



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

INSTITUTO DE INVESTIGACIONES EN  
MATEMÁTICAS APLICADAS Y EN SISTEMAS

COMPARACIÓN DE MODELOS PARA LA PREDICCIÓN  
DE GRASA INTRAHEPÁTICA MEDIDA POR  
RESONANCIA MAGNÉTICA CON ESPECTROSCOPIA

T E S I N A

QUE PARA OBTENER EL GRADO DE:

ESPECIALISTA EN ESTADÍSTICA APLICADA

P R E S E N T A :

OMAR YAXMEHEN BELLO CHAVOLLA

TUTORES

MTRO. JAVIER SANTIBAÑEZ CORTÉS  
DR. CARLOS A. AGUILAR-SALINAS



CIUDAD UNIVERSITARIA, Cd. Mx. A 1 DE MARZO DE 2020



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*Algunas dedicatorias...*

*A mis padres, por su pasión incansable.*

*A mi novia Jessica por su apoyo y cariño.  
Por los horarios extremos y maratónicos juntos.*

*Al Prof. Javier Santibañez, por su paciencia infinita.*

*Al Dr. Carlos Aguilar, por su ejemplo y apoyo.*

*A mis profesores de la EEA, por abrir mis ojos a un nuevo mundo.*

# Agradecimientos

- Me gustaría agradecer a todo el personal involucrado en la Especialización en Estadística Aplicada. Considero que este programa revolucionó mi visión científica de la investigación médica y me fortaleció como médico, investigador y persona. El trabajo incansable de los profesionales de la estadística y la probabilidad involucrados así como el personal de apoyo administrativo ayudó a hacer accesible un campo que a veces se considera incompatible con el ámbito médico, pero que he llegado a considerar como una herramienta vital para entender mejor nuestro quehacer profesional.
- Agradezco al profesor Javier Santibáñez, quien me apoyó con esta modesta pero ambiciosa pregunta de investigación que bajo otra tutoría, no habría podido llevar a buen puerto. Su apoyo me ha ayudado a sentar de forma más práctica muchos de los conceptos que pude aprender durante mi formación como especialista.
- Agradezco el apoyo y comprensión de mi tutor de doctorado el Dr. Carlos Aguilar Salinas, quien me ayudó a empatar horarios y vencer condiciones que pudieron resultar adversas para mi formación simultánea como especialista. Su apoyo resultó en un mejor crecimiento profesional y un sin fin de aplicaciones del conocimiento que he adquirido.
- Agradezco a la Universidad Nacional Autónoma de México por una vez más abrirme las puertas para aprender y cambiar, otra vez más, mi forma de ver y entender el mundo. Mi compromiso sigue firme en aplicar todo el conocimiento adquirido, compartirlo y tratar de contribuir al desarrollo de conocimiento científico de calidad.

*Por mi raza hablará el espíritu.*

# Resumen del trabajo

**Introducción:** La Enfermedad por Hígado Graso no alcohólico (EHGNA) es una de las principales causas de enfermedad hepática crónica en el mundo. Debido a su curso clínico, existe un elevado interés en identificar sujetos con esteatosis sin esteatohepatitis en quienes las intervenciones médicas puedan reducir la carga de morbilidad. El diagnóstico de EHGNA es complicado y se ve limitado por el elevado costo o invasividad de las metodologías más precisas para su identificación. El creciente uso de metodologías de inteligencia artificial ofrece un campo prometedor para mejorar la medicina de precisión y disminuir la complejidad del diagnóstico de EHGNA.

**Metodología:** Se desarrollaron tres modelos en una población de pacientes ( $n = 129$ ) para la predicción de grasa intra-hepática basada en parámetros clínicos en ayuno, incluyendo mediciones séricas de glucosa, perfil de lípidos, pruebas de función hepática y antropometría comparadas con resonancia magnética por espectroscopía mediante tres metodologías distintas de aprendizaje de máquina incluyendo regresión lineal múltiple (RLM), *gradient boosting machine* (GBM) y redes neuronales auto-normalizables (SNNN).

**Resultados:** Los 3 modelos desarrollados tuvieron adecuado desempeño. El modelo RLM tuvo los errores de predicción (MAE, MSE, RMSE) más altos comparados con los modelos GBM y SNNN; sin embargo, su desempeño diagnóstico fue mejor al tener la tasa más baja de falsos negativos. El modelo SNNN fue el modelo con menor sesgo, menor error de predicción y mayor coeficiente de correlación y determinación; sin embargo, fue un estimador de alta varianza y con algunos falsos negativos en su desempeño diagnóstico. El modelo GBM tuvo un desempeño intermedio, fue superado por el modelo RLM en su desempeño diagnóstico y tuvo comparativamente mayor error de predicción comparado con SNNN.

**Conclusiones:** Debido a la facilidad de su implementación y por ser un modelo más parsimonioso el modelo RLM podría ser el modelo con mejor aplicación cotidiana con potenciales aplicaciones para tamizaje preventivo. Mientras tanto, el modelo SNNN podría ser de mayor utilidad cuando el objetivo de la evaluación sea tener valores más precisos para la estimación de grasa intrahepática. En bases con datos limitados el modelo RLM tiene un desempeño adecuado considerando la simplicidad de su implementación comparada con SNNN y GBM. El uso de aprendizaje de máquina debe considerarse cuando el número de observaciones sea grande.

# Índice general

<b>Agradecimientos</b>	<b>II</b>
<b>Resumen del trabajo</b>	<b>III</b>
<b>1. Introducción</b>	<b>1</b>
1. Enfermedad por hígado graso no alcohólico . . . . .	1
1.1. Epidemiología . . . . .	2
1.2. Determinantes de acumulación de grasa intrahepática . . . . .	2
1.3. Estimación clínica de grasa intrahepática por imagen . . . . .	3
1.4. Modelos matemáticos existentes para la estimación de grasa intrahepática . . . . .	4
1.5. Uso de modelos de aprendizaje de máquina e inteligencia artificial en predicción metabólica . . . . .	5
2. Planteamiento del problema . . . . .	6
3. Objetivos . . . . .	7
4. Hipótesis . . . . .	7
<b>2. Metodología</b>	<b>8</b>
1. Origen de los datos para análisis . . . . .	8
1.1. Evaluaciones bioquímicas . . . . .	9
1.2. Antropometría y cuantificación de grasa intrahepática . . . . .	9
2. Descripción de los modelos a implementar . . . . .	9
2.1. Modelo de regresión lineal múltiple . . . . .	10
2.1.1. Implementación del modelo RLM en los datos . . . . .	13
2.2. Modelo Gradient Boosting Machine . . . . .	15
2.2.1. Funcionamiento general del algoritmo GBM . . . . .	17
2.2.2. Implementación del modelo GBM en los datos . . . . .	18
2.2.3. Optimización de hiperparámetros del modelo GBM . . . . .	19
2.3. Modelo de Redes neuronales auto-normalizables . . . . .	20
2.3.1. Principios de funcionamiento de las redes neuronales . . . . .	21
2.3.2. Redes neuronales auto-normalizables . . . . .	23
2.3.3. Implementación del modelo SNNN en los datos . . . . .	25
2.4. Comparación de los modelos RLM, GBM y SNNN . . . . .	27
2.4.1. Análisis de Bland-Altman . . . . .	28
2.4.2. Curvas ROC . . . . .	29

2.4.3.	Matriz de confusión . . . . .	31
<b>3.</b>	<b>Resultados</b>	<b>34</b>
1.	Análisis descriptivos . . . . .	34
2.	Modelo de regresión lineal múltiple . . . . .	38
2.1.	Selección del modelo en la base de entrenamiento . . . . .	38
2.2.	Validación de supuestos . . . . .	41
2.2.1.	Supuesto de linealidad . . . . .	41
2.2.2.	Supuesto de homoscedasticidad de varianza e independencia de los errores . . . . .	42
2.2.3.	Matriz de diseño de rango completo . . . . .	42
2.2.4.	Normalidad conjunta de los errores . . . . .	43
2.3.	Análisis final del modelo . . . . .	44
3.	Modelo <i>Gradient Boosting Machine</i> . . . . .	45
4.	Modelo de Redes neuronales auto-normalizables . . . . .	49
4.1.	Cálculo de la métrica de error . . . . .	49
5.	Comparación de modelos . . . . .	51
5.1.	Análisis de Bland-Altman de los modelos . . . . .	52
5.2.	Evaluación de los modelos para predicción de EHGNA . . . . .	54
<b>4.</b>	<b>Discusión y Conclusiones</b>	<b>57</b>

# Capítulo 1

## Introducción

### 1. Enfermedad por hígado graso no alcohólico

La enfermedad por hígado graso no alcohólico (EHGNA) es actualmente la principal causa de enfermedad hepática crónica. La EHGNA se caracteriza por la acumulación de triacilgliceroles o triglicéridos en los hepatocitos, las células componentes del hígado, atribuible a una causa diferente al consumo crónico de alcohol, medicamentos hepatotóxicos o enfermedades ligadas a la acumulación hepática de grasa [1]. La presentación clínica de la EHGNA es heterogénea y, en la mayoría de los casos, asintomática; esto dificulta su diagnóstico oportuno y reduce la efectividad de los tratamientos encaminados a reducir sus complicaciones y la morbilidad cardiovascular asociada[1, 2].

Desde el punto de vista médico, la acumulación hepática de grasa se denomina esteatosis hepática. A nivel histológico, la presencia de esteatosis hepatocelular mayor a 5% es la característica definitiva para determinar el diagnóstico de EHGNA[3]. El curso clínico de la EHGNA es heterogéneo, pudiendo progresar de una simple esteatosis hepática a esteatohepatitis, caracterizada por inflamación, balonización y daño hepatocelular con o sin fibrosis e incluso progresar a un carcinoma hepatocelular [1, 3]. Debido al curso clínico y a su asociación con la presencia de comorbilidades metabólicas, el estudio de la EHGNA y la esteatosis hepática ha ganado interés en años recientes, sobre todo con un enfoque preventivo que permita detener su progresión



clínica y reducir la carga de morbilidad asociada a su diagnóstico [2, 3].

### 1.1. Epidemiología

El aumento en la prevalencia de EHGNA ha ocurrido en paralelo con el aumento en la prevalencia de obesidad, diabetes mellitus tipo 2 y enfermedad cardiovascular [1]. La prevalencia estimada de EHGNA es heterogénea debido a la diversidad étnica de las poblaciones evaluadas y la heterogeneidad de las metodologías utilizadas para su diagnóstico; sin embargo, se estima una prevalencia entre 17 % y 46 % a nivel mundial [2, 3]. Tan sólo en Estados Unidos, se estima una prevalencia de EHGNA entre 30 % y 40 % en la población adulta de entre los cuales aproximadamente el 5 % padecen esteatohepatitis y el 80 % esteatosis hepática no complicada [3, 4]. Los pacientes con EHGNA y esteatohepatitis tienen un mayor riesgo de enfermedad cardiovascular y de mortalidad por enfermedad cardiovascular prematura, cáncer hepatocelular y complicaciones de enfermedad hepática [4, 5]. A pesar de que sólo 10 % y 30 % de los pacientes con esteatohepatitis progresan a cirrosis, su alta prevalencia poblacional implica un elevado costo económico y una elevada morbi-mortalidad [5].

### 1.2. Determinantes de acumulación de grasa intrahepática

Uno de los principales determinantes de EHGNA está ligado a la presencia de estilos de vida poco saludables y la ocurrencia simultánea de comorbilidades metabólicas, incluyendo síndrome metabólico, obesidad abdominal y diabetes mellitus tipo 2 [4]. La baja actividad física además de las dietas con elevada densidad calórica asociadas a la obesidad abdominal conllevan a una mayor acumulación intrahepática de triglicéridos. Algunos factores de riesgo han sido evaluados en asociación con la acumulación intrahepática de grasa, incluyendo componentes del síndrome metabólico (hiperglucemia, hipertrigliceridemia, hipoalfalipoproteinemia, obesidad abdominal e hipertensión), diabetes mellitus tipo 2, dislipidemia, sexo masculino y edad [1, 2, 5]. Además se han observado asociaciones con componentes antropométricos y bioquímicos, como la circunferencia abdominal, el índice de masa corporal, elevación de

triglicéridos séricos, descenso en colesterol de alta densidad (HDL-C) así como alteraciones en enzimas hepáticas, particularmente aspartato aminotransferasa (AST), alanina aminotransferasa (ALT) y la proporción AST/ALT [1, 4]. Sin embargo, las alteraciones en estudios de laboratorio son directamente dependientes del grado de daño hepático, por lo que pueden manifestarse sin alteraciones serológicas notorias. Un componente significativo que aumenta el riesgo de EHGNA es la resistencia a la acción periférica de la insulina, la hormona encargada de mantener la homeostasis del metabolismo de glucosa y lípidos en el organismo[6]. La relación entre la acumulación intrahepática de triglicéridos y la sensibilidad a la insulina es bidireccional; se ha demostrado afectación en la sensibilidad de la insulina a nivel hepático con acumulación de grasa intrahepática de aproximadamente 1.5 %, con similar afectación en la sensibilidad del músculo estriado a la insulina. Las alteraciones metabólicas caracterizadas por hipertrigliceridemia y HDL-C bajo ocurren cuando el contenido de grasa intrahepática es de al menos 6 % [7]. Se ha identificado además una correlación entre niveles más altos de insulina, péptido C y el contenido intrahepático de triglicéridos [6, 7]. El *Metabolic Score for Insulin Resistance* (METS-IR) es una alternativa para la estimación de resistencia a la insulina que prescinde del uso de mediciones de insulina en ayuno y en estudios previos se identificó como un predictor del contenido de grasa intrahepática, por lo que es un candidato importante para la cuantificación de la resistencia a la insulina como alternativa al *Homeostasis Model Assessment* (HOMA-IR), el cual requiere de mediciones de insulina en ayuno para su estimación [8].

### 1.3. Estimación clínica de grasa intrahepática por imagen

El tamizaje para EHGNA está basado en la identificación de factores de riesgo para la acumulación intrahepática de triglicéridos. En individuos con obesidad, componentes del síndrome metabólico o diabetes mellitus tipo 2, se recomienda la evaluación para EHGNA por determinación de enzimas hepáticas o métodos de imagen [2]. Los métodos de imagen para diagnóstico de EHGNA incluyen al ultrasonido abdominal, la elastografía transitoria ultrasonográfica y la resonancia magnética nuclear con espectroscopía (MRI-S).

- **Ultrasonido abdominal:** El ultrasonido abdominal es una técnica no invasiva altamente sensible para la identificación de esteatosis hepática, es de bajo costo y no incurre en riesgos para el paciente. Sin embargo, es una metodología con alta variabilidad dependiente de operador y con baja sensibilidad para el diagnóstico de fibrosis, además de no ofrecer una medición cuantitativa del contenido de grasa intrahepática [2].
- **Elastografía transitoria:** La elastografía transitoria o *FibroScan*, es una metodología que permite la evaluación simultánea de esteatosis hepática y fibrosis. Es una herramienta no invasiva de costo moderado que sin embargo no tiene la precisión necesaria para la estimación del contenido de grasa intrahepática comparada con la evaluación histológica por biopsia, el patrón de oro para la cuantificación de grasa intrahepática [2].
- **Resonancia magnética con espectroscopía:** La MRI-S es una metodología que cuantifica grasa intrahepática al estimar la fracción protónica de densidad grasa en el tejido hepático. Se considera el método de imagen más preciso para la cuantificación de contenido hepático de grasa con desempeño similar a la evaluación histológica por biopsia. Sin embargo, la dificultad técnica de su implementación, además de su baja disponibilidad en México y su elevado costo ha restringido su uso a instancias de investigación [2, 9].

#### 1.4. Modelos matemáticos existentes para la estimación de grasa intrahepática

La complejidad de establecer el diagnóstico de EHGNA o esteatosis hepática mediante marcadores serológicos y las limitantes en precisión y accesibilidad de los métodos de imagen ha generado la necesidad de estimar la grasa intrahepática mediante modelos matemáticos que incorporen mediciones de bajo costo y permitan su implementación en un primer nivel de atención médica o en estudios epidemiológicos de gran escala [2, 3]. Hasta el momento, existen tres índices validados en diferentes poblaciones para estimar el contenido de grasa intrahepática, ninguno de los cuales

ha sido validado con los patrones de oro de biopsia o MRI-S para su uso en población mexicana [2]. Debido a la falta de disponibilidad de información sobre el modelaje de *SteatoTest* por su naturaleza comercial, sólo se discutirán los dos más relevantes:

- **Fatty Liver Index (FLI):** El FLI es un modelo obtenido a partir de regresión logística utilizando el diagnóstico de esteatosis hepática por MRI-S  $> 5.5\%$ , que estima la probabilidad de EHGNA mediante IMC, circunferencia abdominal, triglicéridos y niveles de la enzima  $\gamma$ -glutamyl transpeptidasa (GGT). La principal limitante de su uso es la contribución significativa del perímetro abdominal en el modelo; debido a que la circunferencia de cintura varía de forma importante según la etnicidad, sexo y edad de los participantes es necesario su validación en los grupos étnicos en los que se utilizará, incluyendo a la población Mexicana [10].
- **NAFLD score:** El *NAFLD score* fue desarrollado por un grupo de investigadores finlandeses mediante regresión lineal utilizando variables dicotómicas (presencia o ausencia de síndrome metabólico y/o diabetes mellitus tipo 2) y continuas (insulina de ayuno, AST, ALST y la proporción AST/ALT). El modelo ha sido validado en diversas poblaciones para la predicción de grasa intrahepática. Una limitante para su uso es la necesidad de la medición de insulina, la cual ofrece una elevada variabilidad dependiendo del inmunoensayo utilizado [11].

## 1.5. Uso de modelos de aprendizaje de máquina e inteligencia artificial en predicción metabólica

La reciente introducción de metodologías de *machine learning* o aprendizaje de máquina y diferentes modelos iterativos dentro del campo de la inteligencia artificial ha modificado el panorama de la práctica médica moderna. El uso de algoritmos de inteligencia artificial para diagnóstico automatizado en el contexto de atención primaria en ausencia de especialistas ha demostrado efectividad como una herramienta de apoyo para la toma de decisiones clínicas [12]. Los avances más trascendentales se han

dado con algoritmos de reconocimiento y clasificación de imágenes y texto provenientes de expedientes clínicos electrónicos, donde la inteligencia artificial ha mejorado algoritmos diagnósticos e incluso el seguimiento y atención de pacientes a largo plazo [13]. A pesar de que el principal reto de estas nuevas tecnologías es generalizar su implementación, evidencia reciente ofrece pruebas irrefutables de su beneficio clínico. El uso de la inteligencia artificial en el campo de la endocrinología y metabolismo ha tenido un desarrollo más incipiente. Las aplicaciones del aprendizaje de máquina en la predicción de interacciones genómicas, metabólicas y ambientales ha devenido en el desarrollo de modelos diagnósticos a través de la medicina computacional ha permitido la personalización de la medicina enfocada en metabolismo, con la identificación de patrones de enfermedad específicos para los pacientes y para aumentar la reproducibilidad de metodologías de análisis poblacional [14]. La generación de modelos complejos para entender métodos de tamizaje, respuestas fisiológicas, evolución clínica, respuesta a tratamiento y pronóstico son algunas de las implicaciones de la inteligencia artificial que podrían mejorar el diagnóstico y manejo de la obesidad, diabetes, síndrome metabólico y otras endocrinopatías así como sus complicaciones en poblaciones susceptibles como los mexicanos.

## 2. Planteamiento del problema

La EHGNA es una de las principales causas de enfermedad hepática crónica en el mundo. Debido a su curso clínico existe un elevado interés en identificar sujetos con esteatosis sin esteatohepatitis en quienes las intervenciones puedan reducir la carga de morbilidad. El diagnóstico de EHGNA es complicado y se ve limitado por el elevado costo o invasividad de las metodologías más precisas para su identificación. Los estimadores subrogados son una alternativa atractiva para estimación de grasa intrahepática; sin embargo, los estimadores existentes han sido validados en poblaciones diferentes a la Mexicana y ofrecen algunas limitaciones técnicas que sugieren que su aplicabilidad en nuestra población podría ser limitada. El desarrollo de nuevas metodologías que incluyan algoritmos de inteligencia artificial podría ofrecer estimadores

más precisos para nuestra población.

### 3. Objetivos

- **Objetivo principal:** Desarrollo de un modelo para la predicción de grasa intrahepática basada en parámetros clínicos en ayuno, incluyendo mediciones séricas de glucosa, perfil de lípidos, pruebas de función hepática y antropometría
  - **Objetivo específico 1:** Identificar los principales predictores de acumulación de grasa intrahepática en población mexicana.
  - **Objetivo específico 2:** Comparación de tres abordajes estadísticos diferentes para la estimación de grasa intrahepática medida, incluyendo regresión lineal múltiple, *gradient boosting machine* y redes neuronales densas.
  - **Objetivo específico 3:** Evaluar la precisión y variabilidad del modelo predictivo para grasa intrahepática para seleccionar el modelo con el mejor desempeño en población mexicana.

### 4. Hipótesis

La resistencia a la insulina es un epifenómeno relevante involucrado en la acumulación de grasa intrahepática. Basado en las consideraciones planteadas, el modelo predictivo estará compuesto de un estimador de resistencia a la insulina (METS-IR), un parámetro de estimación de daño hepático por acumulación de grasa (AST, ALT, AST/ALT) y marcadores de antropométricos y/o bioquímicos de alteraciones metabólicas.

# Capítulo 2

## Metodología

### 1. Origen de los datos para análisis

La población de origen de los datos proviene de una muestra no probabilística por conveniencia de individuos sanos y con diabetes mellitus tipo 2 reclutados de la Unidad de Investigación de Enfermedades Metabólicas, un centro de investigación clínica con sede en el Instituto Nacional de Ciencias Médicas y Nutrición Salvador Zubirán. Para la muestra analizada, se incluyeron hombres y mujeres mestizos mexicanos de 20 a 79 años de edad, con un IMC entre 18 y  $34.9 \text{ kg/m}^2$ . Los individuos con diabetes tipo 2 con una concentración de hemoglobina glucosilada (HbA1c)  $<8\%$  y sin tratamiento con insulina fueron elegibles para el estudio. Ningún sujeto fumaba tabaco, tenía enfermedad cardiovascular, complicaciones de la diabetes o una infección aguda. Sujetos con más de un 3% de pérdida de peso en los últimos tres meses, tomando medicamentos o con afecciones que podrían interferir con la medición de grasa intrahepática por resonancia magnética fueron excluidos. Los sujetos proporcionaron un consentimiento informado por escrito antes de participar en este estudio, que fue aprobado por el Comité de Ética en Investigación del Instituto Nacional de Ciencias Médicas y Nutrición Salvador Zubirán (INCMNSZ). Todos los procedimientos se ajustaron a las normas éticas de la declaración de Helsinki.

### 1.1. Evaluaciones bioquímicas

Con los sujetos en estado de ayuno entre 10 y 12 horas se realizó una toma de muestra sanguínea del antebrazo para la cuantificación de variables bioquímicas. La concentración plasmática de glucosa se midió con un analizador de glucosa automatizado (Yellow Springs Instruments Co.) La concentración sérica de insulina se midió utilizando un inmunoensayo quimioluminiscente (Beckman Coulter Access 2). Las concentraciones de lípidos (colesterol, triglicéridos y colesterol HDL) y enzimas hepáticas se midieron mediante ensayos colorimétricos (Unicel DxC 600 Synchron Clinical System Beckman Coulter).

### 1.2. Antropometría y cuantificación de grasa intrahepática

En estado de ayuno, se realizó cuantificación de talla y peso con una báscula y estadímetro calibradas con precisión de 0.1g y 0.01cm, respectivamente. El IMC se calculó como el peso dividido entre el cuadrado de la talla en metros. El contenido de triglicéridos intrahepáticos se determinó mediante espectroscopia MRI (Philips Achieva 3 Teslas) durante una secuencia en la que también se cuantificó grasa subcutánea e intravisceral.

## 2. Descripción de los modelos a implementar

Para la estimación de grasa intra-hepática se desarrollarán tres metodologías diferentes de estimación:

- Regresión lineal múltiple.
- Gradient Boosting Machine.
- Redes neuronales auto-normalizables.

Le presente sección describirá la teoría detrás de cada uno de los modelos así como las particularidades de su implementación y ajuste.



## 2.1. Modelo de regresión lineal múltiple

El modelo de regresión lineal múltiple (RLM) se basa en el supuesto de modelar una variable respuesta  $Y$  con  $p$  predictores o variables explicativas  $X_1, X_2, \dots, X_p$  [15]. El modelo RLM busca formular la relación entre  $Y$  y  $X_1, X_2, \dots, X_p$  como un modelo lineal de la forma:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \epsilon, \quad (2.1)$$

donde  $\beta_0, \beta_1, \beta_2, \dots, \beta_p$  son términos constantes contenidos en un vector  $\hat{\beta}$  referidos como coeficientes de la regresión y  $\epsilon$  corresponde al error. El modelo RLM asume que para valores de  $X$  que caigan dentro del rango de los datos obtiene una adecuada estimación de la relación entre  $X$  y  $Y$ . El modelo RLM es una generalización de la regresión lineal simple (RLS) o, viceversa, el modelo RLS podría expresarse como un caso especial del modelo RLM cuando  $p = 1$ .

El objetivo del modelo RLM consiste en realizar una predicción para entender la relación de una variable aleatoria dependiente  $Y$  sobre una serie de predictores  $X_1, X_2, \dots, X_n$ , denominadas variables independientes. Para cumplir con este objetivo, el modelo RLM estima los parámetros del vector  $\hat{\beta}$  mediante el método de mínimos cuadrados ordinarios (MCO). El método MCO tiene como objetivo minimizar la suma del cuadrado de los errores que tiene como solución un sistema de ecuaciones normales. Se denomina a  $\hat{\beta}_0$  como el intercepto o constante del modelo, mientras que los coeficientes  $\hat{\beta}_j$  corresponden al estimador del coeficiente de regresión parcial para el predictor  $X_j$ . Al modelo ajustado para  $Y = (y_1, \dots, y_i)$  para cada observación de  $X_j = (x_{j1}, \dots, x_{jn})$  expresamos el modelo RLM como:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \dots + \hat{\beta}_{pn} x_{pn}, \quad (2.2)$$

Dado que la ecuación (2.22) representa los valores ajustados del modelo RLM, los errores o *residuos* ( $\hat{\epsilon}$ ) del modelo deben calcularse mediante:

$$\hat{\epsilon} = y_i - \hat{y}_i \quad i = 1, 2, \dots, n, \quad (2.3)$$

Definimos la SCE como:

$$SCE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n \hat{\epsilon}^2. \quad (2.4)$$

Finalmente, a partir de la ecuación (2.4) expresamos al estimador insesgado de la varianza ( $\sigma^2$ ) como:

$$\hat{\sigma}^2 = \frac{SCE}{n - p - 1}, \quad (2.5)$$

donde  $n - p - 1$  representa los grados de libertad,  $n$  representa el número de observaciones y  $p$  el número de predictores. Si consideramos el modelo RLM en términos de álgebra matricial, donde tenemos un vector  $\mathbf{Y}$ , una matriz de predictores  $\mathbf{X}$ , un vector de estimadores  $\beta$  y un vector de errores  $\epsilon$ , podemos escribir el modelo RLM en forma matricial como:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon, \quad (2.6)$$

donde asumimos que  $\epsilon$  para la estimación por MCO cumple que:

$$E(\epsilon) = 0 \quad y \quad Var(\epsilon) = E(\epsilon\epsilon^T) = \sigma^2\mathbf{I}_n, \quad (2.7)$$

donde  $E(\epsilon)$  es el valor esperado de  $\epsilon$ , la matriz  $\mathbf{I}_n$  es una matriz de identidad de orden  $n$  y  $\epsilon^T$  es la matriz transpuesta de  $\epsilon$ . Al minimizar la matriz  $\epsilon^T\epsilon$  (SCE) por el método MCO, tenemos el sistema de ecuaciones:

$$(\mathbf{X}^T\mathbf{X})\hat{\beta} = \mathbf{X}^T\mathbf{Y}, \quad (2.8)$$

El sistema de ecuaciones normales presentado en (2.9) asume que  $\mathbf{X}^T \mathbf{X}$  es invertible, por lo que el estimador MCO del vector  $\beta$  puede escribirse como:

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (2.9)$$

Los estimadores de MCO tienen las siguientes propiedades:

- El estimador  $\hat{\beta}_j$  y el estimador  $\hat{\sigma}^2$  son estimadores insesgados de  $\beta_j$  y  $\sigma^2$ , respectivamente. Para todos los estimadores lineales los estimadores MCO tienen varianza mínima por lo que se les considera los *mejores estimadores lineales insesgados (MELI)*.
- El estimador  $\hat{\beta}_j$   $j = 0, 1, \dots, p$  tiene distribución normal con media  $\beta_j$  y varianza  $\sigma^2 c_{ij}$ , donde  $c_{ij}$  es el  $j$ -ésimo elemento del inverso de la matriz de covarianza de los estimadores  $\hat{\beta}_j$  y  $\hat{\beta}_i$ .
- El estimador  $W = SCE/\sigma^2$  tiene distribución  $\chi^2$  con  $n - p - 1$  grados de libertad, donde  $\hat{\beta}$  y  $\hat{\sigma}^2$  se distribuyen independiente el uno del otro.

Las propiedades previas de los estimadores MCO son de utilidad cuando se busca hacer inferencias sobre las  $\hat{\beta}_j$  o sobre  $\hat{\sigma}^2$ ; sin embargo solamente se cumplen y permiten hacer inferencias cuando se cumplen los supuestos del modelo RLM, que se enuncian a continuación, como se describe en la clasificación presentada en *Chatterjee et al.* [15]:

- **Supuestos sobre la forma del modelo:** El modelo RLM que relaciona una variable  $Y$  con los predictores  $X_1, X_2, \dots, X_p$  es lineal para los estimadores de regresión  $\beta_1, \beta_2, \dots, \beta_n$ , lo que implica que para la  $i$ -ésima observación se cumple lo descrito en la ecuación (2.22). A éste supuesto se le conoce como supuesto de *linealidad*. Cuando no se cumple el supuesto de linealidad puede llevarse a cabo transformación de los datos para cumplirlo.
- **Supuestos sobre los errores del modelo:** Se asume que los errores del modelo son variables aleatorias normales idéntica e independientemente distribuidas (iid) con media 0 y varianza  $\sigma^2$ . Esto implica que:

- Los errores  $\epsilon_i$  tienen distribución normal con media cero. A este supuesto se le conoce como *distribución normal de los errores*.
  - Los errores  $\epsilon_i$  tienen la misma varianza desconocida  $\sigma^2$ . Esto se conoce como el supuesto de *varianza constante u homoscedasticidad*.
  - Los errores son independientes con covarianzas por pares igual a cero. A este supuesto se le conoce como de *independencia de los errores*.
- **Supuestos sobre los predictores:** Se asume que los valores de las variables predictoras en  $\mathbf{X}$  son no aleatorios y por lo tanto controlados por el investigador. Se asume que éste supuesto se cumple bajo situaciones experimentales pero puede no cumplirse en situaciones observacionales. Para el caso de los datos estudiados, se cuenta con las variables independientes obtenidas de mediciones por laboratorio (AST/ALT, METS-IR, insulina de ayuno), variables antropométricas (cintura, IMC) y variables demográficas (sexo, edad, diagnóstico previo de DM2). La lista detallada de variables se describe en la sección de **Resultados**.

### 2.1.1. Implementación del modelo RLM en los datos

1. De forma inicial, se realizó un ajuste de un modelo RLM preliminar con todas las variables estudiadas, poniendo la variable  $Y = \text{Grasa\_hepatica}$  como variable respuesta. El objetivo de este abordaje fue identificar potenciales observaciones influyentes que pudieran interferir con el modelo. Para identificar observaciones influyentes, se ajustó un modelo inicial con todos los predictores y se identificaron las observaciones influyentes basadas en COVRATIO, DIFFITS Y DIFFBETAS. Una vez identificadas las observaciones influyentes, se evaluó su retiro en el ajuste general de los datos considerando la posibilidad de que los valores fueran mediciones atípicas o que representaran extremos de la población que afectarían la generalización de la predicción de `Grasa_hepatica`.
2. Una de las características más importantes del abordaje de aprendizaje de máquina consiste en la generación de un modelo en una base de entrenamiento que logre un buen desempeño predictivo en una base de validación con la cual no se

realizó el ajuste. Para implementar ésta metodología, se decidió la aplicación de éste abordaje en el modelo RLM. Se generó una base de entrenamiento y una de validación utilizando muestreo aleatorio simple mediante la función `rsample`, con una separación aleatoria de la base en 70 % de entrenamiento y 30 % de validación.

3. Para comenzar con el ajuste y selección del modelo en la base de entrenamiento, se procedió a ajustar un modelo RLM utilizando todas las variables disponibles en la base de entrenamiento, teniendo como variable respuesta `Grasa_hepatica`. Para la selección de modelo un procedimiento exhaustivo por pasos (*step-wise*) utilizando al Criterio de Información Bayesiano (BIC, por sus siglas en inglés) como métrica para selección del modelo utilizando el paquete `leaps` de R.
4. Una vez identificado el modelo que minimizara el BIC se procedió a realizar un modelo RLM únicamente con las variables identificadas en el modelo. Los modelos depurados se compararon con los modelos saturados para comparación de los modelos finales.
5. El modelo final fue evaluado para los supuestos del modelo RLM antes descritos de a siguiente manera:
  - **Linealidad:** Se realizaron pruebas de falta de ajuste utilizando términos cuadráticos en los modelos RLM para cada una de las variables del modelo. Falta de ajuste se definió como una falta de rechazo de la hipótesis nula de  $\beta_i = 0$ .
  - **Homoscedasticidad de varianza:** Para validar el principio de homoscedasticidad de varianza se utilizará la prueba de Breusch-Pagan, así como la evaluación de gráficos de los residuos para comprobarla independencia de los errores.
  - **Matriz de diseño de rango completo:** Para validar éste supuesto, evaluaremos la colinearidad de las variables en el modelo completo y en el modelo final, para validar que la colinearidad no influyó en la selección del

modelo y que ésta no está presente en el modelo final. Para ésto calcularemos el índice de condicióny el factor de inflación de la varianza (VIF).

- **Normalidad conjunta de los errores:** Para evaluar el supuesto de normalidad conjunta de los errores primero exploraremos gráficamente los residuos y procederemos a probar su normalidad con la prueba de Anderson-Darling.

6. Como última instancia, para la evaluación final del modelo se calcularán intervalos de confianza simultáneos al 95 % para los coeficientes del modelo utilizando el método de Hotelling-Scheffé.

## 2.2. Modelo Gradient Boosting Machine

Considere un modelo *gradient boosting machine* (GBM), con un conjunto de datos  $(X, Y)$ , donde  $\mathbf{X} = X_1, X_2, \dots, X_n$  es una matriz de predictores y  $\mathbf{Y}$  es un vector que buscamos predecir mediante el modelo. El objetivo general del modelo GBM del modelo es generar una función  $f : x \rightarrow y$  para obtener un estimador de  $y$  de la forma  $\hat{f}(x) = \hat{y}$  para minimizar una función de pérdida  $\psi(y, f)$  como:

$$\hat{f}(x) = \arg \text{mín } \psi(y, f(x)) \quad (2.10)$$

La variable respuesta para el modelo en cuestión corresponde a una variable cuantitativa continua, donde  $y \in \mathbb{R}^+$ . En este caso la función de pérdida  $\psi$  puede evaluarse mediante el error cuadrático medio (ECM). Podemos reescribir la ecuación (2.10) en términos del valor esperado de la función de pérdida  $\psi$  con respecto a la variable de respuesta condicional al vector  $\mathbf{X}$  de la forma:

$$\hat{f}(x) = \arg \text{mín } E_x[E_y(\psi[y, f(x)]|x)] \quad (2.11)$$

Para restringir la estimación de la función se condiciona a la familia de funciones paramétricas, lo cual generaría un problema de estimación paramétrica [16]:

$$\hat{\theta} = \operatorname{argmin} E_x[E_y(\psi[y, f(x, \theta)]|x)] \quad (2.12)$$

Debido a que *gradient boosting machine* (GBM) es un algoritmo iterativo, utiliza como procedimiento de estimación de parámetros el *steepest gradient descent* (descenso de gradiente más pronunciado), el cual consiste en disminuir la función empírica de pérdida en los datos observados [16, 17]:

$$J(\theta) = \sum_{i=1}^N \psi(y_i, f(x_i, \hat{\theta})) \quad (2.13)$$

*Steepest gradient descent* es un proceso de optimización basado en mejoras consecutivas en la dirección del gradiente de la función de pérdida  $\nabla J(\theta)$ . Consiste en iniciar el algoritmo con un parámetro  $\hat{\theta}_0$ , tras lo cual se evalúa la función de pérdida  $\nabla J(\theta)$ , dados los parámetros obtenidos de la función:

$$\nabla J(\theta) = \left[ \frac{dJ(\theta)}{dJ(\theta_i)} \right]_{\theta=\hat{\theta}_i} \quad (2.14)$$

Basado en el gradiente de la función de pérdida, se calcula el estimador incremental del parámetro  $\hat{\theta}_t \leftarrow -\nabla J(\theta)$  y lo agrega al conjunto de estimadores de la forma:

$$\hat{\theta}^t = \sum_{i=0}^{t-1} \hat{\theta}_i. \quad (2.15)$$

Uno de los principios fundamentales del aprendizaje de máquina es la **optimización**, proceso durante el cual se lleva a cabo una suma iterativa por pasos en el espacio de la función tomando una aproximación inicial ( $\hat{f}_0$ ) con incrementos o *boosts* de la función  $\hat{f}_{i=1}^M$  [16, 17]. Para establecer la optimización, se define un estimador de la función de aprendizaje o  $h(x, \theta)$ . Generalmente se hace uso de árboles de decisión que determinan el tamaño de cada paso del algoritmo [16, 18]. Para la función estimada

en la  $i$ -ésima iteración, se define la regla de optimización como:

$$\hat{f}_t \leftarrow \hat{f}_{i-1} + \rho_t h(x, \theta_i). \quad (2.16)$$

### 2.2.1. Funcionamiento general del algoritmo GBM

Así pues, podemos definir el algoritmo GMB especificando una función de pérdida  $\psi(y, f)$  y una función de aprendizaje  $h(x, \theta)$ . El algoritmo general propuesto por Friedman et al. [16, 17] depende de los diseños de las funciones de pérdida y aprendizaje en:

1. Iniciar con una aproximación  $\hat{f}_0$  constante.
2. Para  $i=1$  a  $M$  iteraciones repetir:
3. Estimar el gradiente negativo de la función de pérdida  $\psi(y, f)$ .
4. Ajustar una nueva función de aprendizaje  $h(x, \theta_t)$ .
5. Encontrar el mejor paso en el gradiente descendiente ( $\rho_t$ ) mediante:

$$\rho_t = \arg \min_{\rho} \sum_{i=1}^N \Psi \left[ y_i, \hat{f}_{t-1}(x_i) + \rho h(x_i, \theta_t) \right] \quad (2.17)$$

Para un caso como el nuestro, donde el predictor  $\mathbf{Y}$  es una variable cuantitativa continua, tenemos las siguientes funciones de pérdida[18]:

1. Función Gaussiana de pérdida  $L_2$ , definida como:  $\Psi(y, f)_{L_2} = \frac{1}{2}(y - f)^2$ . Para esta función la derivada es  $y - f$  por lo que el algoritmo simplemente ejecuta ajustes basado en residuales para penalizar desviaciones grandes de las cantidades a predecir.
2. Función Laplaciana de pérdida  $L_1$  es una función de pérdida robusta de la forma:  $\Psi(y, f)_{L_1} = |y - f|$ . La función es de utilidad en variables con errores de cola larga.



3. Función de pérdida de Huber, especificando  $\delta$ :

$$\Psi(y, f)_{\text{Huber}, \delta} = \begin{cases} \frac{1}{2}(y - f)^2, & |y - f| \leq \delta \\ \delta(|y - f| - \delta/2), & |y - f| > \delta \end{cases} \quad (2.18)$$

Esta función de pérdida se basa en la noción de que alcanzar un error máximo para  $L_2$ , después de lo cual se aplica la función  $L_1$ .

La selección típica de la función de aprendizaje para el algoritmo GBM depende sobre todo en que éste sea computacionalmente eficiente [16]. En la mayoría de los casos la función de aprendizaje más utilizada son las de los *árboles de decisión*. El diseño de un árbol de decisiones se basa en generar particiones del espacio de las variables entrada en áreas de rectángulos homogéneos basado en una regla de decisión definido en la variable de entrada. Cuando los árboles de decisión tienen solamente una partición con dos nodos terminales, se le conoce como un *tocón de árbol*, los cuales generalmente han sido reconocidos como funciones de aprendizaje poco precisas. A pesar de esto, se ha demostrado que el uso de tocones de árboles deviene en resultados más efectivos que árboles con mayor número de particiones [16, 19]. Así pues, el algoritmo GBM basa su funcionamiento en una combinación de *weak learners* para generar un modelo con mayor fortaleza predictiva.

### 2.2.2. Implementación del modelo GBM en los datos

El modelo GBM es un algoritmo diseñado para generar un modelo con mayor capacidad predictiva conformado a partir de modelos con baja capacidad predictiva. Algunos de los hiperparámetros que deben ajustarse en el algoritmo para optimizar la función del algoritmo GBM son:

- **Tasa de aprendizaje:** Codificado como `shrinkage` en el paquete `gbm` de R. Éste hiperparámetro determina el factor por el cual se ajusta el modelo en cada iteración.
- **Profundidad de interacción:** Codificado como `interaction.depth` indica la profundidad máxima de los árboles o tocones utilizados en el modelo.

- **Observaciones en nodos terminales:** Codificado como `obs`, indica el número mínimo de observaciones requeridos en cada nodo terminal durante el ajuste del modelo.
- **Submuestras:** Codificado como `bag.fraction` indica el porcentaje de los datos de entrenamiento a muestrear para cada árbol en cada iteración del modelo.
- **Fracción de entrenamiento:** Codificado como `train.fraction` indica el porcentaje de observaciones a muestrear para cada árbol, utilizando el resto de las observaciones para estimar la función de pérdida.

Ajustaremos el modelo GBM con los valores transformados, siguiendo un protocolo similar al utilizado para el modelo RLM, pero además se procederá a centrar las variables transformadas con media 0 y desviación estándar 1 para optimizar el algoritmo *boosting* el cual facilita la identificación de mínimos cuando los datos se encuentran estandarizados [18] .

### 2.2.3. Optimización de hiperparámetros del modelo GBM

Para la optimización de los parámetros diseñamos una matriz con diferentes valores a probar por *cross-validation* especificando el siguiente rango de valores, previamente explorados en modelos similares [17]:

- `shrinkage`: 0.1, 0.2, 0.4
- `interaction.depth`: 1, 3, 5
- `n.minobsinnode`= 5, 7, 10
- `bag.fraction`= 0.65, 0.8, 1.0

Como podemos observar, se iterará el modelo para un total de 81 combinaciones de hiperparámetros con el objetivo de encontrar la combinación que minimice la función de pérdida estimando un máximo de 5,000 árboles para cada combinación de hiperparámetros, como se muestra en el código desglosado en **Anexos**. Para el

algoritmo de optimización de hiperparámetros se utilizará solamente al 75 % de la base de entrenamiento (`train.fraction`), para disminuir el tiempo computacional requerido para identificación de los hiperparámetros óptimos. La función de pérdida establecida para el modelo GBM fue la raíz cuadrada de la suma del cuadrado de los errores (RMSE, por sus siglas en inglés).

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (2.19)$$

Una vez identificado el modelo se considerará su ajuste en la base de entrenamiento con la combinación óptima de hiperparámetros identificada por *cross-validation*.

### 2.3. Modelo de Redes neuronales auto-normalizables

*Deep Learning* (DL) o aprendizaje profundo es un campo del aprendizaje de máquina que consiste en modelos de aprendizaje basados en representaciones de datos en capas sucesivas de transformaciones numéricas incrementalmente más representativas. El término *profundo* de DL se basa precisamente en ésta organización de capas sucesivas, de forma que modelos más *profundos* son aquellos con un mayor número de capas. Los modelos más frecuentes en DL son las *redes neuronales*, los cuales codifican datos de entrada mediante múltiples capas profundas para la estimación de un desenlace sometiendo a los datos a diferentes formas de transformación por capa. Las redes neuronales pueden subclasificarse dependiendo del algoritmo utilizado; las aplicaciones computacionales más frecuentes han encontrado resultados muy favorables para clasificación de imágenes con redes neuronales convolucionales y para clasificación de texto con redes neuronales recurrentes. La aplicación de las redes neuronales *feed-forward* para clasificación binaria o multiclase o para predicción de variables continuas ha sufrido debido a los procesos de normalización o pre-procesamiento de datos necesarios en las capas profundas de las redes neuronales, causando elevada variabilidad en el cálculo de las funciones de error y poca efectividad en la aplicación de métodos de regularización.

### 2.3.1. Principios de funcionamiento de las redes neuronales

La función de una capa específica se basa en la estimación de *pesos* y el efecto de la transformación de los datos procesados por una capa son parametrizados por sus pesos. El *aprendizaje* llevado a cabo por las capas de una red neuronal consiste en encontrar el valor óptimo de los pesos, que generan el algoritmo que lleva a las predicciones de mayor precisión. En el contexto de una red neuronal los parámetros estimados tienen relación directa con el número de nodos y capas, por lo que puede volverse una tarea computacionalmente demandante optimizar ambos parámetros. Para controlar la salida de un modelo de redes neuronales es necesario cuantificar la divergencia de la cantidad estimada con respecto de la cantidad real (residuos). Para esto, al igual que con el modelo GBM se utiliza una función de pérdida, denominada función objetiva, la cual calcula el score de distancia (o residuo) obtenido de la red. Lo que hace que el DL sea iterativo se basa en generar una señal de retroalimentación para el ajuste de los pesos en una dirección que minimice los residuos del modelo. Esto se lleva a cabo mediante un *optimizador* el cual genera un algoritmo denominado *back-propagation* al cual subyace al funcionamiento del DL. En la **Figura 2.1** se esquematiza el modelo básico de los algoritmos de DL basados en redes neuronales. Brevemente, el algoritmo de una red neuronal funciona de la siguiente manera:

1. Al principio, los pesos de la red neuronal tienen valores aleatorios, por lo que las transformaciones son al azar.
2. Al inicio de las predicciones, la función de pérdida detecta un valor de pérdida alto.
3. Mediante *stochastic gradient descent* se calcula la derivada de la función de pérdida que minimice el calor de pérdida para una siguiente iteración.
4. El optimizador entonces ajusta los pesos de las capas de la red neuronal, mediante *back-propagation*, induciendo un descenso del valor de pérdida en un proceso conocido como *bucle de entrenamiento*.

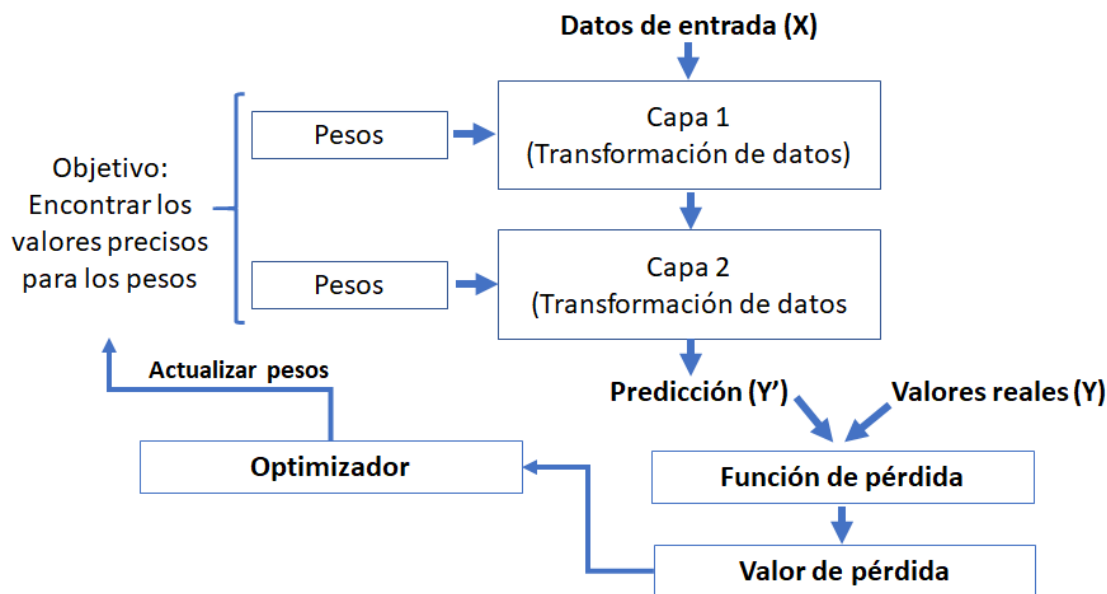


Figura 2.1: Esquema general de una red neuronal. Reproducido de Chollet et al [20]

5. Si se llevan a cabo múltiples iteraciones, se minimiza la función de pérdida. En éste punto se dice que la red está entrenada.

Los principales componentes que determinan la arquitectura de una red neuronal son los inicializadores, las funciones de activación, las funciones de pérdida y la configuración de optimización.

1. Los inicializadores corresponden a los valores de la matriz de pesos  $\mathbf{W}$  con la cual inicia la estimación de la red neuronal. Usualmente son números aleatorios con una distribución estadística conocida. La selección los valores de *inicialización* es dependiente de la función de activación que se elija.
2. Las funciones de pérdida o funciones objetivo representan la cantidad a minimizar mediante *stochastic gradient descent* durante el entrenamiento de la red neuronal y se utiliza como medida del grado de ajuste de la red. Para un problema de regresión a valores arbitrarios las funciones de pérdida más utilizadas son el error cuadrático medio y el error absoluto medio.
3. La función de activación establece las restricciones en la última capa de la red neuronal que determina el tipo de predicción que genera el modelo. Si se busca

un modelo de regresión a valores arbitrarios no es necesario utilizar una función específica, aunque la más comúnmente utilizada es la función ReLu o rectificador que se basa en la función  $f(x) = \max(0, x)$ .

4. La configuración de optimización determina la forma en que la red será actualizada basada en la función de pérdida. Dependiendo el optimizador se especifica la metodología precisa del *stochastic gradient descent*. Los dos optimizadores más frecuentes son RMSProp (que ajusta con mayor rapidez la tasa de aprendizaje) y ADAM (que genera gradientes por parámetro de forma adaptativa).

### 2.3.2. Redes neuronales auto-normalizables

Una limitación en la aplicación de DL en la predicción de desenlaces continuos tiene que ver con la alta variabilidad del error de estimación y de las funciones de pérdida [21]. Una solución propuesta por Klambauer et al. para solucionar éste problema son las *self-normalizing neural networks* o redes neuronales auto-normalizables (SNNN, por sus siglas en inglés). Consideremos una red neuronal con función de activación  $g$  y dos capas consecutivas conectadas por una matriz de pesos  $\mathbf{W}$ . Si la variable de entrada a una red neuronales una variable aleatoria, entonces los valores de entrada de la red  $z = \mathbf{W}x$  y las activaciones  $y = f(z)$  en la capa precedente también son variables aleatorias.

Por lo tanto, se define una SNNN como una red neuronal que posee una función  $g : \Omega \rightarrow \Omega$  para cada activación  $y$  que mapea la media de la varianza de una capa a la siguiente y tiene un punto estable y crítico que depende de la suma de pesos  $\omega = \sum_{i=1}^m w_i$  y de pesos al cuadrado  $\tau = \sum_{i=1}^n w_i^2$ . Cuando se aplica el mapeo de  $g$  a cada punto en  $\Omega$  converge a éste punto fijo.

Por lo tanto, se considera que las activaciones de una red neuronal están normalizadas si tanto su media como su varianza entre muestras se encuentran en intervalos pre-definidos, por lo que la normalización es transitiva entre capas. Por lo tanto, las SNNN mantienen normalización a través de las propagaciones de la red. Para determinar una SNNN se implementarán dos adaptaciones a las redes neuronales previamente

descritas:

- **Función de activación:** Como se había referido en la sección previa, la función de activación más ampliamente utilizada en redes neuronales para casos de regresión es ReLU. Para las SNNN la función de activación debe cambiarse por *scales exponential linear units* (SeLU), definida como:

$$\text{selu}(x) = \lambda \begin{cases} x, & \text{si } x > 0 \\ \alpha e^x - \alpha, & \text{si } x \leq 0 \end{cases} \quad (2.20)$$

La función de activación SeLU es lo que permite la auto-normalización de las SNNN. El uso de *back-propagation* con funciones ReLU, sigmoides o tangentes no puede converger en una normalización. Esto permite que los valores dentro de la red se centren y que se regularice el cambio de la varianza interna de la red neuronal. El valor de  $\lambda > 1$  permite cumplir con las propiedades que llevan a una SeLU.

- **Inicialización:** Se propone que para una SNNN se inicialice la red neuronal valores aleatorios obtenidos de una distribución aproximadamente normal, con media 0 y varianza  $\frac{1}{\text{inputs}}$ , donde *inputs* se refiere al número de predictores introducidos a la red.
- **Dropout:** Para la eliminación de neuronas con poca contribución se utilizan las capas de *dropout*, las cuales permiten disminuir la proporción de ruido en los datos de entrenamiento modelados por la red neuronal. Para las SNNN se utiliza una función *dropout* que cumpla con el principio de  $\lim_{x \rightarrow -\infty} \text{selu}(x) = -\lambda\alpha = \alpha'$ . Éste principio permite mantener la media y varianza de los valores procesados y se denomina *alpha-dropout*. Su aplicación permite asegurar la naturaleza auto-normalizable de las SNNN.

Las SNNN convergen en valores con media cero y varianza conocida. Su aplicación es útil en arquitecturas con múltiples capas, generando resultados robustos altamente reproducibles y tienen un mejor desempeño comparado con redes densas completamente

conectadas, independientemente de la normalización de las variables de entrada o por capa [21].

### 2.3.3. Implementación del modelo SNNN en los datos

Los modelos de redes neuronales auto-normalizables (SNNN) requieren de una especificación de hiperparámetros similar a la aplicada para el algoritmo GBM. En éste caso, la selección de la función de activación y los parámetros de inicialización deben ser los mismos para cumplir con el principio de auto-normalización requerido por las SNNN. Debido a que las combinaciones de hiperparámetros en los modelos SNNN son múltiples, requieren de múltiples ensayos experimentales con diferentes combinaciones de parámetros. El modelo final es resultado de la iteración de diversos modelos, hasta encontrar el modelo que minimizó la función de error en *cross-validation*. La arquitectura final del modelo SNNN se diseñó con una red neuronal con 3 capas, definidas a continuación:

- **Capa de entrada:** 8 neuronas, con una función de activación SeLU y un inicializador aleatorio LeCun normal. Se ajustaron varias redes con diferente número de neuronas, el valor óptimo de acuerdo a la literatura debe no ser mayor al número de predictores, en éste caso ocho.
- **Segunda capa:** Una función *layer\_alpha\_dropout* para eliminar neuronas con contribución <10 %, para prevenir el sobreajuste.
- **Capa de salida:** Se utilizó una función *softplus* para evitar valores negativos. La función *softplus* se define como:  $f(x) = \log(1+e^x)$  y es una función estrictamente creciente.
- **Función de error:** Se utilizó el error absoluto medio (MAE, por sus siglas en inglés) como función de error, definida como:

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (2.21)$$



- **Función de pérdida:** Se utilizó el error cuadrático medio (MSE, por sus siglas en inglés) como función de pérdida, definida como:

$$\text{MSE} = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2 \quad (2.22)$$

- **Optimizador:** Se utilizó como optimizador ADAM, debido a su capacidad de actualizar la tasa de aprendizaje dependiendo de la efectividad del *back-propagation* en reducir el error de predicción.

El código de R utilizado para el ajuste de la red se presenta a continuación:

```
set.seed(123)
build_model <- function() {
  model <- keras_model_sequential() %>%
    layer_dense(units = 8, kernel_initializer='lecun_normal',
    activation="selu", input_shape = dim(train_data)[[2]]) %>%
    layer_alpha_dropout(rate=0.10) %>%
    layer_dense(units = 1, activation='softplus')
  model %>% compile(
    optimizer = "adam",
    loss = "mse",
    metrics = c("mae")
  )
}
```

Las SNNN son redes completamente conectadas en las cuales, considerando el input de la base de entrenamiento, incluye la conexión de cada una de las 8 variables con 8 neuronas en la capa densa de entrada, seguida por una capa de *alpha dropout*, la cual tiene la función de eliminar las neuronas con contribución en la matriz de pesos  $\mathbf{W} < 10\%$ . La salida de las 8 neuronas de la capa de entrada es procesada en una capa de salida mediante la función SoftPlus para resultar en una neurona de salida en la capa densa que arroja los valores transformados de la variable `Grasa_hepatica`.

- Posterior a la determinación de la arquitectura de la SNNN, evaluaremos el número óptimo de épocas que debemos entrenar la red neuronal para minimizar el error cuadrático medio, que elegimos como nuestra función de pérdida. Para esto diseñamos un algoritmo de k-fold cross-validation, con k=4 entrenado durante 50 épocas o iteraciones del modelo. Esto permitirá optimizar el cálculo del error de predicción.

## 2.4. Comparación de los modelos RLM, GBM y SNNN

Para la comparación de los modelos de forma inicial evaluaremos cada modelo (RLM, GBM, SNNN) en la cohorte de validación (n=38). Al comparar el desempeño de los modelos en un conjunto independiente de datos con respecto al original en que fueron ajustados estamos demostrando su reproducibilidad. Para este propósito, se utilizaron como métricas de comparación el MAE, MSE, RMSE y la correlación de Pearson de los valores observados con respecto a los predichos. Finalmente, como para el modelo de regresión lineal simple, la  $R^2$  representa el cuadrado de la correlación de Pearson para un análisis de regresión lineal simple (RLS). Éste último resultado se presentará ésta métrica como un componente adicional de la comparación.

rado de ajuste de los modelos y los errores de predicción, es de interés para determinar la efectividad de los modelos su evaluación en los siguientes dos dominios:

- **Precisión y error:** Se busca que los modelos además de minimizar el error de estimación, obtengan mediciones cuya estimación no tenga divergencias significativas del valor obtenido mediante resonancia magnética. Como los resultados de cada modelo son los valores transformados de `Grasa_hepatica`, se elevarán al cubo los valores predichos para analizarlos en sus unidades originales y estimar con mayor exactitud estos parámetros. Para esto se hará uso de un análisis de Bland y Altman utilizando el paquete `blandr` de R, como se describirá a continuación.
- **Concordancia y relevancia clínica:** Otro objetivo de los modelos es ser de utilidad como herramienta diagnóstica para EHGNA. Para evaluar estos fenó-

menos se hará uso de dos herramientas: 1) Las curvas de características operativas del receptor (ROC, por sus siglas en inglés) y las matrices de confusión. Ambos modelos también se detallan más adelante.

- Para la evaluación de los modelos, se categorizó la variable `Grasa_hepatica` no transformada de acuerdo al punto de corte  $\geq 5.5$ , lo cual se establece como la definición clínica de EHGNA cuando se utiliza MRI-S como método de diagnóstico. Para el cálculo del área bajo la curva ROC se utilizó el paquete `pROC` de R y para el cálculo de métricas de la matriz de confusión se utilizó el paquete `OptimalCutpoints`, también de R.

#### 2.4.1. Análisis de Bland-Altman

El análisis de Bland-Altman tiene el objetivo de evaluar el grado de acuerdo entre dos métodos para la medición de una misma variable. Para explicar la metodología propuesta por Bland y Altman [22] supongamos la medición de una variable aleatoria  $\mathbf{X}$  obtenida de un método como  $X_1$  y la medición por otro método como  $X_2$  medidas en  $n$  observaciones pareadas  $(X_{1k}, X_{2k}), k = 1, 2, \dots, n$ . El análisis de Bland-Altman considera la estimación del sesgo ( $\bar{d}$ ) y de límites de acuerdo los cuales se incluyen en un gráfico de las diferencias  $d = X_1 - X_2$  contra las medias  $(X_1 + X_2)/2$  de las mediciones por ambos métodos. Podemos definir el sesgo ( $\bar{d}$ ) como:

$$\bar{d} = \frac{1}{n} \sum_{k=1}^n d_k \quad (2.23)$$

Los límites de acuerdo se estiman mediante el intervalo de predicción estimado mediante la desviación estándar de las diferencias ( $\bar{s}_d$ ) calculadas como:

$$\bar{s}_d = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (d_k - \bar{d})^2} \quad (2.24)$$

Por lo tanto, los límites de acuerdo (LoA, por sus siglas en inglés), se calculan como el intervalo de predicción al 95 % del sesgo  $\bar{d}$ , como:

$$LoA = \bar{d} \pm z_\alpha * \bar{s}_d \quad (2.25)$$

Donde  $z_\alpha$  representa el cuantil para un tamaño de prueba  $\alpha = 0.05$ . Para hacer inferencias poblacionales sobre el sesgo y LoA mediante intervalos de confianza es necesario calcular los estimadores de varianza de  $\bar{d}$  y de  $\bar{s}_d$ , los cuales se estiman como:

$$\hat{V}(\bar{d}) = \frac{S_d^2}{n} \quad (2.26)$$

$$\hat{V}(LoA) = \left( \frac{1}{n} + \frac{z_\alpha^2}{2(n-1)} \right) S_d^2 \quad (2.27)$$

Debido a que en el análisis de Bland-Altman el sesgo es una medida de diferencia, de esto sigue que los intervalos de confianza al 95 % para  $\bar{d}$  y para LoA se obtienen a partir de cuantiles de la distribución  $t$  de la siguiente manera:

$$\bar{d} \pm t_{1-\alpha/2, n-1} \sqrt{\text{Var}(\bar{d})} \quad (2.28)$$

$$LoA \pm t_{1-\alpha/2, n-1} \sqrt{\text{Var}(\bar{d} \pm 1.96 S_d)} \quad (2.29)$$

El análisis de Bland-Altman busca cuantificar el sesgo de los métodos medición o estimación y establecer límites que razonablemente cubran valores esperados para determinar consistencia en el método. Para que un método demuestre reproducibilidad debe tener un sesgo bajo y límites de acuerdo estrechos.

#### 2.4.2. Curvas ROC

El área bajo la curva ROC (AUROC) es una medida de desempeño de uso frecuente en la validación de modelos de aprendizaje de máquina para clasificación [23]. Para el uso de AUROC consideremos una variable aleatoria  $X \in (0, 1)$ . Para la clasificación de individuos basado en un estimador  $\theta$  y consideremos una regla de probabilidad  $s = s(x)$  para una clase  $k$ , donde  $f_k(s), k = 0, 1$  representa una función de densidad

de probabilidad del puntaje  $s(x)$ . La probabilidad de clasificar un objeto nuevo con la regla de decisión se obtiene al comparar el score  $s$  con un umbral de clasificación  $t$ , de tal forma que  $s > t$  implica que el objeto se clasifique como 1 y  $s \leq t$  implica que se clasifique como 0, la probabilidad de ambos eventos debe ser igual a 1,  $\pi_0 + \pi_1 = 1$ . Para obtener desempeño de la clasificación podemos hacer uso de métricas como sensibilidad y especificidad:

- **Sensibilidad:** La sensibilidad de un clasificador se refiere a la proporción de casos correctamente clasificados como casos,  $F_1(t)$ . En general, la sensibilidad refiere la capacidad de un modelo para clasificar a un caso como tal y se obtiene de dividir la tasa de verdaderos positivos entre la suma de verdaderos positivos más falsos negativos.
- **Especificidad:** La especificidad de un clasificador se refiere a la proporción de no casos correctamente clasificados como no casos,  $1 - F_0(t)$ . En general, la especificidad refiere la capacidad de un modelo para clasificar a un no caso como tal y se obtiene de dividir la tasa de verdaderos negativos entre la suma de verdaderos negativos más falsos positivos.

Conforme cambia el umbral de clasificación, también se cambia el valor de sensibilidad y especificidad. La curva ROC, entonces, es una curva de  $F_1(t)$  en el eje de las ordenadas y  $F_0(t)$  (o 1-Especificidad) en el eje de las abscisas. Un clasificador que tenga una impecable capacidad de discriminación entre dos clases tendrá una curva que se aproxime a una línea horizontal con  $F_1(t) = 1, F_0(t) = 1$ . Por lo tanto, el AUROC será una métrica del desempeño de un clasificador adecuado con valores que oscilen entre 0.5 para un clasificador perfectamente aleatorio y 1.0 para un clasificador perfecto. Si dejamos que  $v = F_0(d)$ , por la definición de la curva ROC tenemos que su área bajo la curva es:

$$\int_0^1 F_1(F_0^{-1}(v)) dv \quad (2.30)$$

Si hacemos un cambio de variable e integramos con respecto a la función de clasificación  $s$ , tenemos que el AUROC se calcula como:

$$AUROC = \int_{-\infty}^{\infty} F_1(s)f_1(s)ds \quad (2.31)$$

Se puede observar por la definición de AUROC que el umbral  $t$  no necesita ser pre-especificado, lo cual mejora la aplicabilidad y reproducibilidad de su uso como métrica de desempeño. La selección de  $t$  puede hacerse con base en distintos criterios para optimizar la clasificación de una forma que maximice el desempeño de la prueba diagnóstica. El índice de Youden es una métrica que permite la optimización de selección de punto de corte para minimizar la probabilidad de errores de clasificación en una prueba diagnóstica al maximizar el valor resultante de la suma de sensibilidad y especificidad menos 1 para todos los posibles valores de sensibilidad y especificidad calculados para una prueba diagnóstica [24]. El índice de Youden ( $J$ ) se calcula como:

$$J = \max(F_1(t) + F_0(t)) \quad (2.32)$$

La interpretación del AUROC puede hacerse como la probabilidad de la correcta clasificación de las clases 0 y 1 o como la sensibilidad promedio considerando todos los valores de especificidad como igualmente probables. Para complementar su uso se utilizan otras métricas derivadas de la matriz de confusión.

### 2.4.3. Matriz de confusión

La matriz de confusión es una matriz de tamaño  $n \times n$  asociada con un clasificador que contrasta la clasificación predicha con respecto a la real, donde  $n$  representa el número de clases [25]. Para el caso de un clasificador con clases  $k, k = 0, 1$  tenemos la siguiente distribución:

Comparación de modelos		
	Predicho positivo	Predicho negativo
Verdadero positivo	a	b
Verdadero negativo	c	d

Basado en la tabla previa, podemos determinar que:

- $a$  representa el número de verdaderos positivos
- $b$  representa el número de falsos negativos
- $c$  representa el número de falsos positivos
- $d$  representa el número de verdaderos negativos

De la matriz de confusión pueden derivarse distintas métricas. La precisión del clasificador se define como la proporción de casos correctamente clasificados y se calcula como:

$$Precisión = \frac{a + d}{a + b + c + d} \quad (2.33)$$

De la misma matriz de confusión es posible calcular la sensibilidad y la especificidad del clasificador utilizados para la construcción de la curva ROC:

$$Sensibilidad = \frac{a}{a + c}, \quad Especificidad = \frac{d}{d + b} \quad (2.34)$$

Otras métricas de utilidad son los valores predictivos positivos (VPP) y los valores predictivos negativos (VPN). Estos indican la probabilidad de que un valor positivo implique clasificación correcta como caso y la probabilidad de que un valor negativo implique clasificación correcta como no caso, respectivamente. Los estimadores se calculan como:

$$VPP = \frac{a}{a + b}, \quad VPN = \frac{d}{d + c} \quad (2.35)$$

Como tanto la precisión, sensibilidad, especificidad, VPN y VPP son proporciones, los intervalos de confianza para éstas métricas pueden calcularse como intervalos de

confianza para proporciones, de la forma:

$$\hat{p} \pm z_{\alpha/2} \sqrt{\frac{\hat{p}(1 - \hat{p})}{n}} \quad (2.36)$$

La evaluación por matriz de confusión para pruebas de diagnóstico en medicina es un indicador de su desempeño en escenarios reales y puede dar una idea de la confiabilidad de su aplicación para la toma de decisiones clínicas y terapéuticas [26].



# Capítulo 3

## Resultados

### 1. Análisis descriptivos

Los datos analizados se obtuvieron de 67 sujetos de investigación sin diabetes y 62 sujetos con diabetes reclutados de la Unidad de Investigación de Enfermedades Metabólicas del Instituto Nacional de Ciencias Médicas y Nutrición Salvador Zubirán, en quienes se realizaron las evaluaciones bioquímicas, antropométricas y por resonancia magnética previamente descritas. Las ocho variables de interés para el análisis son:

1. **Variable desenlace:** Porcentaje de grasa intrahepática obtenido por resonancia magnética nuclear con espectroscopía, en la base codificado como `Grasa_hepatica`. Variable cuantitativa continua.
2. **Insulina de ayuno:** Concentración sérica de insulina obtenida tras 10-12 hrs de ayuno, medido en  $\mu/L$ , en la base codificado como `Insulina`. Variable cuantitativa continua.
3. **Resistencia a la insulina:** Cuantificado mediante el índice METS-IR, representado en la base como `METS_IR`. Variable cuantitativa continua.
4. **Proporción AST/ALT:** Representa la división de valores séricos de la enzima aspartato aminotransferasa (AST) entre los valores séricos de la enzima

alanino aminotransferasa (ALT), codificado en la base como `AST_ALT`. Variable cuantitativa continua.

5. **Circunferencia de cintura:** Evaluada mediante la circunferencia de cintura expresada en centímetros, representada en la base como `Cintura`. Variable cuantitativa continua.
6. **Edad:** Edad de los sujetos evaluados en el estudio, expresada en años cumplidos y codificada en la base como `Edad`. Variable cuantitativa discreta.
7. **Diabetes mellitus tipo 2:** Variable categórica binaria que identifica con 0 a los pacientes sin diabetes y con 1 a los pacientes con diabetes, codificado en la base como `DM2`.
8. **Sexo:** Variable categórica binaria que identifica con 0 a sujetos de sexo femenino y con 1 a sujetos de sexo masculino. Codificada en la base como `Sexo`.

Primero, decidimos investigar la relación existente entre ellas utilizando el método de correlación de Pearson. Como podemos observar, la variable desenlace `Grasa_hepatica` tiene correlaciones positivas de moderada intensidad con `Insulina` ( $r = 0.563$ ) y con `METS_IR` ( $r = 0.519$ ) y una correlación negativa moderada con `AST_ALT` ( $r = -0.519$ ); se observaron además correlaciones bajas con `Cintura` y `Edad`, por lo que no se espera que sean predictores significativos. Tras realizar descripciones gráficas de los datos mediante gráficos de dispersión con ajuste suavizado e histogramas, como podemos observar en la **Figura 2.1**, se concluyó que la distribución de la mayoría de las variables tiene un sesgo positivo a la derecha, evidenciando asimetría en la distribución. Debido a esto, se decidió evaluar transformaciones para mejorar la simetría de las distribuciones previo al análisis. Esto tiene fundamento en que tanto el modelo RLM como los modelos de aprendizaje de máquina tienen mejor desempeño con distribuciones centradas y simétricas [17, 20].

Se probaron diversas transformaciones para disminuir el sesgo y hacer las distribuciones más simétricas incluyendo raíz cúbica, raíz cuadrada, potencias, logaritmo natural y base 10 e inversa. Finalmente, se identificó que la transformación por raíz

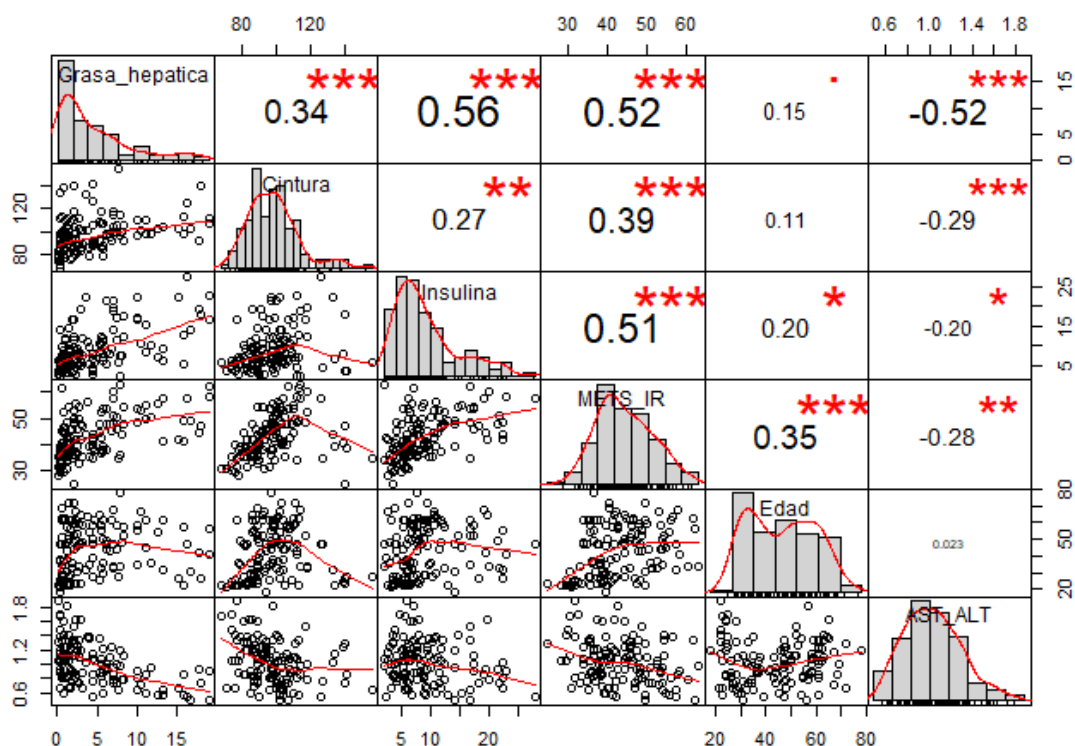


Figura 3.1: Gráficos descriptivos de las variables continuas.

cúbica de *Grasa\_hepatica* y las transformaciones mediante logaritmo natural de *AST\_ALT*, *METS\_IR*, *Cintura* e *Insulina* lograban mayor simetría en las distribuciones de las variables. No se logró simetría con ninguna transformación para *Edad*.

Como puede corroborarse en la **Figura 3.2**, la transformación de las variables mejoró la simetría de las distribuciones y redujo sesgos en las variables; así mismo, hizo más evidentes las relaciones lineales entre las variables evaluadas, lo cual permite postular su influencia en el ajuste del modelo. Finalmente, se evaluó la influencia de las variables categóricas en la variable dependiente *Grasa\_hepatica* utilizando la prueba t de Student. No se observaron diferencias significativas en *Grasa\_hepatica* transformada para un tamaño de prueba  $\alpha = 0.05$  contrastando las categorías de la variable sexo (1.43 vs. 1.48,  $p = 0.6307$ ). Sin embargo, sí se observaron diferencias significativas contrastando el grupo de sujetos con diabetes contra el grupo sin diabetes para los valores transformados de *Grasa\_hepatica* (1.26 vs. 1.63,  $p = 0.000192$ ).

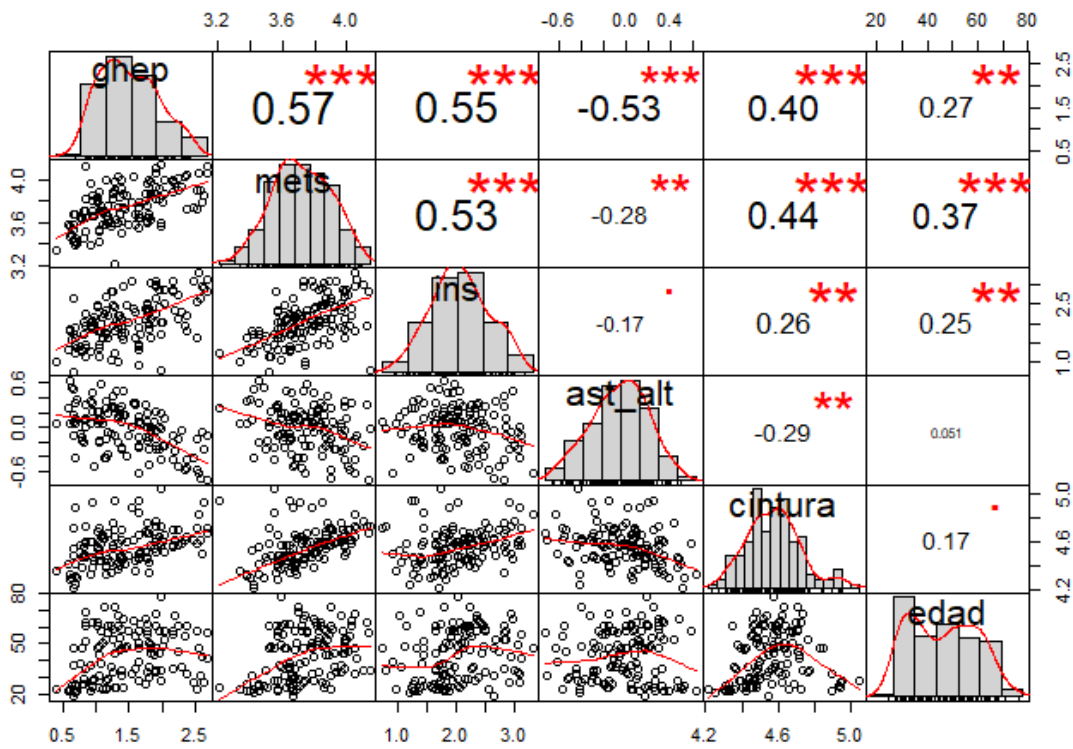


Figura 3.2: Gráficos descriptivos de las variables continuas transformadas.

Esto indica la potencial influencia de la variable DM2 en determinar el contenido de grasa intrahepática con un tamaño de prueba  $\alpha = 0.05$ .

## 2. Modelo de regresión lineal múltiple

Tras realizar el ajuste del primer modelo RLM con todas las variables se calcularon las observaciones potencialmente influyentes, como se detalla a continuación:

Observaciones influyentes				
Observación	DFFITS	COVRATIO	Distancia de Cook	Hat
40	0.54	0.72*	0.03	0.04
112	0.78*	1.16	0.08	0.21*
121	0.85*	1.13	0.09	0.21*

Tabla 1: Análisis de observaciones influyentes del modelo completo RLM.

Como se observa en la **Tabla 1**, las observaciones 40, 112 y 121 se consideraron influyentes basados en DDFITS y COVRATIO, sin identificar influencia de DIFFBETAS (no mostrado). Estos resultados se complementan con las gráficas de DDFITS y COVRATIO (**Figura 3.3**), donde se confirma que muchas observaciones se salen de los límites de confianza. Sin embargo solamente 3 en DDFITS y una en COVRATIO podría considerarse como influyentes utilizando la función `influence.measures`. Al analizar los casos se identifica que el caso 40 tuvo un valor de grasa hepática atípico dado su estado metabólico y los casos 112 y 121 son sujetos con obesidad abdominal extrema, lo que dificulta la medición precisa de grasa hepática por RMN. Debido a estas inconsistencias metodológicas identificadas en el análisis estadístico, se decidió excluir a los sujetos, con el objetivo de tener un mejor ajuste y un modelo más generalizable sin considerar los casos extremos una vez que se separen las bases en entrenamiento y validación.

### 2.1. Selección del modelo en la base de entrenamiento

Se realizó un ajuste inicial del modelo RLM incluyendo todas las variables en estudio en la base de validación. A continuación, se muestra el resumen del primer modelo saturado:

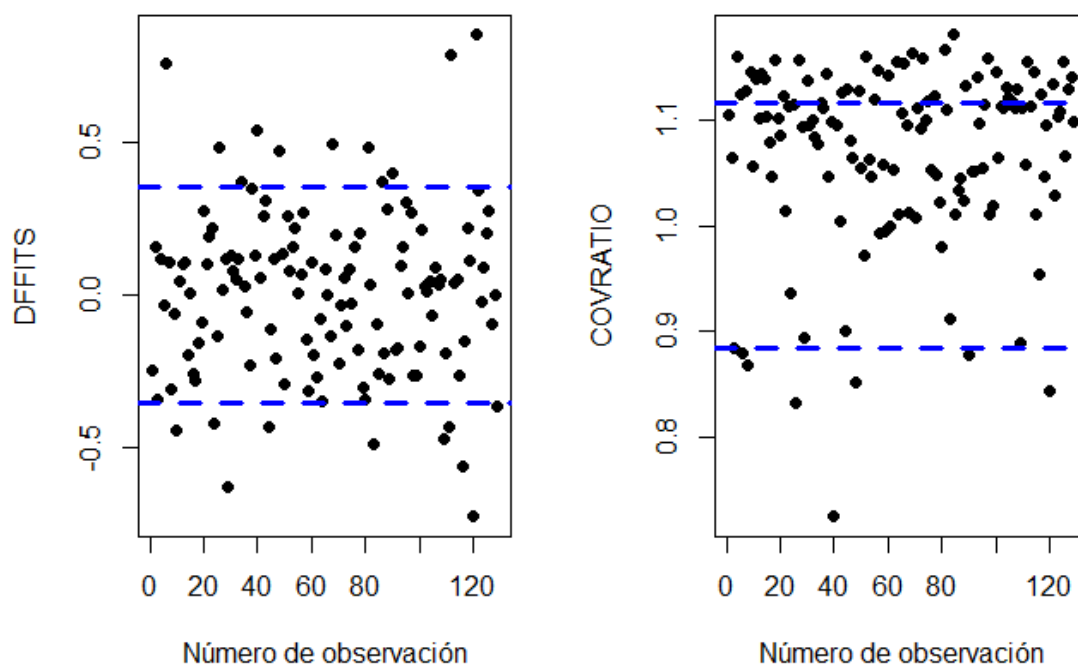


Figura 3.3: DFIITS y COVRATIO de las variables con sus límites de confianza para el modelo RLM con todas las variables.

Modelo RLM saturado				
Coeficientes	$\hat{\beta}$	Error estándar	Valor t	$p - value$
Intercepto	-2.844697	1.335904	-2.129	0.036256
METS-IR	0.632312	0.289842	2.182	0.032040
Insulina	0.332323	0.092468	3.594	0.000558
AST/ALT	-0.714977	0.144706	-4.941	<0.00001
Cintura	0.247922	0.304575	0.814	0.418035
Edad	0.001131	0.003726	0.304	0.762278
Sexo	-0.105276	0.088852	-1.185	0.239547
DM2	0.142338	0.117745	1.209	0.230234
SRE= 0.3648, 81 G.L.	$R^2 = 0.603$	$R^2_{adj} = 0.5687$	F-test=17.58	$p < 0.0001$

Tabla 2: Modelo RLM saturado que incluye todas las variables estudiadas.

Con el modelo presentado en la **Tabla 2** se concluye que la variable METS-IR, el valor de *Insulina* en ayuno y la proporción AST\_ALT están asociados con los valores transformados de grasa intrahepática. Con un tamaño de prueba  $\alpha = 0.05$  no se identificó influencia de *Cintura*, *Edad*, *Sexo* y *DM2* en este primer modelo. El modelo con todas las variables tiene un  $R_{adj}^2 = 0.596$ , lo que implica que explica el 59.6% de la variabilidad en los valores transformados de grasa intrahepática. Para evaluar si el modelo ajustado representa el mejor modelo mediante el paquete `leaps` de R se utilizó como criterio el BIC.

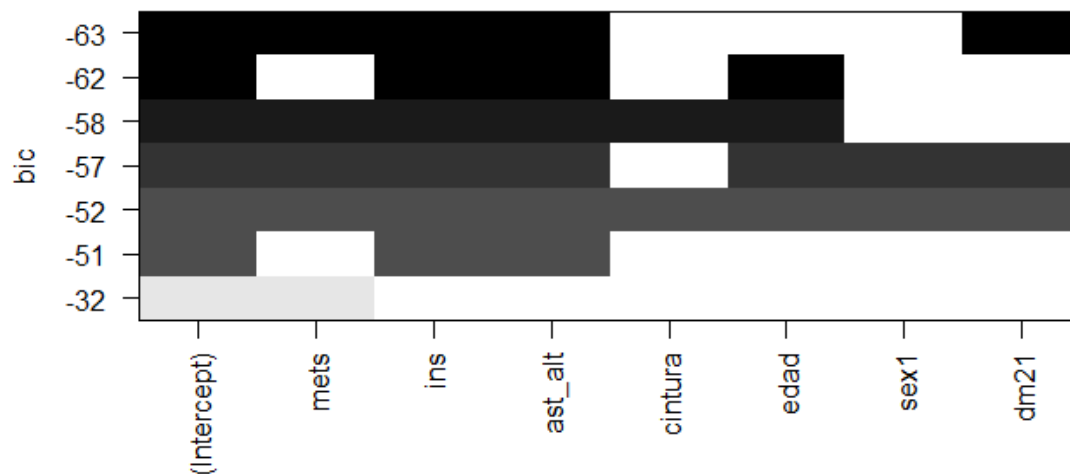


Figura 3.4: Gráfico de selección de modelo utilizando el BIC como criterio de selección

En la **Figura 3.4** podemos apreciar que el mejor modelo de acuerdo al BIC incluye al índice METS-IR de resistencia a la insulina, el valor de *Insulina* en ayuno, la proporción AST\_ALT, el status de diabetes mellitus tipo 2, *DM2*, y el intercepto, con un  $BIC = -63.0$ . Considerando éste como el mejor modelo, se procede a realizar un ajuste únicamente con las variables seleccionadas en el modelo, obteniendo el siguiente resultado:

Modelo RLM saturado				
Coefficientes	$\hat{\beta}$	Error estándar	Valor t	$p - value$
Intercepto	-1.87285	0.87174	-2.148	0.034561
METS-IR	0.68356	0.25473	2.683	0.008774
Insulina	0.28120	0.08164	3.444	0.000895
AST/ALT	-0.70767	0.12505	-5.659	<0.00001
DM2	0.24964	0.08343	2.992	0.003637
SRE= 0.3507, 84 G.L.	$R^2 = 0.6172$	$R_{adj}^2 = 0.5989$	F-test=33.85	p<0.0001

Tabla 3: Modelo RLM depurado que minimiza el BIC.

El modelo resultante presentado en la **Tabla 3** tiene una  $R_{adj}^2 = 0.5989$ , que es ligeramente superior al modelo completo. Para evaluar si el modelo ajusta mejor que el que incluye a todas las variables, se decidió hacer una comparación con BIC de ambos modelos. El modelo saturado tiene un  $BIC_{sat} = 98.73$ , mientras que el modelo depurado tiene un  $BIC_{dep} = 87.85$ . Como podemos ver, la diferencia entre ambos BIC ( $= BIC_{sat} - BIC_{dep}$ ) es considerable ( $\Delta BIC = 10.88$ ), por lo que puede determinarse que el modelo depurado es superior e incluye al intercepto, **METS\_IR**, **Insulina**, la proporción **AST\_ALT** y **DM2**. Como éste modelo se considera el modelo final, se realizará la validación de supuestos para verificar que éste modelo y sus inferencias sean válidas.

## 2.2. Validación de supuestos

### 2.2.1. Supuesto de linealidad

Para validar el supuesto de linealidad, realizaremos pruebas de falta de ajuste en cada uno de los coeficientes continuos, utilizando términos cuadráticos para cada variable incluida en el modelo depurado, como se confirma en la **Tabla 4**.



Pruebas de falta de ajuste			
Variable	Suma de cuadrados	F-value	$p - value$
$METS - IR^2$	0.046956	0.3789	0.5399
$Insulina^2$	0.15895	1.2969	0.2581
$AST/ALT^2$	0.21852	1.7933	0.1842

Tabla 4: Pruebas de falta de ajuste del modelo RLM depurado.

Como se puede apreciar en la tabla precedente, ninguna de las variables viola el principio de linealidad, por lo que puede validarse éste supuesto para nuestro modelo.

### 2.2.2. Supuesto de homoscedasticidad de varianza e independencia de los errores

Para validar el principio de homoscedasticidad de varianza se utilizará la prueba de Breusch-Pagan, así como la evaluación de gráficos de los residuos para comprobar la independencia de los errores. Al realizar la prueba sobre el modelo se identifica un valor del estadístico de Breusch-Pagan de 0.7757 que corresponde a un  $p - value = 0.9417$ . Dado que la prueba no rechaza la hipótesis nula de homoscedasticidad de varianza para el modelo, puede determinarse que se cumple éste supuesto. En los gráficos de residuos vs. variables y valores ajustados se observan gráficos nulos por lo que hay homoscedasticidad de varianza y no hay dependencia en los errores (**Figura 3.5**).

### 2.2.3. Matriz de diseño de rango completo

Para validar éste supuesto, se evaluó la colinearidad las variables en el modelo completo y en el modelo final, para validar que la colinearidad no influyó en la selección del modelo y que ésta no está presente en el modelo final. Para ello se llevó a cabo cálculo del estadístico  $\kappa = 67.85415$  y de los VIF para el modelo depurado ( $AST/ALT=1.087885$ ,  $Insulina=1.536629$ ,  $METS-IR=1.699478$  y  $DM2=1.259042$ ) y el modelo saturado ( $METS-IR=2.406767$ ,  $Insulina=1.819004$ ,  $AST/ALT=1.177854$ ,

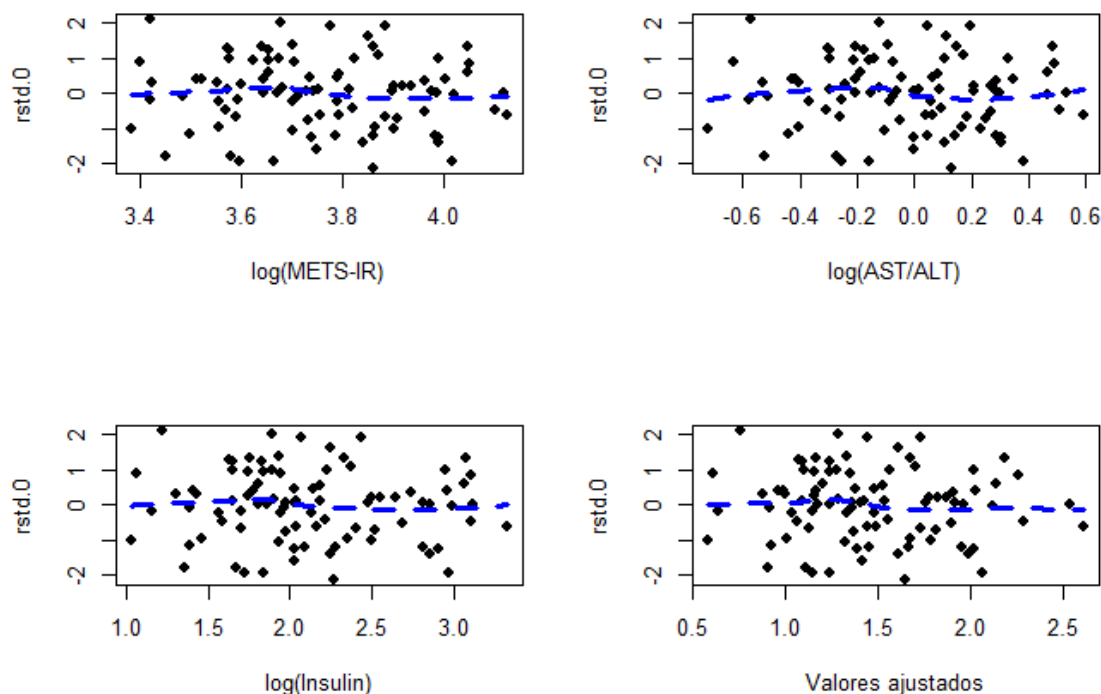


Figura 3.5: Gráfico de residuos vs. variables y valores ajustados del modelo.

Cintura=1.627944, Edad=2.287287, Sexo=1.247851, DM2=2.223877). Como podemos corroborar, el valor del índice  $\kappa$  es bajo y todos los VIF son menores a 5, por lo que puede establecerse que el modelo cumple el supuesto y no hay evidencia de multicolinealidad en el modelo. Además, al comparar los VIF del modelo completo comparado con el depurado, podemos confirmar que no se realizó selección del modelo basado en multicolinealidad.

#### 2.2.4. Normalidad conjunta de los errores

Para evaluar el supuesto de normalidad conjunta de los errores primero exploraremos gráficamente los residuos del modelo RLM depurado. Al realizar la prueba de Anderson-Darling sobre los errores se identifica un estadístico de Anderson-Darling DE 0.34032, que corresponde a un  $p - value = 0.4898$ .

Como puede observarse en los gráficos previos (**Figura 3.6**) y los resultados de

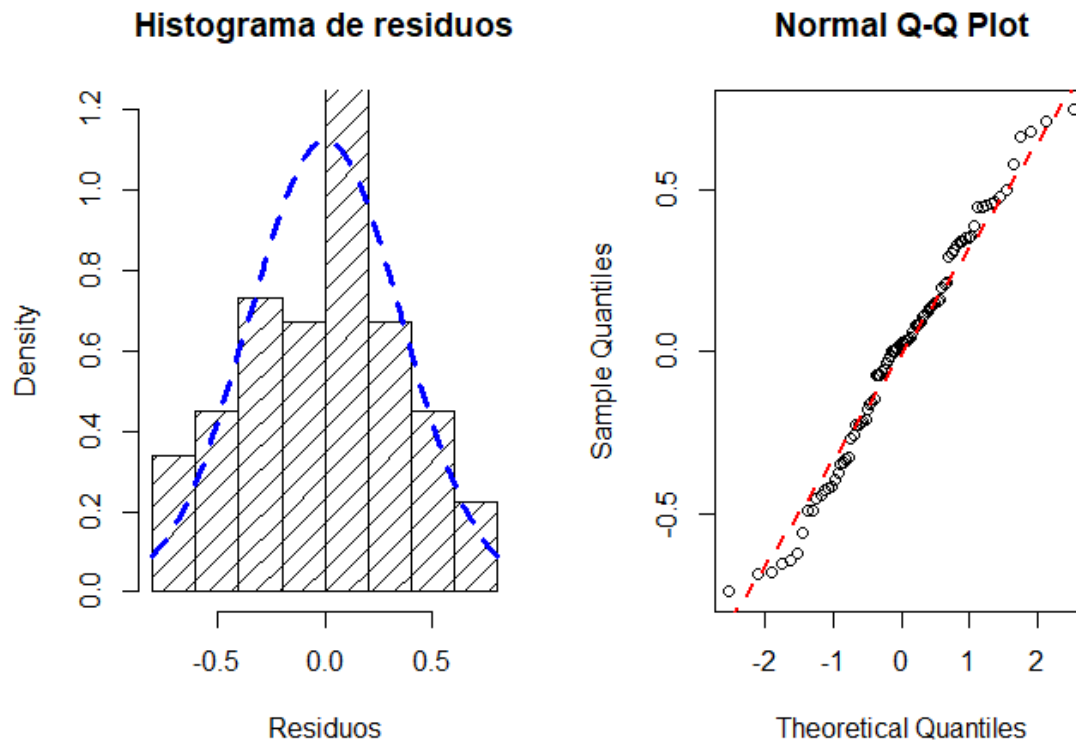


Figura 3.6: Histograma y gráfico cuantil-cuantil de los errores del modelo depurado.

la prueba de Anderson-Darling, no hay evidencia de que los residuos no sigan una distribución normal, por lo que puede determinarse que el modelo cumple con el supuesto de normalidad conjunta de los errores.

### 2.3. Análisis final del modelo

En la **Sección 2.2** se demostró que el modelo depurado además de minimizar el BIC cumple con todos los supuestos del modelo RLM. Ahora, para el análisis final del modelo, se evaluó si los intervalos de confianza simultáneos de los coeficientes son significativos cuando se consideran en conjunto utilizando el método de Hotelling-Scheffé, dado que estimaremos la simultaneidad en 5 coeficientes incluyendo al intercepto.

Intervalos de confianza simultáneos al 95 %			
Variable	Estimador	Límite inferior	Límite superior
Intercepto	-1.95930	-4.23730	0.31870
AST/ALT	-0.70614	-1.04708	-0.36520
Insulina	0.37561	0.15909	0.59213
METS-IR	0.67645	0.01669	1.33621
DM2	0.13754	-0.07733	0.35241

Tabla 5: Intervalos de confianza simultáneos al 95 % para los coeficientes del modelo RLM depurado.

Se puede observar en la **Tabla 5** que, considerando todos los predictores de forma simultánea, solo el intercepto no mantiene significancia con un  $\alpha = 0.05$ . Para comprobar la reproducibilidad del modelo, lo aplicaremos en los datos de validación en la **sección 5** del presente capítulo, donde lo compararemos con los otros dos modelos que ajustaremos a continuación.

### 3. Modelo *Gradient Boosting Machine*

Se llevó a cabo el ajuste de modelos GBM utilizando las 81 combinaciones de hiperparámetros descritos en la sección de **Metodología** mediante *cross-validation*. A continuación, se presentan los resultados de los 10 mejores modelos ajustados con los hiperparámetros elegidos, utilizando como criterio de jerarquización la minimización del RMSE.

Hiperparámetros óptimos para el modelo GBM por <i>cross-validation</i>					
Shrinkage	Max.depth	Min.Obs.	B. fraction	Trees	RMSE.min
0.4	3	10	0.65	12	0.3187722
0.4	5	10	0.65	12	0.3187722
0.4	3	7	0.80	12	0.3285613
0.2	1	5	0.65	35	0.3476695
0.4	5	10	0.80	12	0.3479147
0.4	1	7	0.65	8	0.3538526
0.2	3	10	0.65	37	0.3582650
0.2	5	10	0.65	37	0.3582650
0.2	3	7	0.80	19	0.3589927
0.4	3	7	0.65	8	0.3619064

Tabla 6: Hiperparámetros de los 10 mejores modelos GBM obtenidos de la matriz de optimización de parámetros jerarquizados por minimización del RMSE.

Una vez realizada la búsqueda (**Tabla 6**), concluimos que un `shrinkage=0.40`, `max_depth=3`, `min_obs=10`, `b.fraction=0.65` y `optimal_trees=12` son los hiperparámetros óptimos para minimizar el RMSE, por lo cual ajustamos el modelo corregido. Para mejorar el cálculo del error de predicción, se iteró el modelo mediante *cross-validation*, `k=12`. Tras este procedimiento de 12 iteraciones se identificó que la mejor iteración por *cross-validation* fue la 8; se identificaron 6 predictores cuya influencia en el resultado fue mayor a cero y el RMSE mínimo calculado de las 12 iteraciones fue  $RMSE = 0.4013309$  (**Figura 3.7**). El RMSE equivalió a un valor mayor al obtenido con los hiperparámetros pero vale la pena comentar que en ésta iteración se utilizó toda la muestra y no el 75 % como en la evaluación preliminar de hiperparámetros.

Una vez elegido el modelo GBM final, se evaluó la influencia relativa de cada una de las variables tomando como partida el desempeño condicional del modelo excluyendo la variable en diferentes iteraciones.

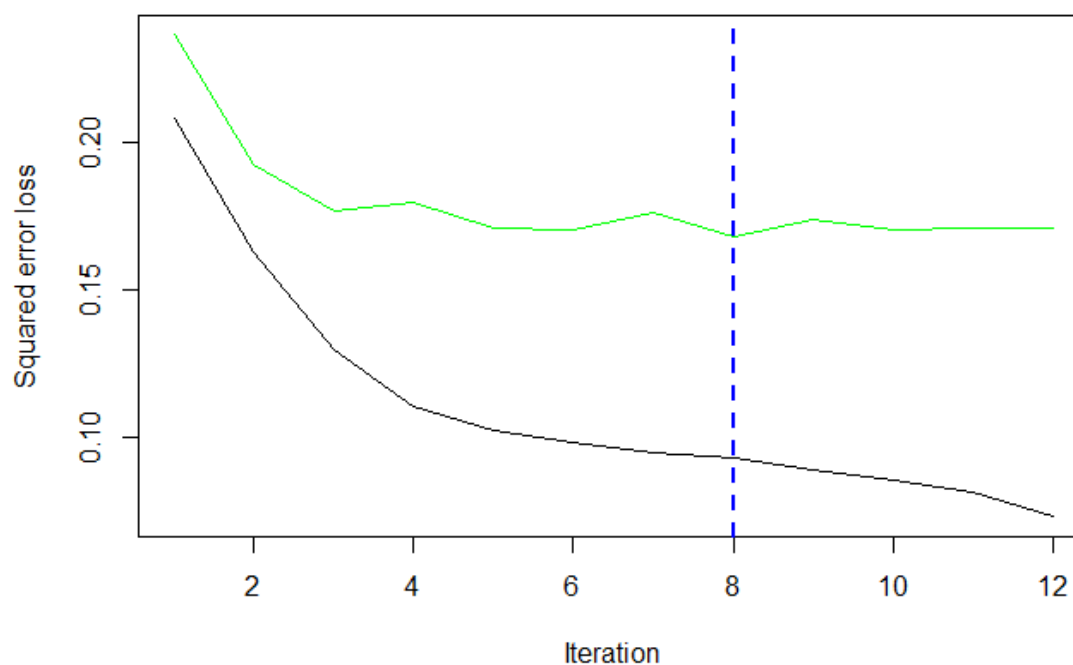


Figura 3.7: Desempeño en *cross-validation* del modelo GBM, indicando la mejor iteración del modelo tras 12 validaciones.

Influencia relativa de las variables en el modelo GBM	
Variable	Influencia relativa
AST/ALT	33.3328565
METS-IR	29.6939008
Insulina	22.9084317
Edad	7.0945843
DM2	5.7257230
Sexo	0.6703748
Cintura	0.5741288

Tabla 7: Influencia condicional de las variables en el modelo GBM.

Como puede observarse en la **Figura 3.8** y en la **Tabla 7**, la proporción AST\_ALT es la variable con mayor influencia sobre la predicción de Grasa\_hepatica, seguida

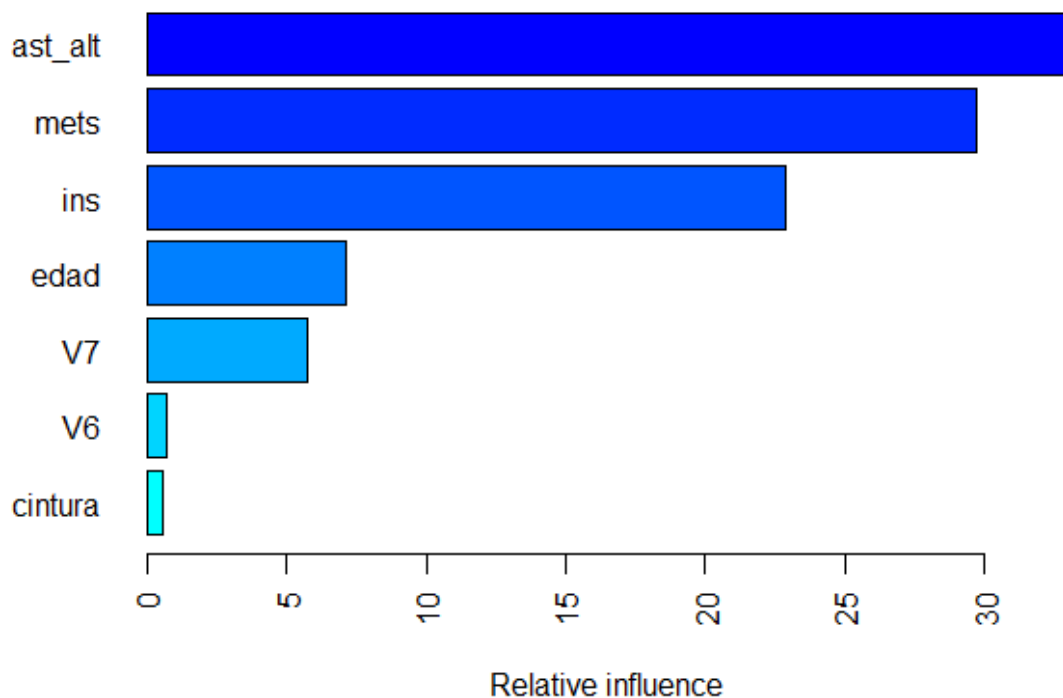


Figura 3.8: Influencia relativa de las variables incluidas en el modelo GBM, V6 representa DM2 y V7 representa Sexo.

por el índice de resistencia a la insulina `METS_IR` y los valores de `Insulina` de ayuno. En contraste con el modelo RLM, el modelo GBM encontró contribución significativa por parte de la `Edad` por sobre `DM2`, indicando la fortaleza del modelo para encontrar relaciones no lineales entre las variables. Al igual que con el modelo RLM, se comparará en la base de validación el desempeño del modelo en la sección **Comparación de modelos**.

## 4. Modelo de Redes neuronales auto-normalizables

La arquitectura de la red neuronal y los parámetros a entrenar por la interfaz `keras` en `R` consideran un modelo con una capa densa de 8 neuronas de entrada, una capa oculta densa con función de activación `SeLU` compuesta por 64 neuronas, una capa de `alpha dropout` para eliminación de parámetros redundantes de 8 neuronas y una capa de salida con una función de activación `SoftPlus`.

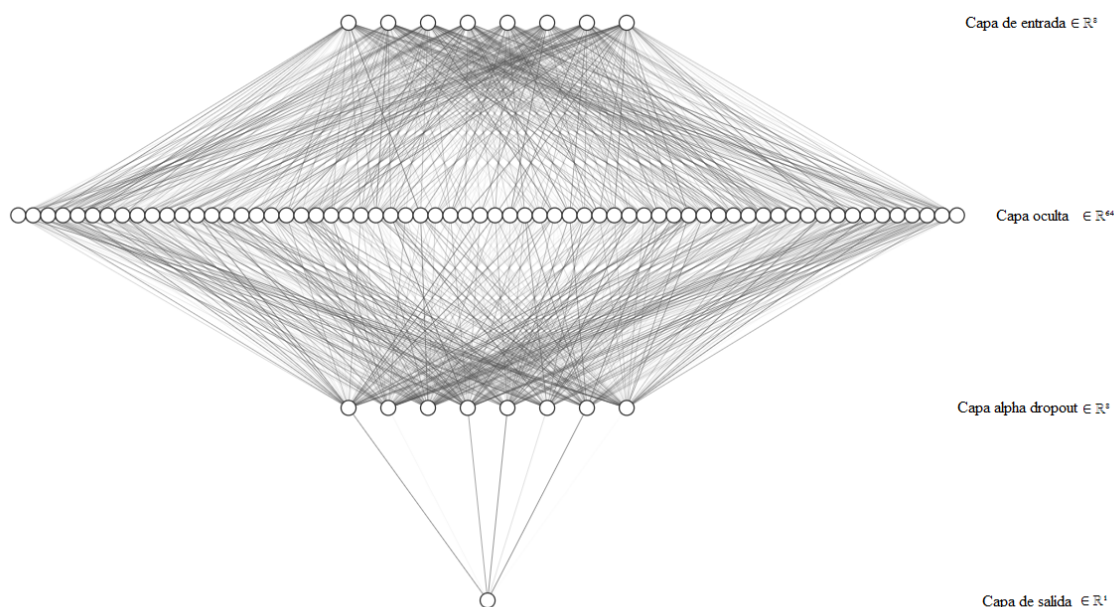


Figura 3.9: Representación gráfica de la red neuronal totalmente conectada utilizando SNNN con el paquete `keras` de `R`

### 4.1. Cálculo de la métrica de error

Una vez procesado el modelo SNNN, se evaluó el resultado del modelo entrenado en la última época en una submuestra de 67 de la base de entrenamiento que representan el 75 % de los casos totales de entrenamiento durante *cross-validation* y validadas en el 25 % restante. El modelo fue iterado durante 50 épocas tras las cuales se encontró una función de pérdida de 0.1123 con un  $MAE = 0.2653$  en la submuestra de entrenamiento y una función de pérdida de 0.1664 con un  $MAE = 0.3091$



en la submuestra de validación dentro de la muestra completa de entrenamiento. La representación gráfica del modelo se presenta en la **Figura 3.9**.

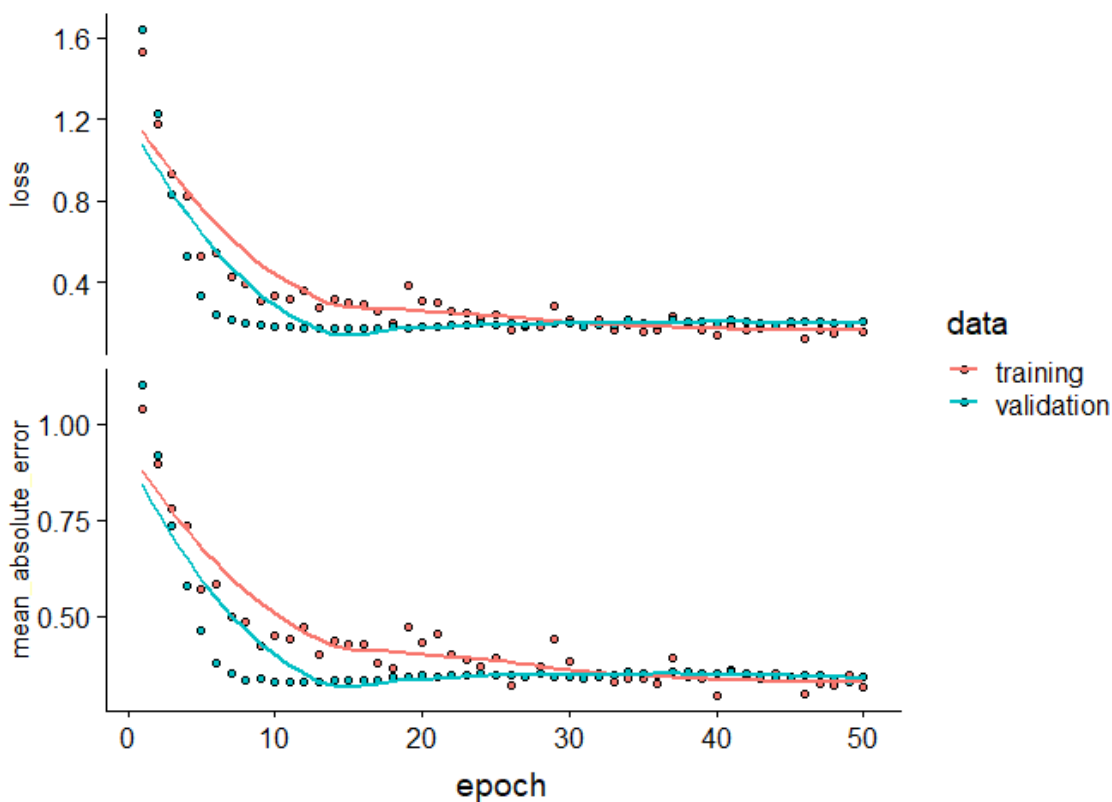


Figura 3.10: Gráfico del error de entrenamiento y la función de pérdida del modelo SNN por k-fold *cross-validation* en la base de entrenamiento y validación.

Como podemos ver en la **Figura 3.10**, identificamos que con  $n = 50$  épocas lográbamos un MSE relativamente bajo. Se observa en la época 50 que el MAE de la base de entrenamiento con todos los datos debería aproximar 0.265 y el MSE (función de pérdida) debería aproximar 0.309 como se describió anteriormente. Para validar éstas observaciones, ahora procedemos a entrenar el modelo en la base completa, entrenando los parámetros por 50 épocas.

Posteriormente, se entrena la red en todos los datos de la base de entrenamiento, omitiendo la separación generada para *cross-validation*. El MAE del modelo SNN ajustado en los datos de entrenamiento completos fue de 0.276 y la función de pérdida de 0.126. Para evaluar el desempeño en la cohorte de validación, comparamos los modelos utilizando la función `evaluate` de `keras`. Podemos observar que en las 37

muestras de validación se obtiene una función de pérdida de 0.1544 y un  $MAE = 0.3227$ . Podemos concluir que tanto el MSE como el MAE obtenidos en la cohorte de entrenamiento y validación por *cross-validation* aproximan razonablemente los MSE y MAE observados, por lo que determinamos que el modelo SNNN tiene validez externa y puede aplicarse para la comparación de modelos con el modelo RLM y GBM.

## 5. Comparación de modelos

Una vez ajustados los modelos RLM, GBM y SNNN, se procedió a evaluar las métricas de error y los coeficientes de correlación y determinación de cada uno en la base de validación ( $n = 38$ ).

Comparación de modelos					
Modelo	MAE	RMSE	MSE	Correlación ( $r$ )	$R^2$
RLM	2.845	4.088	16.711	0.686	0.471
GBM	2.858	3.878	15.041	0.665	0.442
SNNN	2.661	3.554	12.631	0.703	0.494

Tabla 8: Comparación de los modelos RLM, GBM y SNNN en cuanto a métricas de error y correlación lineal en la cohorte de validación  $n = 38$ .

Como podemos observar en el análisis de los modelos previos (**Tabla 8**), el MAE, RMSE y MSE más bajos se observaron para el modelo SNNN indicando un menor error de estimación en todos los casos. El modelo RLM tuvo MAE más bajo aunque RMSE y MSE más alto comparado con el modelo GBM. Esto nos podría indicar que el modelo RLM tiene más valores atípicos altos, los cuales causan cambios más significativos en el RMSE y el MSE debido que se elevan al cuadrado los residuales en lugar de tomar su valor absoluto, como en el caso del MAE.

Para investigar las discrepancias en los valores predichos en la base de validación, se realizaron histogramas sobrepuestos de los valores predichos con respecto a los reales (**Figura 3.11**). Como puede apreciarse en la figura, los valores estimados por SNNN

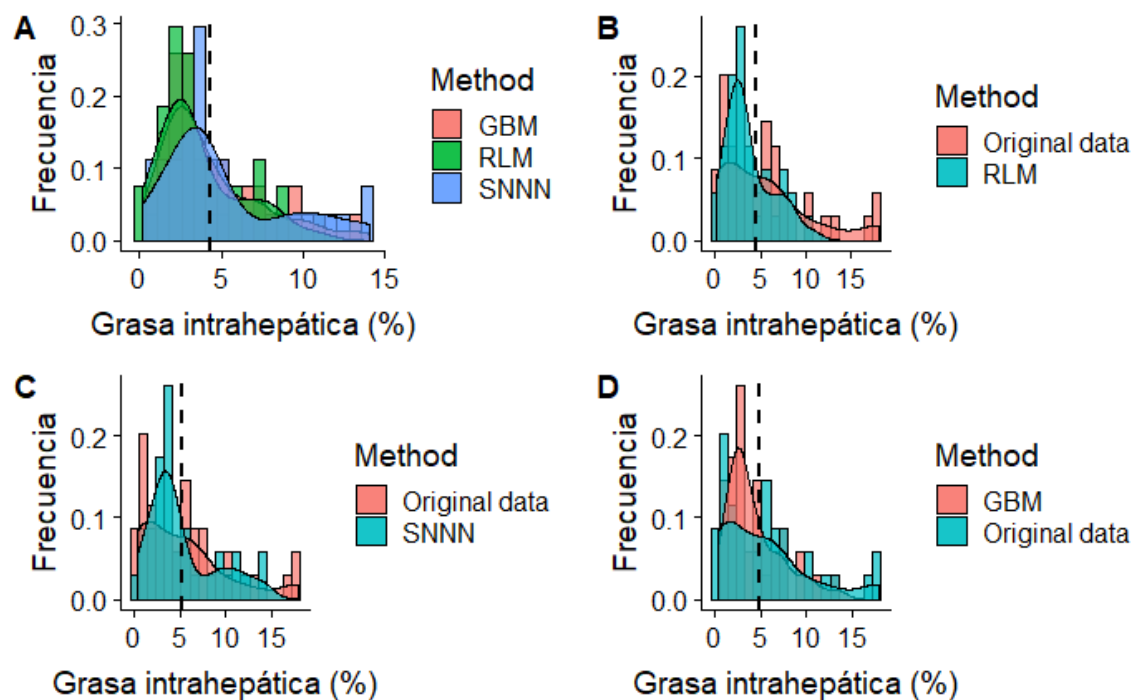


Figura 3.11: Histogramas sobrepuestos de los valores predichos por el modelo RLM, GBM y SNNN.

tienen menos sesgo positivo y mayor cobertura hacia valores altos comparado con los modelos RLM y GBM. Al comparar la varianza de los valores predichos, observamos una menor varianza para el estimador RLM con  $V(X) = 6.626$ , seguido por el modelo GBM con  $V(X) = 9.379$  y los valores predichos con mayor varianza fueron para el modelo SNNN con  $V(X) = 14.47$ .

### 5.1. Análisis de Bland-Altman de los modelos

Para cuantificar el sesgo de las 3 mediciones de los modelos RLM, SNNN y GBM se realizó un análisis de Bland-Altman utilizando el paquete `blandr` de R. Los resultados comparativos de los 3 modelos se presentan a continuación:

Comparación de modelos			
Modelo	$\bar{d}$ , IC95 %	LoA Superior, IC95 %	LoA inferior, IC95 %
RLM	1.70 (0.45 - 2.96)	9.09 (6.92 - 11.25)	-5.68 (-7.85 - -3.51)
GBM	0.24 (-0.96 - 1.43)	7.28 (5.21 - 9.35)	-6.81 (-8.88 - -4.74)
SNNN	1.10 (-0.16 - 2.35)	8.49 (6.32 - 10.66)	-6.30 (-8.46 - -4.13)

Tabla 9: Comparación de los modelos RLM, GBM y SNNN mediante un análisis de Bland-Altman en la base de validación  $n = 38$ .

Como podemos ver en la **Tabla 9**, el modelo SNNN tiene el menor sesgo y el intervalo de predicción más estrecho, seguido por el modelo GBM y el modelo RLM. Esto sugiere mayor precisión en la estimación mediante SNNN. El sesgo del modelo GBM fue más cercano a 0 indicando mayor exactitud, aunque el intervalo de predicción sea más amplio que el modelo SNNN.

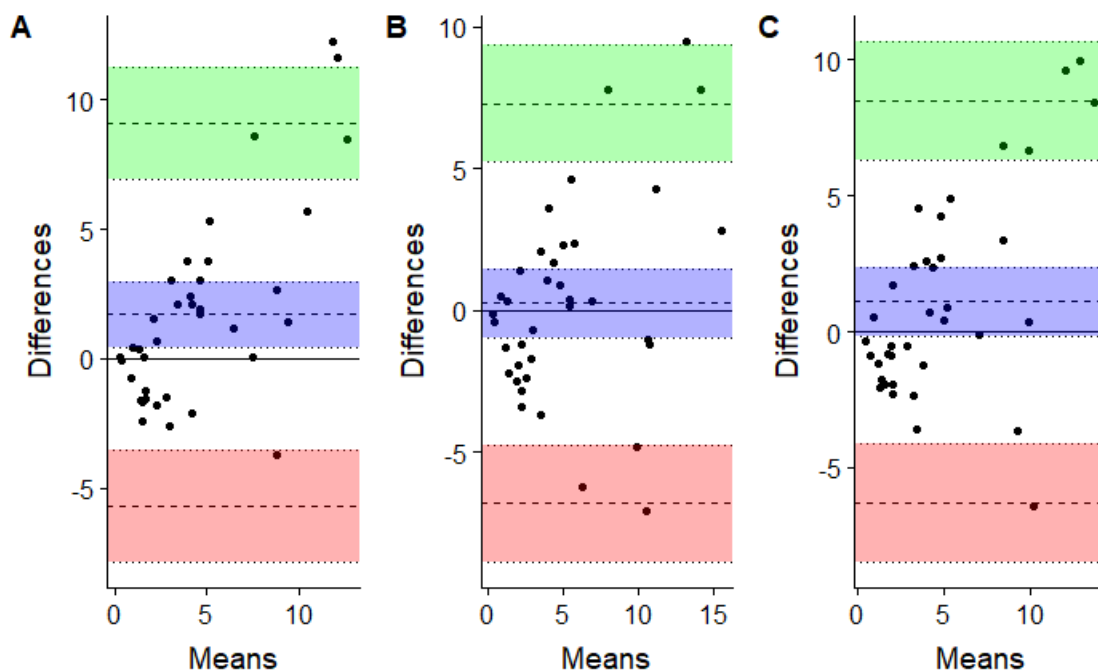


Figura 3.12: Gráficos de Bland-Altman para evaluar el grado de acuerdo y reproducibilidad de los modelos RLM (A), SNNN (B) y GBM (C).

Como puede comprobarse en la **Figura 3.12**, la reproducibilidad de los 3 modelos evaluados es adecuada. Únicamente el modelo RLM tiene 2 observaciones fuera

del intervalo de confianza del límite de acuerdo superior, sugiriendo escenarios de subestimación de la grasa intrahepática utilizando el modelo RLM con valores estimados de grasa hepática mayores a 10 %. Finalmente, evaluamos el ajuste lineal de las predicciones de los tres modelos en conjunto con los valores originales de grasa intrahepática de la base de validación.

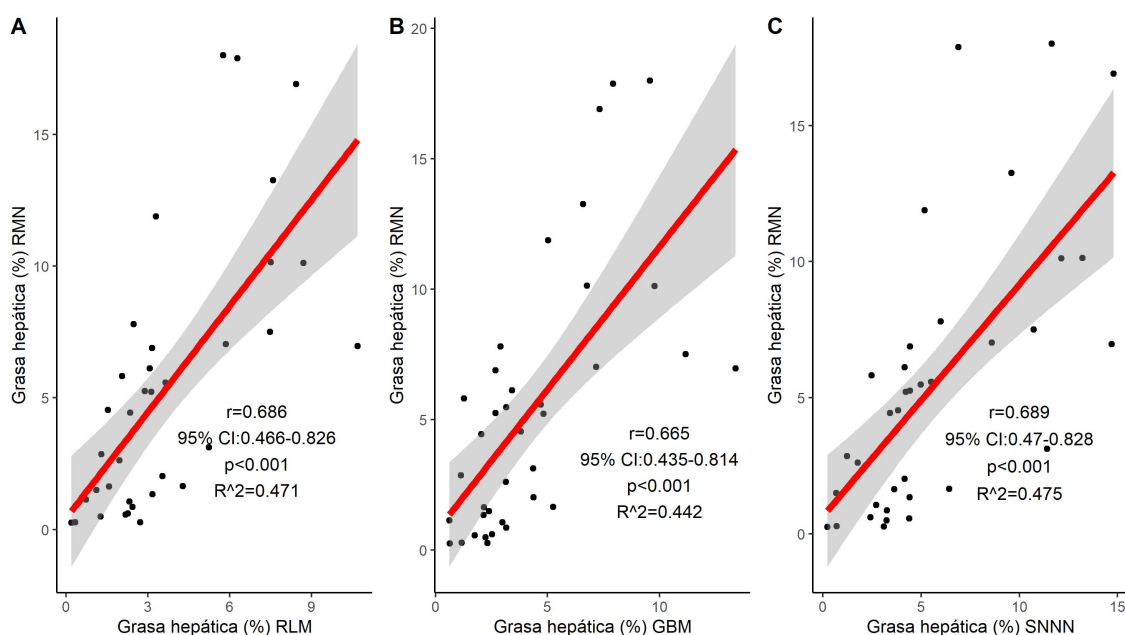


Figura 3.13: Gráficos de dispersión y ajuste lineal con su respectivo intervalo de predicción al 95 % para los modelos RLM (A), GBM (B) y SNNN (C) en comparación con los valores de grasa intrahepática medidos por resonancia magnética.

Como se aprecia en la **Figura 3.13**, el ajuste lineal difiere para cada uno de los modelos. De forma importante, el intervalo de predicción es más estrecho y la  $R^2$  más alta para el modelo SNNN. El comportamiento de los modelos GBM y RLM sugieren desempeño similar a lo observado en los análisis previos.

## 5.2. Evaluación de los modelos para predicción de EHGNA

Finalmente, se evaluó la utilidad clínica de los estimadores en el diagnóstico de EHGNA, para demostrar su aplicabilidad en un contexto clínico donde sea posible la toma de decisiones basada en la información arrojada por los modelos. Para la comparación de los modelos se utilizó la curva Característica Operativa del Receptor

o ROC, por sus siglas en inglés y métricas derivadas de la matriz de confusión. Como podemos observar en la **Tabla 10**, el modelo con el AUROC más alto fue el modelo RLM, el cual también tuvo los valores más altos de sensibilidad y especificidad. El modelo SNNN tuvo el peor desempeño para clasificación, aunque las métricas de la matriz de confusión fueron similares con el modelo GBM.

Comparación de modelos					
Modelo	AUROC, IC95 %	$F_1(t)$ , IC95 %	$1 - F_0(t)$ , IC95 %	VPP, IC95 %	VPN, IC95 %
RLM	0.882 (0.772-0.991)	0.60 (0.32-0.84)	1.00 (0.85-1.00)	1.00 (0.69-1.00)	0.79 (0.54-1.00)
GBM	0.867 (0.736-0.997)	0.73 (0.45-0.99)	0.91 (0.71-0.99)	0.85 (0.57-0.96)	0.83 (0.60-0.98)
SNNN	0.851 (0.736-0.997)	0.73 (0.45-0.99)	0.91 (0.71-0.99)	0.85 (0.57-0.96)	0.83 (0.60-0.98)

Tabla 10: Comparación de los modelos RLM, GBM y SNNN con respecto a las AUROC y las métricas derivadas de la matriz de confusión en la cohorte de validación  $n = 38$ .

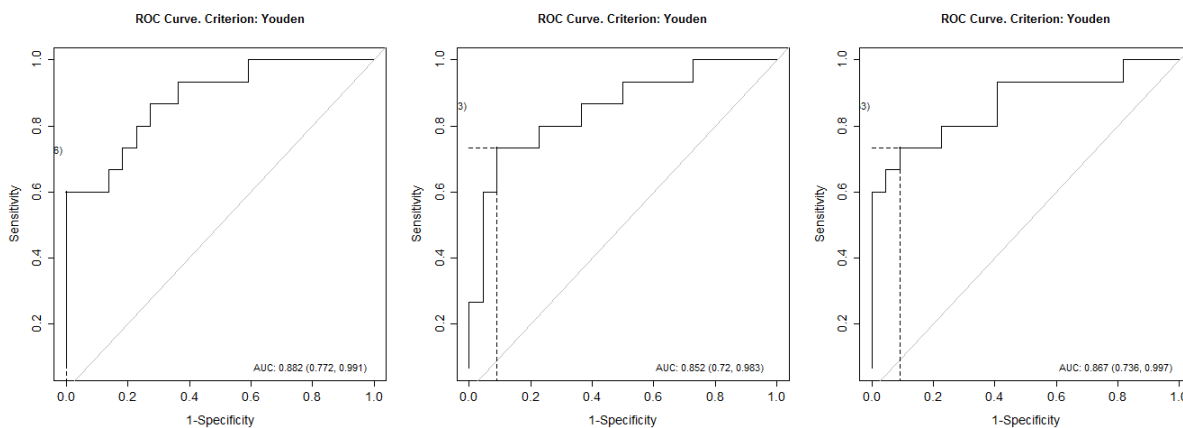


Figura 3.14: Curvas ROC para evaluar el desempeño diagnóstico de los modelos evaluados en la base de validación,  $n = 38$ .

A pesar de las diferencias observadas entre los modelos, considerando el tamaño muestral de la base de validación se observa que los intervalos de confianza al 95 % de

todas las métricas se sobreponen (**Tabla 10**) y sus desempeños son similares (**Figura 3.14**), lo que no permite evidenciar diferencias significativas en clasificación para los modelos. Otra forma de evaluar los modelos consiste en la exploración de falsos positivos y falsos negativos, lo cual da un indicativo de su desempeño en aplicación clínica:

- Para el modelo RLM se clasificó a 6 individuos como falsos positivos y a ninguno como falso negativo, el punto de corte identificado por el índice de Youden para mejorar la clasificación fue de 5.76.
- Para el modelo SNNN se identificó a 2 falsos positivos y 4 falsos negativos, el punto de corte óptimo identificado por el índice de Youden fue 5.87.
- Para el modelo GBM se identificó a 2 falsos positivos y 4 falsos negativos, el punto de corte identificado para mejorar la clasificación por índice de Youden fue 4.71.

Los puntos de corte identificados para cada modelo también difieren entre ellos. En general, los puntos de corte identificados para el modelo RLM y SNNN tienen mayor similitud al utilizado en la práctica clínica, mientras que el identificado para el modelo GBM es comparativamente menor. Esto podría llevar a sobreestimar muchos casos con el modelo GBM, lo cual disminuiría su precisión clínica.

# Capítulo 4

## Discusión y Conclusiones

Como se pudo evaluar en los análisis previos, cuando se cuenta con un número limitado de casos como el nuestro debido a la complejidad metodológica y el costo de la aplicación de la MRI-S, es factible la estimación de grasa intrahepática por diversos métodos de estimación. El advenimiento de la aplicación de modelos de aprendizaje de máquina y *deep learning* en el contexto de la inteligencia artificial ha generado un sinnúmero de aplicaciones en diversas áreas, con aplicaciones incipientes en el área de la medicina y en particular en el metabolismo [14]. Éstas metodologías permiten estimaciones que identifican relaciones no lineares, en datos donde la validación de supuestos puede no ser factible [16, 20]. Basado en éstas nociones, el presente trabajo exploró de diferentes maneras las ventajas y desventajas de los modelos RLM, GBM y SNN para la estimación de grasa intrahepática en población mestiza mexicana, una población con elevada susceptibilidad al desarrollo de complicaciones metabólicas.

### Conclusiones del modelo RLM

El modelo RLM ajustado pudo ser validado por todos los supuestos del modelo RLM; ésta validación asegura que las inferencias poblacionales sean adecuadas y garantiza el desempeño del modelo cuando hay observaciones nuevas previamente no evaluadas por el modelo [15]. Como pudo apreciarse en los resultados comparativos entre modelos, el modelo RLM tuvo errores de predicción más bajos y coeficiente



de correlación comparado con GBM. Estas métricas indican que el modelo RLM es preciso, aunque tiene una elevada propensión a estimación con valores atípicos; sin embargo, la varianza de los valores estimados fue la menor comparada con GBM y SNNN. Al evaluar su desempeño en aplicación clínica utilizando como métricas la AUROC y la matriz de confusión, el modelo RLM tuvo el mejor desempeño al ser el modelo con mayor sensibilidad de entre los tres evaluados. El modelo RLM sería una buena herramienta de tamizaje, aunque con propensión a sobreestimar el riesgo de algunos pacientes. Además, debido a la relativa facilidad de su implementación y el bajo costo computacional, así como el menor número de términos involucrados puede considerarse el modelo más parsimonioso para estimación de grasa intrahepática.

## Conclusiones del modelo GBM

El modelo GBM tiene una implementación basada en la generación de múltiples modelos de bajo aprendizaje que en conjunto generan un modelo con mayor capacidad predictiva [16]. En su implementación en el presente trabajo, el modelo GBM tuvo un adecuado desempeño; cabe resaltar que en comparación con el modelo RLM el modelo GBM fue menos parsimonioso al incluir los 8 predictores en su estimación. Al comparar el modelo GBM con el modelo RLM y SNNN se observó que el sesgo fue bajo y todos los valores predichos cayeron dentro de los límites de acuerdo, demostrando que es un modelo consistente y reproducible. Al compararlo en su desempeño para el diagnóstico de EHGNA, se observó el punto de corte más bajo predicho por índice Youden comparado con el resto de los modelos. Esto sugiere que el modelo GBM podría no ser de mucha utilidad para pacientes en valores intermedios de grasa intrahepática y su uso podría afectar en la decisión clínica de establecer el diagnóstico de EHGNA. El modelo GBM puede ser optimizado al evaluar con mayor precisión un mayor número de hiperparámetros [17, 18]; sin embargo, esta optimización es computacionalmente demandante y requiere de mayores tamaños muestrales. Debido a estas limitaciones, su uso en el conjunto de datos evaluados no fue el óptimo.

## Conclusiones del modelo SNNN

El modelo SNNN es una alternativa recientemente desarrollada para la aplicación de redes neuronales completamente conectadas en tareas de regresión y clasificación [20]. Su desempeño permite la optimización de la predicción aunque con alto riesgo de sobreajuste en muestra pequeñas como la evaluada en el presente estudio. Al comparar el modelo SNNN con el modelo GBM y RLM, el modelo SNNN tuvo el mejor desempeño para tareas de regresión, lo que hace que el modelo SNNN tenga la mayor precisión para la estimación de grasa intrahepática. Sin embargo, al evaluar la compensación sesgo-varianza, que representa un paradigma crucial en el análisis de modelos de aprendizaje de máquina en la actualidad nos encontramos con resultados opuestos. Como evidencia, en el análisis de Bland-Altman el modelo SNNN tiene el menor sesgo y el intervalo de predicción más estrecho comparado con el resto de los modelos; sin embargo esto ocurre a expensas del estimador de más alta varianza, lo cual es un indicativo de sobreajuste o modelaje del ruido en los datos [20, 27]. Esta minimización del sesgo a expensas de aumentos en la varianza ocurre cuando aumenta la profundidad o número de neuronas de las redes ajustadas. Para generar una red que minimice la varianza, es posible experimentar con modelos con diferentes profundidades, inicializadores y número de neuronas, lo cual puede lograr estimadores insesgados de mínima varianza [27]. El resultado de la elevada varianza es un menor desempeño en el diagnóstico clínico de EHGNA, con un AUROC más bajo con respecto a los modelos RLM y GBM. El modelo SNNN ajustado podría ser de utilidad cuando el objetivo de interés sea la estimación con mayor precisión de grasa intrahepática, sobre todo considerando que tiene un mayor número de parámetros comparado con el modelo RLM. La implementación de modelos SNNN funciona mejor en el contexto de bases de datos grandes por lo que el hecho de no ser evidentemente superior en comparación con RLM en el conjunto de datos evaluados no es motivo de sorpresa, sino un indicio de que en bases de datos con tamaños modestos su uso no mejora significativamente la capacidad predictiva.

## Perspectivas

La validación de los modelos para predicción de grasa intrahepática mediante la metodología de aprendizaje de máquina sugiere que su validez externa es significativa, lo cual puede implicar un beneficio clínico en su aplicación. Debido a la facilidad de implementación y por ser un modelo más parsimonioso el modelo RLM podría ser el modelo con mejor aplicación cotidiana, mientras que el modelo SNNN podría ser de mayor utilidad cuando el objetivo de la evaluación sea tener valores más precisos para la estimación de grasa intrahepática. Para asegurar la validez de los modelos en la estimación de desenlaces clínicos sería de utilidad su validación utilizando el estándar de oro absoluto para diagnóstico de EHGNA, la biopsia hepática. Un desenlace secundario de mucho interés también será evaluar su desempeño en la predicción de fibrosis o acumulación de fibrina en el parénquima hepático, el cual es una complicación crónica de EHGNA que tiene la mayor repercusión clínica por su asociación con cirrosis hepática y mayor mortalidad. La comparación con modelos clínicos existentes también podría enriquecer su aplicación y favorecer su reproducibilidad clínica y en estudios de investigación.

## Complicaciones y limitaciones en el desarrollo del proyecto

Una de las dificultades más notorias de implementar el presente trabajo fue adquirir el dominio de los tres métodos evaluados para tareas de regresión. El modelo RLM fue el menos complicado pues el entrenamiento recibido durante el curso de Regresión Múltiple fue más que suficiente para dominar muchas de las particularidades del modelo. El modelo GBM no fue abordado en el curso y su implementación requirió entender algunos conceptos computacionales y de modelaje que con los que no había tenido contacto previamente. Además, para optimizar aún más el modelo y reducir los errores de predicción, la matriz de hiperparámetros debía expandirse a una magnitud tal que la demanda computacional sería demasiada, limitando nuestra capacidad para generar modelos más precisos. El modelo SNNN lo pude aprender durante mi curso optativo de Redes Neuronales en la Especialización; una complicación

de su implementación fue que la profundidad de las redes generaba mucho modelaje del ruido en los datos, implicando bajo desempeño en las bases de entrenamiento. Esto permitió que aprendiera sobre técnicas de regularización, iteración y ajuste de los modelos predictivos con el objetivo de mejorar su desempeño predictivo, no solo en el caso de SNNN, sino también para GMB y RLM.

Una complicación adicional fue experimentar con modelaje el cual vemos de formas distintas en los cursos de la especialización pero que de cualquier manera representan un reto cuando se llevan a la práctica. Pude hacer uso de mi formación médica y científica a la par de la estadística para abordar algunos problemas como la transformación de variables, depuración de modelos y selección de hiperparámetros. De cierto modo, esta tarea de comparación de modelos hubiera sido más complicada si no la hubiera desarrollado en un tema de mi área pues en muchos momentos me encontré pensando de forma simultánea como médico, científico y especialista en estadística. Fue un ejercicio enriquecedor pero repleto de incertidumbre que me obligó a buscar herramientas de ambos campos para mejorar el modelo y hacer comparaciones objetivas. Aunque fue complicado, estoy convencido de que ha influido positivamente en mi formación profesional.

# Bibliografía

- [1] Stefan N, Häring HU, Cusi K. Non-alcoholic fatty liver disease: causes, diagnosis, cardiometabolic consequences, and treatment strategies. *Lancet Diabetes Endocrinol.* 2019;7(4):313-324.
- [2] Bernal-Reyes R, Castro-Narro G, Malé-Velázquez R et al. The Mexican consensus on nonalcoholic fatty liver disease. *Rev Gastroenterol Mex.* 2019;84(1):69-99.
- [3] Wang XJ, Malhi H. Nonalcoholic Fatty Liver Disease. *Ann Intern Med.* 2018;169(9):ITC65-ITC80.
- [4] Lonardo A, Byrne CD, Caldwell SH et al. Global epidemiology of nonalcoholic fatty liver disease: Meta-analytic assessment of prevalence, incidence, and outcomes. *Hepatology.* 2016;64(4):1388-9.
- [5] Kabbany MN, Conjeevaram Selvakumar PK, Watt K et al. Prevalence of Nonalcoholic Steatohepatitis-Associated Cirrhosis in the United States: An Analysis of National Health and Nutrition Examination Survey Data. *Am J Gastroenterol.* 2017;112(4):581-587.
- [6] Allen AM, Therneau TM, Larson JJ et al. Nonalcoholic fatty liver disease incidence and impact on metabolic burden and death: A 20 year-community study. *Hepatology.* 2018;67(5):1726-1736.
- [7] Bril F, Barb D, Portillo-Sanchez P et al. Metabolic and histological implications of intrahepatic triglyceride content in nonalcoholic fatty liver disease. *Hepatology.* 2017;65(4):1132-1144.

- [8] Bello-Chavolla OY, Almeda-Valdes P, Gomez-Velasco D et al. METS-IR, a novel score to evaluate insulin sensitivity, is predictive of visceral adiposity and incident type 2 diabetes. *Eur J Endocrinol*. 2018;178(5):533-544.
- [9] Reeder SB, Cruite I, Hamilton G, Sirlin CB. Quantitative Assessment of Liver Fat with Magnetic Resonance Imaging and Spectroscopy. *J Magn Reson Imaging*. 2011;34(4):spcone.
- [10] Bedogni G, Bellentani , Miglioli L et al. The Fatty Liver Index: a simple and accurate predictor of hepatic steatosis in the general population. *BMC Gastroenterol*. 2006;6:33.
- [11] Kotronen A, Peltonen M, Hakkarainen A et al. Prediction of non-alcoholic fatty liver disease and liver fat using metabolic and genetic factors. *Gastroenterology*. 2009;137(3):865-72.
- [12] Buch VH, Ahmed I, Maruthappu M. Artificial intelligence in medicine: current trends and future possibilities. *Br J Gen Pract*. 2018;68(668):143-144.
- [13] Chen JH, Asch SM. Machine Learning and Prediction in Medicine - Beyond the Peak of Inflated Expectations. *N Engl J Med*. 2017;376(26):2507-2509.
- [14] Gubbi S1, Hamet P, Tremblay J, Koch CA, Hannah-Shmouni F. Artificial Intelligence and Machine Learning in Endocrinology and Metabolism: The Dawn of a New Era. *Front Endocrinol (Lausanne)*. 2019;10:185.
- [15] Chatterjee S, Hadi AS. *Regression analysis by example*. 2012, Hoboken: Wiley-Interscience.
- [16] Naketin A, Knoll A. Gradient boosting machines, a tutorial. *Front. Neurobot*. 2013; 7:21.
- [17] Freund Y, Schapire R. A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci*. 1997, 55, 119–139.

- [18] Friedman J. Greedy boosting approximation: a gradient boosting machine. *Ann Stat*. 2001, 29, 1189–1232.
- [19] Wenxin J. On weak base hypotheses and their implications for boosting regression and classification. *Ann Stat* 2002; 30, 51–73.
- [20] Chollet F, Allaire JJ. *Deep learning with R*. 2018; Shelter Island, NY: Manning Publications.
- [21] Klambauer G, Unterthiner T, Mayr A, Hochreiter S. Self-Normalizing Neural Networks. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. arXiv:1706.02515v5 [cs.LG]
- [22] Bland JM, Altman DG. Measuring agreement in method comparison studies. *Stat Methods Med Res*. 1999;8(2):135-60.
- [23] Hand DJ. Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning*. 2009; 77(1):103-123
- [24] Schisterman EF, Perkins NJ, Liu A, Bondell H. Optimal cut-point and its corresponding Youden Index to discriminate individuals using pooled blood samples. *Epidemiology*. 2005;16(1):73-81.
- [25] Visa S, Ramsay B, Ralescu AL, van der Knaap R. Confusion Matrix-based Feature Selection. *MAICS*. 2011.
- [26] Cuperlovic-Culf M. Machine Learning Methods for Analysis of Metabolic Data and Metabolic Pathway Modeling. *Metabolites*. 2018;8(1). pii: E4. doi: 10.3390/metabo8010004.
- [27] Neal B, Mittal S, Baratin S, Tantia V, Scicluna M, Lacoste-Julien S, Mitliagkas I. A Modern Take on the Bias-Variance Tradeoff in Neural Networks. arXiv:1810.08591 [cs.LG]

# Anexo: Código de R

A continuación, se presenta en extenso el código utilizado para los análisis incluidos en el presente trabajo de investigación. El código y su renderización se encuentran también disponibles en un repositorio digital: [https://github.com/oyaxbell/AI.metabolisMX/tree/master/tesina\\_EAA](https://github.com/oyaxbell/AI.metabolisMX/tree/master/tesina_EAA)

```
library(corrplot); library(corr)
library(PerformanceAnalytics)
library(mice); library(randtests); library(nortest)
library(BlandAltmanLeh); library(rsample)
library(gbm); library(caret); library(h2o)
library(ggplot2); library(lme); library(keras)
library(car); library(leaps); library(MASS)
library(lmtest); library(blandr)
#Cargamos los datos
livergbm<-read.csv("livergbm.csv")
# Analisis de regresion lineal multiple
correlacion_data<-cor(livergbm[c(3:7, 10)])
corrplot(correlacion_data, method="circle")
chart.Correlation(livergbm[c(3:7, 10)], histogram=TRUE, pch=19)
cor(livergbm[c(3:7, 10)])

## Transformacion de los datos
ins<-log(livergbm$Insulina); ast_alt<-log(livergbm$AST_ALT)
mets<-log(livergbm$METS_IR); sex<-as.factor(livergbm$Sexo)
```



```
dm2<-as.factor(livergbm$DM2)
ghep<-(livergbm$Grasa_hepatica)^(1/3)
cintura<-log(livergbm$Cintura); edad<-livergbm$Edad
dat<-data.frame(ghep, mets, ins, ast_alt, cintura,edad,
                sex, dm2)
chart.Correlation(dat[,1:6], histogram=TRUE, pch=19)
t.test(ghep[sex==0], ghep[sex==1])
t.test(ghep[dm2==0], ghep[dm2==1])

### Observaciones influyentes
lm<-lm(ghep~., data=dat)
par(mfrow=c(1, 2))
DFFITS<-dffits(lm);
plot(1:nrow(dat), DFFITS, pch = 19, cex = 1,
     xlab="Numero de observacion")
abline(h=2*(sqrt(4/nrow(dat))), col="blue", lwd=3, lty=2)
abline(h=-2*(sqrt(4/nrow(dat))), col="blue", lwd=3, lty=2)
COVRATIO<-covratio(lm)
plot(1:nrow(dat), COVRATIO, pch = 19,
     cex = 1, xlab="Numero de observacion")
abline(h=((3*5)/nrow(dat))+1, col="blue", lwd=3, lty=2)
abline(h=-((3*5)/nrow(dat))+1, col="blue", lwd=3, lty=2)
l1<-influence.measures(lm)
summary(l1)
##Se identificaron 40, 112 y 121 como influyentes

## Separacion de bases: Entrenamiento y validacion.
livergbm<-dat[-c(40, 112, 121),]
set.seed(45)
#Separamos la base general en una de entrenamiento (70%)
#y una de validacion (30%).
liver1 <- initial_split(livergbm, prop = 0.70)
```

```
liver_train <- training(liver1)
liver_test  <- testing(liver1)
head(liver_train)
dat1<-liver_train

## Ajuste y seleccion del modelo en la base de entrenamiento
#Ajustamos modelo inicial
lm1<-lm(ghep~., data=dat1)
summary(lm1)

##Seleccionamos nuevo modelo
library(leaps)
models <- regsubsets(ghep~., data = dat1, method = "seqrep")
m1<-summary(models)
plot(models)
lm2<-lm(ghep~ast_alt+ins+mets+dm2, data=dat1)
summary(lm2)
lm2<-lm(ghep~ast_alt+ins+mets+dm2, data=dat1)
c(BIC(lm1), BIC(lm2)); BIC(lm1) > BIC(lm2)

## Validacion de supuestos
### Supuesto de linealidad
m1<-lm(ghep~mets+ins+ast_alt+dm2+I(mets^2), data=dat1)
anova(m1)[5,]
m2<-lm(ghep~mets+ins+ast_alt+dm2+I(ins^2), data=dat1)
anova(m2)[5,]
m3<-lm(ghep~mets+ins+ast_alt+dm2+I(ast_alt^2), data=dat1)
anova(m3)[5,]

### Supuesto de homoscedasticidad de varianza
### e independencia de los errores
bptest(lm2, data = dat1, studentize = TRUE)
rstd.0 <- stdres(lm2); par(mfrow=c(2,2)); x<-sort(dat1$mets)
```

```

lw1<-loess(rstd.0 ~ x,data=dat1)
plot(x, rstd.0, xlab="log(METS-IR)", pch = 19, cex = 1)
lines(x, lw1$fitted,col="blue",lwd=3, lty=2)
x1<-sort(dat1$ast_alt); lw2<-loess(rstd.0 ~ x1,data=dat1)
plot(x1, rstd.0, xlab="log(AST/ALT)", pch = 19, cex = 1)
lines(x1, lw2$fitted,col="blue",lwd=3, lty=2)
x2<-sort(dat1$ins); lw3<-loess(rstd.0 ~ x2,data=dat1)
plot(x2, rstd.0, xlab="log(Insulin)", pch = 19, cex = 1)
lines(x2, lw3$fitted,col="blue",lwd=3, lty=2)
x3<-sort(lm2$fitted); lw4<-loess(rstd.0 ~ x3,data=dat1)
plot(x3, rstd.0, xlab="Valores_ajustados", pch = 19, cex = 1)
lines(x3, lw4$fitted,col="blue",lwd=3, lty=2)

### Matriz de diseno de rango completo
##Supuesto de matriz de diseno de rango completo
#Evaluamos el indice kappa
X <- matrix(1, dim(dat1), 5)
X[,2:5] <- as.matrix(dat1[, c('mets', 'ast_alt', 'ins', 'dm2')])
kappa(X)

#Ahora calculamos los VIF utilizando el paquete car
#Compararemos con el modelo con todas las variables.
vif(lm2); vif(lm1)

### Normalidad conjunta de los errores
##Supuesto de normalidad conjunta de los errores
par(mfrow=c(1, 2))
residuos<-lm2$residuals
hist(residuos, density=10, breaks=10, prob=TRUE, xlab="Residuos",
ylim=c(0, 1.2),main="Histograma_de_residuos")
x<-residuos; m<-mean(residuos); std<-sd(residuos)
curve(dnorm(x, m, std), add=TRUE, lty=2, col="blue", lwd=3)
qqnorm(residuos); qqline(residuos, col = 2,lwd=2,lty=2)

```

```
#Prueba de Anderson-Darling
ad.test(residuos)

## Analisis final del modelo
K <- cbind(diag(length(coef(lm2))))
rownames(K) <- names(coef(lm2))
sim <- glht(lm2, linfct = K)
confint(sim, level = 0.95)
#Ajustamos los valores
pred.lm<-(predict(lm2, newdata =liver_test))^3
liver<-liver_test$ghep^3

# Self-normalizing neural network
##Tratamiento de los datos
train_data <- liver_train[,2:8]
test_data <- liver_test[,2:8]
train_targets<-liver_train$ghep
test_targets<-liver_test$ghep
m1 <- apply(train_data[,c(1:5)], 2, mean)
std <- apply(train_data[,c(1:5)], 2, sd)
train_data <- scale(train_data[,c(1:5)], center = m1,
                    scale = std)
test_data <- scale(test_data[,c(c(1:5))], center = m1,
                  scale = std)
train_data<-as.matrix(as.data.frame(cbind(train_data,
                                         liver_train$sex, liver_train$dm2)))
test_data<-as.matrix(as.data.frame(cbind(test_data,
                                         liver_test$sex, liver_test$dm2)))

##Diseno de la red
set.seed(123)
build_model <- function() {
  model <- keras_model_sequential() %>%
```

```
layer_dense(units = 8, kernel_initializer='lecun_normal',
activation="selu",input_shape = dim(train_data)[[2]]) %>%
layer_alpha_dropout(rate=0.10) %>%
layer_dense(units = 1, activation='softplus')

model %>% compile(
  optimizer = "adam",
  loss = "mse",
  metrics = c("mae")
)
}

#Evaluamos el numero de epocas
k <- 4
indices <- sample(1:nrow(train_data))
folds <- cut(indices, breaks = k, labels = FALSE)
num_epochs<-50

set.seed(123)

for (i in 1:k) {
  cat("processing fold", i, "\n")

  # Prepare the validation data: data from partition # k
  val_indices <- which(folds == i, arr.ind = TRUE)
  val_data <- train_data[val_indices,]
  val_targets <- train_targets[val_indices]

  # Prepare the training data: data from all other partitions
  partial_train_data <- train_data[-val_indices,]
  partial_train_targets <- train_targets[-val_indices]

  # Build the Keras model (already compiled)
```

```
model <- build_model()

# Train the model (in silent mode, verbose=0)
history <- model %>% fit(
  partial_train_data, partial_train_targets,
  validation_data = list(val_data, val_targets),
  epochs = num_epochs, batch_size = 1, verbose = 0
)

}
print(history)
plot(history)
#Reiniciamos el modelo, ajustamos por numero optimo de epocas
model <- build_model()
summary(model)
#Se entrena el modelo en la base completa
set.seed(123)
model %>% fit(train_data, train_targets,
             epochs = 50, batch_size = 1, verbose = 0)
print(history)
#Evaluamos el modelo en la base de validacion
result <- model %>% evaluate(test_data, test_targets)
result
#Ajustamos los valores de validacion
pred.nn <- predict(model, as.matrix(test_data))^3

# Gradient Boosting Machine

set.seed(123)
##Evaluacion de los parametros de ajuste
# create hyperparameter grid
hyper_grid <- expand.grid(
```

```
shrinkage = c(.1, .2, .4),
interaction.depth = c(1, 3, 5),
n.minobsinnode = c(5, 7, 10),
bag.fraction = c(.65, .8, 1),
optimal_trees = 0,
min_RMSE = 0
)
# total number of combinations
nrow(hyper_grid)

# grid search
train_data<-as.data.frame(train_data)
for(i in 1:nrow(hyper_grid)) {

  # reproducibility
  set.seed(123)

  # train model
  gbm.tune <- gbm(
    formula = train_targets ~ .,
    distribution = "gaussian",
    data = train_data,
    n.trees = 5000,
    interaction.depth = hyper_grid$interaction.depth[i],
    shrinkage = hyper_grid$shrinkage[i],
    n.minobsinnode = hyper_grid$n.minobsinnode[i],
    bag.fraction = hyper_grid$bag.fraction[i],
    train.fraction = 0.75,
    n.cores = NULL, # will use all cores by default
    verbose = FALSE
  )
# add min training error and trees to grid
```

```
hyper_grid$optimal_trees[i] <- which.min(gbm.tune$valid.error)
hyper_grid$min_RMSE[i] <- sqrt(min(gbm.tune$valid.error))
}
hyper_grid %>%
  dplyr::arrange(min_RMSE) %>%
  head(10)

##Ajustamos el modelo con los parametros corregidos
set.seed(123)
# train GBM model
gbm.fit <- gbm(
  formula = train_targets ~ .,
  distribution = "gaussian",
  data = train_data,
  n.trees = 12,
  interaction.depth = 3,
  shrinkage = 0.4,
  n.minobsinnode = 10,
  bag.fraction = 0.65,
  train.fraction = 1,
  cv.folds = 10,
  n.cores = NULL, # will use all cores by default
  verbose = FALSE
)
# print results
print(gbm.fit)

#MSE y grafico del modelo
sqrt(min(gbm.fit$cv.error))
gbm.perf(gbm.fit, method = "cv")
#Importancia de las variables
par(mar = c(5, 8, 1, 1))
summary(
```



```
gbm.fit,
  cBars = 10,
  method = relative.influence,
  las = 2
)
# predict values for test data
test_data<-as.data.frame(test_data)
pred.gbm <- predict(gbm.fit,
                   n.trees = gbm.fit$n.trees, test_data)^3

# Comparacion de los modelos
#Mean absolute error
mae.lm<-sum(abs(liver-pred.lm))/length(liver)
mae.nn<-sum(abs(liver-pred.nn))/length(liver)
mae.gbm<-sum(abs(liver-pred.gbm))/length(liver)
c(mae.lm, mae.nn, mae.gbm)
#Root of mean squared error
rmse.lm<-sqrt(sum((liver-pred.lm)^2)/length(liver))
rmse.nn<-sqrt(sum((liver-pred.nn)^2)/length(liver))
rmse.gbm<-sqrt(sum((liver-pred.gbm)^2)/length(liver))
c(rmse.lm, rmse.nn, rmse.gbm)
#Mean squared error
mse.lm<-sum((liver-pred.lm)^2)/length(liver)
mse.nn<-sum((liver-pred.nn)^2)/length(liver)
mse.gbm<-sum((liver-pred.gbm)^2)/length(liver)
c(mse.lm, mse.nn, mse.gbm)
#Correlacion y R^2
c(cor(liver, pred.lm), cor(liver, pred.lm)^2)
c(cor(liver, pred.nn), cor(liver, pred.nn)^2)
c(cor(liver, pred.gbm), cor(liver, pred.gbm)^2)
#Correlacion y R^2
var(pred.lm);var(pred.gbm); var(pred.nn)
```

