Diagnóstico diferencial de anemias mediante parámetros hematológicos y técnicas de machine learning

Jorge Ferriz Vivancos

MU Bioinf. i Bioest.
Bioinformática Estadística y Aprendizaje Automático

Tutora de TF:

Romina Astrid Rebrij

Profesor responsable de la asignatura:

Carles Ventura Royo

20/06/2023

Universitat Oberta de Catalunya

UNIVERSITAT DE BARCELONA

Título del trabajo:	*Diagnóstico diferencial de anemias mediante parámetros hematológicos y técnicas de machine learning*
Nombre del autor:	*Jorge Ferriz Vivancos*
Nombre de la consultora:	*Romina Astrid Rebrij*
Nombre del PRA:	*Carles Ventura Royo*
Fecha de entrega:	*06/2023*
Titulación o programa:	*Máster en Bioinformática y Bioestadística*
Área del Trabajo Final:	*Statistical Modelling, Machine Learning and Statistical bioinformatics*
Idioma del trabajo:	*Castellano*
Palabras clave	*aprendizaje automático, datos de población celular, anemia*

Resumen del Trabajo

La anemia es una condición que afecta a millones de personas en todo el mundo, y pese a que su etiología es muy variada, más de la mitad de los afectados presentan déficit nutricional de hierro, vitamina B12 o ácido fólico. Para su diagnóstico es imprescindible la determinación de hemoglobina mediante autoanalizadores hematológicos, seguido de una caracterización con biomarcadores específicos.

Con el diseño de los autoanalizadores hematológicos modernos han surgido una batería de nuevos marcadores que se determinan conjuntamente a la hemoglobina, y cuya utilidad clínica ya ha sido validada en otras enfermedades.

En este trabajo se han desarrollado modelos de aprendizaje automático, mediante el paquete caret de R, que permitan detectar mediante estos nuevos parámetros hematológicos la carencia de los nutrientes necesarios en la eritropoyesis, con el objetivo de un diagnóstico precoz de la anemia sin necesidad de la determinación de estos marcadores, lo que permitiría además una adecuación a la demanda dentro del laboratorio clínico.

Los mejores resultados han sido los generados por modelos entrenados con los algoritmos eXtreme Gradient Boosting, Random Forest y Support Vector Machine, con posterior ensamblado mediante caretEnsemble, obteniendo las mejores predicciones en los casos con anemia por deficiencia de hierro, cercanos al 75% de precisión.

Abstract

Anemia is a condition that affects millions of people worldwide, and although its etiology is highly diverse, more than half of the affected individuals exhibit nutritional deficiencies in iron, vitamin B12, or folic acid. The determination of hemoglobin through hematological autoanalyzers is essential for its diagnosis, followed by characterization using specific biomarkers.

With the design of modern hematological autoanalyzers, a battery of new markers has emerged and are determined in conjunction with hemoglobin, and their clinical utility has already been validated in other diseases.

In this study, machine learning models have been developed using the caret package in R, which enable the detection of deficiencies of nutrients necessary for erythropoiesis through these new hematological parameters. The objective is to achieve early diagnosis of anemia without the need for determining these markers, thus allowing for better adequacy of the demand within the clinical laboratory.

The best results were obtained from models trained using the eXtreme Gradient Boosting, Random Forest, and Support Vector Machine algorithms, with subsequent assembly using caretEnsemble, achieving the highest predictions in cases with iron deficiency anemia, close to 75% accuracy.

Índice

Índice de ilustraciones

Índice de tablas

1. Introducción

1.1 Contexto y justificación del trabajo

Se define anemia a la afección en la que existe una baja concentración de hemoglobina en los hematíes. Esta hemoglobina es necesaria para el transporte de oxígeno a las células, por lo que en pacientes anémicos este signo clínico se manifiesta en forma de piel pálida, fatiga, mareos, cianosis o taquicardia y que, en casos severos, puede ser potencialmente mortal. Actualmente se estima que aproximadamente 1620 millones de personas en todo el mundo padecen anemia, lo que corresponde al 24,8% de la población. Este porcentaje es incluso superior en determinados grupos poblacionales: 42% en niños menores de 5 años y 37% en embarazadas.

Existen múltiples causas para esta afección, como una síntesis inadecuada por déficit de precursores de la eritropoyesis, destrucción de los hematíes en casos de anemias hemolíticas o pérdida de hematíes por hemorragias. De igual forma, las anemias pueden ser clasificadas en función del volumen corpuscular medio de los hematíes (VCM) y de su capacidad regenerativa.

Es, por tanto, que ante la gran diversidad de tipos de anemia que se manejan en la práctica clínica, resulta imprescindible para el personal sanitario contar con herramientas eficaces que permitan una correcta clasificación, como son los biomarcadores analíticos.

	Regenerativas (Reticulocitos)	Arregenerativas (Reticulocitos)
Microcítica (VCM<83)	- Anemia hemolítica congénita (esferocitosis, talasemia, etc.)	- Anemia ferropénica - Anemia procesos crónicos
Normocítica (VCM 84-97)	- Anemia hemolítica extracorpuscular (hiperesplenismo, microangiopatía,etc.) - Hemorragia aguda	- Aplasia medular - Nefropatía crónica
Macrocítica (VCM>98)	- Crisis hemolíticas	- Déficit ácido fólico - Déficit vitamina B12 - Enfermedad hepática - Hipotiroidismo

Tabla 1. Clasificación de anemias en función de tamaño y capacidad de regeneración

Así, muchos tipos de anemia se clasifican en función de la clínica y de uno o varios marcadores, a destacar: hierro en caso de anemias ferropénicas, vitamina B12/ácido fólico en anemias megaloblásticas o pruebas de biología molecular en talasemias. Otros marcadores de utilidad en todos los tipos de anemia y que las caracterizan son los parámetros hematológicos incluidos en el hemograma. A su vez, los analizadores hematológicos (Coulter) proporcionan determinados datos numéricos que definen características cualitativas de diferentes poblaciones celulares, conocido como datos de población celular (Cell Population Data o CPD). Estos datos han sido introducidos recientemente en el estudio de diversas patologías y su utilidad en la clasificación de anemias aún no ha sido validada clínicamente.

Serie roja	**Hematíes, hemoglobina, hematocrito (Hto), volumen corpuscular medio (VCM), hemoglobina corpuscular media (HCM), concentración hemoglobina corpuscular media (CHCM), ancho distribución eritrocitario (ADE) y su SD, eritroblastos.**
Serie blanca	Leucocitos, neutrófilos (%), neutrófilos (absoluto), linfocitos (%), linfocitos (absoluto), monocitos (%), monocitos (absoluto), eosinófilos (%), eosinófilos (absoluto), basófilos (%), basófilos (absoluto).
Serie plaquetar	Plaquetas, volumen plaquetar medio.
Serie reticulocítica	Reticulocitos (%), reticulocitos (absolutos), fracción reticulocitos inmaduros (FRI), volumen reticulocitario medio.

Tabla 2. Parámetros hematológicos realizados en el hemograma

Por lo general, a todo paciente en su seguimiento clínico por parte de atención primaria se le solicita una bioquímica básica y un hemograma, y salvo sospecha de anemia o determinados grupos de pacientes, el clínico no suele solicitar las pruebas complementarias mencionadas anteriormente. Al ser el hemograma una prueba común, rápida y barata, y cuyos parámetros son predictores del tipo de anemia, se propone en este trabajo fin de master, mediante métodos de aprendizaje automático (machine learning o ML), el diseño de modelos que permitan predecir la deficiencia de determinados biomarcadores (hierro, vitamina B12 y ácido fólico), a fin de poder clasificar las anemias más frecuentes en la población únicamente con parámetros hematológicos, de tal forma que se evite la determinación innecesaria de otros marcadores.

Los principales motivos para seleccionar este tema han sido la disponibilidad de un elevado volumen de datos, la alta prevalencia de anemia y la reciente aparición de los CPD, junto al potencial de estos y el resto de parámetros hematológicos de dirigir el diagnóstico diferencial de las anemias de tal forma que se reduzcan tiempos de diagnóstico y operativos, así como costes asociados en el laboratorio clínico. Por otro lado, a título personal me genera un gran interés profundizar en el aprendizaje del machine learning, ya que se considera una herramienta con un presente y futuro prometedor dentro del campo de los laboratorios clínicos, rama en la que desempeño mi trabajo habitual.

1.2 Objetivos del trabajo

Objetivos generales

- Clasificar pacientes con anemia en base a parámetros hematológicos sin necesidad de realizar pruebas complementarias.

Objetivos específicos

- Extraer mediante análisis estadísticos los parámetros hematológicos de mayor influencia en los distintos subtipos de anemia.
- Diseñar un modelo de machine learning que permita predecir la deficiencia de hierro, vitamina B12 y ácido fólico en pacientes anémicos.
- Optimizar el modelo seleccionado con el objetivo de lograr una precisión mínima del 80%.
- Evaluar influencia de los CPD en el modelo y en la clasificación de los pacientes con anemia.

El planteamiento inicial del trabajo incluía como objetivo secundario la elaboración de una aplicación en Shiny para la implementación del modelo final. No obstante, este objetivo fue eliminado en la Fase 2 del proyecto para centrar el trabajo en técnicas de ensemble learning, no incluidas inicialmente en el proyecto, y así intentar aumentar la precisión de los modelos.

1.3 Impacto en sostenibilidad, ético-social y de diversidad

Se ha velado por cumplir los criterios de sostenibilidad, responsabilidad social y diversidad, por lo que este trabajo se considera adecuado para todos los usuarios con independencia de si son hombres o mujeres, sin resultados sesgados ni estereotipados.

Se considera que el impacto general de este proyecto es positivo, en los siguientes aspectos:

- Se produce un impacto positivo respecto a la responsabilidad social, pues se pretende una detección precoz de una de las enfermedades más prevalentes a nivel mundial, con lo que se permitiría un diagnóstico precoz y una reducción de la morbi-mortalidad. Además, la reducción de gastos asociados a la enfermedad, junto a la posibilidad de mejorar la adecuación a la demanda, permitiría un claro beneficio económico para la sociedad.

- Pese a que este trabajo no tiene un impacto positivo directo sobre la diversidad de género, al tratarse de una condición que afecta en situaciones especiales a las mujeres (embarazo o menstruación) podría considerarse un impacto positivo sobre este género, efecto inherente de la propia solución.

- No se espera un impacto relevante sobre la sostenibilidad medioambiental. No obstante, podría considerarse un impacto mínimo positivo ya que la menor realización de pruebas

conllevaría un ahorro energético junto a menor emisión de residuos por parte de los analizadores presentes en el laboratorio.

1.4 Enfoque y método seguido

Para desarrollar este trabajo se ha aplicado un enfoque secuencial en el que se han ido realizando secuencialmente las tareas definidas posteriormente, con el propósito de llevar un control continuo de los resultados. Así, este TFM viene definido en las siguientes etapas:

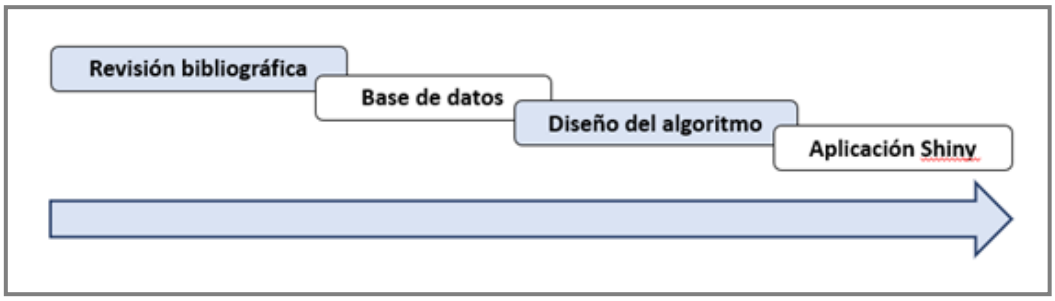


Ilustración 1. Enfoque seguido para la realización del Trabajo

Revisión bibliográfica

Se ha realizado una evaluación del estado del arte actual mediante búsqueda bibliográfica y lectura crítica de temas relacionados con este proyecto. Para ello se ha hecho uso de recursos de libre acceso disponibles en la biblioteca de la UOC, así como aquellos obtenidos en buscadores especializados, como PubMed. La bibliografía obtenida y utilizada para la redacción de la memoria de TFM está citada al final de la presente memoria de acuerdo con las normas APA.

Elaboración base de datos

Se ha elaborado una base de datos propia a partir de todas las determinaciones de parámetros hematológicos realizadas en un hospital terciario mediante analizadores Beckman Coulter DxH900®. Esta base de datos incluye datos brutos e información redundante que debe ser procesada previamente:

- Explorar datos y creación de subsets: para el posterior diseño de los modelos se han creado tres conjuntos de datos específicos para el estudio individual de cada tipo de deficiencia. Así, se disponen de tres bases de datos: una para estudio del déficit de vitamina B12, otra para déficit de ácido fólico y otra para déficit de hierro. Los principales criterios de inclusión para cada caso son la existencia de anemia en adultos (<12g/dl en mujeres y <13g/dl en hombres) y disponer de resultado del biomarcador específico de cada subset.
- Definición de categorías: se clasifican los casos, según punto de corte de vitamina B12 (180 pg/mL), ácido fólico (3 ng/mL) y hierro (60 µg/dL), en “Déficit” y “Normal”.

- Preprocesado de datos y control de calidad: eliminar datos nulos, omitir variables de varianza cero y altamente correlacionadas, normalización de los datos y sobremuestreo de la clase minoritaria, entre otras.

El número de casos para cada uno de los subsets se ha estimado entre n=3500 para la base de datos del hierro, n=2000 para la base de datos de vitamina B12 y n=1000 para la base de datos de ácido fólico, de los que se analizan más de 100 variables de parámetros hematológicos junto a sexo y edad. Las distintas categorías en las que se clasifican los datos se estiman correctamente balanceadas y en caso contrario (posible riesgo), se aplicarían técnicas para su corrección (p.e. data augmentation o transfer learning).

Diseño de los modelos

Para la realización de este proyecto se ha empleado el software R y R Studio ya que se trata de uno de los programas más empleados en este ámbito de la bioinformática y que permite disponer de todas las herramientas necesarias para lograr los objetivos del proyecto, con una gran comunidad y múltiple información disponible online. También se trata de un lenguaje comúnmente empleado durante la realización del máster y la asignatura de "Machine Learning".

En este caso, debido a la disponibilidad de datos ya clasificados, se emplean paquetes de R con algoritmos enfocados al aprendizaje supervisado. Para ello se ha buscado en la literatura los algoritmos más prometedores para el problema a resolver, probando distintos enfoques para así implementar aquellos que mejores resultados se obtengan. Se plantean inicialmente como potenciales algoritmos a emplear: k-NN, SVM, Naive-Bayes, Random Forest y redes neuronales.

Previo al diseño del modelo se lleva a cabo un primer estudio estadístico, también con R, de cada una de las variables que permita conocer qué grado de relación existe entre estas variables y la presencia o no de anemia, así como extraer las características principales.

La estrategia general de diseño de cada modelo ha sido:

- Transformación de los datos en función del algoritmo seleccionado.
- Selección de parámetros más adecuados para cada algoritmo.
- Construcción del modelo.
- Training de muestra de entrenamiento (80%) y testeo en muestra de validación (20%).
- Optimización del modelo.
- Analizar cuál es mejor modelo óptimo en función de los resultados.

1.5 Planificación del trabajo

1.4.1 Tareas

Para la realización de este TFM se han establecido una serie de tareas que se deben llevar a cabo para conseguir los objetivos propuestos. Pudiendo coincidir o no en línea temporal con cada una de las PECS, estas tareas son:

PEC 1 - Definición y plan de trabajo (01/03-20/03 – 20 días)

A. Búsqueda bibliográfica sobre la enfermedad y sobre las distintas técnicas de ML disponibles, así como librerías de R que implementen técnicas de Machine Learning.
B. Obtención, limpieza y filtrado de bases de datos.
C. Desarrollo de la memoria de plan de trabajo.
D. Entrega y revisión del plan de trabajo.

PEC 2 – Desarrollo del trabajo – Fase 1 (21/03-24/04 – 34 días)

A. Preprocesamiento de los datos.
B. Análisis estadístico y de regresión de las variables.
C. Extracción de características que se emplearán en los modelos de ML.
D. Selección de técnicas de ML a emplear según características de las variables.
E. Construcción y entrenamiento de los modelos de ML y análisis de posibles mejoras.
F. Desarrollo, entrega y revisión del documento de Fase 1.
G. Comienzo de redacción de la memoria final

PEC 3 – **Desarrollo del trabajo – Fase 2** (25/04-29/05 – 34 días)

A. Validación del modelo más adecuado y mejora del rendimiento.
B. Interpretación y análisis de los resultados.
C. Desarrollo, entrega y revisión del documento de Fase 2.
D. Integración en la memoria final de los resultados obtenidos y conclusiones.

PEC 4 – **Cierre de la memoria y la presentación** (30/05-20/06 – 22 días)

A. Adecuación y optimización final de la memoria de TFM.
B. Elaboración de la presentación de defensa de TFM.

PEC 5 – **Defensa pública** (03/07-14/07)

En el siguiente apartado “Calendario” se puede observar de forma gráfica la duración temporal asociada a cada una de las tareas.

1.4.2 Calendario

Mediante el diseño de un diagrama de Gantt se establece la duración estimada para cada tarea, seleccionada en función de su aparente complejidad, las características del problema y los conocimientos previos del estudiante.

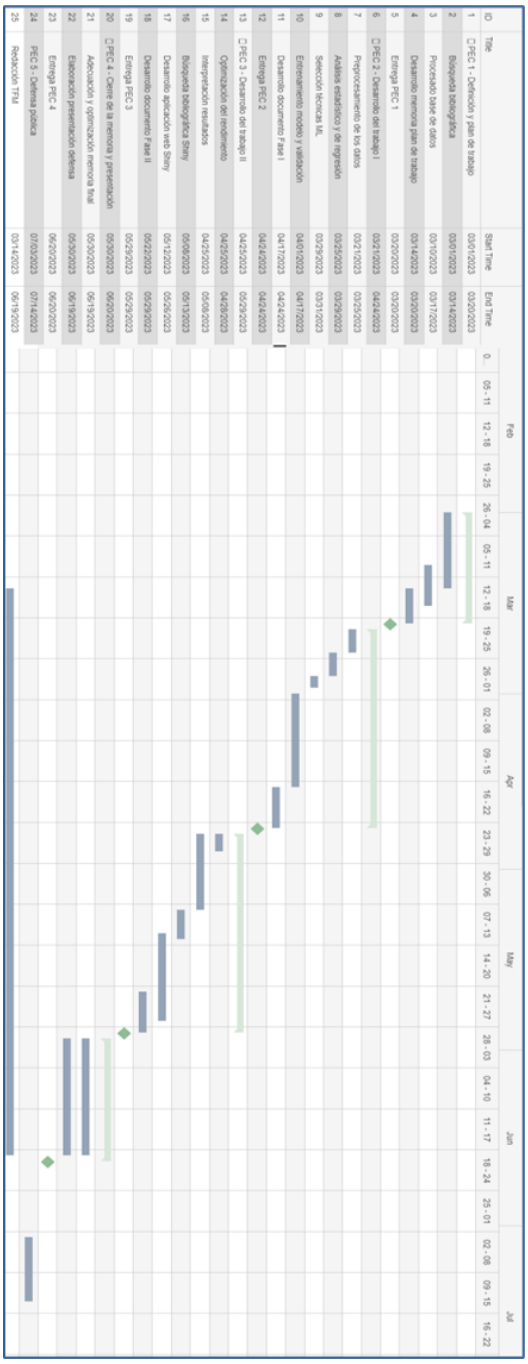


Ilustración 2. Diagrama de Gantt para la realización del TFM

La carga de trabajo de esta asignatura es de 15 ECTS, es decir, 375 horas de trabajo, por lo que se ha estimado una dedicación media de 3-3,5 horas diarias, incluyendo fines de semana, a la realización de las tareas y consecución de los objetivos. Este número de horas es razonable con la duración definida previamente para poder llevar a cabo el proyecto sin alterar los tiempos establecidos.

1.4.3 Hitos

Se establecen dos clases de hitos: aquellos definidos en el plan docente, considerados estrictos, y aquellos internos que siendo más flexibles permitan la consecución de los objetivos y un mejor

control temporal del proyecto. Se definen en la siguiente tabla junto a la fecha prevista de realización:

Hitos plan docente	Hitos específicos internos
PEC 1 – Entrega memoria plan de trabajo 20/03	1B. Obtención base de datos 17/03
PEC 2 – Entrega desarrollo trabajo Fase 1 24/04	2E. Entrenamiento ML 17/04
PEC 3 – Entrega desarrollo trabajo Fase 2 29/05	3A. y 3B. Finalización proyecto 19/06
PEC 4 – Entrega de memoria y presentación 20/06	
PEC 5 – Realización defensa pública	

Tabla 3. Fecha prevista de consecución de hitos

1.4.4 Análisis de riesgos

Se han evaluado los posibles riesgos, y a fin de minimizar su impacto potencial, establecido medidas correctivas, de tal forma que no se alteren sustancialmente los tiempos establecidos. De igual forma y como se define en la PEC1, existen dos factores influyentes y que son inherentes a cualquier proyecto: su alcance, y el tiempo del que se dispone para realizarse.

Posible Riesgo: Problemas en la obtención de datos de CPD (Impacto medio).

Posible Solución: Construcción del modelo con únicamente parámetros hematológicos.

Posible Riesgo: Obtención de datos desbalanceados entre los distintos subgrupos (Impacto medio).

Posible Solución: Empleo de estrategias de corrección.

Posible Riesgo: Fallos informáticos o de software (Impacto alto).

Posible Solución: Búsqueda de alternativas a R o a alguna de sus librerías.

Posible Riesgo: Bajo rendimiento del modelo de ML (Impacto alto).

Posible Solución: Búsqueda de posibles causas y construcción de modelos alternativos, modificando técnicas de aprendizaje.

1.6 Sumario de productos obtenidos

Al finalizar este trabajo se han obtenido varios productos, entre los que destacan la presente memoria y los códigos escritos en R con el desarrollo del modelo. Se presentan por tanto los siguientes entregables:

- Memoria final: en la que se incluye el contexto, la justificación, los objetivos, el enfoque, los resultados y las conclusiones del presente Trabajo Final del Máster.

- Códigos fuente de R: se encuentran disponibles tres códigos de R (uno para cada biomarcador) en un repositorio GitHub creado para tal fin, accesibles a través de: https://github.com/jorgefv23/TFM_anemia_JFV

- Video de presentación: a través del cual se exponen las ideas principales del trabajo realizado por medio de diapositivas. Este video está disponible para visualización pública en la plataforma Present@ de la asignatura.

1.7 Descripción de los otros capítulos de la memoria

Los siguientes capítulos de esta memoria describirán más en profundidad los apartados anteriores, que han servido como un resumen de la justificación y realización del trabajo. Por tanto, este trabajo se ha estructurado de la siguiente manera:

Apartado 2. Estado del arte: Clasificación de anemia

En este capítulo se ha profundizado en los distintos tipos de técnicas disponibles actualmente para el estudio de la anemia, que variarán en función de su clasificación. Se hace especial hincapié en los nuevos parámetros disponibles en autoanalizadores hematológicos y los avances que han generado en el estudio de anemia. Para ello, se comenta, junto a una búsqueda bibliográfica exhaustiva, la situación actual y los avances más recientes obtenidos en este campo.

Apartado 3: Materiales y métodos: Machine learning

Este apartado detalla los materiales empleados para llevar a cabo el diseño de los distintos modelos de aprendizaje automático, junto a una información más detallada de los métodos empleados. Se subdivide en las distintas fases realizadas: elaboración de la base de datos, preprocesado de los datos, entrenamiento del modelo, evaluación del modelo y ajuste y optimización.

Apartado 4: Resultados

Los resultados obtenidos se comentan junto a gráficas y tablas. Aquellos sean de menor importancia pero merezcan ser informados, se anexarán al final de este trabajo (Anexo).

Apartado 5: Discusión y conclusiones

Por último, se discuten los resultados y se exponen las conclusiones del proyecto.

2. Estado del arte: Diagnóstico de anemia

2.1 Clasificación de anemia

La anemia es una afección global que constituye un grave problema de salud pública por su elevada prevalencia, de hasta un 40% según el grupo poblacional, y que afecta especialmente a niños (42%), ancianos (26%) y embarazadas (37%) [1].

		Anemia*		
Población	**Sin anemia***	**Leve***	**Moderada**	**Grave**
Niños de 6 a 59 meses de edad	110 o superior	100-109	70-99	menos de 70
Niños de 5 a 11 años de edad	115 o superior	110-114	80-109	menos de 80
Niños de 12 a 14 años de edad	120 o superior	110-119	80-109	menos de 80
Mujeres no embarazadas (15 años o mayores)	120 o superior	110-119	80-109	menos de 80
Mujeres embarazadas	110 o superior	100-109	70-99	menos de 70
Varones (15 años o mayores)	130 o superior	100-129	80-109	menos de 80

Ilustración 3. Concentración de hemoglobina (g/L) para el diagnóstico de anemia

Según la OMS, se entiende como anemia aquellas situaciones en las que el número de hematíes o la concentración de hemoglobina es inferior a lo normal. La hemoglobina es una proteína presente en los hematíes de la sangre, cuya función es transportar el oxígeno captado en los pulmones hacia el resto de los tejidos del cuerpo. Está compuesta por cuatro subunidades, cada una de las cuáles contiene una porción llamada hemo, con un átomo de hierro esencial para la unión al oxígeno. Las necesidades fisiológicas de hemoglobina pueden variar en función de la edad, el sexo, el tabaquismo o las diferentes etapas del embarazo, entre otras [2].

La anemia no debe ser tratada como enfermedad sino como consecuencia de otras causas subyacentes. De tal forma, no todos los pacientes con anemia serán sintomáticos, sino que dependerá de la etiología de la anemia y la presencia de otras comorbilidades, como la enfermedad oncológica o cardiovascular. En aquellos que presentan síntomas, estos pueden variar desde cansancio, mareos, dificultad respiratoria, taquicardia o en situaciones más graves, ser potencialmente mortal [2].

La aparición de esta condición puede ser debida a uno de los siguientes procesos:

- Descenso en la producción de hematíes: es necesario que la formación de hematíes (hematopoyesis) sea un proceso continuo que supere la vida media de los hematíes (120 días). En el déficit de nutrientes, como hierro, vitamina B12 o ácido fólico, o en aquellas

situaciones en las que la médula no es capaz de producir estos precursores, como la aplasia medular, la cantidad de hematíes producidos disminuirá ya que no se regeneran.

- Aumento en la destrucción de hematíes: algunos ejemplos son las anemias hemolíticas autoinmunes o hereditarias, talasemias o determinadas infecciones como la malaria. En estos casos el organismo es capaz de producir suficiente hemoglobina, pero existe un proceso que destruye los hematíes a mayor velocidad.

- Pérdida de sangre: por traumas y sangrados (por ejemplo, durante la menstruación) o debido al efecto dilucional durante el embarazo.

Según el tipo de anemia presente se producirán una serie de cambios fisiológicos u otros, que serán la base para el diagnóstico definitivo y posterior tratamiento del paciente.

2.2 Métodos de detección

Previo a la realización de pruebas complementarias para determinar la etiología y junto a los signos clínicos presentes, en el diagnóstico de anemia es fundamental una cuantificación de la hemoglobina y/o hematocrito del paciente [3], para así determinar la severidad de la enfermedad.

Pese a que existen numerosos métodos para su cuantificación es la espectrofotometría tras lisis de los hematíes el gold estándar actual [4], que puede encontrarse en prácticamente todos los laboratorios clínicos en los conocidos como autoanalizadores hematológicos. Estos analizadores informan además sobre otros parámetros de la serie roja imprescindibles en la clasificación de las anemias, como el volumen corpuscular medio (VCM) o la hemoglobina corpuscular media (HCM). Por ejemplo, mientras las anemias por déficit de hierro son anemias microcíticas e hipocrómicas (↓VCM y ↓HCM), las producidas por déficit de vitamina B12 o ácido fólico son anemias macrocíticas e hipocrómicas (↑VCM y ↓HCM).

A su vez, los analizadores hematológicos informan mediante tecnología Coulter acerca de otras células sanguíneas como las plaquetas, los leucocitos o los eritroblastos y reticulocitos (precursores de los hematíes). Se ha comprobado que determinadas anemias pueden producir cambios cualitativos o cuantitativos en estas otras líneas celulares, como la hipersegmentación de los neutrófilos en casos de déficit de vitamina B12 [5].

El principio Coulter se fundamenta en la detección y medición de cambios en la resistencia eléctrica generados por una célula al atravesar una pequeña abertura cilíndrica. A su paso, cada célula individual altera momentáneamente la impedancia entre 2 electrodos generando así un impulso eléctrico [6]. Estos impulsos eléctricos se suman obteniéndose el recuento de células totales, mientras que la amplitud del impulso eléctrico generado dependerá del volumen de la célula y permite clasificar los leucocitos en sus distintos subtipos: neutrófilos, linfocitos, monocitos, eosinófilos y basófilos.

Con la aparición de los nuevos analizadores hematológicos de última generación han surgido nuevos parámetros, denominados Cell Population Data (CPD o datos de población celular en español) [7], que miden las características de las células basándose en la dispersión de la luz a distintos ángulos y que permite informar sobre el volumen, la conductividad y la dispersión de las distintas células sanguíneas.

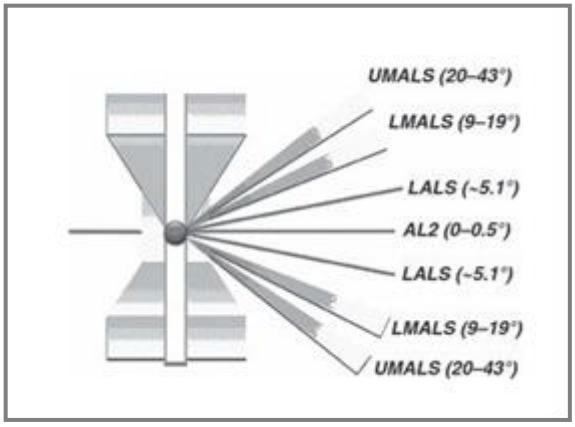


Ilustración 4. Médidas de dispersión empleados por tecnología Coulter para cálculo de CPD [7]

Pese a que estos parámetros no han sido validados ni implementados clínicamente, están siendo estudiados ampliamente ya que presentan la ventaja de ser datos de poco coste que se generan paralelamente a una prueba básica y común como es el hemograma. Se ha demostrado la utilidad de algunos de estos parámetros de CPD en el diagnóstico de múltiples enfermedades, como la leucemia [8] o infecciones víricas [9], siendo el ejemplo más representativo el del ancho de distribución de los monocitos para el diagnóstico de sepsis [10].

	Mean Ne	SD Ne	Mean Ly	SD Ly	Mean Mo	SD Mo	Mean Eo	SD Eo	Mean EGC	SD EGC
V	@MN-V-NE	@SD-V-NE	@MN-V-LY	@SD-V-LY	@MN-V-MO	@SD-V-MO	@MN-V-EO	@SD-V-EO	@MN-V-EGC	@SD-V-EGC
C	@MN-C-NE	@SD-C-NE	@MN-C-LY	@SD-C-LY	@MN-C-MO	@SD-C-MO	@MN-C-EO	@SD-C-EO	@MN-C-EGC	@SD-C-EGC
MALS	@MN-MALS-NE	@SD-MALS-NE	@MN-MALS-LY	@SD-MALS-LY	@MN-MALS-MO	@SD-MALS-MO	@MN-MALS-EO	@SD-MALS-EO	@MN-MALS-EGC	@SD-MALS-EGC
UMALS	@MN-UMALS-NE	@SD-UMALS-NE	@MN-UMALS-LY	@SD-UMALS-LY	@MN-UMALS-MO	@SD-UMALS-MO	@MN-UMALS-EO	@SD-UMALS-EO	@MN-UMALS-EGC	@SD-UMALS-EGC
LMALS	@MN-LMALS-NE	@SD-LMALS-NE	@MN-LMALS-LY	@SD-LMALS-LY	@MN-LMALS-MO	@SD-LMALS-MO	@MN-LMALS-EO	@SD-LMALS-EO	@MN-LMALS-EGC	@SD-LMALS-EGC
LALS	@MN-LALS-NE	@SD-LALS-NE	@MN-LALS-LY	@SD-LALS-LY	@MN-LALS-MO	@SD-LALS-MO	@MN-LALS-EO	@SD-LALS-EO	@MN-LALS-EGC	@SD-LALS-EGC
AL2	@MN-AL2-NE	@SD-AL2-NE	@MN-AL2-LY	@SD-AL2-LY	@MN-AL2-MO	@SD-AL2-MO	@MN-AL2-EO	@SD-AL2-EO	@MN-AL2-EGC	@SD-AL2-EGC

Tabla 4. Algunos CPD obtenidos del canal de leucocitos

(MN= Mean, SD= Standard Deviation, V= Volume, C= Conductivity, EGC= Early Granulated Cells, NE= Neutrophils, LY= Lymphocytes, MO=Monocytes, EO= Eosinophils)

	Mean NRBC	SD NRBC	Mean Non-NRBC	SD Non-NRBC
V	@MN-V-NRBC	@SD-V-NRBC	@MN-V-NNRBC	@SD-V-NNRBC
C	@MN-C-NRBC	@SD-C-NRBC	@MN-C-NNRBC	@SD-C-NNRBC
MALS	@MN-MALS-NRBC	@SD-MALS-NRBC	@MN-MALS-NNRBC	@SD-MALS-NNRBC
UMALS	@MN-UMALS-NRBC	@SD-UMALS-NRBC	@MN-UMALS-NNRBC	@SD-UMALS-NNRBC
LMALS	@MN-LMALS-NRBC	@SD-LMALS-NRBC	@MN-LMALS-NNRBC	@SD-LMALS-NNRBC
LALS	@MN-LALS-NRBC	@SD-LALS-NRBC	@MN-LALS-NNRBC	@SD-LALS-NNRBC
AL2	@MN-AL2-NRBC	@SD-AL2-NRBC	@MN-AL2-NNRBC	@SD-AL2-NNRBC

Tabla 5. CPD obtenidos del canal de eritroblastos (NRBC)

Mientras que el diagnóstico primario se realiza con la hemoglobina, para la clasificación definitiva de la anemia se requieren pruebas específicas destinadas a tal fin. Junto a los parámetros obtenidos del hemograma suele ser necesario tanto una visualización del frotis sanguíneo para caracterizar morfológicamente las células como la determinación de determinados marcadores sanguíneos que influyen en la síntesis y metabolismo de la hemoglobina, siendo los más frecuentemente realizados en un laboratorio clínico el hierro, la vitamina B12 y el ácido fólico. Estos marcadores son pruebas sencillas y automatizadas en los analizadores de bioquímica, y que presentan un coste unitario relativamente bajo. Es por ello por lo que son ampliamente demandadas por el clínico tanto en sospechas de anemia como en controles rutinarios de sus pacientes.

No obstante, ante tanta demanda su coste total dentro del laboratorio está aumentando en los últimos años junto al uso de recursos humanos [11]. Por ello los laboratorios están adoptando estrategias de adecuación a la demanda que permitan rechazar o generar estas pruebas en función de determinados parámetros, como la edad o el sexo [12]. Una de estas estrategias es la aplicación de técnicas de aprendizaje automático, que ya ha mostrado su utilidad junto a los parámetros de población celular para el manejo de otras enfermedades, como el diagnóstico SARS-CoV2 [9], o en la diferenciación de talasemias [13].

Respecto a la bibliografía existente ya han sido publicado algunos artículos que mencionan la utilidad del aprendizaje automático en el diagnóstico de anemia, como el de Rahim et al. que permitiría diferenciar la anemia ferropénica de la β-talasemia [14], o el de Saputra et al. que distingue con hasta un 99% de precisión mediante parámetros hematológicos el déficit de hierro de otras hemoglobinopatías [15]. Otros artículos, como el Vohra et al. utilizan técnicas de machine learning para predecir la severidad de la anemia [16].

Aunque estos trabajos parecen estar muy relacionados a lo propuesto en este proyecto, plantean conceptos diferentes, ya que comparan la anemia con otras enfermedades hematológicas donde suelen producirse cambios característicos significativos, a diferencia de este trabajo donde la causa de anemia de la población considerada como “Normal” puede generarse por muy diversos mecanismos fisiológicos.

No se ha hallado, en la fecha de redacción de este proyecto, ninguna bibliografía en la que se plantee una predicción del déficit de hierro, vitamina B12 o ácido fólico mediante la combinación de

parámetros CPD y técnicas de aprendizaje automático, por lo que podemos considerar este trabajo novedoso, aún más si tenemos en cuenta lo nuevos marcadores emergentes.

Con todo lo anterior, en este trabajo, siguiendo el objetivo de establecer un diagnóstico precoz de los distintos déficits, y de mejorar la adecuación a la demanda de pruebas de laboratorio, se plantea el uso de métodos de machine learning para predecir posibles deficiencias de hierro, vitamina B12 o ácido fólico que pudiesen ser causa conocida (o no) de anemia.

3. Materiales y métodos: machine learning

3.1 Elaboración de la base de datos

En este trabajo se han trabajado con tres bases de datos, una para cada nutriente necesario en la eritropoyetis, a estudiar: subset_Fe, subset_B12 y subset_Fol.

Los datos han sido obtenidos de resultados clínicos de pacientes a los que se les ha solicitado hemograma junto a hierro y/o vitamina B12 y/o ácido fólico, en el periodo comprendido entre el 1 de marzo y el 20 de abril de 2023, previa autorización por parte del Comité Bioético de Investigación del Hospital General de Valencia para tratar de manera anónima los datos de los pacientes.

Los datos relativos a parámetros básicos del hemograma, hierro, vitamina B12 y ácido fólico, junto a los demográficos del paciente, se obtuvieron del sistema informático de laboratorio Modulab de Werfen© en formato .xls. Por el contrario, los parámetros relativos al CPD tuvieron que ser directamente extraídos de los autoanalizadores DxH900 de Beckman Coulter© en formato .csv. Ambos archivos fueron fusionados según número de petición, eliminando aquellos casos en los que faltara algún valor. De esta forma, en el periodo mencionado anteriormente se obtuvieron datos analíticos de más de 25000 pacientes.

Se lleva a cabo un tratamiento preliminar de los datos, a fin de ajustarlos a los criterios de inclusión propuestos. Tras anonimizar los datos, y renombrar variables, se eliminan aquellos pacientes que:

- Tuviesen petición previa en menos de 3 meses, para evitar posibles interferencias por comienzo de tratamiento
- Fueran menores de 18 años
- No presentaran anemia (Hb > 13g/dL en hombres y Hb > 12g/dL en mujeres)
- No dispusieran de datos de CPD por algún problema técnico

Se eliminan a su vez todas las variables relativas a reticulocitos. Si bien habrían sido de utilidad en este estudio al tratarse de precursores de los hematíes, el % de analizados con datos de reticulocitos era inferior al 10%, por lo que se considera que su inclusión hubiera requerido de una imputación de valores perdidos que podría falsear los resultados de los modelos añadiendo ruido.

Finalmente, se categorizan los resultados de los distintos biomarcadores según tuviesen déficit o no, estableciendo como puntos de corte, por debajo del cuál se consideraría déficit:

- Hierro < 60 µg/dL
- Vitamina B12 < 180 pg/mL
- Ácido Fólico < 3ng/mL

Tras esta preparación previa se obtienen 3 subsets en formato .xls, siendo subset_Fe del que mayor número de observaciones se disponía (4275, 16.44% del total de peticiones). Subset_B12 tenía 2122 observaciones (8.16% del total) y subset_Fol 1773 (6.7% del total).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	B12	Edad	Sexo	LEU	ULE	NEUp	NEU	LINp	LIN	MONp	MON	EOSp	EOS	BASp	BAS	ERIT	HGB
2	2	75	1	6,1	6,1	71,1	4,4	14,3	0,9	12,1	0,7	1,8	0,1	0,7	0,0	1,90	5,6
3	2	55	1	11,2	11,2	78,1	8,8	11,2	1,3	7,8	0,9	2,0	0,2	0,9	0,1	3,10	6,4
4	2	90	1	9,4	9,4	78,0	7,3	12,9	1,2	5,6	0,5	2,3	0,2	1,2	0,1	2,10	6,7
5	2	89	1	3,7	3,7	68,1	2,5	20,6	0,8	11,0	0,4	0,1	0,0	0,2	0,0	2,10	6,7
6	2	76	0	7,7	7,7	63,4	4,9	22,7	1,8	10,3	0,8	2,9	0,2	0,7	0,1	3,30	6,6
7	2	85	1	6,2	6,2	65,1	4,0	19,8	1,2	10,1	0,6	3,3	0,2	1,7	0,1	3,30	6,9
8	2	80	1	1,7	1,7	70,7	1,2	20,8	0,4	3,0	0,1	2,9	0,0	2,6	0,0	2,30	7,0
9	2	83	1	6,4	6,4	73,8	4,7	9,1	0,6	12,8	0,8	3,4	0,2	0,9	0,1	2,50	7,1
10	2	60	0	15,8	15,8	81,7	12,9	14,6	2,3	2,7	0,4	0,9	0,1	0,1	0,0	2,30	6,7
11	2	85	1	11,6	11,6	85,2	9,9	10,8	1,3	3,0	0,3	0,2	0,0	0,8	0,1	2,50	7,1

Ilustración 5. Base de datos para la vitamina B12

3.2 Exploración y preprocesado de los datos

Para la realización de este proyecto se ha optado por emplear el paquete CARET de R por presentar múltiples ventajas para usuarios de nivel básico-intermedio en funciones de programación o machine learning. La principal radica en su facilidad para automatizar tareas tanto en el proceso de entrenamiento como en el preprocesado, permitiendo al usuario desde la imputación de valores faltantes hasta la selección de hiperparámetros de los modelos. Por otro lado, es un paquete muy flexible y permite entrenar diversos modelos con la misma interfaz y del que se dispone abundante información y recursos online [17-18].

Este paquete también ha sido el utilizado para muchos pasos del procesado, que se detallan a continuación:

Codificación de variables

Una de nuestras variables (sexo) es bicategórica, por lo que es necesario codificarla como variable dummy para pasarla a formato numérico y así pueda ser incluida en los modelos [19]. Afortunadamente, caret dispone de una función dummyVars() que simplifica este paso:

y <- dummyVars (" ~ Variable" , data = x , fullRank = T)
z <- data.frame (predict(y , newdata = x))

Posteriormente se convierten en factor tanto esta variable sexo (0-1 a etiquetas "Mujer" y "Hombre") como la variable dependiente (1-2 a etiquetas "Déficit" y "Normal"), mediante la función as.factor().

Eliminación de valores ausentes

Mediante map_dbl() del paquete "purrr" se mapearon los subsets en busca de valores ausentes (NA), encontrando algunas observaciones de las que no se disponía algún valor de variable, principalmente

las relacionadas con los eritroblastos. Al tratarse de muy pocas observaciones, se decide eliminarlas del subset mediante la función delete.na() generada para tal fin.

Eliminación de variables con varianza cero

Es importante eliminar variables con varianza cercana a 0 ya que no proporcionan ninguna información significativa al modelo y permite reducir la dimensionalidad de los datos centrando el estudio en variables más importantes [20]. Para eliminarlas, se emplea la función nearZeroVar() de caret, que proporciona información en forma de tabla acerca de freqRatio (proporción del valor más repetido respecto al segundo más repetido, nos interesa que sea lo más bajo posible) y percentUnique (porcentaje de valores únicos, cuanto más alto mejor) (Anexo 1).

Eliminación de variables altamente correlacionadas

En estudios de ML, la presencia de variables altamente correlacionadas entre sí (multicolinealidad) puede ser un problema que genere estimaciones de parámetros inexactos o excluya predictores significativos, ya que, al estar relacionadas, se vuelve difícil determinar la influencia independiente de cada variable sobre la variable respuesta [17]. Para eliminar esta multicolinealidad se recurre a la función findCorrelation() de caret que estima la correlación entre todas las variables y permite establecer un cut-off para seleccionar aquellas más correlacionadas (en este caso se establece en 0.8) y eliminar una de ellas (Anexo 2).

Partición en training y test

Se dividen los tres subsets en una nueva base de datos para entrenamiento y una nueva base de datos para evaluación, con el 80% y 20% de los datos iniciales, respectivamente. Nuevamente una función integrada en caret es de utilidad, createDataPartition().

Escalado y centrado de los datos

Se trata de un paso fundamental en el preprocesado de los datos, ya que la escala y magnitud de las variables, si son muy diferentes, puede llevar a algunos modelos a malinterpretar los datos. Es necesario igualarlos para que aquellos que tengan una escala mayor no enmascaren al resto [20]. En nuestro caso, un ejemplo podrían ser las plaquetas, con valores de media aproximada 300 frente a los valores del resto de variables en torno a 10-20 (Anexo 3). Una técnica para mitigar este problema es realizar un centrado de los datos, restando a cada valor el de su media, y posteriormente un escalado, dividiendo cada característica por su desviación estándar. De esta forma, todos los datos tienen una distribución normal. Para llevarlo a cabo, en caret se dispone de la función preProcess() con método "center" y "scale":

```
function_scale <-preProcess (data , method =c( "center", "scale" ))
data_scaled <-predict (function_scale , data)
```

Selección de variables

Se seleccionan aquellas variables relevantes para la construcción del modelo, con la finalidad de reducir una complejidad de datos que pueda generar ruido o sobreajuste, al mismo tiempo que disminuya el tiempo de entrenamiento. Se prueban y comparan tres tipos de métodos de selección de variables:

- Regularización LASSO (Least Absolute Shrinkage and Selection Operator): se pretende obtener el subconjunto de predictores que presente la mayor certeza en las predicciones, mediante el forzado de los coeficientes a 0 [21] (Anexo 4).
- Análisis de Componentes Principales (PCA): reducción de dimensionalidad que identifica patrones subyacentes en datos de alta dimensionalidad. Transforma las variables iniciales en nuevas variables no correlacionadas llamadas Componentes principales [20,22]. En caret puede realizarse mediante la función preProcess() (Anexo 5 y 6).
- Boruta: basado en el algoritmo Random Forest, realiza una serie de iteraciones en las que comprueba si una característica tiene mayor importancia que la mejor de sus características de sombra y va eliminando aquellas características que no considera importantes [23].

Tras la comparación, los mejores resultados se obtienen con Boruta, por lo que se decide emplear este paquete para seleccionar las variables más importantes.

Tratamiento de datos desbalanceados

Como se ha visto anteriormente, nuestros datos se encuentran desbalanceados hacia la población “Normal", lo que puede afectar considerablemente el rendimiento de los modelos al causar sesgos en las predicciones, generando modelos con una engañosa alta precisión, pues puede clasificar bien la clase mayoritaria generando un elevado % de acierto, pero sin clasificar correctamente la clase minoritaria. Para compensar este problema se pueden llevar a cabo técnicas de “sampling”, como, por ejemplo:

- ROSE (Random Over-Sampling Examples): aumenta el número de casos de la clase minoritaria de forma aleatoria.
- SMOTE (Synthetic Minority Over-sampling Technique): genera nuevas instancias sintéticas interpolando características de las instancias existentes [24].
- Undersampling: al contrario que las anteriores, reduce la clase mayoritaria para igualarla a la minoritaria. También existen métodos mixtos que combinas el undersampling con oversampling.

Aunque alguna técnica anterior se encuentra integrada en el paquete caret() mediante el argumento *sampling= “x”*, de la función train(), no permiten modificar las condiciones de sobremuestreo por lo que se recurre en este trabajo a probar técnicas de ROSE (con el paquete homónimo) y SMOTE (con

el paquete performanceEstimation(). Tras comparar rendimiento de ambas, se selecciona SMOTE como técnica de sobremuestreo de los datos [25].

3.3 Entrenamiento del modelo

Una vez preprocesados los datos se utilizan los predictores más relevantes para entrenar modelos de clasificación que sean capaces de detectar el déficit de hierro, vitamina B12 o ácido fólico en casos de anemia. Una de las ventajas del paquete "caret" es disponer de una interfaz unificada que permite acceder a más de 230 algoritmos de aprendizaje automático de forma sencilla adaptándose a diferentes tipos de conjuntos de datos. De igual forma, como se ha visto en el apartado anterior, ofrece funciones integradas para el preprocesado de datos.

Mediante la función train() de caret hemos entrenado los diferentes algoritmos utilizando una sintaxis similar, donde los principales argumentos generales que hemos modificado en este trabajo han sido:

*train (**y** , **data** = , **method** = , **metric** = , **trControl** = , **tuneGrid** = , ...)*

y: Variable objetivo que se está tratando de predecir: Fe, B12 o Fol
data: Base de datos sobre la que trabajaremos, incluye las variables predictoras
method: Algoritmo que se empleará para entrenar el modelo
metric: Métrica de evaluación utilizada para seleccionar el mejor modelo
trControl: Se especifica cómo se realizará la validación cruzada y el remuestreo
tuneGrid: Configuración de hiperparámetros empleados para ajustar el modelo

Mientras que el argumento "tuneGrid" ha sido definido individualmente para seleccionar los mejores hiperparámetros en cada algoritmo (ver secciones correspondientes), el argumento "trControl" se ha especificado de forma cuasi general para todos ellos. En este argumento, definido mediante la función trainControl(), se indica en qué modo se va a realizar la validación cruzada en la que se entrena y evalúa el modelo repetidamente utilizando distintas submuestras de los datos [18]. En nuestro caso se ha optado por *method = repeatedcv, number = 10, repeats = 5*, donde una validación cruzada constará de 10 submuestras y se realizarán 5 particiones de los datos.

En este trabajo se ha decidido realizar el entrenamiento con los siguientes algoritmos: kNN, SVM, árboles de decisión, Random Forest, redes neuronales y eXtreme gradient boosting.

3.3.1 K-nearest neighbors

El algoritmo k-nearest neighbors (k-NN o k vecinos más cercanos en español) recibe este nombre porque clasifica los datos desconocidos en función de la información obtenida de los puntos más cercanos (vecinos). Calcula la distancia Euclidiana entre la instancia desconocida y el resto de puntos, y tras seleccionar el valor de k, se determina la etiqueta más frecuente en los k-vecinos más cercanos, asignándosela a la variable desconocida [26].

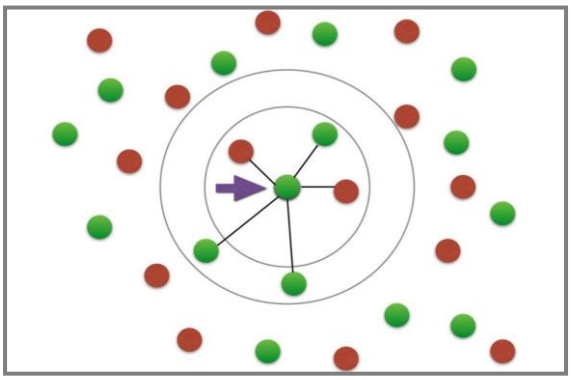


Ilustración 6. Algoritmo de aprendizaje automático: k-Nearest Neighbors
(Extraído de Medium)

Presenta como ventajas la sencillez de implementación y la flexibilidad, pudiendo utilizarse tanto en problemas de clasificación como de regresión. Por contra, son muy sensibles a características irrelevantes y a la elección de k: valores demasiado bajos de k pueden llevar a una clasificación errónea debido al ruido o valores atípicos y valores demasiado altos pueden pasar por alto patrones sutiles y generalizar, sobre todo en datos desbalanceados.

El método a emplear en caret es “knn” y los hiperparámetros a modificar en este algoritmo son:

		Valores empleados
k	Número de vecinos más cercanos	k = seq (1, 100, 2)

Tabla 6. Hiperparámetros empleados en entrenamiento kNN

3.3.2 Support Vector Machine

Las máquinas de soporte vectorial (Support Vector Machine o SVM) tienen como objetivo encontrar el hiperplano en un espacio N-dimensional (donde N es el número máximo de variables) que pueda separar los datos pertenecientes a distintas clases, de tal forma que queden lo más alejados unos de otros y se minimice el error de clasificación [27].

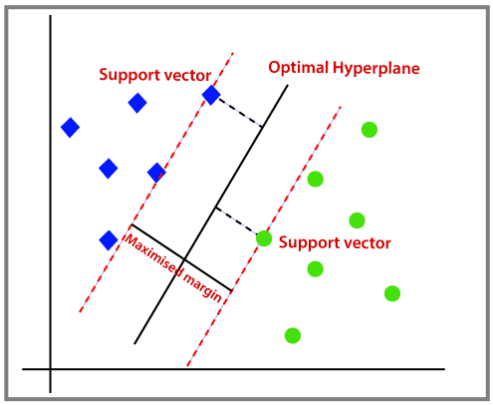


Ilustración 7. Algoritmo de aprendizaje automático: Support Vector Machine Linear
(Extraído de gitconnected.com)

Los datos cercanos a la línea de separación son conocidos como vectores de soporte y son fundamentales para definir el modelo. Si disponemos de mucha información y los vectores de soporte no pueden ser separados linealmente en el plano podemos recurrir a las funciones kernel, las cuáles transforman los datos a un espacio de mayor dimensionalidad donde sea más fácil encontrar una separación. Existen diferentes tipos de kernel, siendo los más empleados el kernel lineal, kernel radial o kernel polinómico, que definiremos específicamente en caret según el método empleado: "svmLinear", "svmRadial" o "svmPoly", respectivamente.

Los hiperparámetros a modificar varían según la función kernel empleada son:

			Valores empleados
Cost	Parámetro de penalización de error. Controla la compensación entre lograr un margen más amplio y minimizar el error de clasificación.	Linear Radial Polinómico	C = 0.01,0.1,1,10,50
Sigma	Controla la forma de la influencia de cada ejemplo de entrenamiento en la formación del margen de división.	Radial	sigma= 0.001,0.01,0.1,1
Degree	Grado del polinomio empleado como función de kernel. Determina la complejidad del margen de decisión.	Polinómico	degree= 2,3,4
Scale	Controla la escala de los predictores antes de aplicar el kernel polinómico.		scale= 0.01,0.1,1

Tabla 7. Hiperparámetros empleados en entrenamiento SVM

3.3.3 Árboles de decisión

Los árboles de decisión son un algoritmo de machine learning basado en un proceso de división del conjunto de datos en subconjuntos más pequeños, agrupando las observaciones según presenten valores similares respecto la variable dependiente.

Un árbol de decisión comienza con un nodo raíz, que representa todo el conjunto de datos, y se selecciona una variable y un umbral para dividir los datos en función de si cumplen o no el criterio de división. Este proceso continúa en cada subconjunto, creando nodos adicionales hasta alcanzar un criterio de parada. Si un nodo contiene datos de diferentes clases, se subdivide nuevamente hasta que las clases estén diferenciadas.

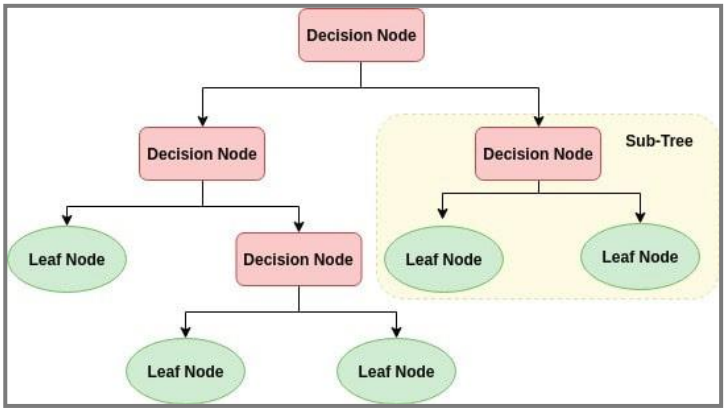


Ilustración 8. Algoritmo de aprendizaje automático: Árbol de decisión
[28]

El paquete base de R para la implementación de árboles de decisión es "rpart", que permite modificar el siguiente hiperparámetro:

		Valores empleados
cp	Complexity Parameter. Es el parámetro de complejidad del árbol generado. Un valor de 1 será un árbol sin divisiones, mientras que el valor 0 presentará la profundidad máxima.	cp= 0.0001,0.001,0. 1,0.5,1

Tabla 8. Hiperparámetros empleados en entrenamiento Árbol de decisión

3.3.4 Random Forest

El algoritmo Random Forest (RF o bosque aleatorio en español) se basa en la construcción de múltiples árboles de decisión en el que cada árbol se entrena de forma independiente con distintas muestras de datos (bagging) y selección aleatoria de características [29]. Posteriormente selecciona la clase mayoritaria según los promedios de los resultados de los distintos árboles de decisión.

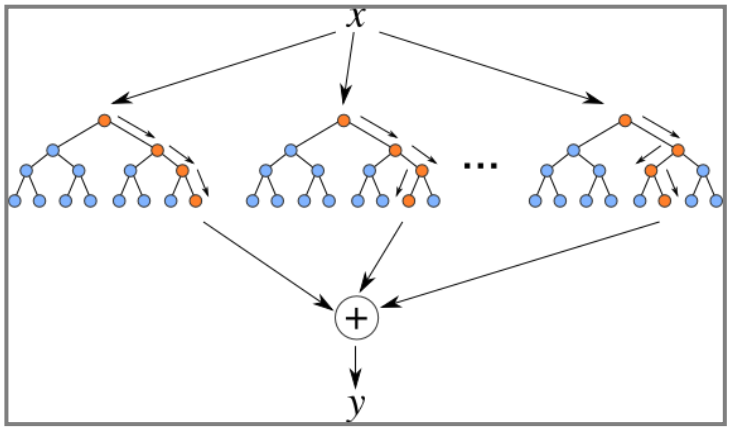


Ilustración 9. Algoritmo de aprendizaje automático: Random Forest
(Extraído de jorgecantero.es)

Se trata de uno de los algoritmos más empleados ya que tienden a reducir considerablemente el sobreajuste siendo menos susceptible a valores atípicos y ruido. Por contra, suele ser más costoso computacionalmente que los árboles de decisión, y su interpretación es más compleja.

En la librería caret existen distintos paquetes para ajustar este tipo de modelos, siendo algunos de los más empleados "randomForest" o "ranger". En nuestro caso se selecciona el método ranger() por su mayor velocidad y la posibilidad de ajustar más hiperparámetros [30]:

		Valores empleados
Mtry	Número de predictores seleccionados aleatoriamente en cada árbol.	mtry= 3,5,7,10,15
Splitrule	Criterio empleado para evaluar la calidad de una división en cada nodo	splitrule= "gini", "extratrees"
Min.node.size	Tamaño mínimo de observaciones requeridas en un nodo para que se realice una división adicional, lo que controla la profundidad y complejidad del bosque	min.node.size= 2,4,6

Tabla 9. Hiperparámetros empleados en entrenamiento Random Forest

3.3.5 Redes neuronales

Las redes neuronales son modelos de machine learning inspirados en el funcionamiento del cerebro humano, donde cada neurona toma una combinación lineal de salidas de las neuronas de la capa anterior, aplica una función de activación no lineal, y produce una salida hacia la siguiente capa de neuronas. Es posible ajustar la estructura de la red neuronal, especificando el número de neuronas de las capas ocultas [31].

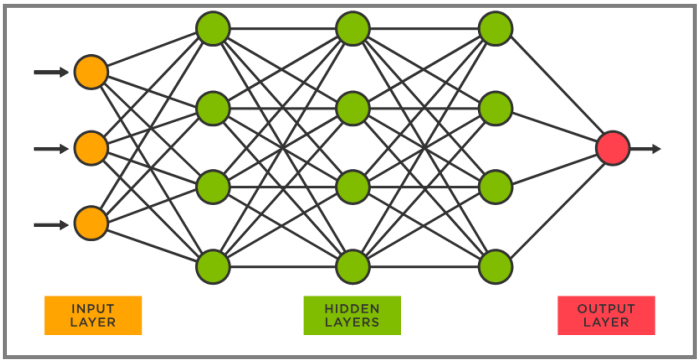


Ilustración 10. Algoritmo de aprendizaje automático: Red neuronal
(Extraído de TIBCO)

Para el entrenamiento en este trabajo de una red neuronal se ha optado por emplear el método "nnet" de la librería "caret", con el tuning de los siguientes hiperparámetros:

		Valores empleados
Size	Cantidad de neuronas en la capa oculta de la red neuronal.	size= 5,10,20,50
Decay	Controla la penalización aplicada a los pesos durante el entrenamiento para evitar un sobreajuste del modelo.	decay= 0.001,0.01,0.1,1

Tabla 10. Hiperparámetros empleados en entrenamiento red neuronal

Además de los hiperparámetros anteriores, el método "nnet" permite modificar otros argumentos que controlan la red neuronal. Uno de ellos es MaxNWts, que determina el número máximo de pesos permitidos, por encima del cuál el entrenamiento se detendrá. En nuestro caso, establecemos por defecto MaxNWts ya que es un valor razonable para una base de datos de tamaño moderado.

3.3.6 eXtreme Gradient Boosting

El último modelo de aprendizaje automático empelado, XGBoost, es una técnica de ensamblado que combina múltiples árboles de predicción débiles entrenándolos sucesivamente de tal forma que cada nuevo modelo corrija los errores cometidos por los modelos anteriores. XGBoost se asemeja al algoritmo de Random Forest, con la diferencia que en este último los árboles se construyen de forma independiente (bagging) mientras que en XGBoost se construyen secuencialmente (boosting) [31]. Además, presenta la ventaja de una mayor optimización que le permite aumentar la velocidad de entrenamiento del modelo.

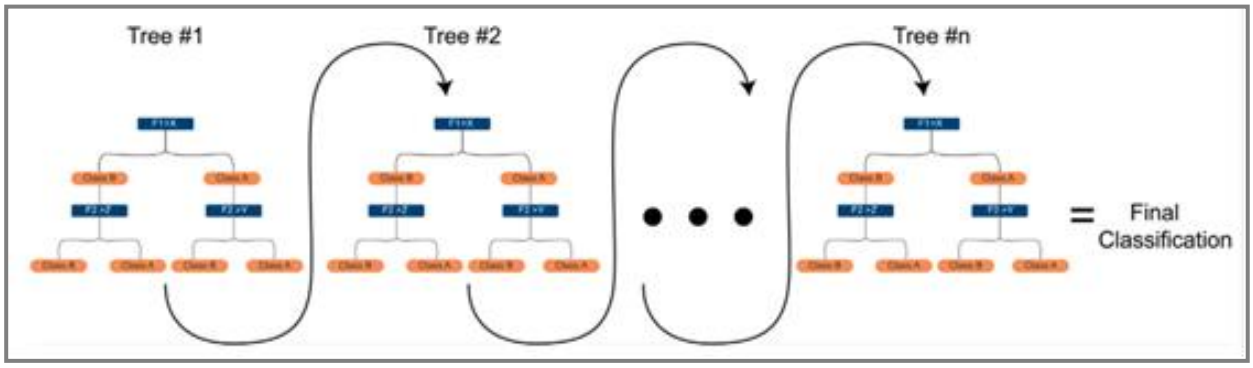


Ilustración 11. Algoritmo de aprendizaje automático: eXtreme Gradient Boosting
[32]

Los hiperparámetros que podemos modificar en este método ("xgbTree") han sido:

		Valores empleados
nrounds	Número de iteraciones. Controla el número máximo de árboles que se construirán.	nrounds= 20,50,70
Max_depth	Profundidad máxima de cada árbol, útil para controlar el sobreajuste de los datos.	max_depth = 7,15
Eta	Tasa de aprendizaje. Controla la contribución de cada árbol al modelo final.	eta= 0.01,0.1,0.5
Gamma	Umbral de reducción de pérdida requerido para dividir un nodo en cada árbol.	gamma= 0.2,0.4
Colsample_bytree	Proporción de columnas a considerar en cada árbol. Ayuda a controlar el sobreajuste.	colsample_bytree =0.8
min_child_weight	Valor mínimo de peso requerido en un nodo hoja. Controla la cantidad mínima de muestras necesarias en cada nodo.	min_child_weight = 1,3
Subsample	Proporción de muestras a considerar en cada árbol.	subsample= 0.8

Tabla 11. Hiperparámetros empleados en entrenamiento eXtreme Gradient Boosting

3.4 Evaluación del modelo

Para una evaluación completa y confiable de los distintos modelos generados se emplea el dataset de prueba que se genera previamente durante el preprocesado de los datos, y cuyos datos no se habían utilizado durante el entrenamiento. Este conjunto de prueba al representar datos no vistos permite evaluar cómo nuestros modelos se desempeñan en situaciones del mundo real. Durante esta evaluación es importante vigilar la aparición de overfitting (sobrejuste en español), un comportamiento de aprendizaje automático que se produce cuando el modelo proporciona predicciones precisas para los datos de entrenamiento, pero no para los datos nuevos. Puede ocurrir cuando el tamaño muestral es pequeño o los datos contienen mucha información irrelevante.

En caret disponemos de una sencilla herramienta para medir el rendimiento de un modelo, la matriz de confusión. Puede realizarse mediante la función confusionMatrix() con fórmula:

confusionMatrix (data= , reference= ,…)

Esta matriz de confusión consta de cuatro celdas en las que se informa:

- Verdaderos positivos (VP): aquellos casos en los que el modelo clasifica correctamente una muestra positiva. En nuestro caso, aquellos pacientes con "Déficit" que clasifica correctamente.
- Verdaderos negativos (VN): aquellos casos en los que el modelo clasifica correctamente una muestra negativa. Serían aquellos pacientes bien clasificados como "Normal".
- Falsos positivos (FP): aquellos casos que no presentan la condición, pero el modelo los clasifica como tal. En nuestro caso, pacientes normales que clasifica como "Déficit".
- Falsos negativos (FN): aquellos casos en los que presentan la condición, pero el modelo no es capaz de clasificarlos como tal. En nuestro caso, pacientes con déficit clasificados como "Normal".

		Predicción	
		Déficit	Normal
Observación	Déficit	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Normal	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Ilustración 12. Matriz de confusión para la evaluación del modelo

A partir de los valores anteriores se pueden calcular otras métricas que nos van a indicar cómo de fiable es nuestro modelo, destacando:

- Precisión (accuracy): mide la proporción de predicciones correctas respecto al total de predicciones → *(TP + TN) / (TP + TN + FP + FN)* y proporciona una visión general de cómo trabaja nuestro modelo. No obstante, este término puede ser engañoso en muestras desbalanceadas donde pueda producirse sobreajuste y clasifique correctamente la clase mayoritaria pero incorrectamente la minoritaria.

- Sensibilidad (o tasa de verdaderos positivos): proporciona información de cómo clasifica los casos positivos → *VP / VP + FN*. En nuestro modelo, interesa una sensibilidad elevada ya que no nos interesa que haya falsos negativos pues se pasaría por alto la enfermedad de muchos pacientes ("Déficit").

- Especificidad (o tasa de verdaderos negativos): informa sobre la proporción de verdaderos negativos entre todos los pacientes sanos → *VN / VN + FP*. En nuestro modelo, valores

elevados serían deseables para una posible adecuación a la demanda en la que se puedan excluir aquellos casos con resultado “Normal”.

- F1 Score: combina la precisión y la sensibilidad en un solo valor, calculando su media armónica, con la ventaja de tener en cuenta tanto falsos positivos como falsos negativos → *2 * ((precisión * sensibilidad) / (precisión + sensibilidad))*. Es especialmente útil en caso de tener datos desbalanceados [33].

En nuestro caso, como hemos balanceado los datos se ha optado por trabajar con estas métricas durante el entrenamiento, teniendo en cuenta un parámetro adicional “kappa”. Este parámetro es una medida que evalúa cómo se desempeña un modelo de clasificación en comparación con una suposición aleatoria, teniendo en cuenta el desbalanceo de clases [34]. Valores cercanos a 0 se asimilarían a una predicción similar al azar, mientras que valores cercanos a 1 indican una concordancia perfecta. No obstante, durante la evaluación del modelo, ya que los datos de test están desbalanceados, se ha optado por comparar los distintos modelos mediante F1-Score.

3.5 Ajuste y optimización

Una vez evaluados los modelos se realiza un proceso de optimización, que puede llevarse a cabo con la configuración tanto de la base de datos de entrenamiento como de los distintos modelos entrenados, todo ello con el objetivo de aumentar la precisión y/o la utilidad clínica. Esto se puede lograr, entre otras formas, mediante la regularización de variables, la selección de distintos hiperparámetros, con técnicas que reduzcan el sobreajuste, o simplemente, añadiendo nuevos datos.

Una estrategia que ha demostrado buenos resultados a la hora de aumentar la eficacia del modelo es el Ensemble Learning (o aprendizaje por conjuntos en español). Esta técnica combina las predicciones de múltiples modelos para obtener una única predicción más precisa y robusta. De esta manera los errores que podrían producirse en cada modelo se ven compensados por las fortalezas del resto de modelos. Algunos ejemplos de Ensemble Learning utilizados en este trabajo son voting, stacking y boosting.

3.5.1 Voting

La técnica de voting (o votación) es una técnica simple en la que se elige aquella predicción que haya sido seleccionada más veces por todos los modelos, pudiendo diferenciar entre dos tipos de votación:

- Hard voting: la clase que obtiene más votos se selecciona como predicción final, sin tener en cuenta las probabilidades de cada modelo para esa clase. Este es el método empleado en este trabajo al disponer de etiquetas, y no probabilidades, como output de las predicciones.
- Soft voting: se promedian las probabilidades de cada modelo y se selecciona la probabilidad más alta como predicción final.

3.5.2 Stacking

El stacking (o ensamblado jerárquico) se diferencia del anterior en la generación de un meta-modelo con características obtenidas a partir de las predicciones de los modelos empleados. Este meta-modelo emplea las predicciones de los otros modelos como datos de entrada, estos datos se agregan, y posteriormente el meta-modelo será el que haga la predicción de las nuevas observaciones.

Uno de los paquetes más conocidos para esta función es “caretEnsemble”. Este paquete consta de dos funciones principales: caretList() y caretStack(). La primera, que presenta los mismos argumentos que train() de caret, permite entrenar mediante validación cruzada distintos algoritmos simultáneamente con las mismas condiciones de remuestreo. Posteriormente, los resultados son agregados mediante la función caretStack(), con el empleo del algoritmo Random Forest [35].

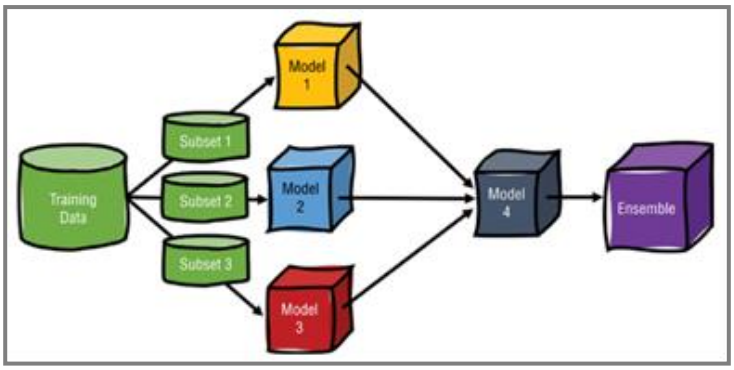


Ilustración 13. Ensemble learning: stacking
[36]

Otro algoritmo para realizar ensemble learning por stacking es el disponible en el paquete SuperLearner, que mediante validación cruzada estima el rendimiento de múltiples modelos y posteriormente les asigna un peso que será el que se apliquen a las predicciones realizadas por cada modelo, pudiendo incluso eliminar aquellos modelos que no contribuyan en el conjunto [37-38].

Además, para optimizar aún más el ensemble, se ha realizado "tuning" de los hiperparámetros de los modelos de SuperLearner con los mejores hiperparámetros obtenidos durante el entrenamiento, mediante la función:

```
SL.method.tune <- function(...){
    SL.method(..., hyperparameter1=..., hyperparameter2=...)}
```

3.5.3 Boosting

A diferencia de los métodos de ensamblado anteriores basado en promedios o votos de varios modelos, el boosting se centra en construir modelos mediante iteración a partir de modelos más débiles. Los modelos se ajustan y se agregan al conjunto secuencialmente, de modo que el segundo modelo intenta corregir las predicciones del primer modelo, el tercero corrige el segundo modelo, y así sucesivamente. De esta forma, sesga los datos de entrenamiento hacia aquellas observaciones que son más difíciles de predecir [39].

Los algoritmos más populares de boosting son Gbm (Gradient Boosting machine), XGBoost (Extreme Gradient Boosting) y AdaBoost (Adaptive Boosting), que será el programa empleado en este trabajo con el paquete “adabag”.

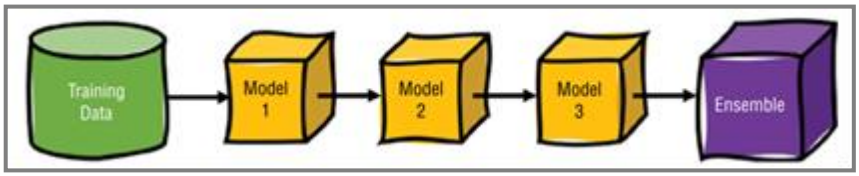


Ilustración 14. Ensemble learning: Boosting
[36]

Finalmente, para comprobar una posible utilidad del modelo en la práctica clínica habitual se modifican los umbrales de decisión al 0.1,0.15 y 0.2 del mejor modelo para la predicción del hierro, ya que fue el que como se verá más adelante proporciona las mejores predicciones. En caso de presentar una probabilidad intermedia se considera que no tiene capacidad predictiva suficiente para ser incluido y se informaría como “No concluyente”, a fin de seleccionar únicamente aquellas predicciones más fiables.

Posteriormente, se calcula sensibilidad y especificidad al aplicar estos umbrales y se estima el beneficio que tendría su aplicabilidad calculando los pacientes que permitiría diagnosticar o el posible ahorro en costes para el laboratorio.

4. Resultados

Se resumen en este apartado los resultados más destacables obtenidos en relación con los objetivos propuestos. Es importante recordar antes de continuar que todos los subsets disponibles previo a su procesado constaban de 110 variables correspondientes con parámetros hematológicos (más edad y sexo) y una variable respuesta, con un número diferentes de observaciones para cada nutriente: 4275 para el hierro, 2122 para la vitamina B12 y 1773 para el ácido fólico. No obstante, estas observaciones se encontraban desbalanceadas hacia la clase "Normal", motivo por el cuál posteriormente se aplicaron técnicas de sobremuestreo de la clase minoritaria.

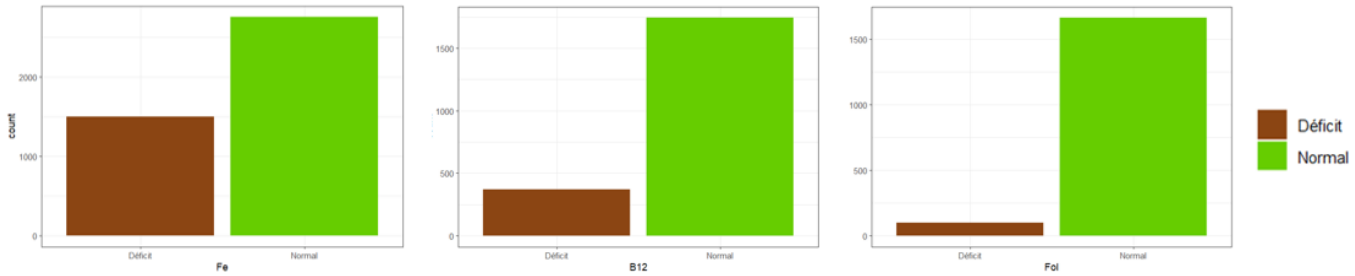


Ilustración 15. Distribución de casos para cada nutriente

Uno de los objetivos del estudio era extraer los parámetros hematológicos de mayor influencia en los distintos subtipos de anemia. Podemos observar cómo al comparar mediante pruebas paramétricas todas las variables hay algunas que presentan un p-valor muy bajo y que nos podrían orientar hacia una futura mayor importancia de la variable en el modelo. Principalmente se observan estas diferencias para el dataset del hierro, mientras que para el ácido fólico únicamente la variable MNVMO es estadísticamente significativa. Para cada tipo de anemia:

t-Student variables Fe		t-Student variables B12		t-Student variables Fol	
HCM	p=4.9686e-50	**MNCNE**	p=0.000034	**MNVMO**	p=0.007815
MNCNE	p=1.25e-35	**CHCM**	p=0.001520		
MNCMO	p=1.149e-31	**SDVNNRBC**	p=0.001925		
MNSMMO	p=7.758e-24	**ERIT**	p=0.021037		
MNCLI	p=2.074e-22	**MNSAEO**	p=0.036624		

Ilustración 16. t-Student de variables más representativas

Se representan gráficamente las principales variables para observar visualmente cómo se distribuyen esos datos, pudiendo comprobar en el caso del hierro como existe una ligera tendencia de ser superior la hemoglobina corpuscular medio (HCM) con los pacientes sin déficit de hierro. En el caso del ácido fólico el volumen medio de los monocitos (MNVMO) parece ser inferior en la población normal.

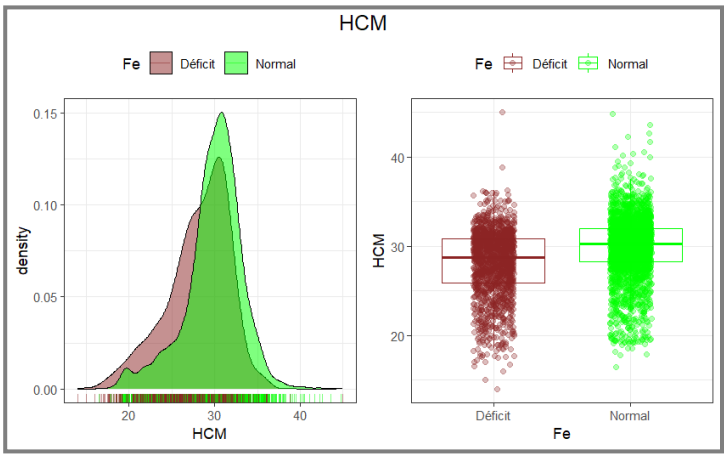


Ilustración 17. Distribución de HCM para el Hierro

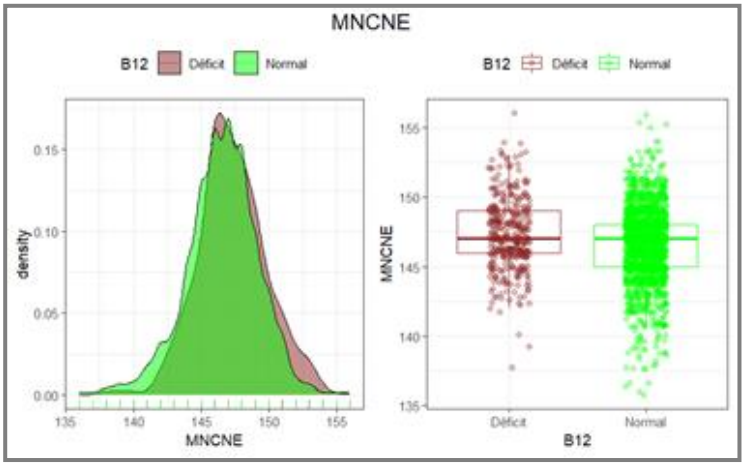


Ilustración 18. Distribución de MNCNE para la Vitamina B12

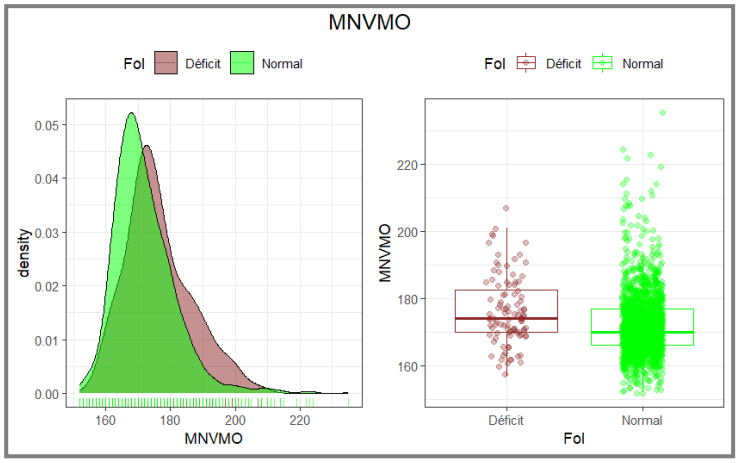


Ilustración 19. Distribución de MNVMO para el Ácido fólico

En la selección de variables realizada mediante el paquete Boruta() podemos observar la importancia de las variables calculadas para el conjunto de datos. Aquellas de mayor importancia se reflejan gráficamente de color verde y en la parte superior derecha de la gráfica. Como podemos comprobar,

todas las variables obtenidas previamente con asociación estadística son reconocidas como importantes para construir el modelo junto a otras nuevas variables como SDSUNE para el hierro, SDVNE para el ácido fólico y el porcentaje de linfocitos para la vitamina B12. En términos generales, en lo observado durante todas las técnicas de selección de predictores los predictores más asociados con los distintos déficits suelen ser los asociados a hemoglobina y hematíes, seguido de aquellos relacionados con neutrófilos y monocitos. Los linfocitos parecen ser relevantes en la predicción de vitamina B12 y ácido fólico, no así para el hierro. Por el contrario, las plaquetas no parecen ser útiles para vitamina B12 y fólico, pero sí para hierro. Los peores resultados se obtienen con los parámetros relacionados con basófilos, y en menor grado, eosinófilos.

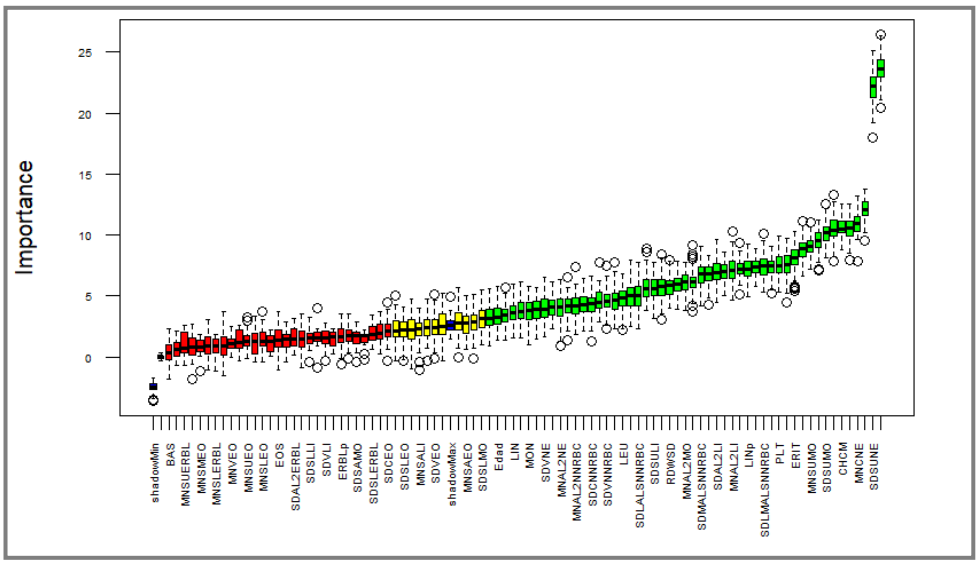


Ilustración 20. Importancia relativa de las variables para el Hierro según Boruta

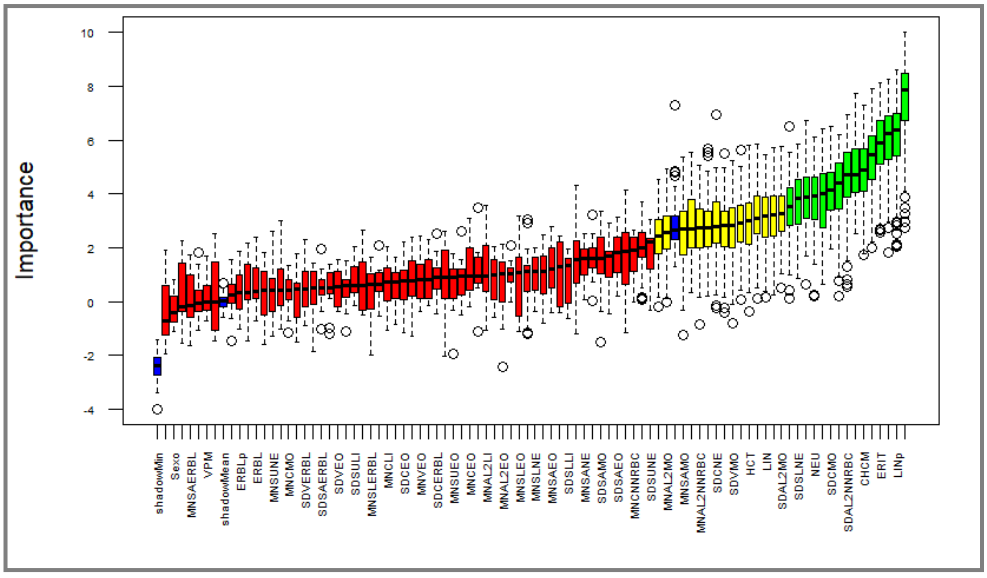


Ilustración 21. Importancia relativa de las variables para la Vitamina B12 según Boruta

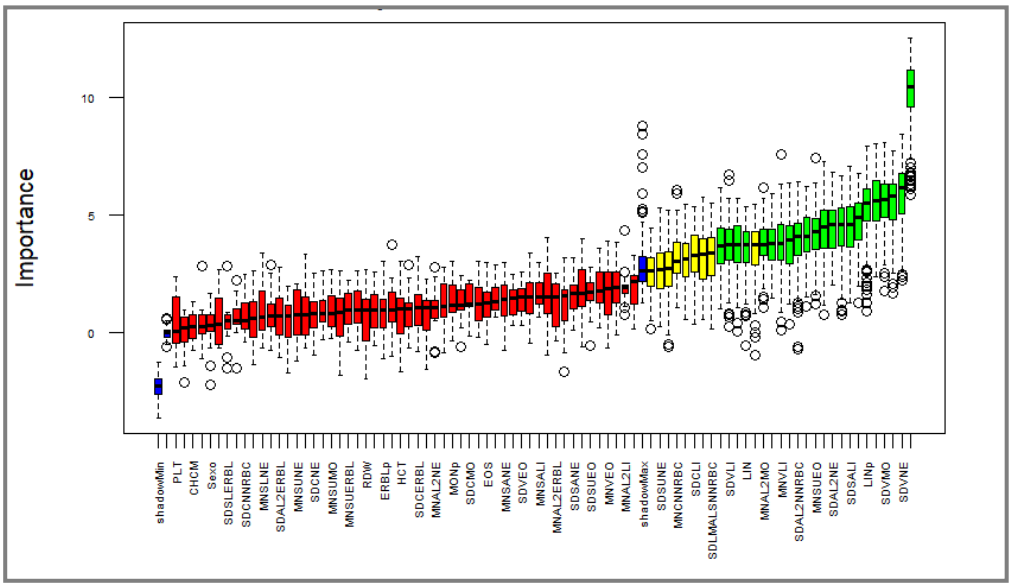
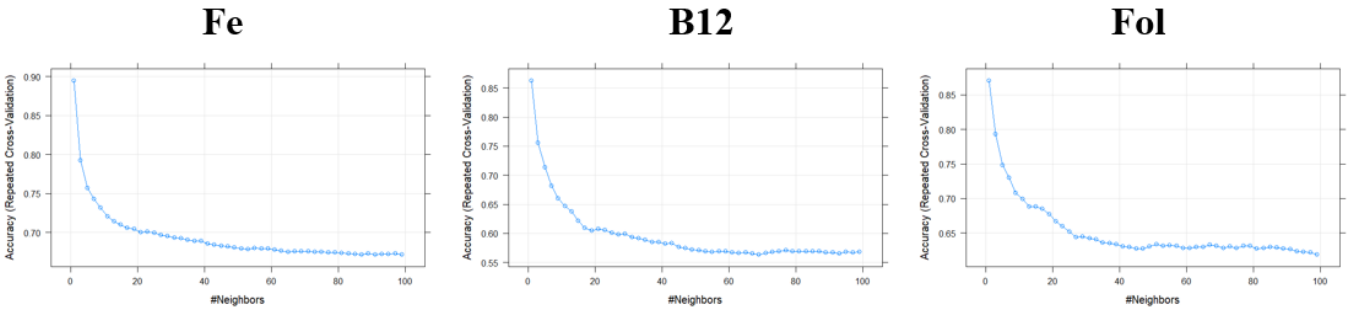


Ilustración 22. Importancia relativa de las variables para el Ácido fólico según Boruta

Una vez elaborados los subsets, preprocesado los datos y estudiado las variables más representativas en Boruta (junto a PCA y regularización Lasso, ver Anexo) los datos están listos para ser empleados en el entrenamiento de los distintos algoritmos seleccionados. Se representan a continuación los resultados de la validación cruzada de entrenamiento junto a los mejores hiperparámetros encontrados.

kNN



Accuracy = 0.8943798 Kappa 0.7887608	Accuracy = 0.8623217 Kappa 0.7246512	Accuracy = 0.8702941 Kappa 0.7439157
k = 1	k= 1	k = 1

Ilustración 23. Métricas de entrenamiento e hiperparámetros kNN

SVM

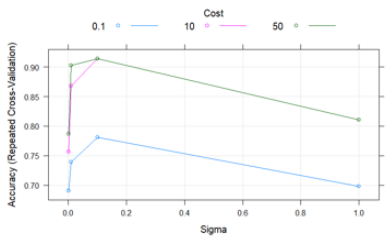
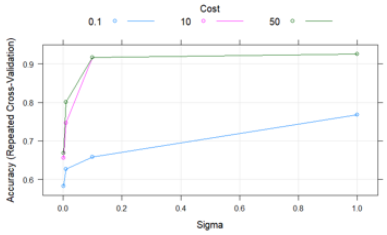
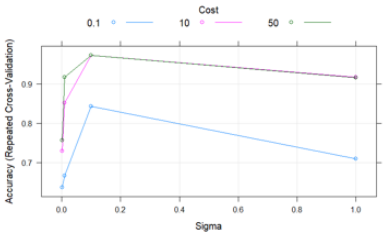
Fe	B12	Fol
		
Accuracy = 0.9145173 **Kappa 0.8290344**	**Accuracy = 0.9260237** **Kappa 0.8520519**	**Accuracy = 0.9730392** **Kappa 0.9458270**
sigma = 0.1 **C = 50**	**sigma = 1** **C = 10**	**sigma = 0.1** **C = 50**

Ilustración 24. Métricas de entrenamiento e hiperparámetros SVM

Árbol de decisión

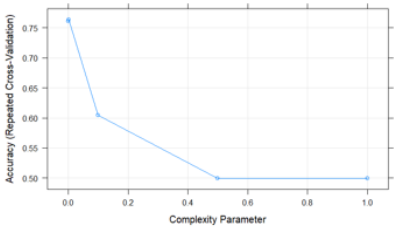
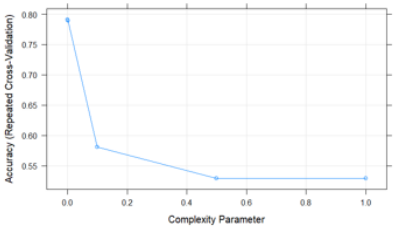
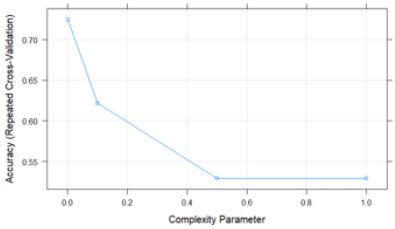
Fe	B12	Fol
		
Accuracy = 0.7632831 **Kappa 0.5265699**	**Accuracy = 0.7908641** **Kappa 0.5791906**	**Accuracy = 0.7244118** **Kappa 0.4471148**
cp = 0.01	**cp = 0.001**	**cp = 0.001**

Ilustración 25. Métricas de entrenamiento e hiperparámetros Árbol de decisión

Random Forest

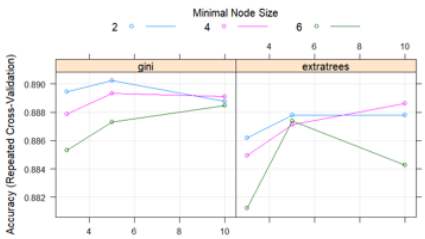
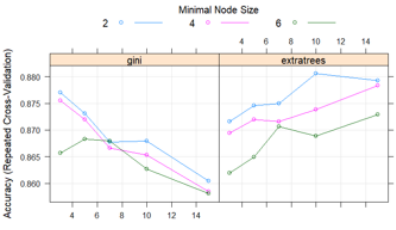
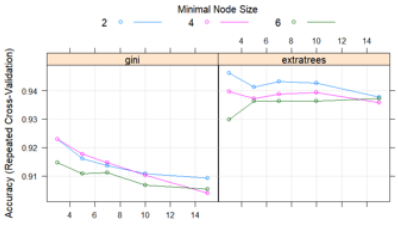
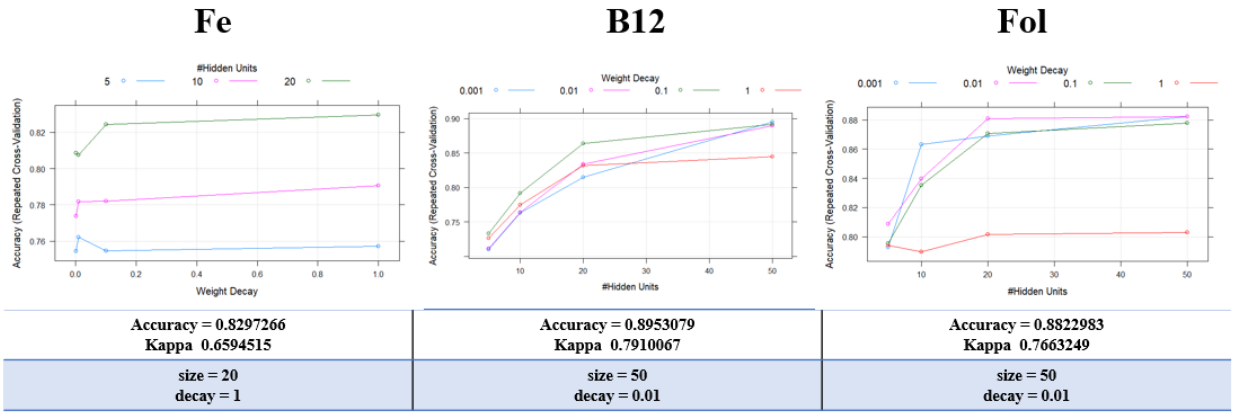
Fe	B12	Fol
		
Accuracy = 0.8902274 **Kappa 0.7804539**	**Accuracy = 0.8805379** **Kappa 0.7610879**	**Accuracy = 0.9460784** **Kappa 0.8918603**
mtry = 5 **splitrule = gini** **min.node.size = 2**	**mtry = 10** **splitrule = extratrees** **min.node.size = 2**	**mtry = 3** **splitrule = extratrees** **min.node.size = 2**

Ilustración 26. Métricas de entrenamiento e hiperparámetros Random Forest

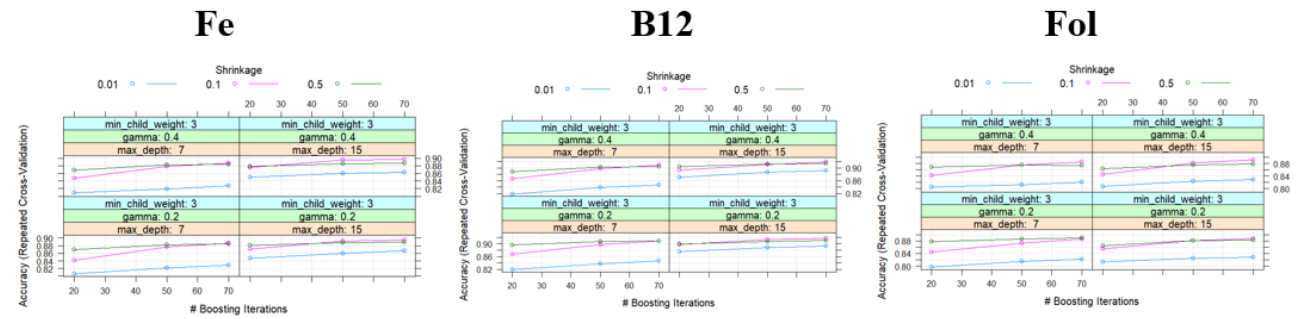
Red neuronal



Fe	B12	Fol
Accuracy = 0.8297266 Kappa 0.6594515	Accuracy = 0.8953079 Kappa 0.7910067	Accuracy = 0.8822983 Kappa 0.7663249
size = 20 decay = 1	size = 50 decay = 0.01	size = 50 decay = 0.01

Ilustración 27. Métricas de entrenamiento e hiperparámetros Red neuronal

XGBoost



Fe	B12	Fol
Accuracy = 0.9003590 Kappa 0.8007162	Accuracy = 0.9253631 Kappa 0.8500689	Accuracy = 0.9078431 Kappa 0.8154451
nrounds = 70 max_depth = 15 eta = 0.1 gamma = 0.4 colsample_bytree = 0.8 min_child_weight = 1 subsample = 0.8	nrounds = 70 max_depth = 15 eta = 0.1 gamma = 0.2 colsample_bytree = 0.8 min_child_weight = 1 subsample = 0.8	nrounds = 70 max_depth = 15 eta = 0.1 gamma = 0.2, colsample_bytree = 0.8 min_child_weight = 1 subsample = 0.8

Ilustración 28. Métricas de entrenamiento e hiperparámetros eXtreme Gradient Boosting

Los resultados obtenidos para el conjunto de entrenamiento parecen demostrar buenos resultados en término general, con valores de accuracy y kappa superiores a 0.9 y 0.8 respectivamente para los 3 nutrientes. En el caso del hierro, el algoritmo de SVM con kernel Radial es el que mejores resultados ofrece. Este algoritmo también presentó el mejor resultado en la validación cruzada para vitamina B12 y ácido fólico, con XGB y RF como segunda opción, respectivamente. Por contra, los resultados de árboles de decisión no han sido efectivos en ninguno de los entrenamientos realizados.

El siguiente paso realizado fue probar todos estos modelos generados anteriormente sobre el 20% de los datos que habíamos reservado para la validación, y así ver cómo se desempeñan en el mundo real. Hay que recordar que estos datos de validación están desbalanceados, principalmente para vitamina B12 y ácido fólico, por lo que se comprueba generalmente la métrica F1-Score.

Se obtienen resultados moderadamente buenos para la predicción de déficit de hierro, con un valor de F1-score de 0.649 para las predicciones mediante RF. En este caso, al estar los datos menos desbalanceados, nos fijamos en el valor de accuracy y kappa que fue de 0.71 y 0.41 respectivamente. En términos generales, se obtiene una especificidad cercana al 75% mientras que la sensibilidad es baja con valores en torno al 30%.

modelo <chr>	F1 <dbl>	accuracy <dbl>	kappa <dbl>
knn	0.6033691	0.6960094	0.35955839
svmlin	0.6430678	0.7159624	0.41234011
svmradial	0.5528169	0.7018779	0.33028818
svmpoly	0.1176471	0.8288973	0.03725698
tree	0.5094067	0.6021127	0.18440581
rf	0.6491994	0.7171362	0.41852720
rn	0.6061538	0.6995305	0.36558464
xgb	0.5346535	0.7241784	0.34836946

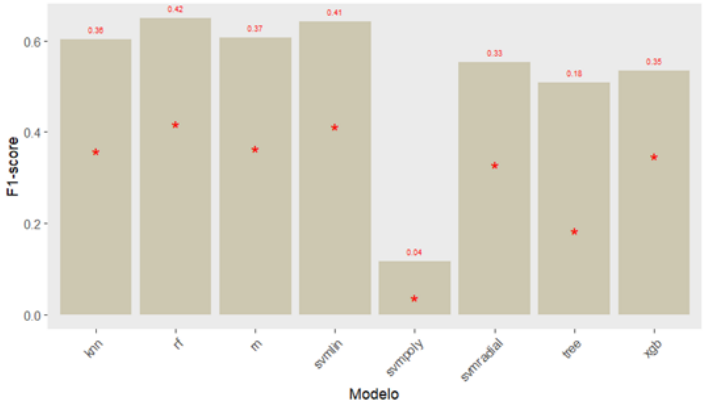


Ilustración 29. Métricas de validación para predicción de hierro

Para la vitamina B12 se puede observar cómo es afectado por el sobreajuste del entrenamiento, con valores de kappa cercanos a 0.

modelo_B12 <chr>	F1_B12 <dbl>	accuracy_B12 <dbl>	kappa_B12 <dbl>
knn	0.3173432	0.5616114	0.08374997
svmlin	0.3257576	0.5781991	0.09811738
svmradial	0.1000000	0.8293839	0.07568752
svmpoly	0.2746781	0.5995261	0.04647432
tree	0.2755556	0.6137441	0.05256054
rf	0.2500000	0.6587678	0.04392700
rn	0.2708333	0.6682464	0.07048458
xgb	0.2251656	0.7227488	0.05641555

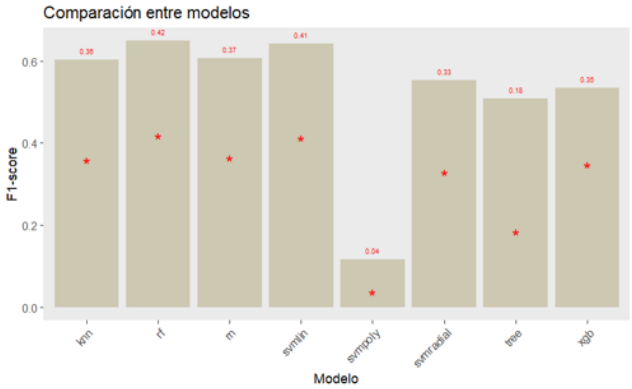


Ilustración 30. Métricas de validación para predicción de vitamina B12

En el caso del ácido fólico se consiguió solucionar parcialmente el problema de sobreajuste. Pese a que se puede observar en la siguiente tabla valores de accuracy de 0.8-0.9, los valores de F1-score únicamente fueron de 0.15, a excepción del algoritmo RF que consiguió aumentarlo hasta 0.28 con un accuracy de 0.91 y kappa de 0.24.

modelo_Fol <chr>	F1_Fol <dbl>	accuracy_Fol <dbl>	kappa_Fol <dbl>
knn	0.15894040	0.6392045	0.06695609
svmlin	0.07142857	0.7045455	-0.02417189
svmradial	0.06896552	0.9232955	0.03493095
svmpoly	0.12500000	0.8409091	0.05084746
tree	0.14925373	0.6761364	0.05820503
rf	0.28571429	0.9147727	0.24050633
rn	0.14117647	0.7926136	0.05944363
xgb	0.17857143	0.8693182	0.11383538

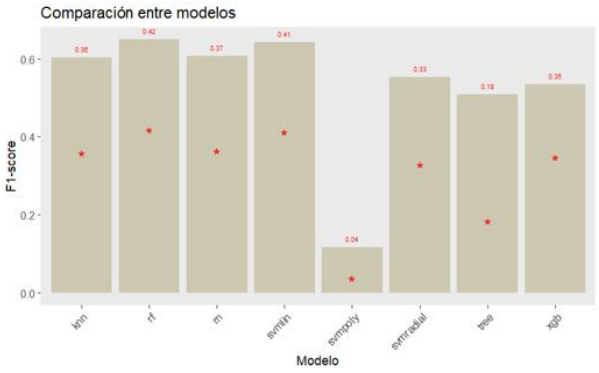


Ilustración 31. Métricas de validación para predicción de ácido fólico

Los métodos de ensemble learning no mejoran por lo general los resultados de validación anteriores. En el caso de hard voting los resultados son claramente inferiores, mientras que tanto con stacking (caretEnsemble y SuperLearner) como con boosting se obtienen algunos valores F1-Score ligeramente superiores, pero muy similares a los comentados anteriormente.

F1-Score	**subset_Fe**	**subset_B12**	**subset_Fol**
caretEnsemble	0.66	0.32	0.24
SuperLearner	0.62	0.29	0.15
adaboost	0.63	0.30	0.22

Tabla 12. F1-Score para métodos de Ensemble Learning

Finalmente, con el empleo del mejor modelo obtenido para la predicción del déficit de hierro (el ensamblado con caretEnsemble) y la modificación de los umbrales de decisión a 0.1, 0.15 y 0.2 se consigue mejorar considerablemente la sensibilidad y especificidad. De un total de 852 observaciones en el dataset de test se obtuvieron los siguientes resultados:

- Umbral 0.1: con un 75.12% de no concluyentes se obtuvo una sensibilidad de 0.99 y una especificidad de 0.86, que permitió clasificar correctamente a 191 pacientes.

- Umbral 0.15: con un 65.38% de no concluyentes se obtuvo una sensibilidad de 0.93 y una especificidad de 0.84, que permitió clasificar correctamente a 257 pacientes.

- Umbral 0.2: con un 54.46 % de no concluyentes se obtuvo una sensibilidad de 0.89 y una especificidad de 0.79, que permitió clasificar correctamente a 318 pacientes.

5. Discusión y conclusiones

El diseño de nuevas herramientas que permitan evaluar con alto grado de fiabilidad la información de datos cuya estructura interna desconocemos, abre un amplio abanico de posibilidades en el campo de la investigación. Durante el desarrollo de este trabajo se han generado, con tal información, distintos modelos y sus consiguientes predicciones con la intención de mejorar el diagnóstico de anemia. Mediante el uso de datos reales de pacientes anémicos se han podido extraer cuáles son aquellos parámetros hematológicos más relevantes en el déficit de precursores eritropoyéticos, y estas variables han sido empleadas posteriormente para clasificar a los pacientes. Se ha hecho hincapié en aquellos parámetros de reciente aparición conocidos como Cell Population Data.

Aunque los resultados no han sido los deseables en términos de objetivos, ya que no se alcanzó en ningún modelo una precisión superior al 80%, se consideran aceptables al cumplir el objetivo de obtener los parámetros de mayor importancia asociados a la enfermedad, así como generar un buen modelo de predicción, para el déficit de hierro, con una precisión cercana al 75%. En términos de planificación los plazos establecidos no fueron los correctos, ya que no se pudo desarrollar una aplicación Shiny que implementara el modelo generado. Este cambio fue motivado por una falta de tiempo al focalizar la fase dos, pensada para generar la aplicación, en encontrar técnicas que pudiesen prevenir el sobreajuste del modelo y mejorar la precisión.

Las variables más importantes obtenidas están en consonancia con lo esperable a la fisiopatología de la enfermedad, lo que demuestra una adecuada selección de variables. Aquellas relacionadas con el tamaño de los hematíes o la concentración de hemoglobina son las más importantes a la hora de predecir los distintos déficits. Como valor añadido este trabajo pretendía encontrar los parámetros de CPD más influyentes en cada tipo de anemia siendo las variables relacionadas con el volumen y conductividad de los neutrófilos y monocitos, como MNCNE, las más destacadas. Estos resultados pueden tener una explicación razonable ya que los nutrientes que estudiamos son también necesarios en la síntesis de precursores de la serie blanca, y su déficit ya se ha comprobado en la literatura puede producir determinados cambios, como la hipersegmentación de neutrófilos. Los parámetros han sido introducidos recientemente y conocer en qué forma influyen puede ser de utilidad para futuros estudios.

Respecto a la evaluación de los resultados de los modelos de aprendizaje automático generados, se han obtenido malos resultados para la predicción de déficit de vitamina B12, donde tras la evaluación las predicciones se veían afectadas por el sobreajuste. Resultados similares se obtenían para la predicción del ácido fólico, aunque en este caso el algoritmo Random Forest parecía escapar de este problema y se obtenían buenas cifras de especificidad. Se considera que hacen falta más datos, principalmente en el caso del ácido fólico en el que se disponían de menos de 100 observaciones con déficit, para intentar mitigar este sobreajuste. De igual forma, tras esto sería necesario comprobar de

forma más exhaustiva otras técnicas de selección de características, así como otras técnicas de remuestreo.

Como aspecto positivo, los resultados obtenidos para predecir el déficit del hierro se aproximan a los esperados, siendo los mejores modelos obtenidos aquellos entrenados mediante Random Forest, eXtreme Gradient Boosting y Support Vector Machine, con posterior ensamblado mediante caretEnsemble. Por el contrario, y como era de esperar debido a la complejidad de los datos, no se obtuvieron buenos resultados con los árboles de decisión ni con kNN. Respecto a las redes neuronales se esperaban mejores resultados, y tras leer bibliografía al respecto se llega a la conclusión de que hubieran sido necesarias técnicas de Deep-learning más potentes y que permitan más modificaciones que el método "nnet" de R. Por último, la aplicación de técnicas de Ensemble Learning, aunque se esperaba aumentaran considerablemente la precisión, únicamente se consiguió (ligeramente) tras realizar stacking por caretEnsemble.

Al aplicar umbrales de decisión a las probabilidades de predicción, se esperaba lograr un aumento de sensibilidad y especificidad que permitiera utilizar el modelo como herramienta de apoyo en el diagnóstico de anemia. Aunque estos puntos de corte excluían a una gran parte de la población a estudio, mediante su uso se pudo clasificar con más certeza pacientes, que si bien eran minoría, seguía siendo un número considerable si se tiene en cuenta la cantidad de pacientes que pasan a diario por un laboratorio clínico. De esta forma, aplicando el punto de corte de <0.1 para "Normal" y >0.9 para "Déficit" se consiguió que en casi un 25% de los pacientes se pudiese predecir el déficit de hierro con una sensibilidad del 99% y una especificidad del 86%.

Valores tan elevados de sensibilidad permitirían detectar, casi sin error (sin falsos negativos), a pacientes con déficit de hierro sin necesidad de la cuantificación del biomarcador. Si estos resultados fuesen aplicados al total de pacientes de los que disponemos datos (4261 observaciones) para el periodo de estudio, del 1 de marzo al 20 de abril (51 días), se habrían pronosticado con un 99% de acierto a 390 pacientes con déficit de hierro, 7.56 pacientes de media al día. Estos valores son aún mayores si tenemos en cuenta que se parte de 25000 observaciones pero muchas tuvieron que ser eliminadas al no cumplir el criterio de inclusión "Disponer de la prueba del marcador específico".

No obstante, es importante comentar que con estos resultados no se pretende sustituir la determinación de hierro, ya que es imprescindible para el diagnóstico definitivo de anemia ferropénica; sino ser un complemento de aviso en aquellos casos que con mayor probabilidad tengan un déficit de hierro, permitiendo un diagnóstico precoz de la enfermedad y así contribuyendo a disminuir su morbi-mortalidad.

Para finalizar y en función de las conclusiones obtenidas, este trabajo presenta algunos puntos de mejora que se podrían ampliar en futuros estudios:

- Ampliación de la base de datos a fin de conseguir más datos de clase minoritaria, que en nuestro caso ha influenciado claramente el entrenamiento del modelo para predecir déficit de vitamina B12 y principalmente, ácido fólico.

- Estudio de otros algoritmos y/o métodos de ensamblaje, así como de selección de predictores.

- Ampliar el estudio a niños o adolescentes, no incluidos en este estudio, o centrarlo a determinados grupos poblacionales, como embarazadas o ancianos.

- Predicción de déficit de otros parámetros bioquímicos, considerando la ferritina como mejor alternativa al hierro para la clasificación de anemia ferropénica.

Por último, se considera que este trabajo presenta un impacto positivo general relacionado con los impactos ético-sociales, de sostenibilidad y diversidad comentados en el punto 1.3

6. Glosario

CARET: Classification And REgression Training

CPD: Cell Population Data

GBM: Gradient Boosting Machines

Hb: Hemoglobina

HCM: Hemoglobina Corpuscular Media

k-NN: k-Nearest Neighbors

LASSO: Least Absolute Shrinkage and Selection Operator

ML: Machine Learning

MNCNE: Mean Neutrophils Conductivity

MNVMO: Mean Monocyte Volume

OMS: Organización Mundial de la Salud

PCA: Análisis de Componentes Principales

RF: Random Forest

RN: Red Neuronal

ROSE: Random Over-Sampling Examples

SMOTE: Synthetic Minority Over-sampling Technique

SVM: Support Vector Machine

VCM: Volumen Corpuscular Medio

XGB: Extreme Gradient Boosting

7. Bibliografía

[1] World Health Organization: WHO. (2019). Anaemia. https://www.who.int/health-topics/anaemia

[2] StatPearls. 2022. Anemia Screening https://www.statpearls.com/ArticleLibrary/viewarticle/17529

[3] An, R., Hasan, M., Man, Y., & Gurkan, U. A. (2019). Integrated Anemia Detection and Hemoglobin Variant Identification Using Point-of-Care Microchip Electrophoresis. Blood, 134(Supplement_1), 378.

[4] Karakochuk, C. D., Hess, S., Moorthy, D., Namaste, S., Parker, M., Rappaport, A. I., Wegmüller, R., & Dary, O. (2019). Measurement and interpretation of hemoglobin concentration in clinical and field settings: a narrative review. Annals of the New York Academy of Sciences, 1450(1), 126-146.

[5] Singh, I., Weston, A., Kundur, A., & Dobie, G. (2017). Haematology Case Studies with Blood Cell Morphology and Pathophysiology. Academic Press.

[6] Green, R., & Wachsmann-Hogiu, S. (2015). Development, History, and Future of Automated Cell Counters. Clinics in Laboratory Medicine, 35(1), 1-10.

[7] Jean, A., Boutet, C., Lenormand, B., Callat, M., Buchonnet, G., Barbay, V., Basuyau, J., & Vasse, M. (2010). The new haematology analyzer DxH 800: an evaluation of the analytical performances and leucocyte flags, comparison with the LH 755. International Journal of Laboratory Hematology, 33(2), 138-145.

[8] Haider, Z., Ujjan, I. D., & Shamsi, T. (2020). Cell Population Data–Driven Acute Promyelocytic Leukemia Flagging Through Artificial Neural Network Predictive Modeling. Translational Oncology, 13(1), 11-16.

[9] Boyero, F. C., Rojas, S., Segura, G. P., Jimenez, A. G., Jimenez, M., Gomez-Tarragona, G. C., Ureña, B. B., & Martinez-Lopez, J. (2020). A Machine Learning Approach for the Differential Diagnosis between Sars-COV19 Infection and Influenza Viruses with Hematological Morphologic DATA (CELL MORPHOLOGIC DATA). Blood, 136 (Supplement 1), 43.

[10] Hausfater, P., Boter, N. R., Indiano, C. M., De Abreu, M. C., Marin, A. M., Pernet, J., Quesada, D., Castro, I., Careaga, D. B., Arock, M., Tejidor, L., & Velly, L. (2021). Monocyte distribution width (MDW) performance as an early sepsis indicator in the emergency department: comparison with CRP and procalcitonin in a multicenter international European prospective study. Critical Care, 25(1).

[11] Ko, S. Q., Quah, P., & Lahiri, M. (2019). The cost of repetitive laboratory testing for chronic disease. Internal medicine journal, 49(9), 1168–1170.

[12] Salinas, M., Lopez-Garrígos, M., Rodriguez-Borja, E., Blasco, Á., & Carratalá, A. (2016). Laboratory Test requesting Appropriateness and Patient Safety. Walter de Gruyter GmbH & Co KG.

[13] Ambayya, A., Sahibon, S., Yang, T. W., Zhang, Q. Y., Hassan, R., & Sathar, J. (2021). A Novel Algorithm Using Cell Population Data (VCS Parameters) as a Screening Discriminant between Alpha and Beta Thalassemia Traits. Diagnostics (Basel, Switzerland), 11(11), 2163.

[14] Rahim, F., Kazemnejad, A., Jahangiri, M., Malehi, A. S., & Gohari, K. (2021). Diagnostic performance of classification trees and hematological functions in hematologic disorders: an application of multidimensional scaling and cluster analysis. BMC Medical Informatics and Decision Making, 21(1).

[15] Saputra, D. C. E., Sunat, K., & Ratnaningsih, T. (2023). A New Artificial Intelligence Approach Using Extreme Learning Machine as the Potentially Effective Model to Predict and Analyze the Diagnosis of Anemia. Healthcare (Basel, Switzerland), 11(5), 697.

[16] Vohra, R., Hussain, A., Dudyala, A. K., Pahareeya, J., & Khan, W. (2022). Multi-class classification algorithms for the diagnosis of anemia in an outpatient clinical setting. PLOS ONE, 17(7), e0269685.

[17] Lantz, B. (2015). Machine Learning with R: Expert techniques for predictive modeling to solve all your data analysis problems. Packt Publishing Ltd.

[18] Kuhn, M. (2019, 27 marzo). The caret Package. https://topepo.github.io/caret/

[19] Amunategui, M. (s. f.). Data Exploration & Machine Learning, Hands-on. https://amunategui.github.io/dummyVar-Walkthrough/

[20] RPubs - Machine Learning con R y caret. (s. f.). https://rpubs.com/Joaquin_AR/383283

[21] McNeish, D. (2015). Using Lasso for Predictor Selection and to Assuage Overfitting: A Method Long Overlooked in Behavioral Sciences. Multivariate Behavioral Research, 50(5), 471-484.

[22] RPubs - Análisis de componentes principales (PCA). (s. f.). https://rpubs.com/Cristina_Gil/PCA

[23] Reyes, S. A. (s. f.-a). Chapter 1 Preface | A Machine Learning Compilation. https://f0nzie.github.io/machine_learning_compilation/index.html

[24] Blagus, R., & Lusa, L. (2013). SMOTE for high-dimensional class-imbalanced data. BMC Bioinformatics, 14(1).

[25] Ramos-Pérez, I., Arnaiz-González, Á., Rodríguez, J. M., & García-Osorio, C. (2022). When is resampling beneficial for feature selection with imbalanced wide data? Expert Systems with Applications, 188, 116015.

[26] Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. Annals of Translational Medicine, 4(11), 218.

[27] GeeksforGeeks. (2022). Support Vector Machine Classifier Implementation in R with Caret package. GeeksforGeeks. https://www.geeksforgeeks.org/support-vector-machine-classifier-implementation-in-r-with-caret-package/

[28] Bazazeh, D., & Shubair, R. M. (2016). Comparative study of machine learning algorithms for breast cancer detection and diagnosis.

[29] Machine Learning | Google for Developers. (s. f.). Google for Developers. https://developers.google.com/machine-learning/decision-forests/random-forests?hl=es-419

[30] Probst, Philipp, Marvin N Wright, and Anne-Laure Boulesteix. 2019. “Hyperparameters and Tuning Strategies for Random Forest.” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. Wiley Online Library, e1301.

[31] Rubén Fernández Casal8.3 Implementación en R | Aprendizaje Estadístico. https://rubenfcasal.github.io/aprendizaje_estadistico/implementación-en-r-2.html

[32] Abualdenien, J., & Borrmann, A. (2022). Ensemble-learning approach for the classification of Levels Of Geometry (LOG) of building elements. Advanced Engineering Informatics, 51, 101497.

[33] Wegier, W., & Ksieniewicz, P. (2020). Application of Imbalanced Data Classification Quality Metrics as Weighting Methods of the Ensemble Data Stream Classification Algorithms. Entropy, 22(8), 849.

[34] Widmann, M. (2021, 31 octubre). Cohen’s Kappa: What It Is, When to Use It, and How to Avoid Its Pitfalls. The New Stack. https://thenewstack.io/cohens-kappa-what-it-is-when-to-use-it-and-how-to-avoid-its-pitfalls/

[35] Dalpiaz, D. (2020, 28 octubre). Chapter 27 Ensemble Methods | R for Statistical Learning. https://daviddalpiaz.github.io/r4sl/ensemble-methods.html

[36] Nwanganga, F., & Chapple, M. (2020). Practical Machine Learning in R. John Wiley & Sons.

[37] Chris Kennedy, University of California, Berkeley. (2017, 16 marzo). Guide to SuperLearner. https://cran.r-project.org/web/packages/SuperLearner/vignettes/Guide-to-SuperLearner.html

[38] Gremmell, D. (2018). Ensemble Learning in R with SuperLearner. https://www.datacamp.com/tutorial/ensemble-r-machine-learning

[39] Schapire, R. E. (2003). The Boosting Approach to Machine Learning: An Overview. En Springer eBooks (pp. 149-171).

[40] Wong, J., Manderson, T., Abrahamowicz, M., Buckeridge, D. L., & Tamblyn, R. (2019). Can Hyperparameter Tuning Improve the Performance of a Super Learner? Epidemiology, 30(4), 521-531.

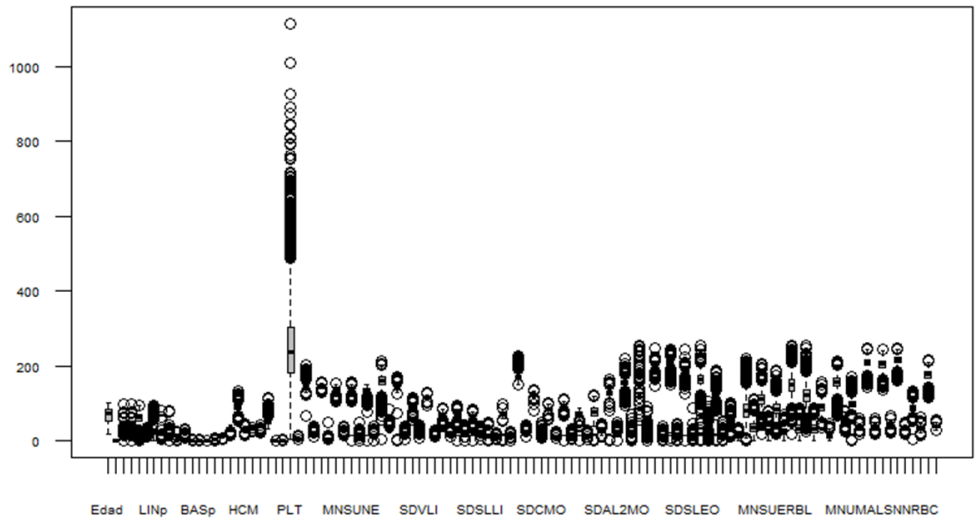
8. Anexo

	freqRatio	percentUnique	zeroVar	nzv
MON	1.038564	0.77446609	FALSE	FALSE
EOSp	1.025424	3.47336306	FALSE	FALSE
EOS	1.372180	0.56324806	FALSE	FALSE
BASp	1.110664	0.77446609	FALSE	FALSE
BAS	1.697242	0.09387468	FALSE	FALSE
ERIT	1.037152	1.45505750	FALSE	FALSE
HGB	1.082524	1.80708754	FALSE	FALSE
HCT	1.076923	5.09270124	FALSE	FALSE
VCM	1.066667	12.03942736	FALSE	FALSE
HCM	1.057143	5.32738794	FALSE	FALSE

Anexo 1. Output de la función nearZeroVar de caret

Fe	[1] "ULE"	"MNSULI"	"SDSUERBL"	"MNMALSNNRBC"	"MNUMALSNNRBC"
	[6] "SDUMALSNNRBC"	"MNLALSNNRBC"	"NEUp"	"EOSp"	"HGB"
	[11] "VCM"	"MNSMNE"	"SDSMNE"	"SDSMLI"	"MNSMLI"
	[16] "SDSMMO"	"MNSMERBL"	"SDSMERBL"	"MNVNNRBC"	
B12	[1] "HCM"	"MNSULI"	"SDSUERBL"	"MNMALSNNRBC"	"MNUMALSNNRBC"
	[6] "SDUMALSNNRBC"	"MNLALSNNRBC"	"LEU"	"NEUp"	"EOSp"
	[11] "HGB"	"MNSMNE"	"SDSMNE"	"SDSMLI"	"MNSMLI"
	[16] "SDSMMO"	"MNSMEO"	"MNSMERBL"	"SDSMERBL"	"MNVNNRBC"
	[21] "SDMALSNNRBC"				
Fol	[1] "MNSULI"	"SDSLEO"	"SDSUERBL"	"MNMALSNNRBC"	"MNUMALSNNRBC"
	[6] "SDUMALSNNRBC"	"MNLALSNNRBC"	"LEU"	"NEUp"	"EOSp"
	[11] "HGB"	"VCM"	"ERBL"	"MNSMNE"	"SDSMNE"
	[16] "SDSLNE"	"SDSMLI"	"MNSMLI"	"MNSMMO"	"SDSMMO"
	[21] "MNSMEO"	"MNSMERBL"	"SDSMERBL"	"MNVNNRBC"	"SDMALSNNRBC"

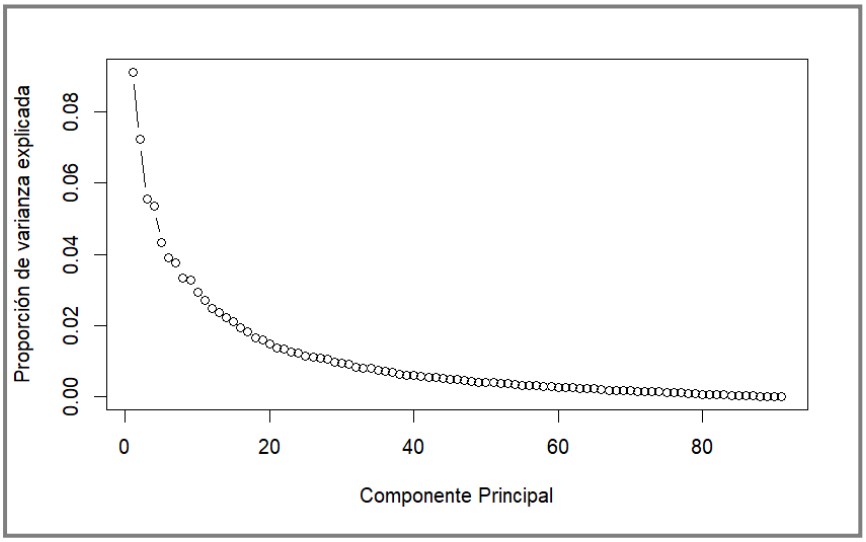
Anexo 2. Variables eliminadas tras análisis de correlaciones entre variables



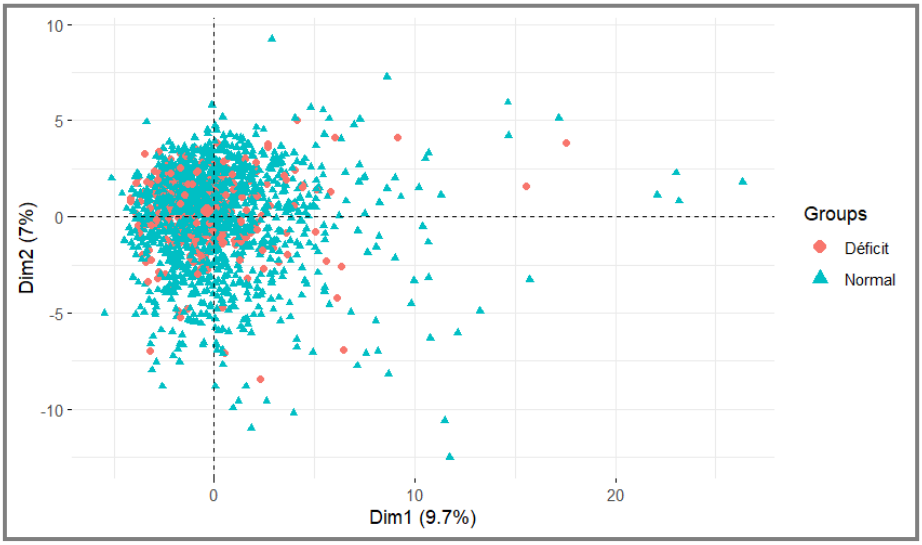
Anexo 3. Distribución de las variables para el dataset del hierro: motivo por el cuál se escalan los datos. Existen diversas variables (principalmente las plaquetas) que presentan valores muy elevados en comparación al resto, lo que podría tener impacto al sesgar las características más importantes

	s0
(Intercept)	0.7573768645
Edad	-0.0786279866
Sexo	0.2190120922
LEU	.
NEU	0.1027192265
LINp	.
LIN	.
MONp	-0.1823353833
MON	0.1286993354
EOS	-0.0954288688
BASp	0.0006866927

Anexo 4. Outpout de coeficientes de aportes de las variables mediante regularización Lasso



Anexo 5. Disminución de la proporción de varianza explicada al disminuir el componente principal



Anexo 6. Análisis de componentes principales para la Vitamina B12 (comparación componente 1 y componente 2)