mayor cantidad de tareas posibles y en caso extremo no realizar la aplicación web.

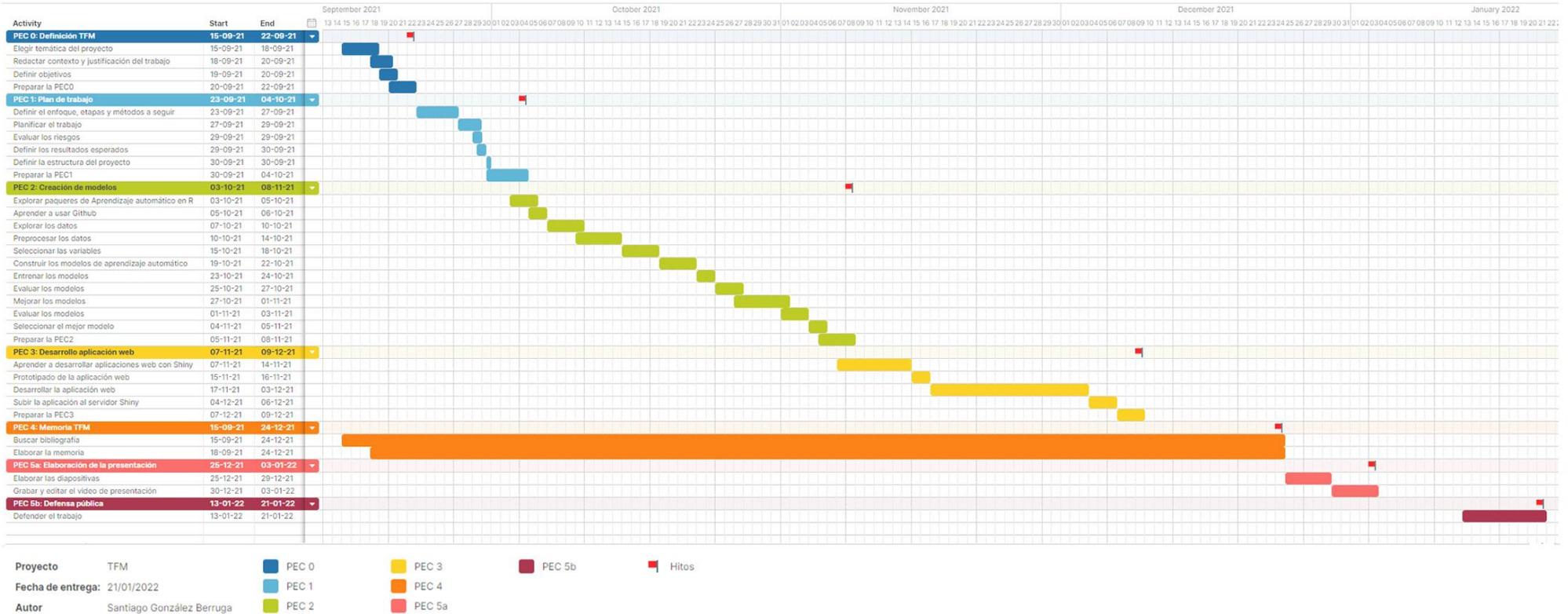


Figura 4. Diagrama de Gantt con la planificación del Trabajo Final de Máster elaborado con la aplicación Tom's Planner. Para ver el diagrama con más detalle pulse el siguiente enlace: <https://plan.tomsplanner.com/public/tfmsantiogob>.

2.5. Breve resumen de contribuciones y productos obtenidos

Las contribuciones y productos esperados en este trabajo son:

- **Plan de trabajo:** Informe del contexto, justificación, objetivos, enfoque, método y la planificación de las diferentes tareas a realizar en el trabajo.
- **Memoria:** Informe con el desarrollo del Trabajo Final del Máster de Bioinformática y Bioestadística de la Universitat Oberta de Catalunya (UOC).
- **Código fuente:** Disponible en el repositorio GitHub. Puede acceder con el enlace <https://github.com/santigberruga/Predict-Liver-Disease>.
- **Aplicación web:** Predicción de pacientes hepáticos rellenando un formulario sencillo cuyos campos son las variables empleadas para entrenar el modelo de aprendizaje automático. Puede acceder con el enlace https://gonzalezberrugasantiago.shinyapps.io/Predict_Liver_Disease/.
- **Presentación virtual:** Video de presentación donde se expondrá el trabajo realizado por medio de diapositivas y su explicación oral.

2.6. Breve descripción de los otros capítulos de la memoria

A continuación, se enumeran y describen el resto de capítulos principales que componen la memoria del trabajo:

- a) **Estado del arte:** En este capítulo se muestra una revisión bibliográfica del empleo de algoritmos de aprendizaje automático para el diagnóstico enfermedades hepáticas, una breve explicación teórica de los principales algoritmos empleados y la importancia de las aportaciones del trabajo.
- b) **Metodología:** En este capítulo se enumeran y explican los materiales y métodos empleados para cumplir con los objetivos del trabajo, que son: generar un modelo de aprendizaje automático para enfermedad hepática e implementarlo en una aplicación web.
- c) **Resultados:** En este capítulo se presentan los principales resultados del trabajo.
- d) **Discusión:** En este capítulo se discuten los resultados del trabajo justificando el modelo de clasificación escogido.
- e) **Conclusiones:** En este capítulo se presentan las conclusiones del trabajo y el trabajo futuro. También se realiza una revisión crítica del seguimiento de la planificación durante el trabajo.
- f) **Glosario:** Este capítulo es una recopilación ordenada de forma alfabética de las definiciones de los términos empleados en el trabajo.
- g) **Bibliografía:** Este capítulo contiene una lista numerada de las fuentes de información mencionadas en esta memoria ordenadas por orden de aparición en el texto.
- h) **Anexos:** Este capítulo contiene información extra que complementa el resto de capítulos de la memoria.

3. Estado del arte

Como se menciona en el contexto del trabajo, actualmente numerosas investigaciones se centran en el desarrollo de modelos de aprendizaje automático para el diagnóstico de enfermedades. En cuanto al presente caso, se han realizado diferentes estudios previos enfocados a detectar pacientes que sufren enfermedad hepática, ya sea una detección general, como en el presente estudio, o una detección de enfermedades hepáticas concretas. Algunos de los estudios previos son:

Akter et al. ^[2] evaluaron la progresión natural de enfermedad hepática a partir de datos de pruebas bioquímicas y los algoritmos RF y CART. Los algoritmos obtuvieron una tasa de éxito del 94% y 95% respectivamente, demostrando que los algoritmos de aprendizaje automático son capaces de predecir y verificar los mismos resultados que las pruebas bioquímicas.

Atabaki-Pasdar et al. ^[17] estudiaron cómo predecir NAFLD para realizar un diagnóstico precoz de dicha enfermedad y evitar complicaciones en los pacientes. Aplicaron los algoritmos LASSO y RF para desarrollar 18 modelos a partir de diferentes combinaciones de datos clínicos y ómicos. Sus resultados resaltaron que los modelos que utilizan datos clínicos comunes demostraron ser adecuados para la predicción de NAFLD, indicando que su rendimiento mejoraba considerablemente con la adición de datos ómicos.

Alice Auxilia ^[38] empleó el conjunto de datos ILPD y los algoritmos DT, NB, RF, SVM Y ANN para hacer un modelo predictivo de enfermedad hepática. Sus resultados mostraron una tasa de éxito máximo del 81% correspondiente al modelo de DT.

Venkata Ramana et al. ^[6] emplearon datos de pacientes hepáticos y los algoritmos de clasificación KNN, NB, C4.5, Back propagation Neural Network y SVM para predecir pacientes hepáticos. Sus resultados indican que los algoritmos KNN, Back propagation y SVM presentan mejores resultados, siendo el mejor de ellos el Back propagation con una tasa de éxito máximo del 98%.

Castro et al. ^[14] hicieron uso de redes neuronales para predecir y clasificar pacientes de hepatitis C a partir de biomarcadores presentes en el paciente. El mejor modelo obtuvo una tasa de éxito del 99.31%, demostrando la capacidad predictiva de este modelo sobre la enfermedad de Hepatitis C y que las redes neuronales artificiales son herramientas adecuadas para generar predicciones de diagnóstico médico.

Gatos et al. ^[25] propusieron el uso de SVM para predecir enfermedad hepática crónica utilizando imágenes de elastografía de ondas de corte por ultrasonido (SWE por sus siglas en inglés) obteniendo una tasa de éxito del 87,3%.

Gogi & Vijayalakshmi ^[26] emplearon los algoritmos SVM, Logistic Regression (LR) y DT para predecir enfermedad hepática a partir de parámetros de laboratorio de los pacientes. El mejor modelo fue LR con una tasa de éxito del 95,8%.

Hashem & Mabrouk ^[7] emplearon SVM para clasificar enfermedad hepática a partir de datos de pacientes con diabetes y/o enfermedad hepática, ya que ésta última es una causa importante de muerte en algunos tipos de pacientes diabéticos. El resultado demuestra que SVM es una gran herramienta, con una tasa de éxito medio entorno al 74%, para predecir pacientes que requieren especial atención médica.

Mohaimenul Islam et al. ^[29] utilizaron datos de pacientes hepáticos y los algoritmos RF, SVM, ANN y LR para predecir pacientes con la enfermedad del hígado graso (FLD por sus siglas en inglés), siendo el mejor modelo LR con una tasa de éxito del 76.3%.

Khan et al. ^[30] emplearon el conjunto de datos de pacientes hepáticos BUPA y los algoritmos Multi-Layer Perceptron (MLP), KNN, SVM, J48, RF, LR y Decision Stump (DS) para predecir pacientes hepáticos. Los mejores modelos fueron RF y ANN con una tasa de éxito del 72.2% y 71.6%.

Kuzhippallil et al. ^[3] propusieron el uso de los algoritmos LR, KNN, DT, RF y ANN para predecir pacientes hepáticos. El mejor modelo fue RF con una tasa de éxito del 88%.

Lin et al. ^[20] emplearon los algoritmos Linear Discriminant Analysis (LDA), SVM, NB, DT, RF y Adaptive Boosting (AdaBoost) para crear un sistema de monitoreo que pueda predecir la mortalidad o clasificar a los pacientes con enfermedad hepática en etapa terminal (ESLD por sus siglas en inglés) y compararlo con los modelos estadísticos tradicionales. Los resultados demostraron que los modelos de aprendizaje automático presentaban mejor rendimiento que los modelos estadísticos tradicionales, siendo el mejor RF con una tasa de éxito del 82,4%.

Ma et al. ^[23] investigaron 11 técnicas de aprendizaje automático de vanguardia para evaluar el mejor modelo clínico predictivo de NAFLD, siendo el mejor LR con una tasa de éxito del 83,4%.

Perveen et al. ^[45] propusieron un modelo de predicción basado en DT para investigar el riesgo de desarrollar NAFLD. Sus resultados indican que J48 es una técnica prometedora para el modelado predictivo de pacientes con NAFLD.

Banu Priya et al. ^[31] analizaron enfermedad hepática utilizando los algoritmos RF, SVM, J48, ANN y Bayesian Network con el conjunto de datos ILPD. El mejor modelo fue J48 con una tasa de éxito del 95,4%.

Rahman et al. ^[32] evaluaron el desempeño de seis algoritmos LR, KNN, DT, SVM, NB y RF con el conjunto de datos ILPD para reducir el alto coste del diagnóstico de enfermedad hepática crónica. El mejor modelo fue LR con una tasa de éxito del 75%.

Singh et al. ^[33] emplearon seis algoritmos de clasificación J48, RF, LR, SVM, KNN y NB con el conjunto de datos ILPD para predecir pacientes hepáticos. El mejor resultado fue el modelo LR seguido del RF con unas tasas de éxito del 74,4% y 71,9%.

Thirunavukkarasu et al. ^[36] tenían como objetivo principal predecir enfermedad hepática utilizando diferentes algoritmos de clasificación a partir del conjunto de datos ILPD. Los algoritmos utilizados para este propósito fueron LR, KNN y SVM. Los modelos LR y KNN presentaron la mayor tasa de éxito de 74%, pero la LR tenía mayor sensibilidad, siendo el modelo LR el más apropiado para predecir pacientes hepáticos.

Wu et al. ^[37] emplearon los algoritmos RF, NB, ANN y LR para predecir FLD. La tasa de éxito de RF, NB, ANN y LR fue 87.48, 82.65, 81.85 y 76.96%, indicando que el modelo RF es bastante adecuado para predecir FLD utilizando variables clínicas mínimas.

Fathi et al. ^[46] proponen un método basado en SVM lineal, cuadrática y gaussiana para predecir pacientes hepáticos a partir de los conjuntos de datos ILPD y BUPA. El mejor modelo fue SVM gaussiano con una tasa de éxito del 90,9%.

El presente trabajo busca un modelo predictivo que mejore los resultados obtenidos en investigaciones previas. Además, se diferencia del resto ya que proporciona un uso práctico y directo de sus resultados de forma sencilla mediante una aplicación web accesible a todo tipo de público, siendo una herramienta con gran potencial para apoyar a los profesionales sanitarios en la toma de decisiones sobre el diagnóstico de pacientes con posible enfermedad hepática.

3.1. Algoritmos de aprendizaje automático

A continuación, se explican brevemente los principales grupos de algoritmos utilizados en este trabajo, así como sus fortalezas y debilidades (Tabla 3) ^[24,39]:

- **K-nearest neighbor (KNN)**: Es un algoritmo supervisado que aplica el enfoque de los vecinos más próximos para clasificar un nuevo individuo. Comienza con un conjunto de datos de entrenamiento de categorías o clases conocidas. Por otra parte, tenemos un conjunto de datos de prueba que contiene ejemplos no etiquetados del mismo tipo que los datos de entrenamiento. Para cada registro en el conjunto de datos de prueba, KNN identifica k registros en los datos de entrenamiento que son los "más cercanos" en similitud, donde k es un valor positivo especificado de antemano. Al registro no etiquetado se le asigna la clase de la mayoría de los k vecinos más cercanos.
- **Naïve Bayes (NB)**: Es un algoritmo supervisado que aplica el teorema de Bayes a los problemas de clasificación. Aunque no es el único método de aprendizaje automático que utiliza métodos bayesianos, es el más común.
- **Artificial Neural Network (ANN)**: Son algoritmos supervisados que se inspiran en las redes neuronales como las que se tiene en el cerebro. Las neuronas se sustituyen por nodos que reciben y envían señales (información). Se crea una red con diferentes capas interconectadas para procesar la información. Cada capa está formada por un grupo de nodos que transmite la información a los otros nodos de las siguientes capas. El algoritmo de entrenamiento establece la importancia de cada conexión para transmitir o no la señal a los nodos correspondientes. El más usado es el algoritmo "backpropagation", que para corregir los errores de la predicción va hacia atrás de la red corrigiendo los pesos de los nodos.
- **Support Vector Machine (SVM)**: Son un conjunto de algoritmos de aprendizaje supervisado dirigido, indistintamente, tanto a la resolución de problemas de clasificación como de regresión. Se basa en buscar el hiperplano que tenga mayor margen posible y de forma homogénea entre las clases. Formalmente, una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) para crear particiones bastante homogéneas a cada lado. Cuando los datos no son separables de forma lineal, es necesario el uso de kernels, o funciones de similitud y especificar un parámetro C para minimizar la función de coste. La elección de este parámetro es una base de ensayo/error, pero se buscan valores que no sean extremos en la búsqueda del equilibrio sesgo/varianza. Los kernels más populares son el lineal y el gaussiano, aunque hay otros como el polinomial, string kernel, chi-square kernel, etc.

- **Decision tree (DT):** Son algoritmos supervisados con gran poder de clasificación, que utilizan una estructura de árbol para modelar las relaciones entre las características y los resultados potenciales. Obtuvo su nombre debido a que, al igual que un árbol, comienza en un tronco ancho que se divide en ramas cada vez más estrechas. De la misma manera, un DT utiliza una estructura de decisiones de ramificación con un valor de clase predicho final. Su estructura general comienza en el nodo raíz, pasando a través de los nodos de decisión que requieren que se tomen decisiones basadas en los atributos. Estas opciones dividen los datos en las ramas, que indican los posibles resultados de una decisión, aunque en algunos casos puede haber más de dos posibilidades. En el caso de que se pueda tomar una decisión final, el árbol es terminado por nodos hoja (también conocidos como nodos terminales) que denotan la acción a tomar como resultado de la serie de decisiones. Algunos de los DT más usados son C5.0 y J48 también llamado C4.5.
- **Random Forest (RF):** Es un meta-aprendiz que genera un conjunto de árboles (bosque). El modelo usa un voto para combinar las predicciones de los árboles. Combinan versatilidad y potencia en un solo enfoque de aprendizaje automático. Como el conjunto usa solo una pequeña porción aleatoria del conjunto completo de características, los bosques aleatorios pueden manejar conjuntos de datos extremadamente grandes, donde la llamada "maldición de la dimensionalidad" podría hacer que otros modelos fallen. Al mismo tiempo, sus tasas de error para la mayoría de las tareas de aprendizaje están a la par con casi cualquier otro método.
- **Logistic Regression (LR):** Es un algoritmo de clasificación empleado para predecir resultados binarios en un conjunto dado de variables independientes. En la regresión logística sólo observamos la probabilidad de la variable dependiente como resultado. Cuando la probabilidad es mayor de 0.5, un registro se clasificaría como clase 1, si no, sería de clase 0.

Tabla 3. Fortalezas y debilidades de los grupos de algoritmos empleados en el trabajo: *k-Nearest Neighbor* (kNN), *Naïve Bayes* (NB), *Artificial Neural Network* (ANN), *Support Vector Machine* (SVM), *Decision tree* (DT), *Random Forest* (RF), *Logistic Regression* (LR).

Algoritmo	Fortalezas	Debilidades
KNN	<ul style="list-style-type: none"> - Simple y efectivo. - No hace suposiciones sobre la distribución de datos subyacente. - Fase de entrenamiento rápida. 	<ul style="list-style-type: none"> - No produce un modelo, lo que limita la capacidad de encontrar nuevos conocimientos en las relaciones entre las características. - Fase de clasificación lenta. - Requiere una gran cantidad de memoria - Las variables nominales y los datos faltantes requieren un procesamiento adicional.
NB	<ul style="list-style-type: none"> - Simple, rápido y muy efectivo. - Funciona bien con ruido de fondo y datos faltantes. - Requiere de relativamente pocas muestras para el entrenamiento, pero también funciona bien cuando hay un gran número de muestras. - Es fácil de obtener la estimación de probabilidad para una predicción. 	<ul style="list-style-type: none"> - Se basa en una asunción que no es siempre cierta: todas las características son igualmente importantes e independientes. - No es ideal para conjuntos con muchas variables numéricas. - Las probabilidades estimadas son menos fiables que las clases predichas.
ANN	<ul style="list-style-type: none"> - Adaptable a clasificación o problemas de predicción numérica. - Capaz de modelar patrones más complejos que casi cualquier otro algoritmo. - No necesita muchas restricciones acerca de las relaciones subyacentes de los datos. 	<ul style="list-style-type: none"> - Requiere de gran potencia computacional y en general es de aprendizaje lento, particularmente si la topología es compleja. - Propenso a sobreajustar los datos de entrenamiento. - Es un modelo de caja negra complejo y difícil de interpretar.
SVM	<ul style="list-style-type: none"> - Se puede usar para problemas de clasificación o predicción numérica. - Funciona bastante bien con datos ruidosos y no es muy propenso al sobreajuste. - Puede ser más fácil de usar que las redes neuronales, en particular debido a la existencia de varios algoritmos SVM bien soportados. - Gana popularidad debido a su alta precisión y ganancias de alto perfil en competiciones de minería de datos. 	<ul style="list-style-type: none"> - Encontrar el mejor modelo requiere probar diferentes kernels y parámetros del modelo (prueba y error). - Lento de entrenar, sobre todo a medida que aumenta el número de características. - Es un modelo de caja negra complejo y difícil de interpretar.
DT	<ul style="list-style-type: none"> - Clasificador multiuso que funciona bien en la mayoría de los problemas. - Proceso de aprendizaje altamente automático, pudiendo manejar variables numéricas o nominales, así como datos faltantes. - Excluye características sin importancia. - Se puede usar en conjuntos de datos pequeños y grandes. - Resultados que pueden interpretarse sin una base matemática. - Más eficiente que otros modelos complejos. 	<ul style="list-style-type: none"> - Los modelos de árboles de decisión a menudo están sesgados hacia divisiones en entidades que tienen una gran cantidad de niveles. - Es fácil sobreajustar el modelo. - Puede tener problemas para modelar algunas relaciones debido a la dependencia de las divisiones paralelas al eje. - Pequeños cambios en los datos de entrenamiento pueden generar grandes cambios en la lógica de decisión. - Los árboles grandes pueden ser difíciles de interpretar y sus decisiones parecer contradictorias.
RF	<ul style="list-style-type: none"> - Clasificador multiuso que funciona bien en la mayoría de los problemas. - Puede manejar datos faltantes o con ruido, así como variables categóricas o numéricas. - Selecciona solo las características más importantes. - Se puede usar en conjuntos de datos con muchas variables y registros. 	<ul style="list-style-type: none"> - El modelo no es fácilmente interpretable. - Puede requerir ajustar el modelo a los datos.
LR	<ul style="list-style-type: none"> - El enfoque más común para modelar datos numéricos. - Se puede adaptar para modelar casi cualquier tarea de modelado. - Proporciona estimaciones de la fuerza y el tamaño de las relaciones entre las características y el resultado. 	<ul style="list-style-type: none"> - Hace fuertes suposiciones sobre los datos. - La fórmula del modelo debe ser especificada previamente por el usuario. - No maneja datos faltantes. - Solo funciona con funciones numéricas, por lo que los datos categóricos requieren procesamiento adicional. - Requiere algunos conocimientos de estadística para comprender el modelo.

4. Metodología

Los objetivos principales de este trabajo son la creación de un modelo de aprendizaje automático que clasifique pacientes de enfermedad hepática y su implementación en una aplicación web. Para conseguirlo, se ha entrenado y comparado el rendimiento de 13 modelos. A continuación, se desarrollan los pasos seguidos para cumplir los objetivos marcados.

4.1. Software

Se ha optado por la utilización de Windows como entorno de trabajo ya que tiene compatibilidad con el resto de los programas empleados. Para la creación de los modelos de aprendizaje automático se ha empleado el software estadístico R versión 4.1.1 ^[47] junto al entorno de desarrollo integrado (IDE) RStudio versión 2021.09.0+351 ^[48], que permite trabajar cómodamente con el lenguaje de programación R.

Los principales paquetes de R que se han empleado para crear los modelos de aprendizaje automático son: *dplyr* ^[49] para el manejo del conjunto de datos, *ggplot2* ^[50] para la creación de gráficos, *recipes* ^[51] para el preprocesado de los datos y *caret* ^[52] para entrenar y evaluar los modelos de aprendizaje automático.

Por otro lado, los principales paquetes que se han empleado para la creación de la aplicación web son: *shiny* ^[53] y *shinydashboard* ^[54] para la creación de la aplicación web, *ggplot2* ^[50] y *plotly* ^[55] para generar los gráficos y *shinyjs* ^[56], *shinyFeedback* ^[57], *shinyalert* ^[58] y *waiter* ^[59] para agregar diferentes funcionalidades a la aplicación web.

4.2. Modelos de aprendizaje automático

4.2.1. Datos

En nuestro caso, se ha trabajado con el conjunto de datos ILPD (Indian Liver Patient Dataset) ^[6] obtenido en el noreste de Andhra Pradesh, India y que se puede descargar desde el repositorio de aprendizaje automático de la Universidad de California en Irvine (UCI) ^[42]. La descripción de este conjunto de datos indica que contiene 583 registros, de los cuales 416 pertenecen a pacientes con enfermedad hepática y 167 a pacientes sin enfermedad hepática. Por otro lado, 441 registros son de pacientes masculinos y 142 de pacientes femeninas. Cada registro está formado por 11 variables (Tabla 4) que son importantes para realizar un correcto diagnóstico de enfermedad hepática ^[1,5,13]: edad (Age), género (Gender) y las pruebas de función hepática Bilirrubina total (TB), Bilirrubina directa (DB), Fosfatasa alcalina (Alkphos), Alanina aminotransferasa (Sgpt), Aspartato aminotransferasa (Sgot), Proteínas totales (TP), Albúmina (ALB), Ratio albúmina/globulina (AG) y Clase de registro en función de si el paciente sufre enfermedad hepática o no (Class). Además, para los cálculos posteriores hay que tener en cuenta que, en este conjunto, cualquier paciente cuya edad sea mayor de 89 años se registró como 90 años.

Tabla 4. Variables del conjunto de datos ILPD.

Variable	Descripción	Tipo
Age	Edad del paciente (años)	Numérica
Gender	Género del paciente	Categórica
TB	Bilirrubina total (mg/dL)	Numérica
DB	Bilirrubina directa (mg/dL)	Numérica
Alkphos	Fosfatasa alcalina (unidades/L)	Numérica
Sgpt	Alanina aminotransferasa (unidades /L)	Numérica
Sgot	Aspartato aminotransferasa (unidades /L)	Numérica
TP	Proteínas totales (g/dL)	Numérica
ALB	Albúmina (g/dL)	Numérica
AG	Relación Albúmina-Globulina (g/dL)	Numérica
Class	Clase: enfermedad hepática (LD) o no (H)	Categórica

4.2.2. Métodos

Análisis exploratorio y preparación de los datos:

Lo primero que se hizo fue nombrar correctamente las variables del conjunto de datos y guardarlas con el objeto de R adecuado en cada caso. También se realizó un breve resumen estadístico de las variables comprobando que los resultados se correspondían con la descripción del conjunto de datos. En el caso de las variables numéricas, también se estudiaron algunas de sus características, como su media, mediana y rango (Tabla 5) donde se observa que el rango de las variables es muy distinto, lo que indica que se deberán normalizar o estandarizar los datos. Un factor importante a destacar es que el conjunto de datos ILPD no está balanceado (Figura 5) ya que presenta mayor número de muestras de pacientes hepáticos (416) que no hepáticos (167).

Tabla 5. Análisis estadístico de las variables numéricas del conjunto ILPD.

Variable	Media	Mediana	Mínimo	Máximo
Age	44.78	45.00	4.0	90.0
TB	3.32	1.00	0.4	75.0
DB	1.49	0.30	0.1	19.7
Alkphos	291.37	208.00	63.0	2110.0
Sgpt	81.13	35.00	10.0	2000.0
Sgot	110.41	42.00	10.0	4929.0
TP	6.48	6.60	2.7	9.6
ALB	3.14	3.10	0.9	5.5
AG	0.95	0.93	0.3	2.8

Durante la exploración de los datos se observó que la variable AG presenta cuatro valores ausentes. En nuestro caso, al tratarse solamente cuatro valores se decidió eliminar dichos registros, ya que se trata de datos clínicos que, si se imputan, se correría el riesgo de introducir valores que tengan mucha influencia en los modelos, generando así resultados poco fiables. Dos de estos valores ausentes pertenecen a pacientes

hepáticos y dos a no hepáticos. Además, dos pertenecen a pacientes masculinos y dos a femeninos. Por lo tanto, al eliminar dichos registros se mantiene el porcentaje de clases original en todas las variables categóricas del conjunto de datos.

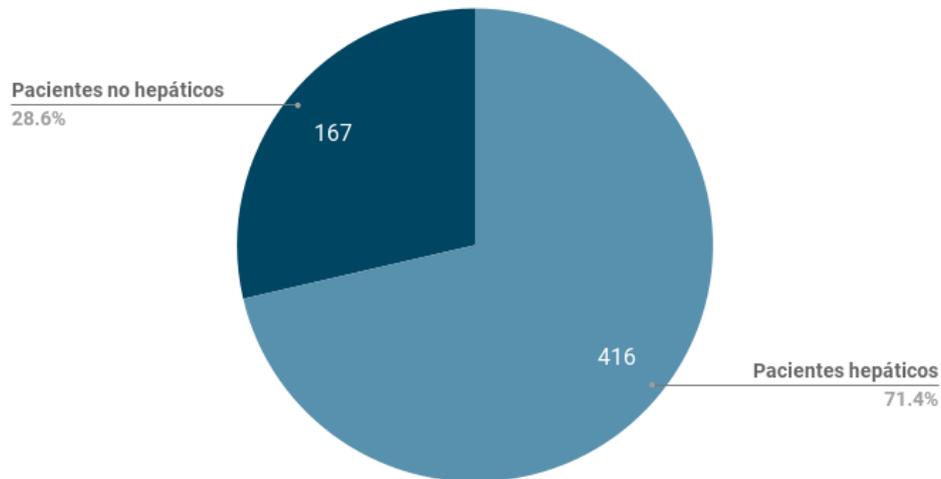


Figura 5. Gráfico de tarta de la variable Class. Se observa que el conjunto de datos no está balanceado.

Para realizar una mejor exploración de los datos, se realizó una exploración visual mediante diferentes gráficos, como diagramas de dispersión, histogramas, diagramas de cajas (de aquí en adelante boxplots) y gráficos de barras.

En los boxplots (Figura 6) se observa cómo el rango de variabilidad entre variables es muy grande. Por eso en algunos casos con rangos de valores muy estrechos las cajas quedan reducidas a una línea. También hay que prestar atención a las variables Alkphos, Sgpt y Sgot ya que presentan bastantes valores atípicos.

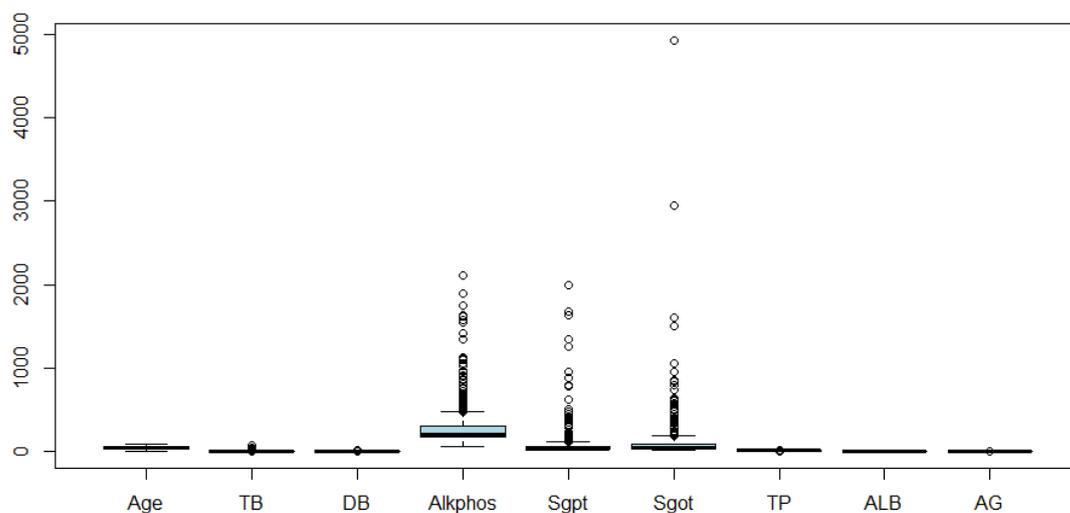


Figura 6. Boxplots de las variables numéricas del conjunto ILPD.

En los gráficos de dispersión (Figura 7) se observa que algunas variables parecen estar correlacionadas por lo que se calcularon las correlaciones entre las variables (Figura 8) donde se observan tres casos con altos valores significativos superiores a 0,7: *DB vs TB*, *Sgot vs Sgpt* y *ALB vs TP*. Cuando dos variables numéricas están muy correlacionadas, puede indicar que alguna de las variables implicadas no aporte nueva información al modelo, por lo que hay que prestar atención a las variables *DB*, *Sgot* y *ALB* ya que presentan una correlación superior a 0,7 y son menos específicas de enfermedad hepática, ya que pueden ser producidas por otras partes del cuerpo que no son el hígado y su liberación no tiene por qué ser causada por enfermedad hepática.

En los histogramas (Figura 9) se observa que las variables *TB*, *DB*, *Alkphos*, *Sgpt* y *Sgot* tienen una distribución asimétrica. Este tipo de distribución suele visualizarse mejor tras una transformación logarítmica. Por lo tanto, se volvió a estudiar la distribución de las variables teniendo esto en cuenta, pero en lugar de realizar un histograma, se representó cada variable en un gráfico de densidad y un boxplot en función de la clase a la que pertenecen (Anexo: Gráficos complementarios).

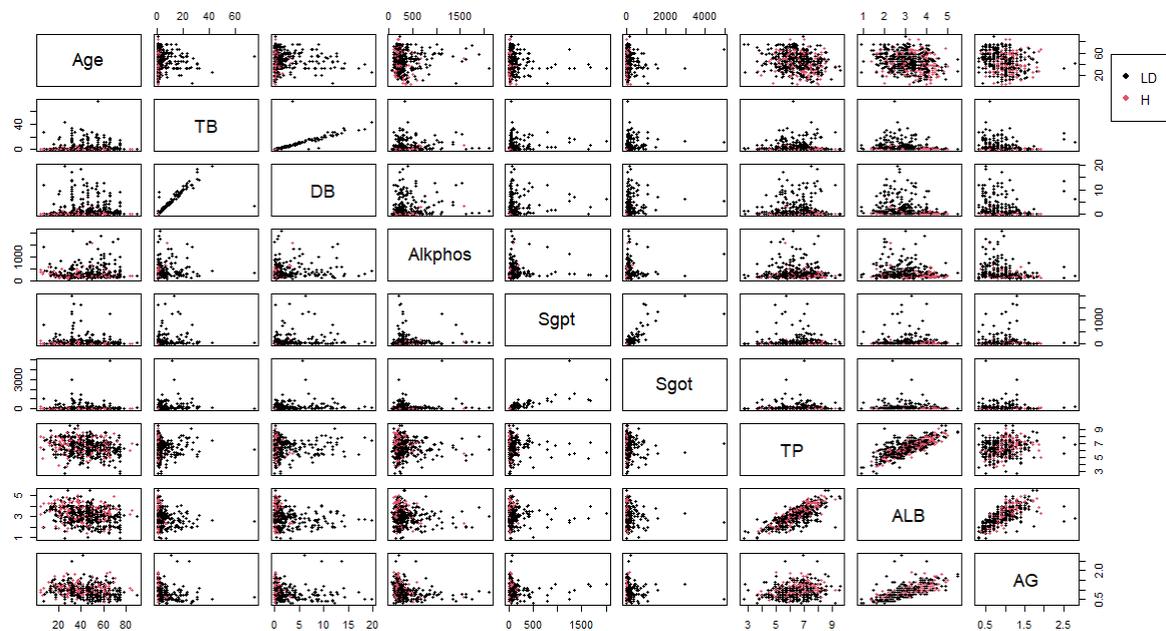


Figura 7. Gráficos de dispersión de las variables numéricas del conjunto ILPD.



Figura 8. Correlograma de las variables numéricas del conjunto ILPD.

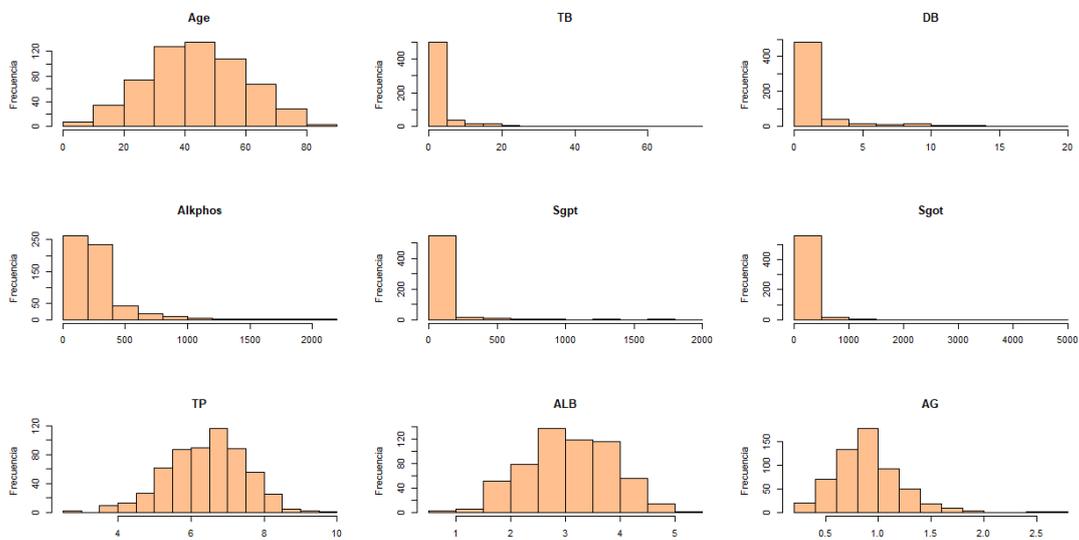


Figura 9. Histogramas de las variables numéricas del conjunto ILPD.

Por otro lado, se comprobó que no había variables con varianza cero o próxima a cero, ya que si una variable tiene varianza igual o próxima a cero (su valor es el mismo o casi el mismo para todas las observaciones) añade al modelo más ruido que información, por lo que suele ser conveniente excluirla.

Para terminar con la exploración de los datos, se realizó un ranking de la importancia de las variables empleando el algoritmo Random Forest (Figura 10), ya que este algoritmo permite realizar una clasificación de los predictores según su importancia en base a la reducción de la tasa de éxito (accuracy en inglés) [52]. El análisis indica que la variable con mayor importancia es *DB* seguida de las variables *Sgpt*, *TB* y *Sgot*.

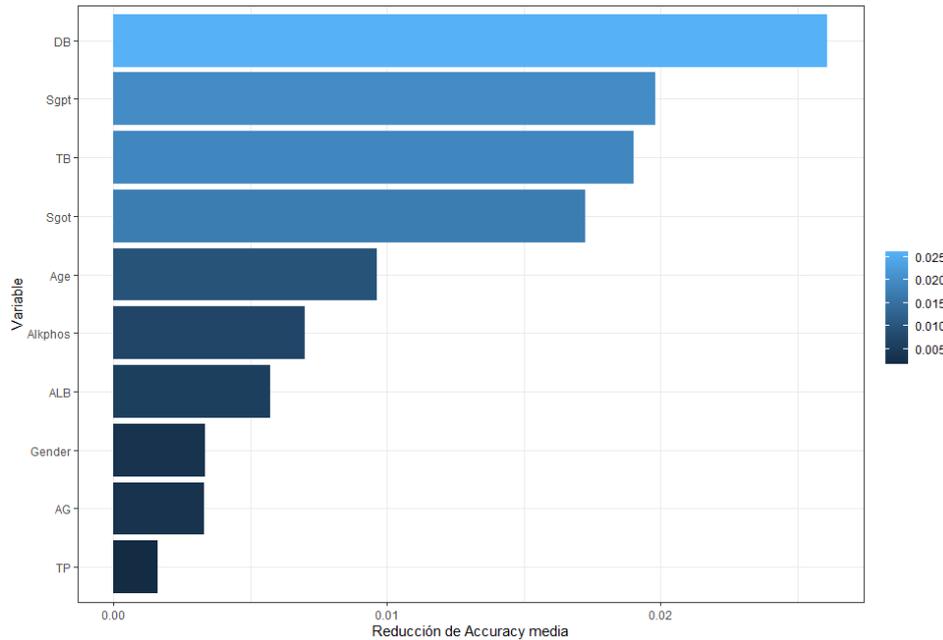


Figura 10. Importancia de las variables del conjunto ILPD para predecir la variable Class.

Tras explorar los datos, un paso importante antes de comenzar con el preprocesado de datos fue dividir el conjunto de datos en los subconjuntos de entrenamiento (70% de los datos) y prueba (30% de los datos). Para llevar a cabo esta división se empleó el paquete *caret* que permite realizar un reparto estratificado asegurando que el conjunto de entrenamiento y el de prueba mantengan una proporción similar de registros de cada clase de la variable respuesta y manteniendo el porcentaje de cada clase en el conjunto original de datos.

Por lo tanto, tras la exploración de los datos se comprobó que stos eran correctos y se eliminaron los registros con valores ausentes, pasando de 583 a 579 registros, de los cuales 406 formaban el conjunto de entrenamiento y 173 el conjunto de prueba.

Preprocesado de los datos:

Tras explorar los datos y obtener los subconjuntos de entrenamiento y prueba se deben preprocesar los datos de forma que sean adecuados para los algoritmos de aprendizaje automático empleados y obtener los mejores resultados posibles. El preprocesado de datos debe realizarse por separado para los conjuntos de entrenamiento y prueba, de modo que ningún registro ajeno al conjunto de entrenamiento afecte al entrenamiento de los modelos.

En este trabajo se probaron diferentes combinaciones de pasos a seguir durante el preprocesado de datos buscando el mejor resultado. Los diferentes pasos que se emplearon para buscar el mejor resultado fueron:

- a) Como el conjunto de datos ILPD está desbalanceado, se probaron 4 opciones: trabajar con datos no balanceados y corregir los datos mediante las técnicas Down-sampling, Up-sampling y SMOTE ^[60].
- b) Conservar o eliminar las variables con una correlación absoluta superior a 0,7.

- c) Para el tratamiento de valores atípicos (en adelante outliers) se probaron 3 opciones: conservar o eliminar sus registros e imputar su valor.
- d) Eliminar registros duplicados o conservarlos.
- e) Normalizar los datos por medio de la estandarización max-min para que las variables predictoras numéricas se encuentren en un rango [0,1].
- f) Binarizar las variables cualitativas.

Sin embargo, en el apartado de resultados solo se muestran los resultados de la combinación de pasos que presentó los mejores resultados:

- a) Trabajar con los datos no balanceados.
- b) Conservar todas las variables.
- c) Conservar todos los registros.
- d) Normalizar las variables por medio de la estandarización max-min para que las variables predictoras numéricas se encuentren en un rango [0,1].
- e) Binarizar las variables categóricas

Para realizar estas transformaciones se empleó el paquete *recipes*. Primero se realizó el preprocesado del conjunto de entrenamiento guardando los parámetros empleados para luego aplicarlos de nuevo sobre el conjunto de prueba, de modo que ningún registro ajeno al conjunto de entrenamiento afectara al entrenamiento de los modelos.

Selección de predictores:

Para la selección de predictores, el paquete *caret* incorpora métodos *Wrapper*, basados en la eliminación recursiva de predictores. Ésta es una buena estrategia para evitar tener que comprobar todas las posibles combinaciones de variables, lo que supondría un alto coste computacional. En este trabajo, se realizó una eliminación recursiva mediante el algoritmo *Random Forest* y *Bootstrapping* con 30 repeticiones, empleando la métrica de tasa de éxito de los modelos generados para seleccionar los mejores predictores. Los resultados indicaron que el mejor modelo empleaba 10 variables, es decir, en este caso todas las variables predictoras que contiene el conjunto de datos original son importantes para predecir el tipo de paciente.

Entrenamiento y optimización de los modelos:

El paquete *caret* proporciona una gran gama de algoritmos de aprendizaje automático adecuados para tareas de clasificación permitiendo crear y entrenar diferentes modelos de clasificación. También permite proporcionar diferentes hiperparámetros con los que entrenar y optimizar los modelos. A partir de los hiperparámetros proporcionados, *caret* genera automáticamente un modelo para cada combinación de hiperparámetros e indica que combinación es la mejor en función de la tasa de éxito de los modelos.

En nuestro caso, se han generado diferentes modelos a partir de 13 algoritmos de aprendizaje automático orientados a tareas de clasificación que podemos englobar en

los 7 grupos explicados en el estado del arte (Tabla 6). Los modelos se han entrenado utilizando validación cruzada repetida con 10 particiones como método de validación (*10-fold cross-validation* en inglés).

Tabla 6. Algoritmos de aprendizaje automático empleados para generar los modelos de clasificación de pacientes hepáticos.

Grupo	Algoritmo	Abreviatura
1	k-Nearest Neighbor	KNN
2	Naive Bayes	NB
3	Artificial neural network	ANN
	Multilayer perceptron	MLP
4	Support vector machine linear kernel	SVM-lineal
	Support vector machine polynomial kernel	SVM-poly
	Support vector machine radial kernel	SVM-radial
5	Decision tree C5.0 sin hiperparámetros	C5.0
	Decision tree C5.0 con hiperparámetros	C5.0-param
	Decision tree J48	J48
6	Random forest	RF
7	Logistic regression	LR
	Penalized Multinomial Logistic Regression	PMLR

Evaluación de los modelos:

Para evaluar los modelos se ha empleado el paquete *caret* para obtener diferentes métricas del rendimiento de un clasificador ^[24]:

- **Falsos negativos (FN):** Indica el número de muestras clasificado incorrectamente como la clase de no interés en la matriz de confusión (Figura 3).
- **Falsos positivos (FP):** Indica el número de muestras clasificado incorrectamente como la clase de interés en la matriz de confusión (Figura 3).
- **Tasa de éxito:** Es la proporción de muestras clasificadas correctamente.

$$Tasa\ de\ éxito = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Tasa de error:** Es la proporción de muestras clasificadas incorrectamente

$$Tasa\ de\ error = \frac{FP + FN}{TP + TN + FP + FN} = 1 - Tasa\ de\ éxito$$

- **Estadístico kappa:** Ajusta la tasa de éxito teniendo en cuenta la posibilidad de una predicción correcta solo por casualidad. Su valor va desde 0 hasta un máximo de 1, que indica un ajuste perfecto del modelo.

$$kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

En esta fórmula, $\Pr(a)$ se refiere a la proporción de acierto obtenido y $\Pr(e)$ se refiere a la proporción de acierto esperada.

- **Sensibilidad:** También llamada tasa de verdaderos positivos, es una métrica del rendimiento de modelos de aprendizaje automático que mide la proporción de muestras positivas que se clasifican correctamente.

$$Sensibilidad = \frac{TP}{TP + FN}$$

- **Especificidad:** También llamada tasa de verdaderos negativos, es una métrica del rendimiento de modelos de aprendizaje automático que mide la proporción de muestras negativas que se clasifican correctamente.

$$Especificidad = \frac{TN}{TN + FP}$$

- **Precisión:** Indica la proporción de predicciones positivas que son verdaderamente positivas.

$$Precisión = \frac{TP}{TP + FP}$$

- **Recall:** Indica cómo de completos son los resultados.

$$Recall = \frac{TP}{TP + FN}$$

- **F1-score:** Combina precisión y recall en un único valor usando la media armónica

$$F1 = \frac{2 * precisión * recall}{recall + precisión} = \frac{2 * TP}{2 * TP + FP + FN}$$

4.3. Aplicación web

El objetivo general de la aplicación web es permitir a los usuarios realizar predicciones, para determinar si un paciente sufre enfermedad hepática o no, utilizando un modelo de aprendizaje automático sin necesidad de tener que programar nada o de tener conocimientos técnicos de aprendizaje automático.

4.3.1. Funcionalidades de la aplicación web

Se ha optado por el inglés como idioma de la aplicación web por ser un idioma vehicular en la comunidad científica y en el mundo en general. Por otro lado, a partir del objetivo de la aplicación, se establecen las diferentes funcionalidades que debe tener:

- **Realizar predicciones:** Se usa el modelo de clasificación generado que presente mejor rendimiento. Se pueden realizar dos tipos de predicciones: predicción única cuando solo hay que realizar la predicción para un registro y predicción múltiple cuando hay que realizar predicciones para múltiples registros simultáneamente.
- **Subida de los datos por el usuario:** En caso de predicción única, se pueden subir los datos mediante un formulario manual o un fichero *csv*, mientras que para las predicciones múltiples solo es posible mediante fichero *csv*.
- **Validaciones:** Se realizan diferentes validaciones comprobando que los datos proporcionados por el usuario sean correctos.
- **Resultados de las predicciones:** Se muestra un panel con los resultados de las predicciones y diferentes tipos de gráficos interactivos para que el usuario pueda explorar sus datos y las predicciones generadas. Además, se puede descargar un fichero *csv* con los resultados de la predicción y un informe completo en *pdf*.
- **Comunicación con el usuario:** En caso de que no se cumpla alguna validación, cuando se realice la predicción o si el usuario necesita más información, se generan pop-ups informando al usuario de forma que sepa qué ocurre en cada momento y no se desoriente perdiendo interés por la aplicación. También se proporcionan ficheros *csv* de ejemplo para cada tipo de predicción, una plantilla *csv* para rellenar con datos del usuario y una guía de la aplicación. Por último, durante la descarga del informe de resultados en *pdf*, se emplea una barra de progreso para indicar al usuario el porcentaje de la descarga.

4.3.2. Estructura de la aplicación web

La interfaz de usuario está formada por tres grandes bloques: *Header*, *Sidebar* y *Body*. El *Header* inicialmente solo contiene el título de la aplicación y un botón de ayuda, sin embargo, tras realizar una predicción también aparece un botón para descargar el informe en *pdf*. El *Sidebar* contiene un enlace con plantillas e información sobre la aplicación y el formulario para introducir los datos y realizar la predicción. El *Body* contiene los resultados de las predicciones, los datos introducidos por el usuario y los gráficos interactivos.

4.3.3. Bocetos

Una vez definido el objetivo de la aplicación web, sus funcionalidades y su estructura, se realizaron diferentes bocetos (*sketches* en inglés) de la aplicación (Figura 11). En esta fase se comenzó a concretar la plataforma, incorporando de forma gráfica los requisitos definidos e iterando rápidamente entre bocetos hasta que se consiguió una estructura adecuada.



Figura 11. Bocetos de la aplicación web. A la izquierda se observa el aspecto general de la aplicación y a la derecha un pop-up informativo.

4.3.4. Implementación en Shiny

Para desarrollar la aplicación web se empleó el paquete *shiny* ^[44,53] por su versatilidad y comodidad al permitir incorporar directamente código y resultados obtenidos en R en una aplicación web. Las aplicaciones Shiny están formadas por dos componentes principales, un objeto de interfaz de usuario (UI) y una función de servidor (Server), que en combinación crean un objeto de aplicación Shiny ^[44]. El contenido de ambos componentes se indica a continuación.

Interfaz de usuario (UI):

El componente UI se encarga de generar la interfaz de usuario de la aplicación de forma que cumpla los requisitos deseados. En nuestro caso, se hizo uso del paquete *shinydashboard* para generar la estructura principal de la interfaz de usuario, formada por el *Header*, *Sidebar* y *Body*.

Para cada parte de la aplicación web, se definieron las entradas de información (inputs) y las áreas donde se muestran los resultados generados (outputs). Además, este paquete permite que la aplicación web sea responsive, facilitando su visualización y uso desde cualquier dispositivo. También se empleó código en HTML y CSS para aportar funcionalidades y mejorar la apariencia de la aplicación web.

Por lo tanto, con el componente UI se generó la interfaz del usuario pero aún no presentaba ninguna de las principales funcionalidades esperadas.

Función de servidor (Server):

El componente Server se encarga de configurar todas las funcionalidades del lado del servidor de la aplicación. En esta parte, se emplean los datos proporcionados por los usuarios, a través de los inputs definidos en la UI, para generar los outputs de forma dinámica, lo que se conoce en Shiny como reactividad. A continuación, enumeramos las principales funcionalidades que presenta el componente Server:

- **Validaciones:** Comprueba que todos los datos necesarios (Tabla 7) para realizar la predicción sean proporcionados y sean correctos.

Tabla 7. Variables necesarias para realizar predicciones en la aplicación web.

Variable	Descripción	Tipo	Valor
ID	Identificación del paciente	Cualquiera	-
Age	Edad del paciente (años)	Numérica	[1,130]
Gender	Género del paciente	Catégorica	Female/Male
TB	Bilirrubina total (mg/dL)	Numérica	Mayor de 0
DB	Bilirrubina directa (mg/dL)	Numérica	Mayor de 0
Alkphos	Fosfatasa alcalina (unidades/L)	Numérica	Mayor de 0
Sgpt	Alanina aminotransferasa (unidades /L)	Numérica	Mayor de 0
Sgot	Aspartato aminotransferasa (unidades /L)	Numérica	Mayor de 0
TP	Proteínas totales (g/dL)	Numérica	Mayor de 0
ALB	Albúmina (g/dL)	Numérica	Mayor de 0
AG	Relación Albúmina-Globulina (g/dL)	Numérica	Mayor de 0

- **Preprocesado de los datos:** Realiza un preprocesado de los datos empleando los mismos pasos y parámetros empleados durante el preprocesado del conjunto ILPD para el entrenamiento del modelo de clasificación seleccionado.
- **Predicciones:** A partir del modelo de clasificación seleccionado y tras preprocesar los datos del usuario se realizan las predicciones.
- **Resultados:** Se generan tablas y gráficos a partir de los resultados de las predicciones y los datos del usuario y se pasan al componente UI como outputs para ser visualizados por el usuario. También se genera un fichero *csv* con los resultados que el usuario puede descargar en cualquier momento.
- **Gráficos:** Se usan los paquetes *ggplot2* y *plotly* de R para generar gráficos interactivos. Estos gráficos se basan en los datos empleados para entrenar el modelo de clasificación y se les superponen los datos y resultados del usuario. Para las variables numéricas generamos *distplots* y *boxplots*, mientras que para las variables categóricas generamos *pie-plots*. Para cada gráfico, el usuario puede seleccionar la variable que desea visualizar permitiendo de este modo que pueda analizar gráficamente los datos y comprender los resultados obtenidos.
- **Informe de resultados:** Se ha generado un script de R Markdown que genera informes *pdf* dinámicos con el resultado de la predicción, los datos proporcionados por el usuario y los gráficos generados para todas las variables.

- **Comunicación con el usuario:** Se emplean códigos de colores en el formulario y pop-ups informativos para comunicarse con el usuario durante su interacción con la aplicación. También se informa al usuario del estado de la descarga del informe en *pdf* indicando el porcentaje del proceso.

4.3.5. Publicar la aplicación web

Una vez que se desarrolló y se comprobaron las funcionalidades de la aplicación Shiny localmente, se publicó para que estuviera a disposición de cualquier usuario que tenga un navegador web. Para ello, se utilizó la herramienta *Shinyapps.io* de RStudio para alojar la aplicación Shiny en los servidores proporcionados por la propia herramienta. Se eligió esta herramienta porque permite alojar las aplicaciones Shiny en la web de forma fácil, segura y escalable. Además, no se requiere ni hardware ni instalación y proporciona opciones gratuitas muy completas. Para poder hacer esto, se realizó una copia de todo el código y contenido de la aplicación en los servidores de RStudio.

5. Resultados

5.1. Rendimiento de los modelos

Para determinar el mejor modelo, se evaluó el rendimiento de los modelos en base a las métricas tasa de éxito, falsos negativos (FN), falsos positivos (FP), tasa de error, estadístico kappa, sensibilidad, especificidad, precisión, recall y F1-score. En la tabla 8 los modelos están ordenados de mayor a menor precisión. No se ha conseguido que los modelos MLP y SVM-lineal se adapten correctamente a los datos ya que predicen todos los registros como paciente hepático, por lo que no se han considerado como resultados válidos.

Todos los modelos presentan una tasa de éxito superior a 0,6 siendo el mayor ANN con un valor de 0,751. El número de falsos negativos es variable desde NB con 56 hasta ANN, LR, PMLR y RF con 8, 8, 8 y 5 falsos negativos respectivamente. El número de falsos positivos es variable desde MLP y SVM-lineal con 49 hasta NB con 7 falsos positivos. A pesar de estas diferencias entre falsos negativos y positivos la tasa de error es muy parecida para todos los modelos en torno al 30%. El estadístico Kappa es bastante bajo para todos los modelos, siendo J48 el mejor con un valor de 0,328. La sensibilidad es muy variable, siendo mayor para los modelos RF, ANN, LR y PMLR con valores de 96%, 93,5%, 93,5% y 93,5% respectivamente. La especificidad también varía entre los modelos, siendo mayor para el modelo NB con valor de 85,7%. La precisión es bastante alta para todos los modelos desde el 70% hasta el 90%, siendo los mejores modelos NB, J48 y C5.0 con un valor del 90,7%, 82,9% y 82,3%. El recall es variable entre los modelos, siendo los mejores RF, ANN, LR y PMLR con valores de 96%, 93,5%, 93,5% y 93,5% respectivamente. El F1-score es bastante alto en general, siendo mayor en los modelos RF, ANN, LR y PMLR con valores de 96%, 93,5%, 93,5% y 93,5% respectivamente.

Tabla 8. Rendimiento de los modelos de aprendizaje automático entrenados con el conjunto ILPD. Las métricas empleadas son: tasa de éxito, falsos negativos (FN), falsos positivos (FP), tasa de error, estadístico kappa, sensibilidad, especificidad, precisión, recall y F1-score.

Modelo	Tasa de Éxito	FN	FP	Tasa de Error	Kappa	Sensibilidad	Especificidad	Precisión	Recall	F1-score
NB	0,636	56	7	0,364	0,311	0,548	0,857	0,907	0,548	0,683
J48	0,705	32	19	0,295	0,328	0,742	0,612	0,829	0,742	0,783
C5.0-param	0,705	31	20	0,295	0,320	0,750	0,592	0,823	0,75	0,785
C5.0	0,705	31	20	0,295	0,320	0,750	0,592	0,823	0,75	0,785
ANN	0,751	8	35	0,249	0,265	0,935	0,286	0,768	0,935	0,844
SVM-radial	0,711	17	33	0,289	0,210	0,863	0,327	0,764	0,863	0,811
KNN	0,728	12	35	0,272	0,220	0,903	0,286	0,762	0,903	0,827
LR	0,740	8	37	0,26	0,220	0,935	0,245	0,758	0,935	0,838
PMLR	0,740	8	37	0,26	0,220	0,935	0,245	0,758	0,935	0,838
RF	0,746	5	39	0,254	0,207	0,960	0,204	0,753	0,960	0,844
SVM-poly	0,705	14	37	0,295	0,154	0,887	0,245	0,748	0,887	0,812
MLP	0,717	0	49	0,283	0	1	0	0,717	1	0,835
SVM-lineal	0,717	0	49	0,283	0	1	0	0,717	1	0,835

5.2. Aplicación web

A continuación, se muestran algunas de las pantallas de la aplicación web obtenida llamada “Liver Disease Prediction” (Figuras 12, 13, 14, 15 y 16) en la que se ha empleado el modelo ANN (véase el apartado Discusión). Puede acceder a la aplicación a través del enlace https://gonzalezberrugasantiago.shinyapps.io/Predict_Liver_Disease/.



Figura 12. Aspecto inicial al abrir la aplicación web. Por defecto, la opción inicial está configurada para realizar una predicción única y subir los datos mediante un fichero csv.

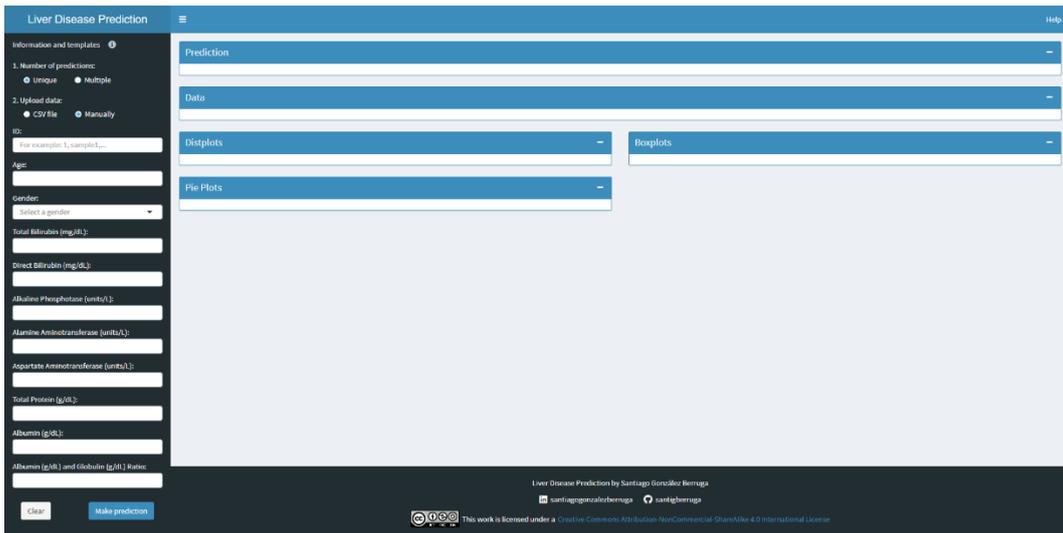


Figura 13. Configuración de la aplicación web para realizar una predicción única e introducir manualmente los datos mediante un formulario.

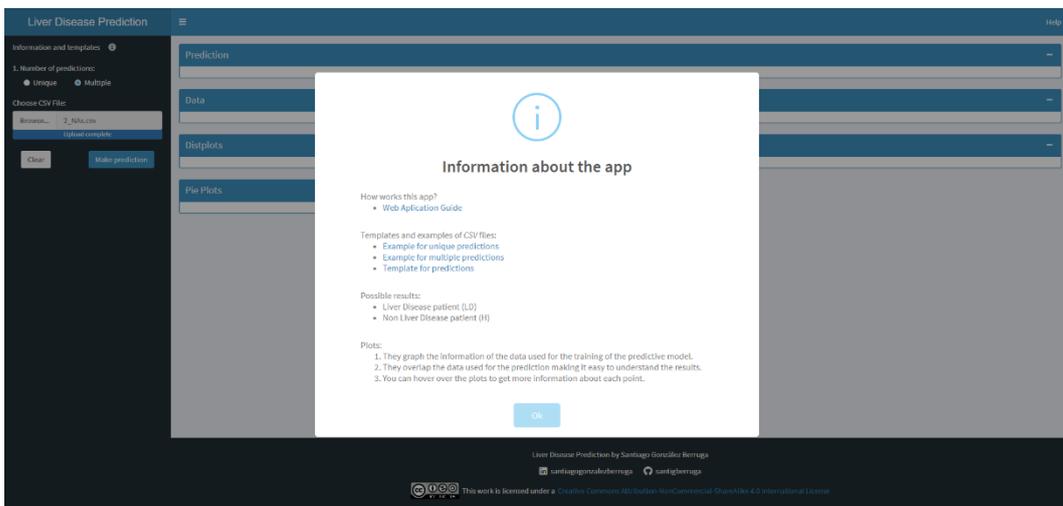


Figura 14. Pop-up de ayuda de la aplicación web.

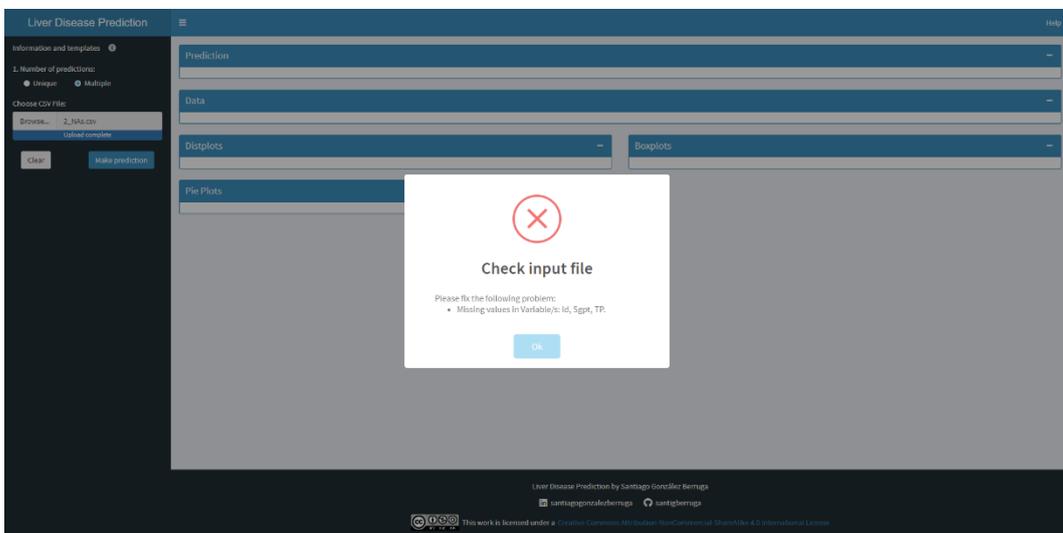


Figura 15. Ejemplo de pop-up indicador de error de la aplicación web.

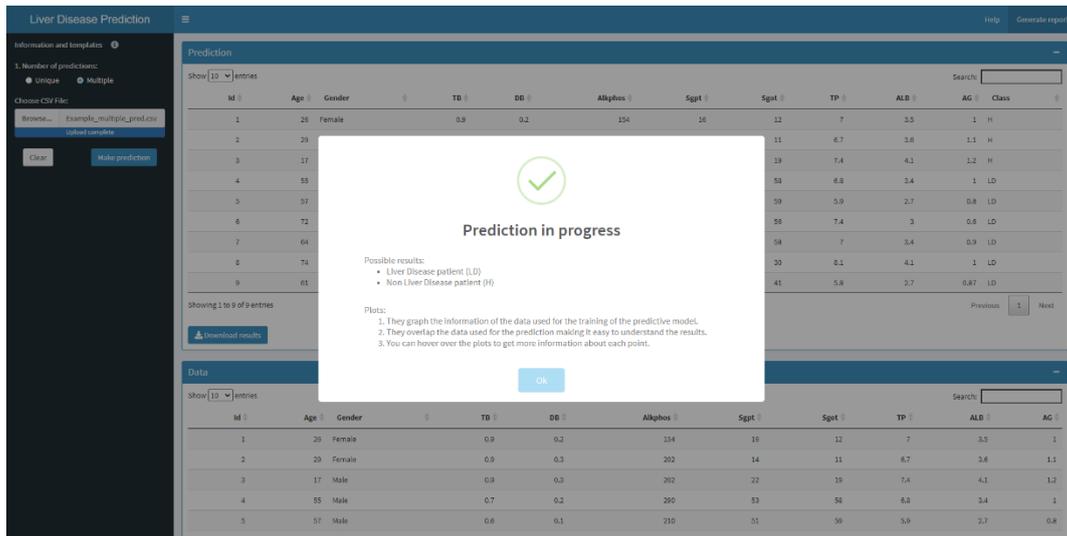


Figura 16. Pop-up de la aplicación web para indicar que la predicción está en progreso.

En la aplicación web, se ha conseguido implementar todas las funcionalidades definidas de forma exitosa. En las figuras 12 y 13 se muestran las dos formas posibles de cargar los datos, fichero csv o formulario manual, y en las figuras 14, 15 y 16 se muestran algunos de los pop-ups informativos generados para que el usuario sepa qué está ocurriendo en la aplicación. Además de los pop-ups, en el formulario manual también se han introducido mensajes informativos para cada campo (Figura 17).

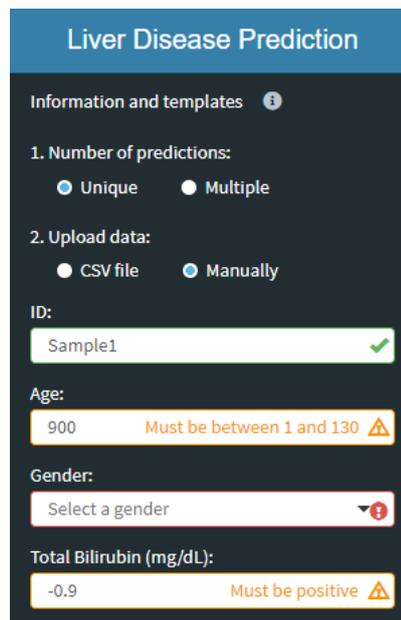


Figura 17. Mensajes informativos del formulario manual: éxito (verde), datos erróneos (naranja) y datos requeridos (rojo).

Por otro lado, también se ha comprobado que la aplicación web realiza predicciones de pacientes hepáticos, tanto únicas (Figura 18) como múltiples (Figura 19), y realiza gráficos interactivos con los que explorar los datos y comprender mejor los resultados obtenidos.

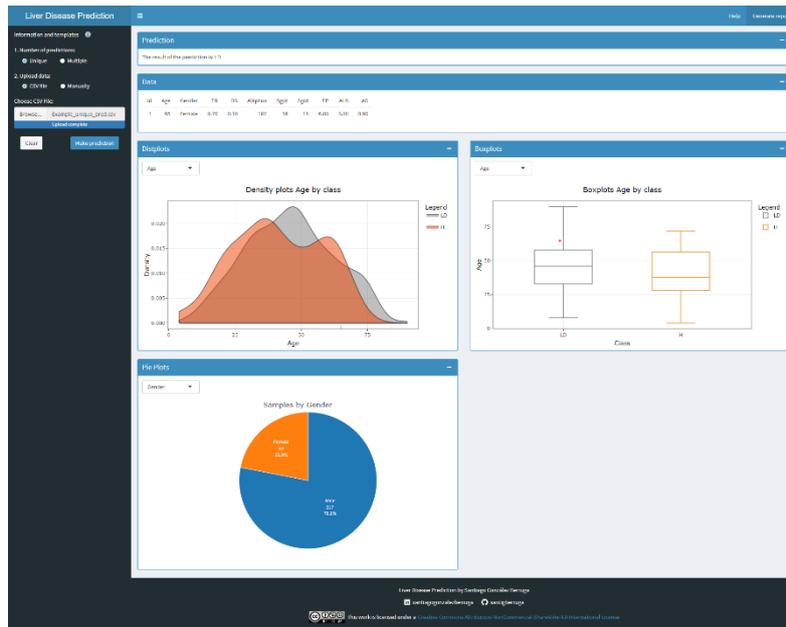


Figura 18. Resultados de una predicción única. Se observa la predicción para el paciente, los datos introducidos y los gráficos interactivos.

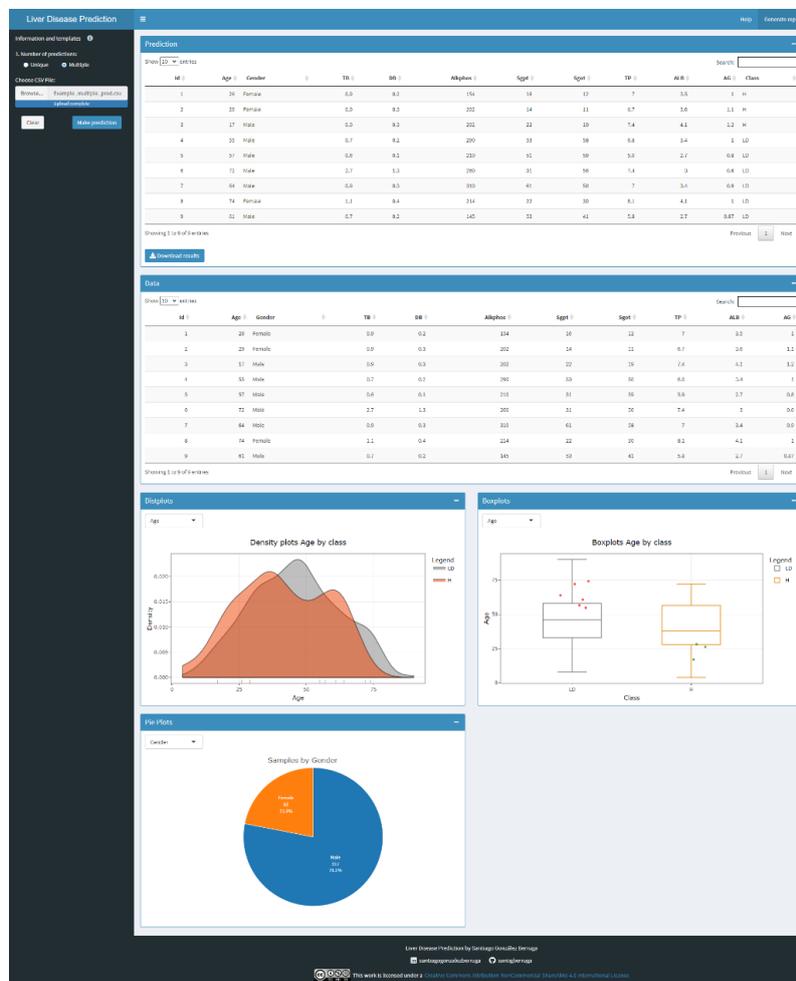


Figura 19. Resultados de una predicción múltiple. Se observa la predicción para el paciente, los datos introducidos y los gráficos interactivos.

Por último, la aplicación web genera informes en *pdf* automáticamente (Figura 20) a partir de un script escrito en RMarkdown. Estos informes contienen los resultados de la predicción, los datos proporcionados por el usuario y los gráficos generados en la aplicación. En la esquina superior derecha de la figura 20 se observa un código QR que se puede leer para acceder rápidamente a la aplicación web desde la que se han obtenido los resultados. Se puede acceder desde cualquier dispositivo (Figura 21) ya que se ha conseguido que la plataforma web sea responsive, es decir, que se ajuste correctamente a cualquier resolución de pantalla.

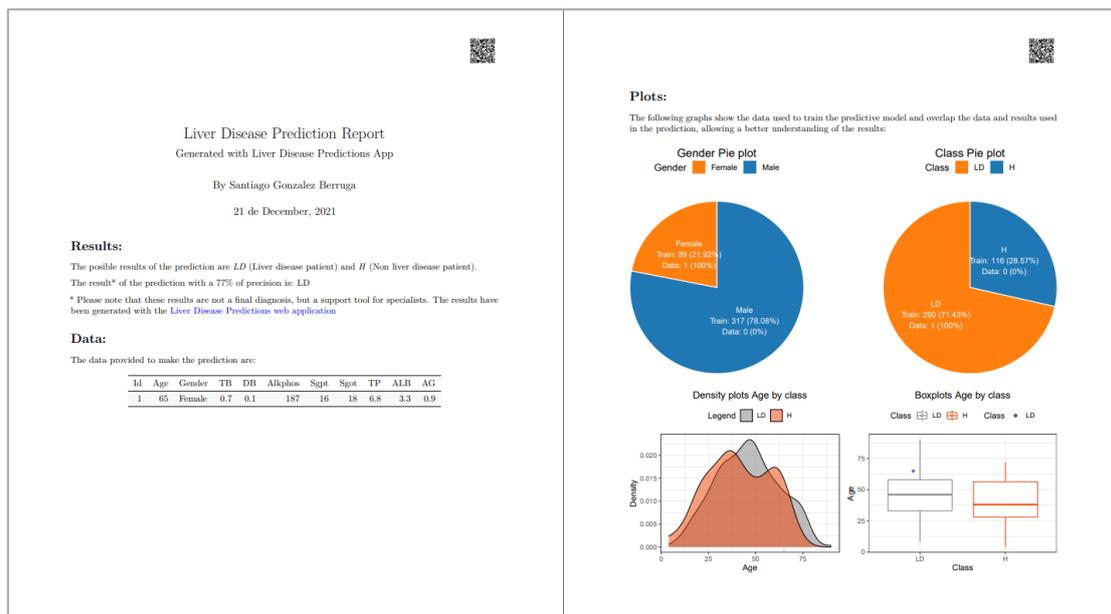


Figura 20. Informe en *pdf* generado automáticamente en la aplicación web. En este caso, se muestran los resultados de una predicción única, los datos empleados para la predicción y los primeros gráficos generados.

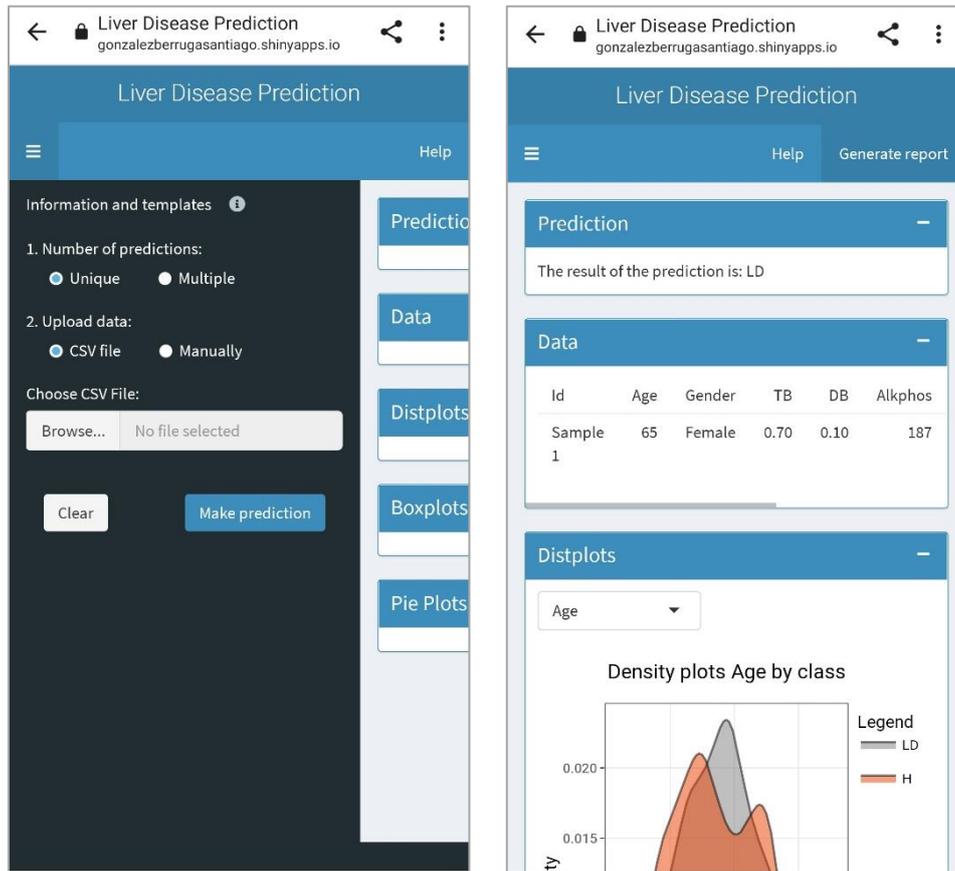


Figura 21. Aplicación web abierta desde un dispositivo móvil. A la izquierda se observa el panel lateral (*Sidebar*) con el formulario y a la derecha los resultados obtenidos.

6. Discusión

Como se observa en la tabla 8, los modelos MLP y SVM-lineal no se han adaptado correctamente a los datos de entrenamiento ya que predicen todos los registros como paciente hepático, por lo que no son aptos como modelos de clasificación.

Por otro lado, el resto de los modelos presentan mejores resultados. En este caso, el objetivo de los modelos es predecir si un paciente sufre enfermedad hepática o no, por lo que, al tratarse de una predicción clínica, en caso de fallar es preferible que sea un falso positivo a un falso negativo, es decir, es preferible prestar más atención a un paciente sano que no prestar atención a un paciente enfermo [24].

Primero, se observaron los modelos que cumplen el objetivo de una precisión mínima del 80%, pero ninguno es adecuado para la predicción de enfermedad hepática ya que presentan un gran número de falsos negativos y bajo recall. Tras esto, se observaron el resto de modelos y se seleccionaron los modelos ANN, LR, PMLR y RF por presentar menor número de falsos negativos y también buenos valores en la mayoría de las métricas de rendimiento. Estos cuatro modelos presentan menor número de falsos negativos a cambio de aumentar ligeramente los falsos positivos y, además, presentan

los mejores valores de tasa de éxito, recall y F1-score respecto al resto de modelos. En ninguno de los cuatro modelos se cumple el mínimo de precisión del 80% pero se decidió que, en base al objetivo de predecir pacientes con enfermedad hepática, es preferible sacrificar un poco de precisión con tal de reducir el número de falsos negativos y obtener un modelo más adecuado para realizar predicciones clínicas. De estos cuatro modelos, ANN y RF presentan los mejores rendimientos en conjunto. Sin embargo, se seleccionó el modelo ANN como mejor modelo por presentar un mejor equilibrio entre falsos negativos y positivos. Por lo tanto, el modelo seleccionado para la realización de las predicciones en la aplicación web es el modelo ANN con una precisión del 76,8%, una tasa de éxito del 75,1% y un recall del 93,5%.

Sin embargo, a la hora de usar estos resultados hay que tener en cuenta sus limitaciones, ya que el conjunto de datos ILPD presenta varias limitaciones. Primero, el conjunto ILPD es un conjunto de datos de mediano tamaño y, además, está desbalanceado, algo bastante común en datos clínicos pero que dificulta su uso ^[24]. Para corregir las clases desbalanceadas se emplearon algunas de las técnicas más utilizadas, como SMOTE, Up-sampling y Down-sampling ^[60] sin llegar a obtener resultados satisfactorios. Segundo, los modelos se han entrenado y validado únicamente con datos pertenecientes a pacientes de un área geográfica concreta, la India, pero hay estudios que demuestran que la incidencia y características de los pacientes hepáticos varían en función de la etnia y la región geográfica ^[11,12] por lo que hay que tener esto en cuenta a la hora de extrapolar los resultados globalmente.

7. Conclusiones

7.1. Conclusiones

En este trabajo, se han estudiado un total de 13 algoritmos de clasificación, con el objetivo de clasificar pacientes con enfermedad hepática, siendo ANN y RF los que presentan mejor rendimiento con una tasa de éxito del 75,1% y 74,6% y una precisión del 76,8% y 75,3% respectivamente. Se tomó la decisión de sacrificar precisión respecto al objetivo inicial de un mínimo del 80% para conseguir el mejor modelo posible. Se consideró que es una buena decisión ya que, de este modo, se reduce considerablemente el número de falsos negativos a cambio de aumentar ligeramente los falsos positivos y reducir ligeramente la precisión. De este modo, se indica una preferencia por diagnosticar a un paciente no enfermo como enfermo antes que diagnosticar a un paciente enfermo como no enfermo, lo que implicaría que no fuera atendido correctamente.

Por lo tanto, este trabajo demuestra que es posible detectar y diagnosticar pacientes hepáticos a partir de modelos de clasificación generados mediante técnicas de aprendizaje automático a partir de variables clínicas sencillas de obtener sin tener que recurrir a métodos invasivos para el paciente.

Además, se ha conseguido implementar exitosamente el modelo ANN en una aplicación web que permite su uso por cualquier usuario, generando una herramienta única y con gran potencial para apoyar a profesionales sanitarios durante el diagnóstico de

enfermedades hepáticas y reduciendo el uso de técnicas intrusivas peligrosas para los pacientes.

En cuanto a los objetivos planteados en el trabajo, se han cumplido los objetivos principales y secundarios, salvo el objetivo de precisión mínima del 80%, que se modificó hasta el 70% para obtener un modelo que se ajustara mejor al resto de objetivos de predicción de pacientes con enfermedad hepática. Por lo tanto, con este proyecto se ha logrado crear un modelo de aprendizaje automático para clasificar pacientes de enfermedad hepática que se ha implementado en una aplicación web sencilla e intuitiva a disposición de todo el público.

7.2. Líneas de futuro

Los resultados del proyecto son muy buenos y prometedores, por lo que son un gran primer paso para seguir desarrollando el proyecto y obtener mejores resultados futuros. Algunos de los puntos de trabajo futuro son:

- Repetir el entrenamiento de los modelos con un conjunto de datos mayor y con más variables importantes en la incidencia de enfermedades hepáticas como, por ejemplo, la diabetes ^[12] y la gestación ^[61], ya que estudios previos han demostrado la relación e importancia de estas variables sobre las enfermedades hepáticas.
- Buscar otros algoritmos que se adapten mejor al conjunto de datos y presenten mejor rendimiento.
- Emplear técnicas de Deep Learning para generar los modelos de clasificación.
- Ampliar las funcionalidades de la aplicación web para que sea más completa y tenga mayor visibilidad. Por ejemplo, permitiendo que el usuario suba los datos desde archivos con diferentes formatos y no solo desde archivos csv.
- Contactar con profesionales del mundo sanitario para que utilicen la aplicación y obtener su feedback directamente, para mejorar los resultados.

7.3. Seguimiento de la planificación

Respecto al seguimiento de la planificación y la metodología empleadas, en todo momento se ha seguido la planificación y metodología inicial sin encontrar grandes contratiempos que alteraran el ritmo de trabajo, por lo que son una buena elección para el tiempo disponible y el ámbito académico del proyecto.

Glosario

ANN: Acrónimo inglés de Red Neuronal Artificial (*Artificial Neural Network*). son algoritmos de aprendizaje automático supervisado inspirados en las redes neuronales como las que se tienen en el cerebro.

Aprendizaje automático: Es un campo de la inteligencia artificial que permite generar modelos predictivos de manera eficiente y efectiva a través de la detección de patrones ocultos dentro de grandes conjuntos de datos.

C5.0: Es un tipo de algoritmo de árbol de decisiones.

CRISP-DM: Acrónimo inglés del Proceso Estándar Multisectorial para la Minería de Datos (*Cross Industry Standard Process for Data Mining*). Es un modelo de proceso aplicado en la minería de datos

DT: Acrónimo inglés de Árbol de Decisiones (*Decision Tree*). Son algoritmos de aprendizaje automático supervisado que utilizan una estructura de árbol para modelar las relaciones entre las características y los resultados potenciales.

Enfermedad hepática: Es un término que engloba todas las enfermedades que afectan al correcto funcionamiento del hígado.

Especificidad: También llamada tasa de verdaderos negativos, es una métrica del rendimiento de modelos de aprendizaje automático que mide la proporción de muestras negativas que se clasifican correctamente. Se obtiene dividiendo los verdaderos negativos (TN) entre el número total de negativos (TN + FP).

Estadístico kappa: Es una métrica del rendimiento de modelos de aprendizaje automático. Ajusta la tasa de éxito teniendo en cuenta la posibilidad de una predicción correcta solo por casualidad. Su valor va desde 0 hasta un máximo de 1, que indica un ajuste perfecto del modelo.

F1-score: Es una métrica del rendimiento de modelos de aprendizaje automático. Combina precisión y recall en un único valor usando la media armónica.

FN: Acrónimo inglés de falsos negativos (*False Negatives*). Indica el número de muestras clasificado incorrectamente como la clase de no interés.

FP: Acrónimo inglés de falsos positivos (*False Positives*). Indica el número de muestras clasificado incorrectamente como la clase de interés.

ILPD: Acrónimo inglés del conjunto de datos de Pacientes Hepáticos de la India (Indian Liver Patient Dataset).

J48: Es un tipo algoritmo de árbol de decisiones basado en C4.5.

KNN: Acrónimo inglés de K vecinos más próximos (*k-nearest neighbors*). Es un algoritmo supervisado que aplica el enfoque de los vecinos más próximos para clasificar un nuevo individuo.

LR: Acrónimo inglés de Regresión Logística (*Logistic Regression*). Es un algoritmo de clasificación que se utiliza para predecir el resultado binario en un conjunto dado de variables independientes.

- Matriz de confusión:** es una tabla que clasifica las predicciones según coincidan con el valor real. En ella se observan los valores de TP, TN, FP y FN.
- MLP:** Acrónimo inglés de Perceptrón Multicapa (*Multilayer Perceptron*). Es una red neuronal artificial formada por múltiples capas.
- NB:** Acrónimo inglés de Clasificador Bayesiano Ingenuo (*Naive Bayes*). Es un algoritmo supervisado que describe un método simple para aplicar el teorema de Bayes a los problemas de clasificación.
- PMLR:** Acrónimo inglés de Regresión Logística Multinomial Penalizada (*Penalized Multinomial Logistic Regression*). Es un tipo de algoritmo de Regresión logística.
- Precisión:** Es una métrica del rendimiento de modelos de aprendizaje automático que indica la proporción de predicciones positivas que son verdaderamente positivas. Se obtiene dividiendo los verdaderos positivos (TP) entre el número total de predicciones positivas (TP + FP).
- Recall:** Es una métrica del rendimiento de modelos de aprendizaje automático que indica cómo de completos son los resultados. Se obtiene dividiendo los verdaderos positivos (TP) entre el número total de muestras positivas (TP + FN).
- RF:** Acrónimo inglés de Bosques Aleatorios (*Random Forest*). Es un algoritmo meta-aprendiz que genera un conjunto de árboles (bosque) combinando las predicciones de los diferentes árboles.
- Sensibilidad:** También llamada tasa de verdaderos positivos, es una métrica del rendimiento de modelos de aprendizaje automático que mide la proporción de muestras positivas que se clasifican correctamente. Se obtiene dividiendo los verdaderos positivos (TP) entre el número total de positivos (TP + FN).
- SVM:** Acrónimo inglés de Máquinas de Vectores de Soporte (*Support Vector Machines*). Son un conjunto de algoritmos de aprendizaje supervisado, dirigido tanto a la resolución de problemas de clasificación como de regresión, indistintamente. Se basa en buscar el hiperplano que tenga mayor margen posible y de forma homogénea entre las clases.
- Tasa de error:** Es la proporción de muestras clasificadas incorrectamente. Se obtiene dividiendo las predicciones incorrectas (+ FP + FN) entre el número de predicciones totales (TP + TN + FP + FN).
- Tasa de éxito:** Llamada *accuracy* en inglés, es una métrica del rendimiento de modelos de aprendizaje automático. Se obtiene dividiendo las predicciones correctas (TP + TN) entre el número de predicciones totales (TP + TN + FP + FN).
- TN:** Acrónimo inglés de verdaderos negativos (*True Negatives*). Indica el número de muestras clasificado correctamente como la clase de no interés.
- TP:** Acrónimo inglés de verdaderos positivos (*True Positives*). Indica el número de muestras clasificado correctamente como la clase de interés.
- UI:** Acrónimo inglés de Interfaz de Usuario (*User Interface*). Es el medio por el que un usuario interactúa de manera efectiva con una aplicación, equipo o dispositivo.

Bibliografía

1. Planas, R., & Salmeron, J. (2007). *Enfermedades Hepáticas. Consejos prácticos*. Publicaciones Permanyer.
2. Akter, S., Shekhar, H. U., & Akhteruzzaman, S. (2021). Application of Biochemical Tests and Machine Learning Techniques to Diagnose and Evaluate Liver Disease. *Advances in Bioscience and Biotechnology*, 12(06), 154-172. <https://doi.org/10.4236/abb.2021.126011>
3. Kuzhippallil, M. A., Joseph, C., & Kannan, A. (2020). Comparative Analysis of Machine Learning Techniques for Indian Liver Disease Patients. *2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020*, 778-782. <https://doi.org/10.1109/ICACCS48705.2020.9074368>
4. Cainelli, F. (2012). Liver diseases in developing countries. *World Journal of Hepatology*, 4(3), 66-67. <https://doi.org/10.4254/wjh.v4.i3.66>
5. Fernández Daza, E., Fernández Juan, E., Moreno Mejía, I., & Moreno Mejía, M. (2008). Aproximación al diagnóstico de enfermedades hepáticas por el laboratorio clínico. *Medicina & Laboratorio*, 14(11-12), 533-546.
6. Venkata Ramana, B., Babu, M. S. P., & Venkateswarlu, N. . (2011). A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis. *International Journal of Database Management Systems*, 3(2), 101-114. <https://doi.org/10.5121/ijdms.2011.3207>
7. Hashem, E. M., & Mabrouk, M. S. (2014). A Study of support vector machine algorithm for liver disease diagnosis. *American Journal of Intelligent Systems*, 4(1), 9-14.
8. Asrani, S. K., Devarbhavi, H., Eaton, J., & Kamath, P. S. (2019). Burden of liver diseases in the world. *Journal of Hepatology*, 70(1), 151-171. <https://doi.org/10.1016/j.jhep.2018.09.014>
9. Moon, A. M., Singal, A. G., & Tapper, E. B. (2020). Contemporary Epidemiology of Chronic Liver Disease and Cirrhosis. *Clinical Gastroenterology and Hepatology*, 18(12), 2650-2666. <https://doi.org/10.1016/j.cgh.2019.07.060>
10. Seitz, H. K., Bataller, R., Cortez-Pinto, H., Gao, B., Gual, A., Lackner, C., Mathurin, P., Mueller, S., Szabo, G., & Tsukamoto, H. (2018). Alcoholic liver disease. *Nature Reviews Disease Primers*, 4(1). <https://doi.org/10.1038/s41572-018-0014-7>
11. Jepsen, P., Grønbaek, L., & Vilstrup, H. (2015). Worldwide Incidence of Autoimmune Liver Disease. *Digestive Diseases*, 33(suppl 2), 2-12. <https://doi.org/10.1159/000440705>
12. Mitra, S., De, A., & Chowdhury, A. (2020). Epidemiology of non-alcoholic and alcoholic fatty liver diseases. *Translational Gastroenterology and Hepatology*, 5, 1-17. <https://doi.org/10.21037/TGH.2019.09.08>
13. Moreno Borque, A., González Moreno, L., Mendoza-Jiménez, J., García-Buey, L., & Moreno Otero, R. (2007). Utilidad de los parámetros analíticos en el diagnóstico de las enfermedades hepáticas. *Anales de Medicina Interna*, 24(1), 38-46. <https://doi.org/10.4321/s0212-71992007000100010>

14. Castro, C., Benítez-Mata, B., Vargas, E., & Flores, L. (2020). Detección temprana de la enfermedad causada por el virus de Hepatitis C usando redes neuronales artificiales. *Congreso Nacional de Ingeniería Biomédica*, 7. <http://memorias.somib.org.mx/index.php/memorias/article/view/791/467>
15. Takkar, S., Singh, A., & Pandey, B. (2017). Application of machine learning algorithms to a well defined clinical problem: Liver disease. *International Journal of E-Health and Medical Communications*, 8(4), 38-60. <https://doi.org/10.4018/IJEHMC.2017100103>
16. Shaheamlung, G., Kaur, H., & Kaur, M. (2020). A Survey on machine learning techniques for the diagnosis of liver disease. *Proceedings of International Conference on Intelligent Engineering and Management, ICIEM 2020*, 337-341. <https://doi.org/10.1109/ICIEM48762.2020.9160097>
17. Atabaki-Pasdar, N., Ohlsson, M., Viñuela, A., Frau, F., Pomares-Millan, H., Haid, M., Jones, A. G., Louise Thomas, E., Koivula, R. W., Kurbasic, A., Mutie, P. M., Fitipaldi, H., Fernandez, J., Dawed, A. Y., Giordano, G. N., Forgie, I. M., McDonald, T. J., Rutters, F., Cederberg, H., ... Franks, P. W. (2020). Predicting and elucidating the etiology of fatty liver disease: A machine learning modeling and validation study in the IMI DIRECT cohorts. *PLoS Medicine*, 17(6), 1-27. <https://doi.org/10.1371/journal.pmed.1003149>
18. Katsiki, N., Gastaldelli, A., & Mikhailidis, D. P. (2019). Predictive models with the use of omics and supervised machine learning to diagnose non-alcoholic fatty liver disease: A “non-invasive alternative” to liver biopsy? *Metabolism: Clinical and Experimental*, 101(xxxx). <https://doi.org/10.1016/j.metabol.2019.154010>
19. Younossi, Z. M., Stepanova, M., Younossi, Y., Golabi, P., Mishra, A., Rafiq, N., & Henry, L. (2020). Epidemiology of chronic liver diseases in the USA in the past three decades. *Gut*, 69(3), 564-568. <https://doi.org/10.1136/gutjnl-2019-318813>
20. Lin, Y. J., Chen, R. J., Tang, J. H., Yu, C. S., Wu, J. L., Chen, L. C., & Chang, S. S. (2020). Machine-learning monitoring system for predicting mortality among patients with noncancer end-stage liver disease: Retrospective study. *JMIR Medical Informatics*, 8(10), 1-16. <https://doi.org/10.2196/24305>
21. Spann, A., Yasodhara, A., Kang, J., Watt, K., Wang, B., Goldenberg, A., & Bhat, M. (2020). Applying Machine Learning in Liver Disease and Transplantation: A Comprehensive Review. *Hepatology*, 71(3), 1093-1105. <https://doi.org/10.1002/hep.31103>
22. Fatima, M., & Pasha, M. (2017). Survey of Machine Learning Algorithms for Disease Diagnostic. *Journal of Intelligent Learning Systems and Applications*, 09(01), 1-16. <https://doi.org/10.4236/jilsa.2017.91001>
23. Ma, H., Xu, C. F., Shen, Z., Yu, C. H., & Li, Y. M. (2018). Application of Machine Learning Techniques for Clinical Predictive Modeling: A Cross-Sectional Study on Nonalcoholic Fatty Liver Disease in China. *BioMed Research International*, 2018. <https://doi.org/10.1155/2018/4304376>
24. Lantz, B. (2015). Machine Learning with R. En *Machine Learning with R* (Second Edi). Packt Publishing. <https://doi.org/10.1007/978-981-10-6808-9>

25. Gatos, I., Tsantis, S., Spiliopoulos, S., Karnabatidis, D., Theotokas, I., Zoumpoulis, P., Loupas, T., Hazle, J. D., & Kagadis, G. C. (2017). A Machine-Learning Algorithm Toward Color Analysis for Chronic Liver Disease Classification, Employing Ultrasound Shear Wave Elastography. *Ultrasound in Medicine and Biology*, 43(9), 1797-1810. <https://doi.org/10.1016/j.ultrasmedbio.2017.05.002>
26. Gogi, V. J., & Vijayalakshmi, M. M. (2018). Prognosis of Liver Disease: Using Machine Learning Algorithms. *2018 International Conference on Recent Innovations in Electrical, Electronics and Communication Engineering, ICRIEEECE 2018*, 875-879. <https://doi.org/10.1109/ICRIEEECE44171.2018.9008482>
27. Hamid, K., Asif, A., Abbasi, W., Sabih, D., & Minhas, F. U. A. A. (2017). Machine Learning with Abstention for Automated Liver Disease Diagnosis. *Proceedings - 2017 International Conference on Frontiers of Information Technology, FIT 2017, 2017-Janua*, 356-361. <https://doi.org/10.1109/FIT.2017.00070>
28. Hashem, S., ElHefnawi, M., Habashy, S., El-Adawy, M., Esmat, G., Elakel, W., Abdelaziz, A. O., Nabeel, M. M., Abdelmaksoud, A. H., Elbaz, T. M., & Shousha, H. I. (2020). Machine Learning Prediction Models for Diagnosing Hepatocellular Carcinoma with HCV-related Chronic Liver Disease. *Computer Methods and Programs in Biomedicine*, 196. <https://doi.org/10.1016/j.cmpb.2020.105551>
29. Mohaimenul Islam, M., Wu, C. C., Poly, T. N., Yang, H. C., & Li, Y. C. (2018). Applications of machine learning in fatty live disease prediction. *Studies in Health Technology and Informatics*, 247, 166-170. <https://doi.org/10.3233/978-1-61499-852-5-166>
30. Khan, B., Naseem, R., Ali, M., Arshad, M., & Jan, N. (2019). Machine learning approaches for liver disease diagnosing. *International Journal of Data Science and Advanced Analytics (ISSN 2563-4429)*, 1(1), 27-31.
31. Banu Priya, M., Laura Juliet, P., & Tamilselvi, P. R. (2018). Performance Analysis of Liver Disease Prediction Using Machine Learning Algorithms. *International Research Journal of Engineering and Technology(IRJET)*, 5(1), 206-211. www.irjet.net
32. Rahman, A. K. M. S., Javed Mehedi Shamrat, F. M., Tasnim, Z., Roy, J., & Hossain, S. A. (2019). A comparative study on liver disease prediction using supervised machine learning algorithms. *International Journal of Scientific and Technology Research*, 8(11), 419-422.
33. Singh, J., Bagga, S., & Kaur, R. (2020). Software-based Prediction of Liver Disease with Feature Selection and Classification Techniques. *Procedia Computer Science*, 167(2019), 1970-1980. <https://doi.org/10.1016/j.procs.2020.03.226>
34. Sontakke, S., Lohokare, J., & Dani, R. (2017). Diagnosis of liver diseases using machine learning. *2017 International Conference on Emerging Trends and Innovation in ICT, ICEI 2017*, 129-133. <https://doi.org/10.1109/ETICT.2017.7977023>
35. Sorino, P., Caruso, M. G., Misciagna, G., Bonfiglio, C., Campanella, A., Mirizzi, A., Franco, I., Bianco, A., Buongiorno, C., Liuzzi, R., Cisternino, A. M., Notarnicola, M., Chiloiro, M., Pascoschi, G., & Osella, A. R. (2020). Selecting the best machine learning algorithm to support the diagnosis of Non-Alcoholic Fatty Liver Disease: A

- meta learner study. *PLoS ONE*, 15(10 October), 1-15. <https://doi.org/10.1371/journal.pone.0240867>
36. Thirunavukkarasu, K., Singh, A. S., Irfan, M., & Chowdhury, A. (2018). Prediction of liver disease using classification Algorithms. *2018 4th International Conference on Computing Communication and Automation, ICCCA 2018*, 6(9), 1-3. <https://doi.org/10.1109/CCAA.2018.8777655>
37. Wu, C. C., Yeh, W. C., Hsu, W. D., Islam, M. M., Nguyen, P. A. (Alex), Poly, T. N., Wang, Y. C., Yang, H. C., & (Jack) Li, Y. C. (2019). Prediction of fatty liver disease using machine learning algorithms. *Computer Methods and Programs in Biomedicine*, 170, 23-29. <https://doi.org/10.1016/j.cmpb.2018.12.032>
38. Alice Auxilia, L. (2018). Accuracy Prediction using Machine Learning Techniques for Indian Patient Liver Disease. *2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, 45-50.
39. Müller, A. C., & Guido, S. (2017). *Introduction to Machine Learning with Python*. O'Reilly Media.
40. Duda O., R., Hart E., P., & Stork G., D. (2001). *Pattern Classification* (Second Edi). Wiley John + Sons.
41. Schröer, C., Kruse, F., & Gómez, J. M. (2021). A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, 181(2019), 526-534. <https://doi.org/10.1016/j.procs.2021.01.199>
42. Dua, D. and Graff, C. (2019). *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science. <http://archive.ics.uci.edu/ml>
43. Tang, J., Alelyani, S., & Liu, H. (2014). Feature Selection for Classification: A Review. *Data Classification: Algorithms and Applications*, 37.
44. RStudio. (s. f.). *Shiny*. Recuperado 30 de septiembre de 2021, de <https://shiny.rstudio.com/>
45. Perveen, S., Shahbaz, M., Keshavjee, K., & Guergachi, A. (2018). A Systematic Machine Learning Based Approach for the Diagnosis of Non-Alcoholic Fatty Liver Disease Risk and Progression. *Scientific Reports*, 8(1), 1-12. <https://doi.org/10.1038/s41598-018-20166-x>
46. Fathi, M., Nemati, M., Mohammadi, S. M., & Abbasi-Kesbi, R. (2020). A machine learning approach based on svm for classification of liver diseases. *Biomedical Engineering - Applications, Basis and Communications*, 32(3), 1-9. <https://doi.org/10.4015/S1016237220500180>
47. Team, R. C. (2021). *R: A language and environment for statistical computing* (4.1.1). R Foundation for Statistical Computing. <https://www.r-project.org/>
48. RStudio Team. (2020). *RStudio: Integrated Development for R* (2021.09.0+351). RStudio, PBC. <https://www.rstudio.com/>
49. Wickham, H., François, R., Henry, L., & Müller, K. (2021). *dplyr: A Grammar of Data Manipulation. R package version 1.0.7*. <https://cran.r-project.org/package=dplyr>

50. Wickha, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. R package version 3.3.5. Springer.
51. Kuhn, M., & Wickham, H. (2021). *recipes: Preprocessing and Feature Engineering Steps for Modeling*. R package version 0.1.17. <https://cran.r-project.org/package=recipes>
52. Kuhn, M. (2021). *caret: Classification and Regression Training*. R package version 6.0-90. <https://cran.r-project.org/package=caret>
53. Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., & Borges, B. (2021). *shiny: Web Application Framework for R*. R package version 1.7.1. <https://cran.r-project.org/package=shiny>
54. Chang, W., & Borges Ribeiro, B. (2021). *shinydashboard: Create Dashboards with «Shiny»*. R package version 0.7.2. <https://cran.r-project.org/package=shinydashboard>
55. Sievert, C. (2020). *Interactive Web-Based Data Visualization with R, plotly, and shiny*. R package version 4.10.0. <https://cran.r-project.org/web/packages/plotly/>
56. Attali, D. (2020). *shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds*. R package version 2.0.0. <https://cran.r-project.org/package=shinyjs>
57. Merlino, A., & Howard, P. (2021). *shinyFeedback: Display User Feedback in Shiny Apps*. R package version 0.4.0. <https://cran.r-project.org/package=shinyFeedback>
58. Attali, D., & Edwards, T. (2020). *shinyalert: Easily Create Pretty Popup Messages (Modals) in «Shiny»*. R package version 2.0.0. <https://cran.r-project.org/package=shinyalert>
59. Coene, J. (2021). *waiter: Loading Screen for «Shiny»*. R package version 0.2.4. <https://cran.r-project.org/package=waiter>
60. Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling with Applications in R. En *Springer* (Vol. 26). Springer Science+Business Media. <https://doi.org/10.1007/978-1-4614-6849-3>
61. Lorente Pérez, S., & Serrano Aulló, T. (2010). Enfermedades hepáticas propias del embarazo. *Revista Española de Enfermedades Digestivas*, 102(8), 2010. <https://doi.org/10.4321/s1130-01082010000800010>

Anexo: Gráficos complementarios

Durante la exploración de los datos en el 4.2.2, en los histogramas de las variables continuas (Figura 9) se observa que las variables *TB*, *DB*, *Alkphos*, *Sgpt* y *Sgot* tienen una distribución asimétrica. Como este tipo de distribución suele visualizarse mejor tras una transformación logarítmica, se han realizado gráficos complementarios que permitan una mejor visualización de la distribución de estas variables. Para ello, se representa cada variable en un gráfico de densidad y un boxplot en función de la clase a la que pertenecen los datos.

En la figura 22 se observa que la distribución de la edad (*Age*) es parecida para pacientes hepáticos y sanos. Sin embargo, parece que entre los 40 y 60 hay mayor número de pacientes hepáticos y disminuye el número de pacientes sanos. En general, hasta los 40 años parecen predominar los pacientes sanos y a partir de 40 años la situación se invierte con dominancia de pacientes hepáticos. Esto tiene sentido, porque se ha demostrado que una edad superior a 45 años es un factor relacionado con cirrosis ^[1].

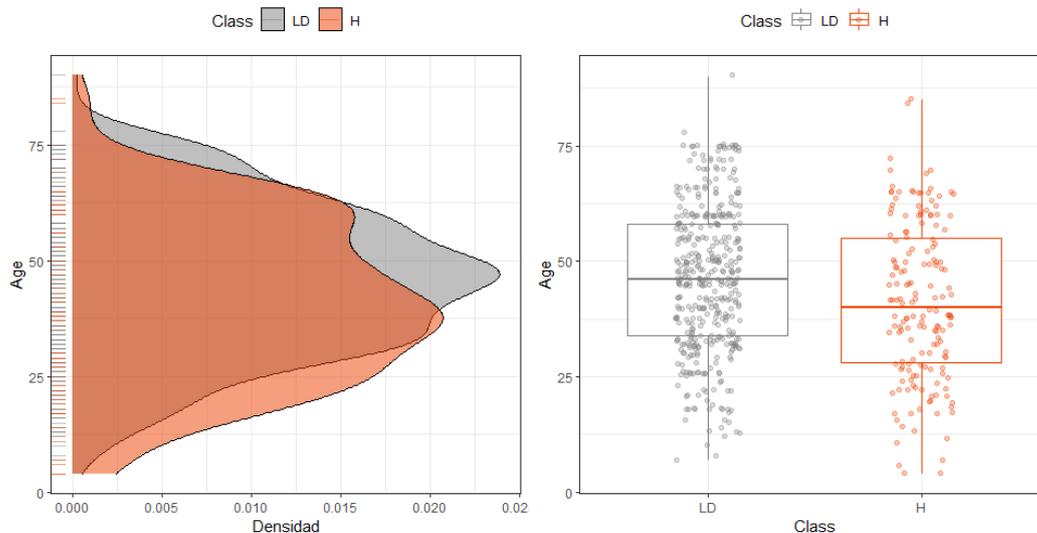


Figura 22. Gráficos de densidad y boxplots por clase de la edad (*Age*) del conjunto de datos ILPD. LD: Enfermedad hepática; H: No enfermedad hepática.

En las figuras 23 y 24 se observa que las distribuciones de bilirrubina directa (*DB*) y bilirrubina total (*TB*) son diferentes para pacientes hepáticos y sanos. Parece que la gran mayoría de pacientes sanos presentan niveles bajos de bilirrubina total y directa, mientras que los pacientes hepáticos presentan una curva más suave en ambos casos. Esto tiene sentido ya que el aumento de *TB* y *DB* se asocia con necrosis y colestasis ^[1,5,13].

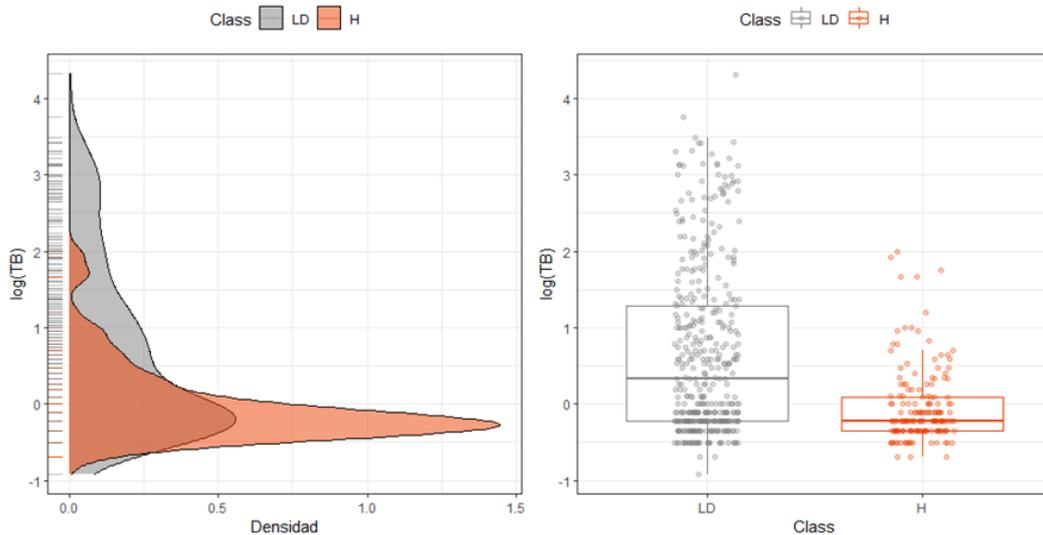


Figura 23. Gráficos de densidad y boxplots por clase del logaritmo de la bilirrubina total (TB) del conjunto de datos ILPD. LD: Enfermedad hepática; H: No enfermedad hepática.

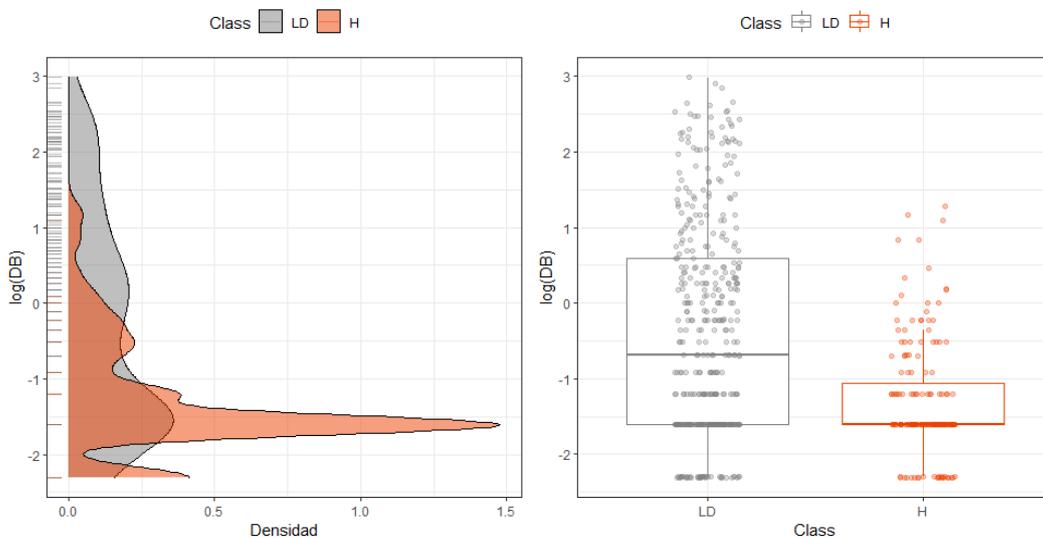


Figura 24. Gráficos de densidad y boxplots por clase del logaritmo de la bilirrubina directa (DB) del conjunto de datos ILPD. LD: Enfermedad hepática; H: No enfermedad hepática.

En la figura 25 se observa que la distribución de la fosfatasa alcalina (Alkphos) es parecida para pacientes hepáticos y sanos. Sin embargo, los pacientes sanos presentan mayor densidad en valores intermedios de Alkphos, mientras que los pacientes hepáticos presentan mayor densidad con valores altos de Alkphos. Esto tiene sentido porque el valor de Alkphos producida por el hígado puede aumentar hasta 10 veces por obstrucción en vías biliares, procesos infecciosos, cirrosis y enfermedades intestinales [1,5].

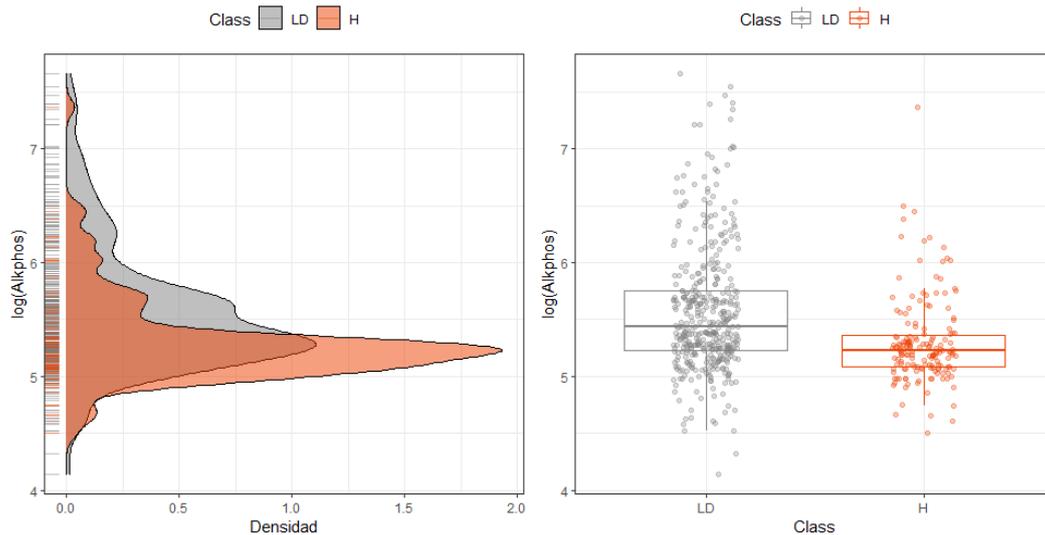


Figura 25. Gráficos de densidad y boxplots por clase del logaritmo de la fosfatasa alcalina (Alkphos) del conjunto de datos ILPD. LD: Enfermedad hepática; H: No enfermedad hepática.

En las figuras 26 y 27 se observa que las distribuciones de alanina aminotransferasa (Sgpt) y aspartato aminotransferasa (Sgot) son parecidas para pacientes hepáticos y sanos. Sin embargo, al igual que ocurre con la Alkphos, los pacientes sanos presentan mayor densidad en valores bajos y medios de Sgpt y Sgot, mientras que a partir de valores medios encontramos mayor densidad de pacientes hepáticos. Esto tiene sentido ya que valores elevados de Sgpt y Sgot es indicador de necrosis hepática. La Sgpt es específica del hígado, sin embargo, la Sgot no solo se encuentra en el hígado sino también la podemos encontrar en otros órganos, como el corazón y los músculos, por lo que Sgot no es un indicador específico para enfermedades hepáticas mientras que Sgpt sí lo es [1,5].

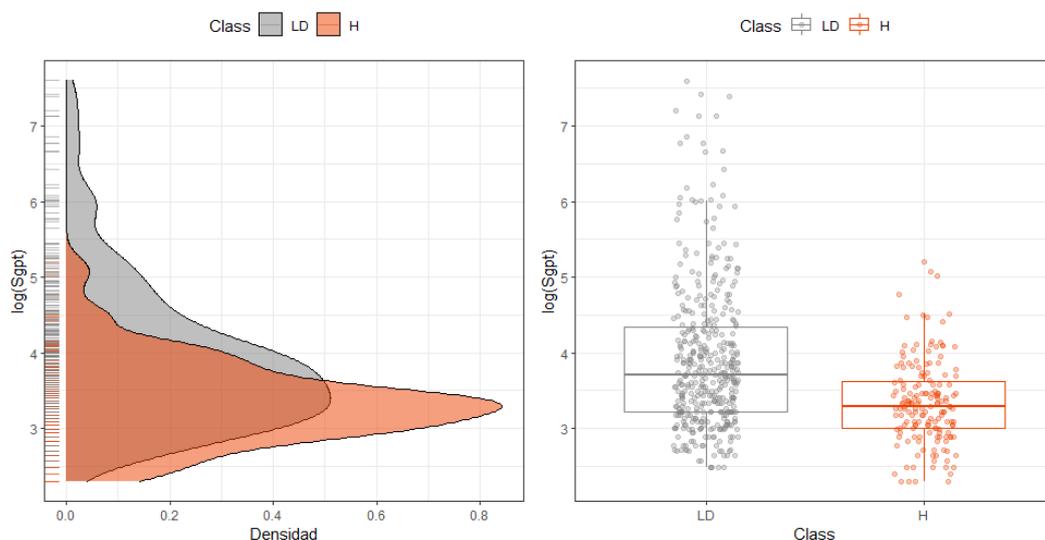


Figura 26. Gráficos de densidad y boxplots por clase del logaritmo de la alanina aminotransferasa (Sgpt) del conjunto de datos ILPD. LD: Enfermedad hepática; H: No enfermedad hepática.

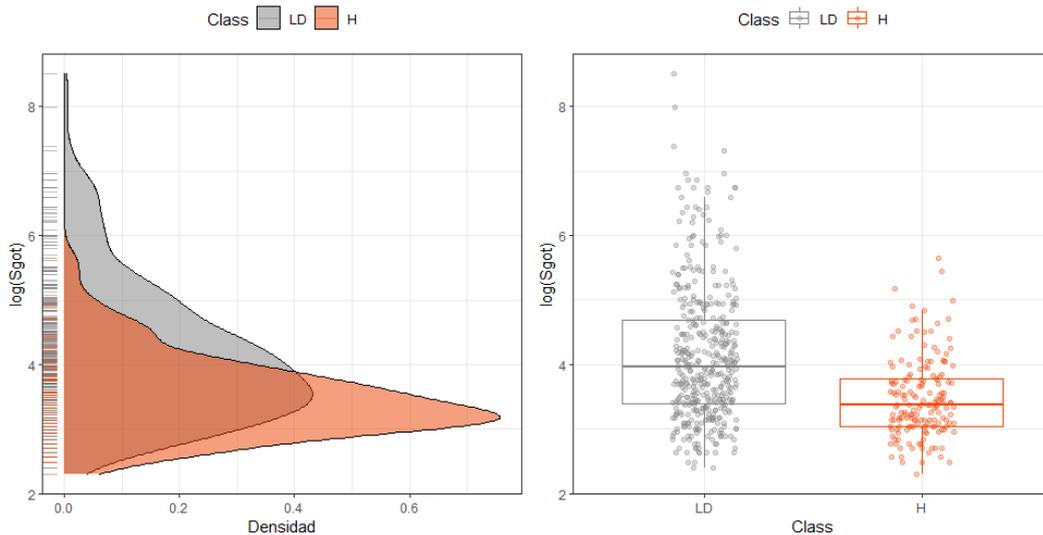


Figura 27. Gráficos de densidad y boxplots por clase del logaritmo de la aspartato aminotransferasa (Sgot) del conjunto de datos ILPD. LD: Enfermedad hepática; H: No enfermedad hepática.

En la figura 28 se observa que la distribución de proteínas totales (TP) es muy parecida para pacientes hepáticos y sanos. En este caso no encontramos grandes diferencias, salvo que en torno a una concentración de 7 g/dL de TP hay mayor densidad de pacientes hepáticos y para concentraciones mayores la densidad de pacientes sanos es ligeramente mayor. Un aumento de las TP es indicativo de enfermedad hepática crónica [5], sin embargo, en nuestro caso no conocemos el estado de los pacientes hepáticos del conjunto de datos, por lo que no podemos extraer una conclusión clara.

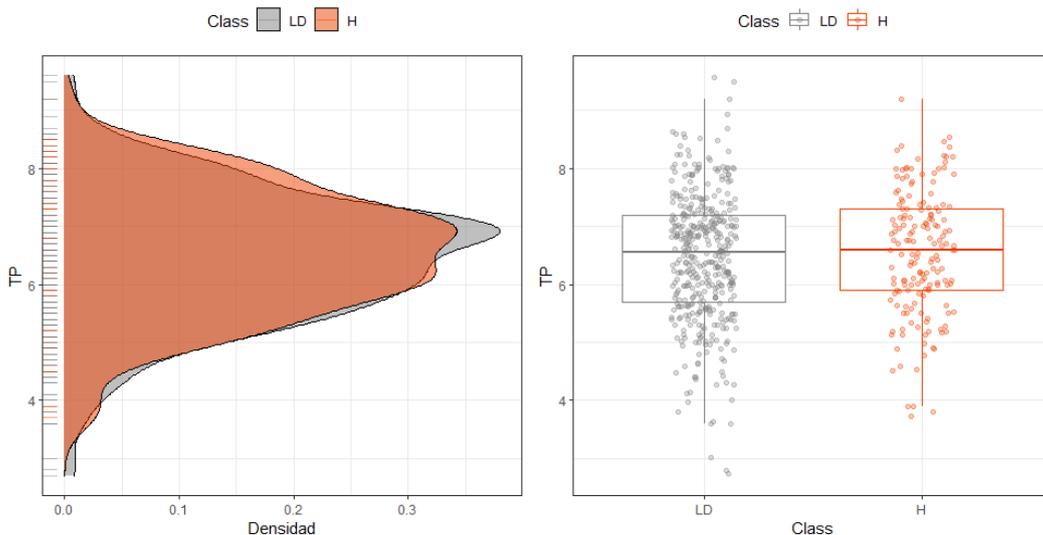


Figura 28. Gráficos de densidad y boxplots por clase de las proteínas totales (TP) del conjunto de datos ILPD. LD: Enfermedad hepática; H: No enfermedad hepática.

En la figura 29 se observa que la distribución de la albúmina (ALB) es parecida para pacientes hepáticos y sanos para concentraciones intermedias de ALB. Sin embargo, los pacientes sanos presentan mayor densidad en valores altos de ALB, mientras que los pacientes hepáticos presentan mayor densidad en valores bajos de ALB. Esto tiene sentido ya que la ALB es la principal proteína producida por el hígado y su concentración disminuye cuando hay daño en tejidos hepáticos, siendo uno de los principales indicadores de cirrosis. Sin embargo, la concentración de ALB también puede verse afectada por otros procesos como estados catabólicos y desnutrición, por lo que no es un indicador específico para enfermedades hepáticas [1,5,13].

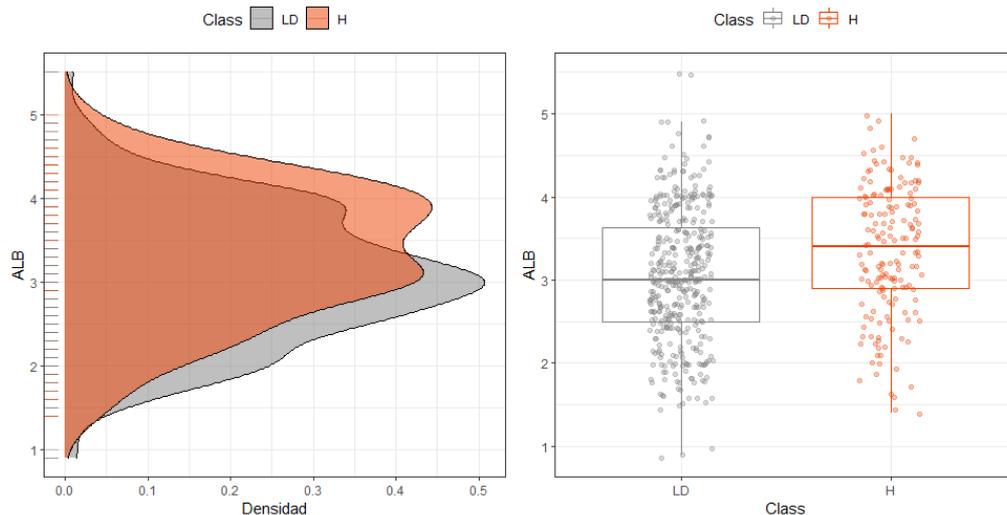


Figura 29. Gráficos de densidad y boxplots por clase de la albúmina (ALB) del conjunto de datos ILPD. LD: Enfermedad hepática; H: No enfermedad hepática.

En la figura 30 se observa que la distribución de la proporción entre albúmina y globulina (AG) es parecida para pacientes hepáticos y sanos. Sin embargo, los pacientes sanos presentan mayor densidad en valores intermedios y altos de AG, mientras que los pacientes hepáticos presentan mayor densidad en valores bajos de AG.

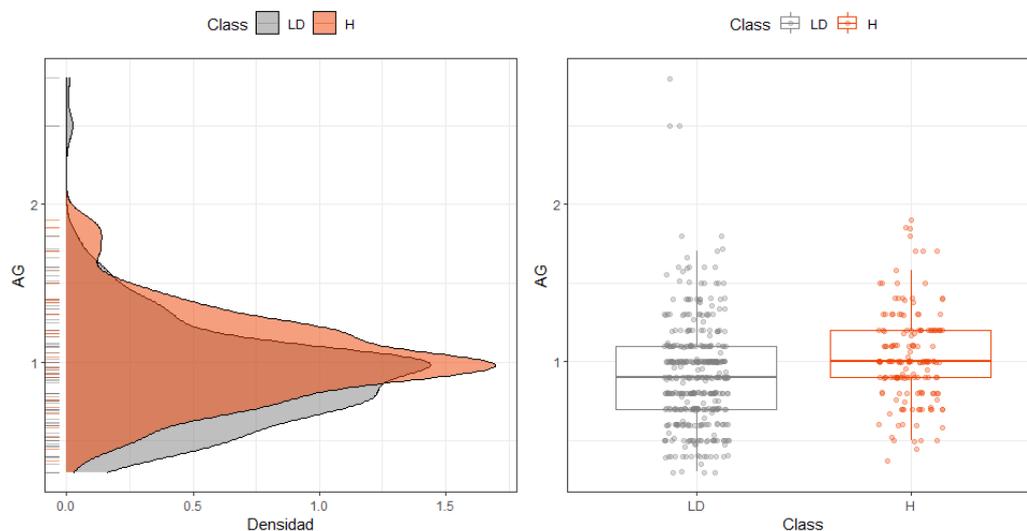


Figura 30. Gráficos de densidad y boxplots por clase de la proporción entre albúmina y globulina (AG) del conjunto de datos ILPD. LD: Enfermedad hepática; H: No enfermedad hepática.