



CARRERA DE ESPECIALIZACIÓN EN INTELEGENCIA ARTIFICIAL

MEMORIA DEL TRABAJO FINAL

Predicción de eventos graves en pacientes hipertensos basado en informes de presurometrías a partir de la aplicación de técnicas de inteligencia artificial

Autor:

Ing. Trinidad Monreal

Director:

Dr. Ing. Roberto A. Bunge (UdeSA)

Jurados:

Nombre del jurado 1 (pertenencia)

Nombre del jurado 2 (pertenencia)

Nombre del jurado 3 (pertenencia)

*Este trabajo fue realizado en la Ciudad Autónoma de Buenos Aires,
entre octubre de 2022 y agosto de 2023.*

Resumen

En esta memoria se describe el diseño y la implementación de un software de inteligencia artificial desarrollado para el Hospital Alemán. El modelo utiliza datos obtenidos de presurometrías realizadas a pacientes hipertensos, con el propósito de predecir el riesgo de sufrir eventos cardiovasculares graves. Como resultado, este trabajo permite al servicio de cardiología e hipertensión definir estrategias de tratamiento personalizadas para cada grupo de riesgo.

Para llevar a cabo este trabajo se emplearon técnicas de estadística, análisis de datos y criterios de selección de características, así como modelos de aprendizaje profundo para lograr la clasificación.

Índice general

Resumen	I
1. Introducción general	1
1.1. Conceptos básicos de la presión arterial y sus métodos de medición	1
1.1.1. Presión arterial normal e hipertensión arterial	1
1.1.2. Presurometrías	3
1.2. Contexto y motivación	4
1.3. Objetivos, alcance y requerimientos	4
1.3.1. Objetivos	4
1.3.2. Alcance	5
1.3.3. Requerimientos	5
1.4. Estado del arte	6
2. Introducción específica	7
2.1. Enfoque del problema	7
2.2. Algoritmos de clasificación	8
2.2.1. Fundamentos de los modelos de clasificación	8
2.2.2. Estrategias para tratar el desbalance de clases	9
2.2.3. Evaluación de modelos de clasificación	10
<i>Precision</i>	11
<i>Recall</i>	11
<i>F1 score</i>	11
<i>Accuracy</i>	12
<i>Specificity</i>	12
Curva ROC y AUC	12
2.3. Perceptrón multicapa	13
2.3.1. Perceptrón	13
2.3.2. Arquitectura del perceptrón multicapa	14
2.3.3. Algoritmo de entrenamiento	16
Propagación hacia adelante	16
Retropropagación	16
2.3.4. Validación del modelo	17
2.3.5. Optimización del modelo	18
2.3.6. Hiperparámetros del modelo	19
3. Diseño e implementación	21
3.1. Adquisición de datos	21
3.1.1. Descripción del conjunto de datos	22
3.2. Preprocesamiento de datos	24
3.2.1. Conjunto de datos de presurometrías	24
Distribuciones de las variables numéricas	24
Tratamiento de clases desbalanceadas	28

	Normalización de características	28
	Selección de características	30
3.2.2.	Conjunto de datos de presurometrías y datos clínicos	32
	Distribuciones de las variables numéricas	32
	Imputación de valores faltantes	35
	Tratamiento de clases desbalanceadas	35
	Normalización de características	35
3.3.	Diseño y desarrollo de modelos	37
3.3.1.	Definición de métricas	37
3.3.2.	Arquitectura de los modelos	38
	Definición de la estructura de capas	38
	Selección de funciones de activación, función de pérdida y algoritmo de optimización	43
	Ajuste de hiperparámetros	44
3.3.3.	Estrategias implementadas para evitar el <i>overfitting</i>	44
	Bibliografía	45

Índice de figuras

1.1. Representación esquemática del pulso de presión registrado en la aorta ascendente ¹	2
1.2. Esquema de un <i>holter</i> de presión arterial.	4
2.1. Representación esquemática de las etapas del trabajo.	7
2.2. Representación esquemática del algoritmo SMOTE ²	10
2.3. Matriz de confusión ³	11
2.4. Representación gráfica de la curva ROC y las distribuciones de TP y TN ⁴	13
2.5. Representación de un perceptrón ⁵	14
2.6. Representación de un perceptrón multicapa <i>fully connected</i> ⁶	15
2.7. Representación esquemática de <i>k-fold cross validation</i> con $k = 5$ ⁷	18
3.1. Representación esquemática de la etapa de adquisición de datos.	21
3.2. Gráficos Q-Q de variables cuantitativas.	26
3.3. Histogramas de variables cuantitativas.	27
3.4. Conjunto de datos antes y después de aplicar SMOTE y <i>RandomUnderSampler</i>	28
3.5. Distribuciones de la PAS, PAD y FC durante 24 h antes y después de la normalización.	30
3.6. Combinaciones lineales de la presión arterial media y presión de pulso con las presiones sistólicas y diastólicas.	31
3.7. Gráficos Q-Q e histogramas de la talla, peso, IMC y edad.	34
3.8. Distribuciones de la talla, peso, IMC y edad antes y después de la normalización.	36
3.9. Valores promedio de AUC obtenidos mediante $k = 5$ iteraciones de validación cruzada.	40
3.10. Arquitectura elegida para los modelos.	43

Índice de tablas

1.1. Clasificación de la presión arterial por niveles	2
1.2. Valores de referencia para definir HTA por MAPA	3
3.1. Configuraciones de modelos explorados	39
3.2. Configuraciones de modelos explorados para el conjunto de datos de MAPA.	41
3.3. Configuraciones de modelos explorados para el conjunto de datos de MAPA y datos clínicos.	42
3.4. Hiperparámetros elegidos para el <i>dataset</i> 1 (datos MAPA) y <i>dataset</i> 2 (datos MAPA y datos clínicos).	44

Capítulo 1

Introducción general

En este capítulo se presentan los conceptos básicos de la presión arterial, hipertensión y presurometrías. Además, se menciona el contexto y la motivación que impulsan este trabajo de investigación. También, se establecen los objetivos, el alcance y los requerimientos, y se revisa el estado del arte en el campo de estudio.

1.1. Conceptos básicos de la presión arterial y sus métodos de medición

En esta sección se abordan las definiciones de presión arterial normal e hipertensión. Además, se discute el uso de las presurometrías como herramienta complementaria en el diagnóstico y seguimiento de paciente hipertensos.

1.1.1. Presión arterial normal e hipertensión arterial

La presión arterial (PA) es la fuerza por unidad de superficie ejercida por la sangre contra las paredes de las arterias [1]. Sin profundizar en los principios físicos, parece relevante destacar que el corazón bombea la sangre de forma pulsátil. Por este motivo, la PA alterna entre una presión arterial sistólica (PAS) y una presión arterial diastólica (PAD). En la figura 1.1 se expone un registro típico de las pulsaciones de la presión en la raíz de la arteria aorta [2] [3].

En un adulto joven sano, la presión máxima en cada pulso (PAS) es de 120 mmHg y la presión mínima (PAD) es de 80 mmHg. La diferencia entre estas dos presiones se conoce como la presión de pulso (PP) y ronda unos 40 mmHg. Por otro lado, la presión arterial media (PAM) está determinada en un 60 % por la PAD y en un 40 % por la PAS, dado que se invierte una mayor fracción del ciclo cardíaco en la diástole que en la sístole [2] [3]. Así, la expresión matemática que describe a la PAM es la siguiente:

$$PAM = \frac{(2 \cdot PAD + PAS)}{3} \quad (1.1)$$

¹Imagen adaptada de Klabunde, R. E. (2013). Normal and Abnormal Blood Pressure (Physiology, Pathophysiology and Treatment). Kindle Edition.



FIGURA 1.1. Representación esquemática del pulso de presión registrado en la aorta ascendente¹.

La PA alta o hipertensión arterial (HTA) es uno de los principales factores de riesgo para las enfermedades cardiovasculares, como la enfermedad cerebrovascular, insuficiencia cardíaca, cardiopatía isquémica, enfermedad renal terminal, arteriopatía periférica y retinopatía. La HTA se clasifica en diferentes grados según los niveles de presión arterial registrados. De esta forma, permite evaluar y categorizar la gravedad de la HTA en los pacientes y determinar las medidas terapéuticas más adecuadas para su manejo [4] [5]. Actualmente, se utiliza como guía la clasificación de HTA de la tabla 1.1.

TABLA 1.1. Clasificación de la presión arterial por niveles².

Grado	Presión arterial sistólica		Presión arterial diastólica
Óptima	< 120 mmHg	y	< 80 mmHg
Normal	120-129 mmHg	y/o	80-84 mmHg
Normal-alta	130-139 mmHg	y/o	85-89 mmHg
HTA de grado I	140-159 mmHg	y/o	90-99 mmHg
HTA de grado II	160-179 mmHg	y/o	100-109 mmHg
HTA de grado III	≥ 180 mmHg	y/o	≥ 110 mmHg

²Tabla realizada con información de López Farré, A. y Macaya Miguel, C. (2009). Libro de la salud cardiovascular del hospital clínico San Carlos y la fundación BBVA (pp. 121-129). Editorial Nerea, S.A.

1.1.2. Presurometrías

Las mediciones de la PA en el consultorio médico son necesarias pero insuficientes para un adecuado diagnóstico, tratamiento y seguimiento de la HTA. El monitoreo ambulatorio de presión arterial (MAPA), también conocido como presurometría, es un examen complementario que permite evaluar la PA en el contexto de la vida cotidiana del paciente. A diferencia de las mediciones de PA en el consultorio, que se realizan en condiciones estandarizadas, el MAPA obtiene un gran número de mediciones a lo largo de un día habitual del paciente. Esto proporciona la capacidad de medir la tensión arterial durante el reposo, sueño, actividad física y mental, trabajo y período postprandial. Conocer la PA ambulatoria permite identificar diferentes patrones de HTA, tales como: HTA diurna, HTA nocturna, HTA durante todo el día, HTA durante el sueño e HTA de guardapolvo blanco (solo presente en el consultorio médico). Por lo tanto, este estudio puede ser un predictor más efectivo de la mortalidad y eventos cardiovasculares que simplemente basarse en la medición de la PA en el consultorio médico [5] [6].

Actualmente, el MAPA es el único método disponible para medir la PA durante la noche, y diversos estudios demuestran que la PA nocturna tiene un mayor valor pronóstico que la PA diurna. Por esta razón, se recomienda incluir el MAPA como parte del diagnóstico de la HTA. Es especialmente útil cuando los valores de PA en el consultorio se encuentran en un rango limítrofe (grado normal-alta según la tabla 1.1) en varias consultas consecutivas [5]. Las indicaciones actuales para definir la HTA mediante el MAPA se detallan en la tabla 1.2.

TABLA 1.2. Valores de referencia para definir HTA por MAPA³.

	Presión arterial sistólica		Presión arterial diastólica
PA de 24 horas	≥ 130 mmHg	y/o	≥ 80 mmHg
PA diurna	≥ 135 mmHg	y/o	≥ 85 mmHg
PA nocturna	≥ 120 mmHg	y/o	≥ 70 mmHg

El MAPA se lleva a cabo mediante dispositivos conocidos como presurómetros o *holters* de presión arterial. Un técnico en cardiología coloca el presurómetro en el brazo del paciente, y se retira al día siguiente. El instrumento consiste en un manguito de presión arterial conectado a una grabadora, como se muestra en la figura 1.2. Esta realiza un inflado periódico cada 20 a 30 minutos, y los datos se almacenan en una memoria de estado sólido, generalmente en una tarjeta SD. Luego, los datos pueden ser analizados mediante el software del dispositivo [3] [5].

³Tabla realizada con información de Cuffaro, P. E., Morales, M. S. y Waisman, G. D. (2013). MONITOREO AMBULATORIO DE PRESIÓN ARTERIAL. En SAHA (Ed.), Hipertension Arterial, Epidemiología, Fisiología, Fisiopatología, Diagnóstico y Tratamiento. (pp. 391-395). Buenos Aires.



FIGURA 1.2. Esquema de un *holter* de presión arterial.

1.2. Contexto y motivación

La incorporación de técnicas de inteligencia artificial (IA) en la medicina cardiovascular es una de las principales motivaciones en la realización de este trabajo. A medida que la IA se desarrolla en distintos campos, su aplicación en el ámbito de la salud experimenta un crecimiento significativo. Sin embargo, uno de los desafíos clave que la IA enfrenta en la medicina es la limitación de datos disponibles para entrenar modelos de aprendizaje profundo [7].

En el caso particular del Hospital Alemán de Buenos Aires, el servicio de cardiología posee una gran cantidad de datos de informes de presurometrías de pacientes hipertensos. Aunque el personal carece de experiencia en herramientas de aprendizaje automático o aprendizaje profundo, los profesionales médicos comprenden el potencial de estos datos de MAPA para facilitar la identificación de patrones, factores de riesgo y respuestas al tratamiento de HTA. Además, es importante destacar que este trabajo se alinea con la misión del Hospital Alemán, que busca la actualización constante de tecnologías para garantizar una mejor atención médica basada en la investigación [8].

1.3. Objetivos, alcance y requerimientos

1.3.1. Objetivos

El propósito de este trabajo fue el desarrollo de un software de IA que permite predecir el riesgo de sufrir un evento grave en pacientes hipertensos. En particular, se busca predecir la ocurrencia de eventos cardiovasculares adversos mayores (MACE, por sus siglas en inglés). Este es un criterio de valoración compuesto que es empleado con frecuencia en la investigación cardiovascular. A pesar del uso generalizado del término en ensayos clínicos, las definiciones de MACE pueden diferir [9]. Para este trabajo, se definió como la combinación de: accidente cerebrovascular no fatal, infarto agudo de miocardio, insuficiencia cardíaca, insuficiencia

renal crónica o muerte. De este modo, un valor de MACE equivalente a la unidad indica la existencia de alguno de los eventos graves mencionados anteriormente, mientras que un valor nulo se refiere a la ausencia de estos.

Como resultado, este trabajo permite al servicio de cardiología e hipertensión del Hospital Alemán definir estrategias de tratamiento integral personalizadas para cada grupo de riesgo.

1.3.2. Alcance

Se encuentra dentro del alcance del trabajo elaborar un *dataset* con variables provenientes de presurometrías. El conjunto de datos debe estar en cumplimiento con la ley 25.326 para garantizar el derecho al honor y a la intimidad de los pacientes en cuestión. Además, es parte del alcance del trabajo seleccionar y elaborar un modelo de inteligencia artificial que prediga MACE en pacientes hipertensos. Asimismo, se incluye en el alcance la definición de métricas para evaluar el correcto desempeño del modelo.

Sin embargo, no se encuentra dentro del alcance del proyecto la instalación del software desarrollado dentro de los establecimientos del Hospital Alemán. Tampoco se encuentra dentro del alcance la implementación de una aplicación visual e interactiva para que utilice el usuario final.

1.3.3. Requerimientos

A continuación, se listan los requerimientos principales del trabajo agrupados por afinidad:

1. Requerimientos funcionales:

- a) El código desarrollado deberá ser capaz de estratificar a los pacientes en diferentes grupos de riesgo.
- b) El área bajo la curva ROC (acrónimo de *Receiver Operating Characteristic*, o característica operativa del receptor) del modelo deberá ser superior a 85 %.

2. Requerimientos de documentación:

- a) El código desarrollado deberá estar bien documentado. En consecuencia, se deben incluir comentarios que permitan a cualquier persona comprender qué se está haciendo y por qué.

3. Requerimiento de *testing*:

- a) Los resultados del código desarrollado deberán ser aprobados por el cliente y los usuarios finales.

4. Requerimientos reglamentarios:

- a) Los datos clínicos deberán ser de carácter anónimo, cumpliendo con la ley 25.326 que reglamenta la protección de datos personales. Para ello, se distinguirá a cada paciente mediante un número de identificación.

1.4. Estado del arte

Se llevó a cabo una exhaustiva revisión de la literatura relacionada con la predicción de MACE a partir de datos de presurometrías. Después de examinar la literatura, no se encontró ningún trabajo que tuviera como objetivo predecir eventos cardíacos mayores a partir de informes de presurometrías. Sin embargo, se hallaron trabajos que si bien no tienen el mismo objetivo que este, proporcionan una perspectiva útil y algunas ideas interesantes para el desarrollo.

En primer lugar, existen trabajos que buscan predecir MACE a partir de otras fuentes, como historias clínicas o resultados de pruebas de laboratorio. Tal es el caso de Huang *et al.* [10], Zhang *et al.* [11] y Wang *et al.* [12], quienes utilizan información proveniente de historiales médicos para predecir eventos cardíacos mayores. Estos trabajos desarrollan modelos de aprendizaje de máquina con resultados aceptables. Sin embargo, es importante tener en cuenta que al emplear información de historiales clínicos se requiere recopilar datos durante un período prolongado. Esto contrasta con el uso de datos de MAPA, que brindan una visión más inmediata y detallada de la PA de los pacientes, permitiendo realizar inferencias en tiempo real.

En segundo lugar, existen numerosos estudios que en vez de centrarse en la predicción de MACE, se enfocan en predecir el riesgo de enfermedades cardiovasculares particulares. Se tuvieron en cuenta un total de 17 artículos científicos relevantes [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30]. Si bien los objetivos de estos estudios varían en cada caso, las técnicas de IA utilizadas son similares. Se puede destacar que los modelos de aprendizaje profundo demostraron el mejor rendimiento en la predicción de insuficiencia cardíaca y diabetes. Por otro lado, los modelos de aprendizaje de máquina clásicos obtuvieron los mejores resultados para la predicción de HTA.

Si bien existe literatura académica que aborda el tema de la predicción de eventos cardiovasculares, la mayoría de los trabajos se centran en investigaciones de carácter teórico y experimental. Sin embargo, este trabajo se destaca por su enfoque práctico y su objetivo de desarrollar una herramienta concreta que pueda ser implementada de manera efectiva en el Hospital Alemán. Mediante el uso de la presurometría y la carga inmediata de datos, se espera obtener una inferencia inmediata sobre el riesgo individual de MACE. Esta iniciativa brinda una solución real y aplicable en el entorno clínico, mejorando así la atención y el seguimiento de los pacientes hipertensos.

Capítulo 2

Introducción específica

El objetivo de este capítulo es proporcionar una base teórica sólida sobre las herramientas y métodos utilizados en el desarrollo de este trabajo. En particular, se presentan las principales etapas de trabajo, se explican los conceptos elementales de los algoritmos de clasificación binaria y se exploran los fundamentos del perceptrón multicapa.

2.1. Enfoque del problema

El presente trabajo se estructura en torno a tres etapas principales, las cuales se encuentran ilustradas en la figura 2.1.

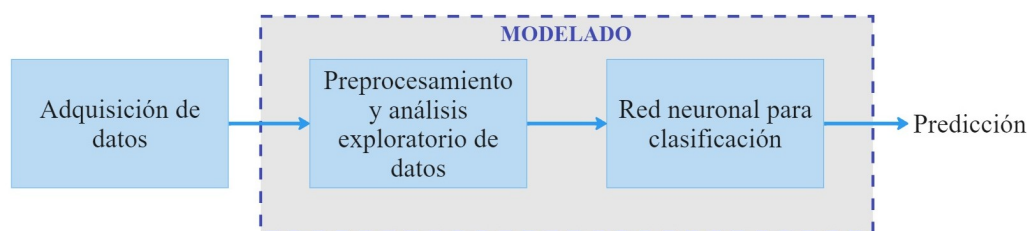


FIGURA 2.1. Representación esquemática de las etapas del trabajo.

Para comenzar, se llevó a cabo la adquisición de datos. Esta etapa fue realizada en colaboración con el servicio de cardiología del Hospital Alemán. Para ello, se recopiló información proveniente de presurometrías realizadas a pacientes hipertensos a partir del año 2013. Además, se efectuó un análisis exhaustivo de la historia clínica de cada paciente con el objetivo de identificar la ocurrencia de eventos cardiovasculares adversos mayores. De esta manera, se logró completar el *dataset* con la variable a predecir (MACE) para cada paciente. Asimismo, se enriqueció el conjunto de datos con información clínica adicional con el fin de contar con variables explicativas más amplias para abordar el problema de investigación.

A continuación, se efectuó una etapa de preprocesamiento. En primer lugar, se procedió a realizar una limpieza de los datos con el propósito de seleccionar las variables más representativas del problema. Este proceso de selección permitió eliminar datos redundantes, ruidosos o de poca utilidad para el análisis posterior. También, se realizó un análisis exploratorio de datos para comprender de manera más profunda las características y patrones presentes en el *dataset*. Esto proporcionó una visión más completa y significativa del conjunto de datos, permitiendo así una mejor comprensión de la naturaleza del problema y orientando la toma de decisiones posteriores en el desarrollo de este trabajo.

Por último, se procedió al diseño y entrenamiento de dos modelos de clasificación utilizando redes neuronales. El primer modelo se construyó exclusivamente con los datos de presurometrías, mientras que el segundo también incluyó los datos clínicos. El propósito de esta estrategia fue analizar y comparar el rendimiento de los dos modelos y evaluar si los datos de MAPA tienen un sustento suficiente para predecir la variable objetivo MACE. De esta manera, se buscó determinar si la inclusión de información clínica adicional contribuye a mejorar las métricas de desempeño, y si tiene implicancias clínicas relevantes para el seguimiento de los pacientes.

2.2. Algoritmos de clasificación

En esta sección se presentan los fundamentos de los modelos de clasificación, junto con las estrategias utilizadas para abordar el desbalance de clases, con especial énfasis en la técnica SMOTE. Además, se exploran las métricas utilizadas para evaluar el desempeño de los modelos desarrollados.

2.2.1. Fundamentos de los modelos de clasificación

Un algoritmo de aprendizaje automático es una herramienta que permite resolver tareas que suelen ser complejas de abordar para los seres humanos. La principal ventaja del *machine learning* (ML) es su capacidad de aprender de los datos y extraer información relevante de estos. Las tareas de ML se describen en términos de cómo deben procesar un ejemplo, que se compone de una colección de características cuantitativas o *features* que se miden de un objeto o evento que se desea procesar. En general, se representa un ejemplo como un vector $x \in \mathbb{R}^n$, donde cada entrada x_i del vector corresponde a una característica del objeto o evento en cuestión. De esta manera, el algoritmo de aprendizaje de máquina es capaz de analizar los datos y extraer patrones que permiten tomar decisiones o realizar predicciones precisas [31].

En el campo del aprendizaje automático, los modelos de clasificación desempeñan un papel fundamental [32]. Estos modelos se encargan de asignar etiquetas discretas o categorías a ejemplos específicos según sus características [33]. Dependiendo del enfoque utilizado durante el entrenamiento, los modelos pueden ser de dos tipos: supervisados o no supervisados [31].

En un modelo de clasificación no supervisado, las etiquetas no se proporcionan en el conjunto de datos de entrenamiento [34]. En lugar de ello, el algoritmo busca patrones y estructuras en los datos sin la guía explícita de las etiquetas. Como resultado, el objetivo es agrupar las instancias en clases basadas en similitudes [33].

Por otro lado, en un modelo de clasificación supervisado se proporciona al algoritmo un *dataset* de entrenamiento que incluye tanto las características de las instancias como las etiquetas correspondientes [33]. El modelo utiliza esta información para aprender a asociar las *features* con las etiquetas, lo que le permite clasificar nuevas instancias de manera precisa. Algunos ejemplos de algoritmos supervisados ampliamente utilizados son los árboles de decisión, las máquinas de vectores de soporte y las redes neuronales [34]. En este trabajo en particular, se empleó un modelo de clasificación supervisado basado en redes neuronales con

el propósito de predecir MACE. Es importante señalar que se trata de una clasificación binaria puesto que la variable *target* tiene dos posibles valores: ausencia ("0") o presencia ("1") de eventos cardiovasculares adversos mayores.

2.2.2. Estrategias para tratar el desbalance de clases

Un conjunto de datos se considera desbalanceado cuando presenta una proporción significativamente mayor de observaciones pertenecientes a una clase en comparación con la otra. Este desequilibrio puede tener un impacto considerable en el rendimiento de los modelos de aprendizaje automático. En el caso específico de una red neuronal entrenada con un *dataset* desbalanceado, es probable que encuentre dificultades para discriminar de manera adecuada entre las diferentes clases. En otras palabras, el desequilibrio de clases crea un sesgo en el cual el modelo tiende a predecir la clase mayoritaria, lo que dificulta la correcta clasificación de los ejemplos pertenecientes a la clase minoritaria [35] [36].

Abordar el desequilibrio se convierte en un desafío crucial en la tarea de modelado y actualmente existen dos enfoques principales con este fin. El primero consiste en asignar diferentes pesos a los ejemplos de entrenamiento, mientras que el segundo implica realizar un remuestreo del conjunto de datos original. Es posible crear un nuevo muestreo del *dataset* efectuando un sobremuestreo de la clase minoritaria y/o un submuestreo de la clase mayoritaria. El submuestreo implica una reducción del número de instancias de la clase mayoritaria, lo cual puede resultar en la pérdida de información valiosa que sea representativa de la distribución de los datos. Por otra parte, el sobremuestreo suele incluir réplicas de ejemplos de la clase minoritaria, lo cual puede generar un sobreajuste. Cabe señalar que un modelo sobreajustado concede predicciones muy precisas para el conjunto de entrenamiento pero no logra generalizar adecuadamente cuando se presentan nuevos datos [36].

Para mitigar las desventajas mencionadas anteriormente, Nitesh Chawla *et al.* [36] introducen una nueva técnica que implica una forma especial de sobremuestrear la clase minoritaria. El algoritmo *Synthetic Minority Over-sampling Technique* (SMOTE) implica la creación de muestras sintéticas de la clase minoritaria. Se expone una representación gráfica del algoritmo en la figura 2.2. En primer lugar, el método escoge un ejemplo de la clase minoritaria al azar y encuentra a sus k vecinos más cercanos. Luego, se selecciona uno de los k vecinos más cercanos y se crea una conexión mediante un segmento de línea en el espacio de características con la muestra elegida al azar. Finalmente, se crea una instancia sintética en un punto aleatorio entre los dos ejemplos. Este procedimiento se repite para todas las muestras minoritarias [37].

En suma, SMOTE permite generar ejemplos sintéticos que sean relativamente cercanos en el espacio de características con respecto a las muestras existentes de la clase minoritaria. Además, las muestras pueden crearse en la cantidad que sea requerida. Por lo tanto, en escenarios donde el conjunto de datos no solo está desbalanceado sino que también presenta una escasez de muestras, la aplicación de SMOTE resulta particularmente efectiva [36] [37].

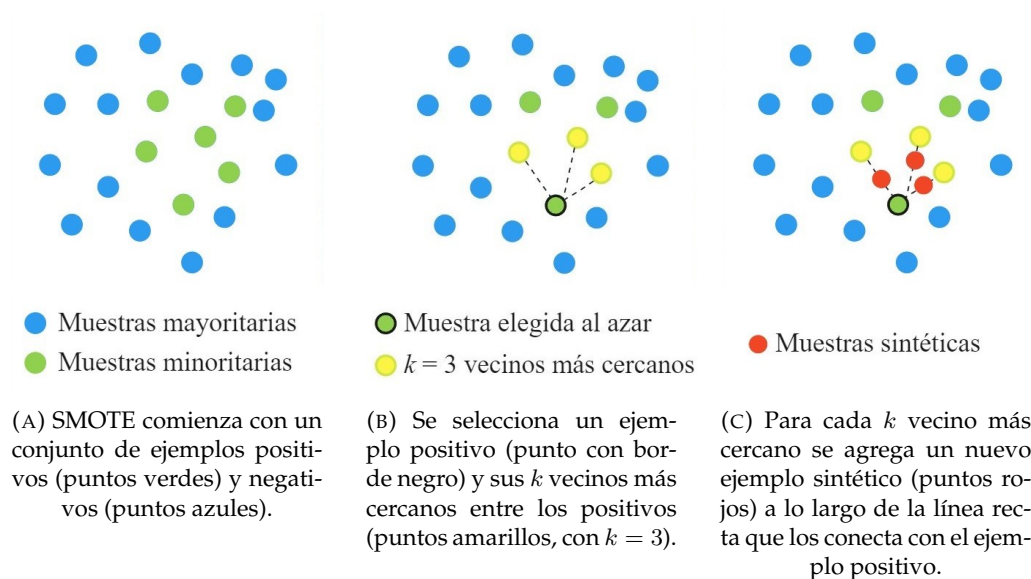


FIGURA 2.2. Representación esquemática del algoritmo SMOTE¹.

2.2.3. Evaluación de modelos de clasificación

Para evaluar el desempeño de un modelo de clasificación binaria se suele emplear una matriz de confusión. Esta matriz se ilustra en la figura 2.3, donde las columnas representan la clase predicha y las filas corresponden a las clases reales [38]. A partir de esta tabla, se definen cuatro indicadores:

- Verdaderos positivos (TP, por sus siglas en inglés): asignaciones positivas correctas.
- Verdaderos negativos (TN, por sus siglas en inglés): asignaciones negativas correctas.
- Falsos positivos (FP, por sus siglas en inglés): asignaciones positivas incorrectas, es decir que indican la presencia de una condición cuando la misma en realidad no está presente.
- Falsos negativos (FN, por sus siglas en inglés): asignaciones negativas incorrectas, lo que significa que indican la ausencia de una condición cuando en realidad está presente.

En el ámbito médico, el falso negativo puede ocasionar las consecuencias más graves para la salud del paciente. En este trabajo en particular, implica el riesgo de no tomar las medidas necesarias para un tratamiento adecuado, lo cual puede incurrir en que el paciente experimente un evento cardiovascular adverso mayor. En consecuencia, es de gran importancia minimizar la tasa de falsos negativos al máximo, aún si esto implica un aumento de falsos positivos. Si bien este incremento puede implicar la necesidad de destinar más recursos médicos para realizar pruebas adicionales o tratamientos innecesarios, se garantiza que menos casos sean pasados por alto [39].

¹Imagen adaptada de Schubach, Max; Re, Matteo; Robinson, Peter y Valentini, Giorgio. (2017). Imbalance-Aware Machine Learning for Predicting Rare and Common Disease-Associated Non-Coding Variants. Scientific Reports. 7. 10.1038/s41598-017-03011-5.

		Predicción	
		Positivo	Negativo
Observación	Positivo	Verdadero positivo (TP)	Falso negativo (FN)
	Negativo	Falso positivo (FP)	Verdadero negativo (TN)

FIGURA 2.3. Matriz de confusión².

A partir de los valores de TP, TN, FP y FN se definen diferentes métricas para evaluar el desempeño de un modelo de clasificación. A continuación, se presentan las que fueron utilizadas en este trabajo.

Precision

La precisión o *precision* es una métrica que permite conocer la proporción de ejemplos clasificados correctamente como positivos sobre el total de ejemplos clasificados como positivos. Una alta precisión indica que el modelo tiene una baja tasa de falsos positivos y clasifica correctamente la mayoría de las instancias positivas. La expresión matemática que describe este comportamiento es la siguiente:

$$Precision = \frac{TP}{TP + FP} \quad (2.1)$$

Recall

La sensibilidad, *recall* o tasa positiva real (TPR) es una métrica que mide la proporción de ejemplos clasificados correctamente como positivos en relación con el total de ejemplos positivos. Por lo tanto, se mide la capacidad de identificar la mayoría de las muestras positivas presentes. La ecuación que describe a la sensibilidad es:

$$Recall = \frac{TP}{TP + FN} \quad (2.2)$$

F1 score

El *f1 score* representa la media armónica entre *precision* y *recall*. Dado que tiene en cuenta tanto a los FP como los FN, se emplea para evaluar un modelo cuando el conjunto de datos es desequilibrado. La fórmula matemática que describe este comportamiento es la siguiente:

²Imagen adaptada de Portilla, Jose. 2018. Data Science and Machine Learning Bootcamp with R. <https://www.udemy.com/data-science-and-machine-learning-bootcamp-with-r/>.

$$F1\ score = \frac{2}{1/Precision + 1/Recall} = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \quad (2.3)$$

Accuracy

La exactitud o *accuracy* es una métrica que mide la proporción de muestras clasificadas correctamente sobre el total de muestras evaluadas y se describe matemáticamente como:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.4)$$

Specificity

La especificidad, *specificity* o tasa negativa real (TNR) se define como la proporción de falsos positivos sobre el total de negativos reales. Es una métrica exactamente opuesta a la sensibilidad y esta dada por:

$$Specificity = \frac{TN}{TN + FP} \quad (2.5)$$

Curva ROC y AUC

La curva ROC es una representación gráfica que muestra la tasa de verdaderos positivos (sensibilidad) en función de la tasa de falsos positivos (1 - especificidad) a medida que se varía el umbral de clasificación. Si bien puede ser de utilidad visualizar la curva ROC, en muchos casos la información puede resumirse en la métrica AUC (acrónimo de *Area Under the Curve*), que mide el área bajo la curva ROC. En general, cuanto más alto sea el AUC, mejor es el desempeño del clasificador [36] [38].

En la figura 2.4a, se puede observar un clasificador ideal en el cual las curvas de TP y TN no se superponen en absoluto, lo que resulta en un AUC igual a 1. Por el contrario, en la figura 2.4c se muestra un clasificador de predicción aleatoria, donde la curva ROC es una línea recta con una pendiente unitaria. En este caso, las curvas de TP y TN se superponen por completo, lo que da como resultado un AUC de 0.5. En la figura 2.4b se observa un caso intermedio en el que hay cierta superposición entre las curvas de TP y TN. En general, la mayoría de los modelos de clasificación binaria tienen un AUC entre 0.5 y 1. Es poco común encontrar un clasificador con un AUC menor que 0.5, ya que esto indicaría que las predicciones están invertidas [38].

La curva ROC y, en particular, el AUC son métricas ampliamente utilizadas en el ámbito clínico. Esto se debe a su capacidad para evaluar la habilidad de un modelo en la discriminación entre clases positivas y negativas. Esto resulta fundamental en la detección de enfermedades, diagnósticos médicos y evaluación de la efectividad de tratamientos.

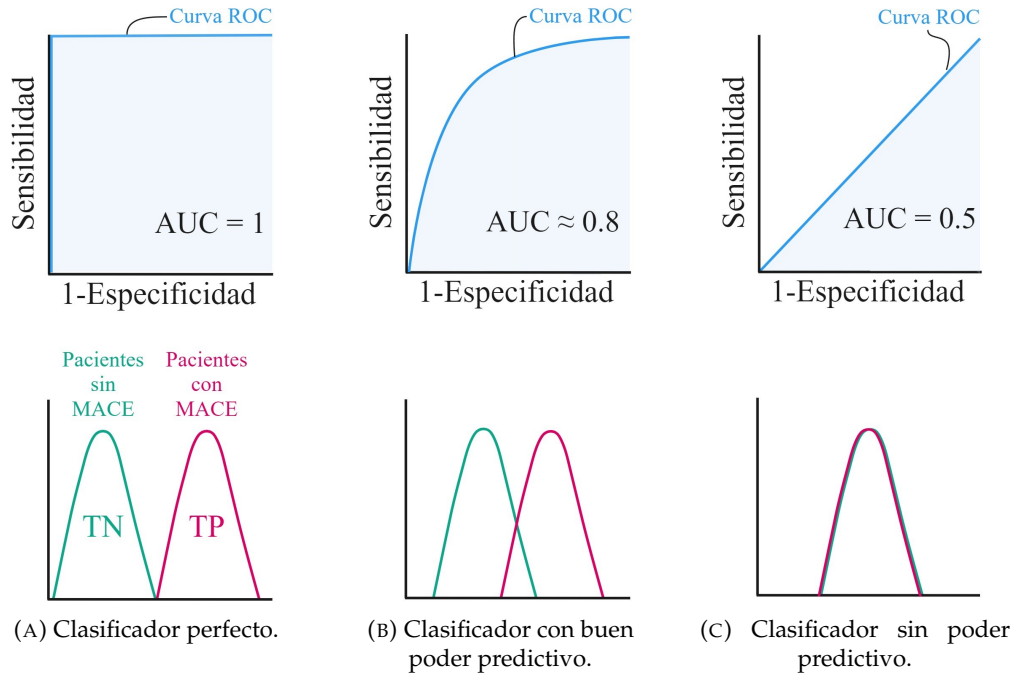


FIGURA 2.4. Representación gráfica de la curva ROC y las distribuciones de TP y TN³.

2.3. Perceptrón multicapa

Las redes neuronales artificiales (RNA), propuestas inicialmente por McCulloch y Pitts en 1943 [40], son modelos computacionales que se inspiran en el funcionamiento del cerebro humano. Si bien existen diversos tipos de RNA, esta sección se centra en los perceptrones multicapa, puesto que es la red empleada en este trabajo. En particular, se abordan varios aspectos teóricos clave relacionados con los perceptrones multicapa, incluyendo los bloques fundamentales que componen esta red, el algoritmo de entrenamiento, la validación del modelo, los hiperparámetros y los optimizadores.

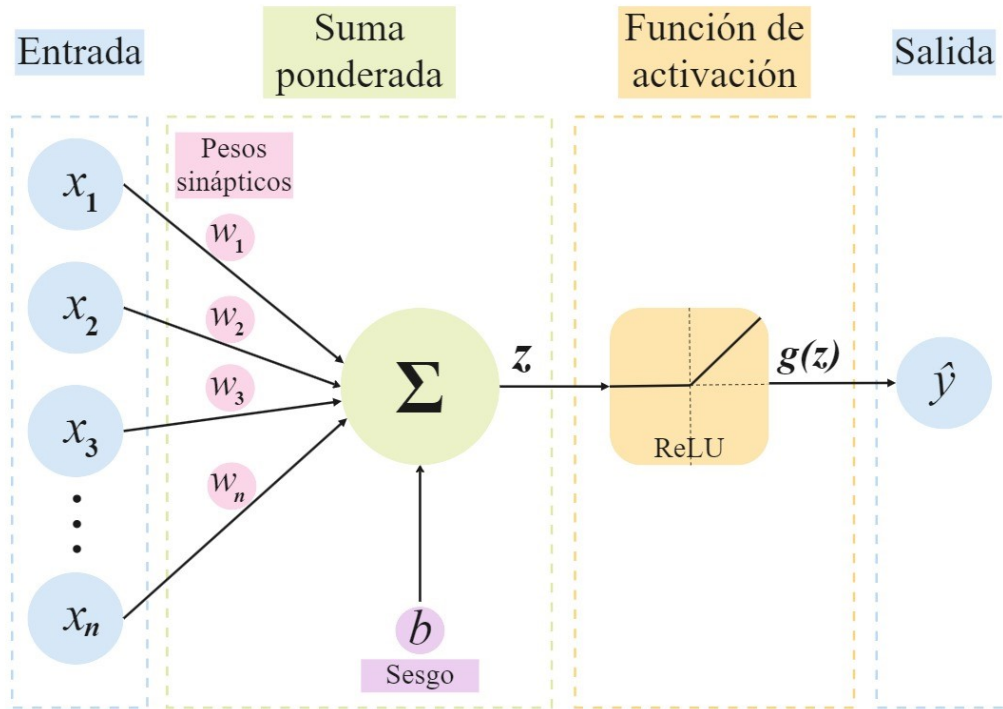
2.3.1. Perceptrón

Un perceptrón, también conocido como neurona artificial o nodo, es el bloque fundamental de las RNA y fue propuesto por Frank Rosenblatt en 1957 [41]. En la figura 2.5 se puede observar cómo un perceptrón recibe n señales de entrada $x = (x_1, x_2, \dots, x_n)$, las cuales se multiplican por un peso $W = (w_1, w_2, \dots, w_n)$. Este comportamiento es análogo a la manera en que los pesos sinápticos influyen en las neuronas biológicas, ya que algunas entradas tienen mayor relevancia que otras. Por este motivo, las entradas se combinan como una suma ponderada por sus pesos ($\sum_{i=1}^n W_i \cdot x_i$) y se añade un término independiente conocido como *bias* o sesgo, con el fin de otorgar flexibilidad en la modelización de patrones complejos. Posteriormente, se aplica una transformación no lineal mediante una función de activación g , la cual determina la salida \hat{y} del perceptrón [31] [42]. La descripción matemática de este comportamiento se expresa de la siguiente manera:

³Imagen adaptada de Glen, S. (2019). ROC Curve Explained in One Picture. Data Science Central. <https://www.datasciencecentral.com/roc-curve-explained-in-one-picture/>.

$$\hat{y} = g(z) = g\left(\sum_{i=1}^n W_i \cdot x_i + b\right) \quad (2.6)$$

$$\hat{y} = g(x_1 \cdot w_1 + x_1 \cdot w_1 + \dots + x_n \cdot w_n + b)$$

FIGURA 2.5. Representación de un perceptrón⁴.

2.3.2. Arquitectura del perceptrón multicapa

Un perceptrón multicapa (MLP, por sus siglas en inglés), también conocido como *feedforward neural network* es una arquitectura de RNA ampliamente utilizada en el campo del aprendizaje profundo. Como se aprecia en la figura 2.6, un MLP está compuesto por tres capas: capa de entrada o *input layer*, capa oculta o *hidden layer* y capa de salida o *output layer*. A su vez, cada capa está formada por neuronas artificiales. La arquitectura del MLP es conocida como *feedforward* puesto que la información siempre fluye desde la capa de entrada hacia la capa de salida a través de las capas ocultas, sin conexiones de retroalimentación. Cuando una RNA tiene dos o más capas ocultas, se lo conoce como una red neuronal profunda o *deep neural network* (DNN) [31] [38].

⁴Imagen adaptada de Hange, A. (2021). Target Prediction using Single-layer Perceptron and Multilayer Perceptron. Nerd For Tech. <https://medium.com/nerd-for-tech/flux-prediction-using-single-layer-perceptron-and-multilayer-perceptron-cf82c1341c33/>.

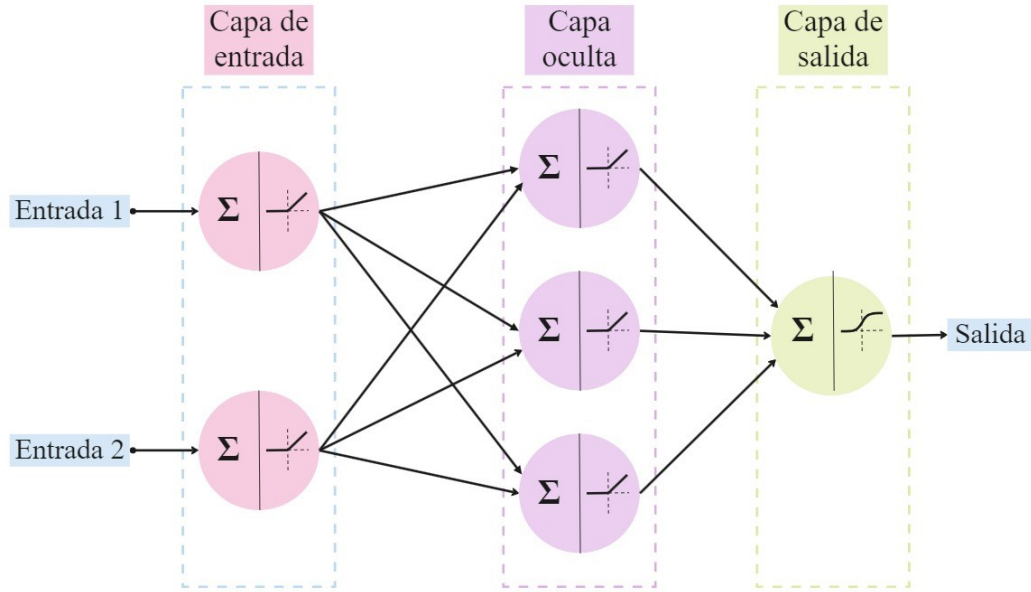


FIGURA 2.6. Representación de un perceptrón multicapa *fully connected*⁵.

Por otro lado, el número de neuronas en la capa de entrada está definido por la cantidad de variables independientes del modelo, mientras que el número de nodos de la salida es la cantidad de variables dependientes. Además, el número de capas ocultas y su cantidad de neuronas dependen de la complejidad del modelo en cuestión, y son parámetros importantes a tener en cuenta durante el desarrollo de la red [42].

La configuración del perceptrón multicapa de la figura 2.6 se conoce como *fully connected*, ya que todos los nodos de una capa están conectados con todos los nodos de las capas adyacentes. Sin embargo, existen otros tipos de capas que poseen diferentes estructuras de conexión. Por ejemplo, las de abandono o *dropout* introducen la omisión de algunos nodos durante el entrenamiento de forma aleatoria para reducir el sobreajuste [38] [43].

En general, la función de activación comúnmente empleada en las capas ocultas es la unidad lineal rectificada (ReLU, por sus siglas en inglés). La elección de ReLU se basa en dos ventajas principales: su implementación es rápida y no tiene un valor máximo de salida. En consecuencia, se evitan ciertos problemas durante el descenso del gradiente. En la ecuación 2.7, se observa que si el valor de entrada z es negativo, la función retorna un 0; de lo contrario, devuelve el valor de z [38].

$$ReLU(z) = \max(0, z) \quad (2.7)$$

Por otro lado, en la capa de salida de un modelo de clasificación binaria, es común utilizar la función sigmoide o *sigmoid*. Esta función, definida en la ecuación 2.8, toma valores reales como entrada y genera valores de salida en el rango de 0 a

⁵Imagen adaptada de Hange, A. (2021). Target Prediction using Single-layer Perceptron and Multilayer Perceptron. Nerd For Tech. <https://medium.com/nerd-for-tech/flux-prediction-using-single-layer-perceptron-and-multilayer-perceptron-cf82c1341c33/>.

1. Así, la función sigmoide es útil para predecir la probabilidad de una variable binaria [38].

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.8)$$

2.3.3. Algoritmo de entrenamiento

El perceptrón multicapa se entrena con el fin de minimizar el error entre la salida deseada y y la salida computada por el modelo \hat{y} empleando una función de pérdida o *loss function*. En el caso de problemas de clasificación binaria, se utiliza una función llamada *binary cross entropy* (BCE), que se define como:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \quad (2.9)$$

Como resultado, cuando la salida verdadera es $y_i = 1$, la probabilidad de clasificarlo como 1 es \hat{y}_i , lo que hace que el segundo término se anule. Por el contrario, cuando la salida verdadera es $y_i = 0$, la probabilidad de clasificarlo como 1 es $1 - \hat{y}_i$, lo que hace que se cancele el primer término. De esta forma, al aplicar el logaritmo, la BCE penaliza en mayor medida cuando el modelo clasifica incorrectamente con alta confianza [44]. En consecuencia, si la red produce una respuesta equivocada, los pesos se actualizan para minimizar el error. Esto conduce a una reducción de los errores y aumenta la probabilidad de que las salidas futuras de la red sean correctas. En la práctica, el algoritmo de aprendizaje de un MLP involucra una etapa de propagación hacia adelante (*forward propagation*) seguida de una etapa de propagación hacia atrás (*backward propagation* o *backpropagation*) [31]. A continuación, se describen brevemente estas etapas.

Propagación hacia adelante

La fase de propagación hacia adelante implica el flujo de información desde la capa de entrada hacia la de salida de la red neuronal, con el objetivo de realizar una predicción. La salida de la n -ésima capa se define en la ecuación 2.10, donde $h^{(n)}$ es la salida, $g^{(n)}$ es la función de activación, $W^{(n)}$ es la matriz de pesos y $b^{(n)}$ es el vector de sesgos [31] [38].

$$h^{(n)} = g^{(n)}(W^{(n)T} \cdot h^{(n-1)} + b^{(n)}) \quad (2.10)$$

El algoritmo 1 describe el proceso de *forward propagation* en una red neuronal de profundidad n . Luego de calcular la predicción \hat{y} , se calcula la función de costo total J durante la fase de entrenamiento. Esta se define como la suma de la función de pérdida \mathcal{L} y un término de regularización $\lambda\Omega(w)$.

Retropropagación

Durante muchos años, los investigadores se enfrentaron al desafío de encontrar una forma efectiva de entrenar a los perceptrones multicapa sin éxito. Sin embargo, en 1986, D. E. Rumelhart *et al.* publicaron un artículo revolucionario que

introdujo el método de *backpropagation* [45]. La fase de retropropagación se describe en el algoritmo 2. Esta implica el cálculo del gradiente de la función de costo respecto a todos los parámetros entrenables empleando la regla de la cadena. Luego, este gradiente se propaga desde la salida hacia la entrada de la red neuronal [31] [42] [38].

Algoritmo 1 Propagación hacia adelante de una RNA con n capas.

Require: $W^{(i)}, i \in 1, \dots, n$

Require: $b^{(i)}, i \in 1, \dots, n$

Require: x

Require: y

$h^{(1)} \leftarrow x$

for $m \leftarrow 1, \dots, n$ **do**

$a^{(m)} \leftarrow W^{(m)T} h^{(m-1)} + b^{(m)}$

$h^{(m)} \leftarrow g(a^{(m)})$

end for

$\hat{y} \leftarrow h^{(n)}$

$J \leftarrow \mathcal{L}(\hat{y}, y) + \lambda \Omega(w)$

▷ Si es fase de entrenamiento.

Algoritmo 2 Propagación hacia atrás de una RNA con n capas.

$g \leftarrow \nabla_{\hat{y}} J = \nabla_{\hat{y}} \mathcal{L}(\hat{y}, y)$

▷ Cálculo del gradiente en la capa de salida.

for $m \leftarrow n, n-1, \dots, 1$ **do**

$g \leftarrow \nabla_{a^{(m)}} J = g \odot f'(a^{(m)})$

▷ Actualiza el gradiente g .

$\nabla_{b^{(m)}} J = g + \lambda \nabla_{b^{(m)}} \Omega(w)$

▷ Cálculo del gradiente respecto a $b^{(m)}$.

$\nabla_{W^{(m)}} J = g h^{(m-1)} + \lambda \nabla_{W^{(m)}} \Omega(w)$

▷ Cálculo del gradiente respecto a $W^{(m)}$.

$g \leftarrow \nabla_{h^{(m-1)}} J = W^{(m)T} g$

▷ Propagación del gradiente.

end for

2.3.4. Validación del modelo

Para el desarrollo de una RNA, es necesario dividir al conjunto de datos en dos subconjuntos: uno para el entrenamiento y otro para la validación o prueba. El conjunto de entrenamiento se emplea para entrenar a la red, mientras que el conjunto de prueba se utiliza para evaluar su desempeño. Es crucial asegurarse de que el conjunto de entrenamiento sea lo suficientemente grande para ser representativo del problema, mientras que el de prueba sea lo suficientemente grande para facilitar una correcta validación de la red. En casos donde el número de ejemplos disponible no es suficiente para realizar la división representativa, se pueden utilizar otras estrategias [42]. El procedimiento mayormente empleado es la validación cruzada de k iteraciones, también conocida como *k-fold cross validation*, que se ilustra en la figura 2.7.

La validación cruzada de k iteraciones consiste en dividir el conjunto de datos en k subconjuntos. Durante el proceso de entrenamiento, se toma cada k subconjunto como el conjunto de prueba, mientras que los restantes $(k - 1)$ se utilizan como conjunto de entrenamiento. Este proceso se repite k veces, donde en cada iteración se toma un conjunto de prueba diferente y los datos remanentes forman el conjunto de entrenamiento. Al finalizar las k iteraciones, se calcula el promedio de las métricas y los errores obtenidos para cada subconjunto de prueba [46].

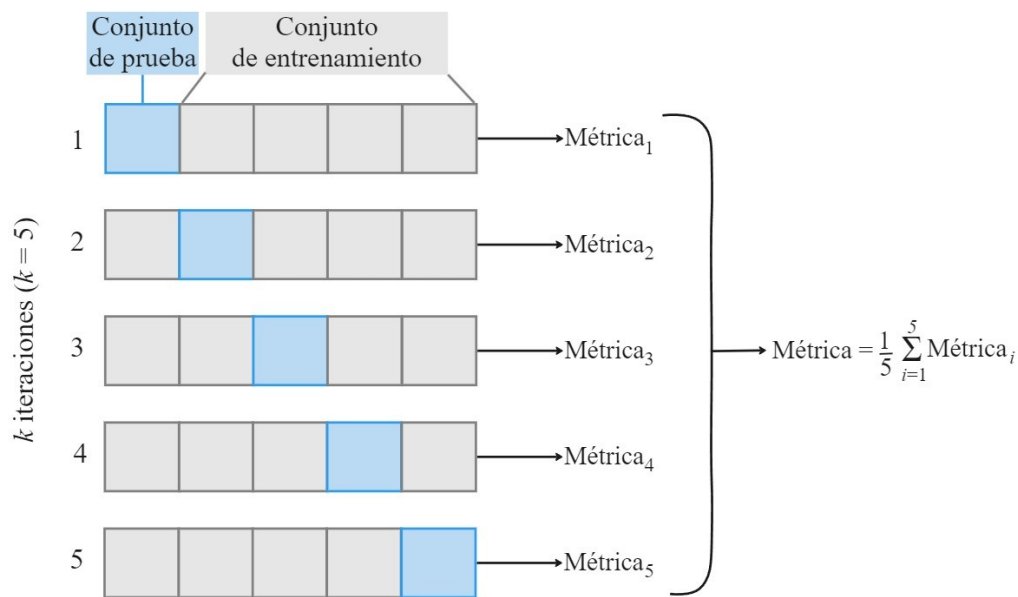


FIGURA 2.7. Representación esquemática de k -fold cross validation con $k=5$ ⁶.

2.3.5. Optimización del modelo

Es fundamental comprender el papel de los optimizadores y su impacto en el entrenamiento de una red neuronal. Los optimizadores son algoritmos empleados para ajustar los parámetros de la red, con el fin de minimizar la función de pérdida. Como resultado, logran mejorar la capacidad de aprendizaje y rendimiento del modelo. Algunos optimizadores comúnmente utilizados incluyen: descenso de gradiente estocástico, AdaGrad, Adam y RMSprop. Cada uno de ellos posee sus propias características y métodos de actualización de parámetros. Por lo tanto, la elección del optimizador es un paso importante en el desarrollo de un modelo, ya que puede influir significativamente en la velocidad y calidad de convergencia durante el entrenamiento [38].

En particular, el algoritmo de optimización empleado en este trabajo fue propuesto por Diederik Kingma y Jimmy Ba en su artículo "*Adam: A Method for Stochastic Optimization*" [47]. Si bien la elección del optimizador depende de diversos factores, Adam se destaca por múltiples ventajas, que incluyen:

- Implementación sencilla.
- Eficiencia computacional.
- Requisitos de memoria reducidos.
- Invarianza a la reescala diagonal de los gradientes.
- Adecuación para problemas con muchos datos y/o parámetros.
- Idoneidad para problemas con gradientes muy ruidosos o dispersos.

⁶Imagen adaptada de Ashfaq Maat, Johar y Iqbal, Amer. (2019). Introduction to Support Vector Machines and Kernel Methods.

- Hiperparámetros con interpretación intuitiva que generalmente requieren poco ajuste.

2.3.6. Hiperparámetros del modelo

En los modelos de aprendizaje automático, los parámetros son las variables que se estiman durante el proceso de entrenamiento con el conjunto de datos. Por otro lado, los hiperparámetros del modelo corresponden a los valores de configuración que se emplean durante dicho proceso. A diferencia de los parámetros, los hiperparámetros no son afectados por el algoritmo de aprendizaje. Por lo tanto, deben ser establecidos antes del entrenamiento y se mantienen constantes durante el mismo [38].

Usualmente, gran parte del diseño de una RNA implica la selección y ajuste de los hiperparámetros. Una elección adecuada puede tener un impacto significativo en el desempeño del modelo, ya que estos determinan el comportamiento y la capacidad de aprendizaje de la red. Algunos hiperparámetros que pueden ser ajustados incluyen: cantidad y tipo de capas ocultas, cantidad de neuronas en cada capa, función de activación, función de pérdida, optimizador, tasa de aprendizaje, tamaño del lote y número de épocas de entrenamiento [31] [42] [38].

Capítulo 3

Diseño e implementación

En este capítulo, se presentan de manera concisa las decisiones de diseño del sistema. Se abordan aspectos clave como la adquisición de datos, las técnicas de preprocesamiento utilizadas y el diseño del modelo de aprendizaje profundo.

3.1. Adquisición de datos

Para llevar a cabo el presente trabajo, se emplearon datos proporcionados por el servicio de cardiología e hipertensión del Hospital Alemán. La figura 3.1 exhibe un diagrama en bloques que ilustra la etapa de adquisición de datos utilizada. Es importante destacar que se utilizaron dos conjuntos de datos diferentes.

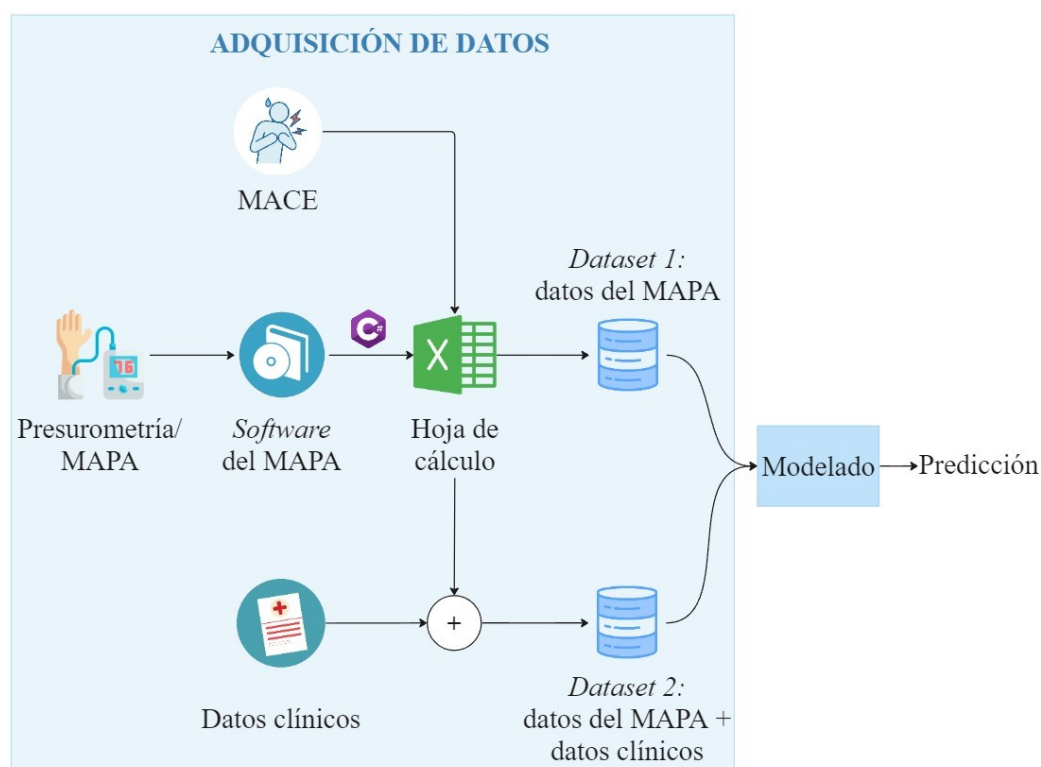


FIGURA 3.1. Representación esquemática de la etapa de adquisición de datos.

El primer *dataset* se compone exclusivamente de información recopilada de las presurometrías realizadas a los pacientes hipertensos a partir del año 2013. Luego

de realizar un MAPA, los datos se almacenan en un *software* especializado de los presurómetros. Posteriormente, se extraen utilizando un programa desarrollado en C# para ser registrados en una planilla de cálculo. Sumado a esto, se incluyó la variable a predecir (MACE) para cada paciente mediante un análisis exhaustivo de su historial clínico. Específicamente, se registró la ocurrencia de un accidente cerebrovascular no fatal, infarto agudo de miocardio, insuficiencia cardíaca, insuficiencia renal crónica o muerte. En caso de que correspondiera, también se incluyó la fecha correspondiente en la que tuvo lugar el evento.

Por otro lado, el segundo conjunto de datos contiene información adicional obtenida del historial clínico de cada paciente. Estos datos clínicos se agregan a la misma hoja de cálculo mencionada anteriormente, complementando así la información proveniente de las presurometrías.

El objetivo de utilizar dos *datasets* es evaluar la calidad de los datos de las presurometrías para generar inferencias de manera independiente. En otras palabras, se buscó determinar en qué medida los datos del MAPA pueden ser utilizados por sí solos para obtener predicciones significativas y precisas de MACE. Al mismo tiempo, se procuró evaluar si la incorporación de información clínica contribuye a un mejor desempeño del modelo en términos de tasa de falsos negativos, AUC y otras métricas relevantes.

3.1.1. Descripción del conjunto de datos

El primer conjunto de datos comprende las variables derivadas del MAPA, junto con el valor de MACE y la fecha del evento asociado. A continuación, se brinda una descripción detallada de las variables provenientes de las presurometrías:

- Fechaest: corresponde a la fecha en la cual se realizó la presurometría.
- PASm24: representa la presión arterial sistólica media durante todo el MAPA.
- PADm24: representa la presión arterial diastólica media durante todo el MAPA.
- FCm24: representa la frecuencia cardíaca media durante todo el MAPA.
- PAMm24: representa la presión arterial media durante todo el MAPA.
- PPm24: representa la presión de pulso media durante todo el MAPA.
- PASsd24: indica el desvío estándar de la presión arterial sistólica media durante todo el MAPA.
- PADsd24: indica el desvío estándar de la presión arterial diastólica media durante todo el MAPA.
- FCsd24: indica el desvío estándar de la frecuencia cardíaca media durante todo el MAPA.
- PAMsd24: indica el desvío estándar de la presión arterial media durante todo el MAPA.
- PPsd24: indica el desvío estándar de la presión de pulso media durante todo el MAPA.
- PASmDIA: representa la presión arterial sistólica media diurna.

- PADmDIA: representa la presión arterial diastólica media diurna.
- FCmDIA: representa la frecuencia cardíaca media diurna.
- PAMmDIA: representa la presión arterial media diurna.
- PPmDIA: representa la presión de pulso media diurna.
- htaloadsbpDIA: se refiere al porcentaje de lecturas de la presión arterial sistólica que exceden el valor de referencia de 135 mmHg durante el día.
- htaloaddbpDIA: se refiere al porcentaje de lecturas de la presión arterial diastólica que exceden el valor de referencia de 85 mmHg durante el día.
- HTAdia01: se utiliza para determinar si un paciente presenta hipertensión diurna (HTAdia01 = 1) o normotensión diurna (HTAdia01 = 0). Se considera hipertensión diurna cuando la presión arterial sistólica es superior a 135 mmHg y la presión arterial diastólica es superior a 85 mmHg durante el día, mientras que se define como normotensión diurna cuando no se cumplen estos criterios.
- PASmNOCHE: representa la presión arterial sistólica media nocturna.
- PADmNOCHE: representa la presión arterial diastólica media nocturna.
- FCmNOCHE: representa la frecuencia cardíaca media nocturna.
- PAMmNOCHE: representa la presión arterial media nocturna.
- PPmNOCHE: representa la presión de pulso media nocturna.
- htaloadsbpNOCHE: representa el porcentaje de lecturas de la presión arterial sistólica que exceden el valor de referencia de 125 mmHg durante la noche.
- htaloaddbpNOCHE: se refiere al porcentaje de lecturas de la presión arterial diastólica que exceden el valor de referencia de 70 mmHg durante la noche.
- HTAnoche01: se define como hipertensión nocturna (HTAnoche01 = 1) cuando la presión arterial sistólica es superior a 120 mmHg y la presión arterial diastólica es superior a 70 mmHg durante la noche. En caso contrario, se considera normotensión nocturna (HTAnoche01 = 0).
- Nondipper01: se utiliza para clasificar a los pacientes en función de la disminución de la presión arterial nocturna. Cuando la presión arterial no disminuye en un 10 % durante la noche, se considera un indicador de alto riesgo cardiovascular y se denominan *non-dippers* (Nondipper01 = 1). Por otro lado, aquellos pacientes cuya presión arterial disminuye en un 10 % durante la noche se denominan *dippers* (Nondipper01 = 0).

El segundo conjunto de datos, por su parte, incluye todas las variables mencionadas anteriormente, junto con 9 variables adicionales extraídas del historial clínico de cada paciente. Estas incluyen:

- Talla: se refiere a la medida de la longitud del cuerpo de una persona y se expresa en centímetros.
- Peso: se refiere a la medida de la masa corporal del paciente expresada en kilogramos.

- IMC: el índice de masa corporal (IMC) es una medida utilizada para evaluar la relación entre el peso y la estatura de una persona. Se calcula dividiendo el peso en kilogramos por el cuadrado de la estatura en metros.
- Edad: representa el tiempo transcurrido desde el nacimiento de un individuo, medido en años.
- Sexo: indica si el paciente es hombre (Sexo = 0) o mujer (Sexo = 1).
- DBT: indica la presencia (DBT = 1) o ausencia (DBT = 0) de diabetes en un paciente, una enfermedad metabólica crónica caracterizada por niveles elevados de glucosa en sangre.
- Tabaquismo: indica si un paciente tiene adicción a la nicotina (Tabaquismo = 1) o no presenta dicha adicción (Tabaquismo = 0).
- Dislipemia: señala la presencia (Dislipemia = 1) o ausencia (Dislipemia = 0) de una alteración en los niveles de lípidos en sangre en un paciente.
- HVI: indica la presencia (HVI = 1) o ausencia (HVI = 0) de un engrosamiento de la pared del ventrículo izquierdo como consecuencia de la hipertensión en el paciente.

3.2. Preprocesamiento de datos

En la siguiente sección, se proporciona una descripción detallada del preprocesamiento de datos realizado en este trabajo. El objetivo principal de esta etapa fue la obtención de un conjunto de datos final de alta calidad y utilidad para su posterior análisis y modelado. En la subsección 3.2.1 se describe el procesamiento realizado al conjunto de datos de presurometrías. Posteriormente, en la subsección 3.2.2 se aborda el procesamiento llevado a cabo en el conjunto de datos que combina los datos de MAPA y los datos clínicos.

3.2.1. Conjunto de datos de presurometrías

El conjunto de datos utilizado consta de 491 informes de MAPA y se caracteriza por no tener datos faltantes. Este *dataset* incluye: 3 variables categóricas (HTA durante el día, HTA durante la noche y *non-dipper*), 2 variables de fecha (fecha del estudio y fecha de MACE) y 25 variables numéricas continuas.

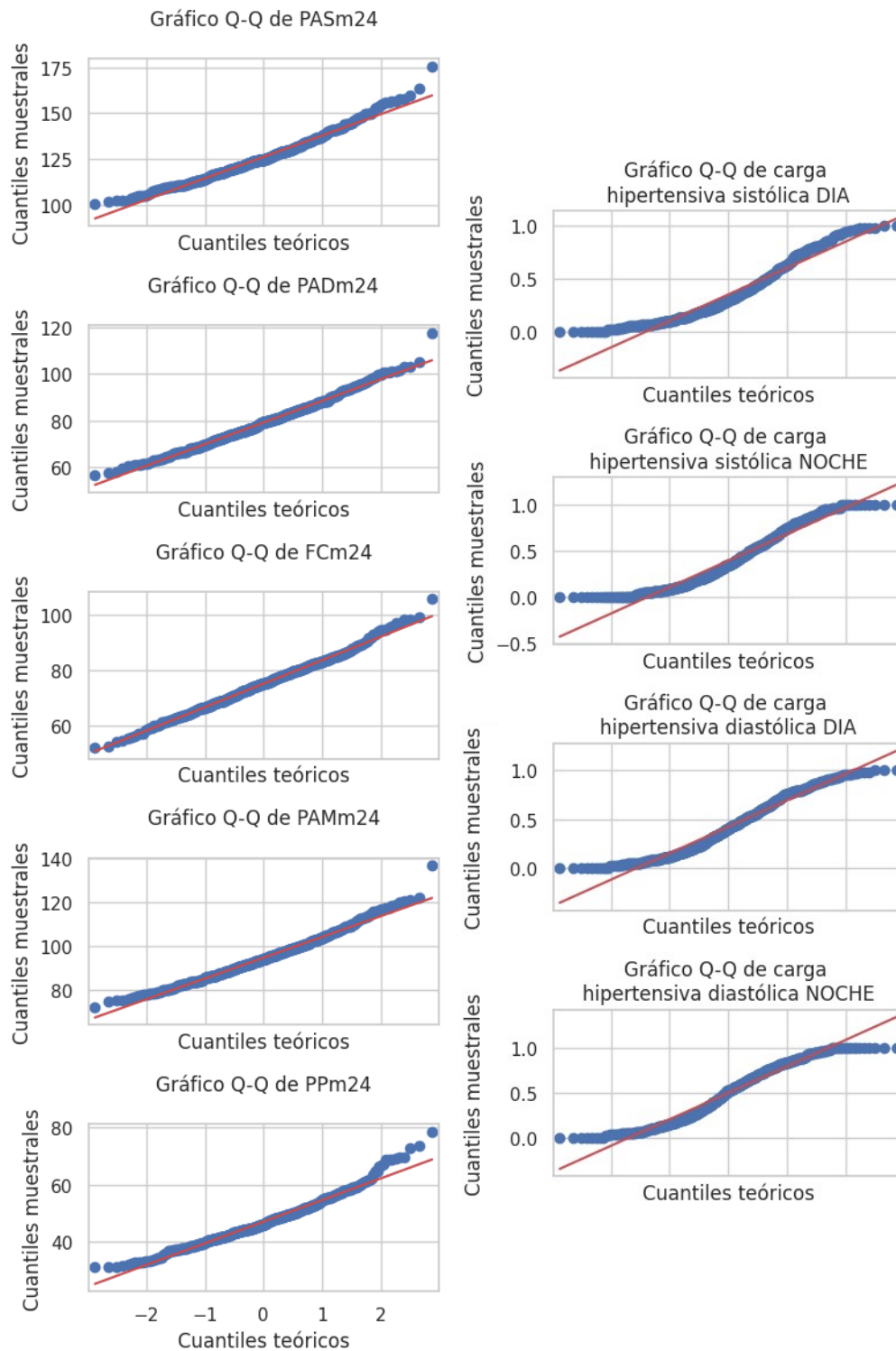
Distribuciones de las variables numéricas

Se realizó un análisis de las distribuciones de los datos con el objetivo de determinar si siguen una distribución normal y si existen valores extremos.

En primer lugar, se llevaron a cabo pruebas visuales utilizando gráficos Q-Q (cuantil-cuantil) e histogramas. El gráfico Q-Q es una herramienta utilizada para comparar dos distribuciones de probabilidad trazando sus cuantiles uno contra el otro. Aunque no es una prueba estadística formal, proporciona una forma sencilla e intuitiva de verificar qué distribución sigue un conjunto de datos. Por otro lado, el histograma permite visualizar la distribución de frecuencias de los datos de una variable.

En la figura 3.2a se representan los gráficos Q-Q de las presiones y la frecuencia cardíaca media durante un período de 24 horas. Por su parte, en la figura 3.2b

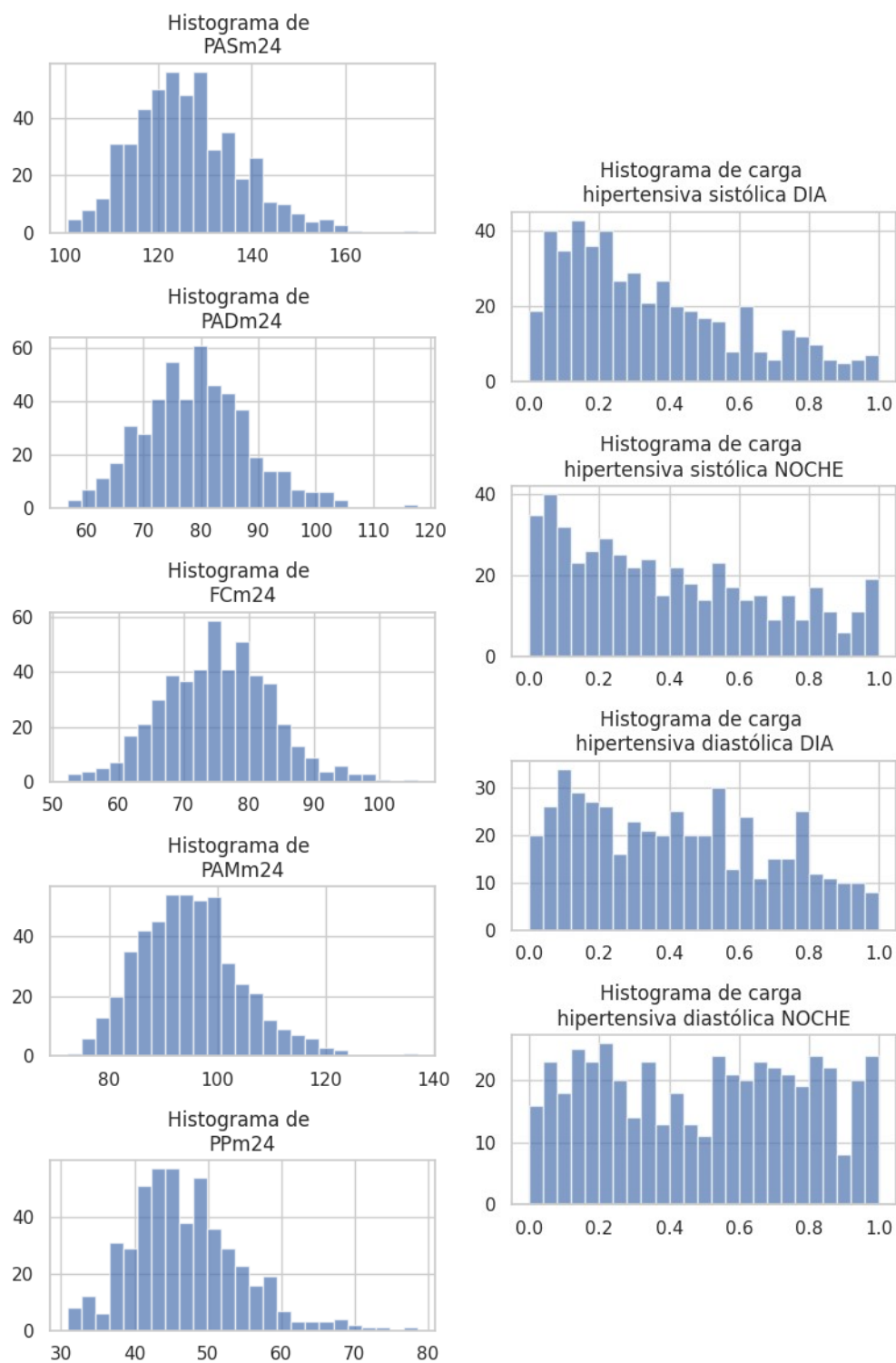
se muestran los gráficos Q-Q de las cargas hipertensivas sistólicas y diastólicas durante el día y la noche. Además, en la figura 3.3 se presentan los histogramas correspondientes a las mismas variables. Al examinar los gráficos, se observa que las variables PAD, PAM y frecuencia cardíaca (FC) siguen una distribución normal, mientras que las variables PAS y PP muestran una ligera sesgadura hacia la derecha. Cabe mencionar que las presiones y la FC mantienen la misma distribución tanto durante el día como durante la noche, aunque estos gráficos no se muestran en las figuras 3.2 y 3.3. En cuanto a las cargas hipertensivas sistólicas, se aprecia una distribución asimétrica hacia la derecha, mientras que las cargas hipertensivas diastólicas presentan una distribución casi uniforme.



(A) Gráficos Q-Q de la PAS, PAD, FC, PAM y PP durante 24 h.

(B) Gráficos Q-Q de las cargas hipertensivas sistólicas y diastólicas durante el día y la noche.

FIGURA 3.2. Gráficos Q-Q de variables cuantitativas.



(A) Histogramas de la PAS, PAD, FC, PAM y PP durante 24 h.

(B) Histogramas de las cargas hipertensivas sistólicas y diastólicas durante el día y la noche.

FIGURA 3.3. Histogramas de variables cuantitativas.

Tratamiento de clases desbalanceadas

En el contexto de este trabajo, se identificó un desbalance de clases, lo cual significa que una de las clases está representada por una fracción muy pequeña de las muestras totales. En la figura 3.4a se puede observar que solamente un 7.54 % de las muestras pertenecen a la clase positiva, lo que equivale a 37 pacientes con $MACE = 1$. Este desequilibrio en las clases es un fenómeno intrínseco y esperable, ya que refleja la naturaleza del proceso que genera los datos. Sin embargo, el desbalance de clases puede crear un sesgo en el modelo de aprendizaje automático, ya que tiende a favorecer la predicción de la clase mayoritaria en lugar de capturar adecuadamente las instancias de la clase minoritaria.

Para mitigar este problema, existen diversas técnicas que permiten equilibrar las clases en el conjunto de datos. En particular, para este trabajo se empleó la técnica SMOTE, la cual consiste en generar de forma sintética nuevos ejemplos de la clase minoritaria. Para lograr un balance más preciso, se combinó el sobremuestreo de la clase minoritaria utilizando SMOTE con el submuestreo de la clase mayoritaria mediante el uso de *RandomUnderSampler* [48], tal como recomienda la publicación científica "SMOTE: Synthetic Minority Over-sampling Technique" [36]. Esta combinación de técnicas permitió abordar de manera más precisa y equitativa el desafío de la predicción en un escenario desbalanceado. El nuevo conjunto de datos balanceado se exhibe en la figura 3.4b.

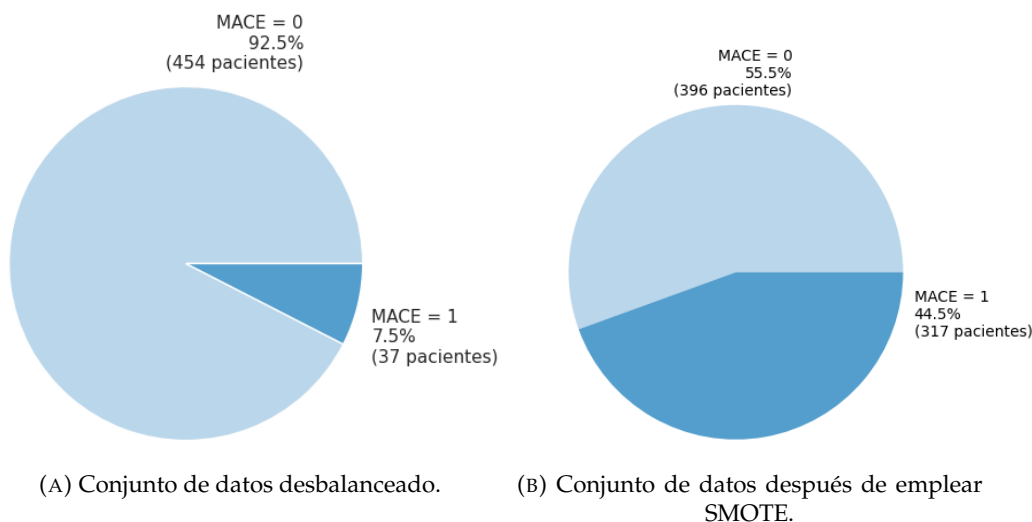


FIGURA 3.4. Conjunto de datos antes y después de aplicar SMOTE y *RandomUnderSampler*.

Normalización de características

La normalización de los datos es un paso crucial antes de entrenar una red neuronal, y tiene como objetivo principal igualar la escala de las características de entrada. Esto es importante puesto que las redes neuronales son sensibles a las diferencias de escala entre las variables. En otras palabras, cuando los datos presentan diferentes escalas, algunas características con valores numéricos más altos pueden tener un impacto desproporcionado en el proceso de entrenamiento. Esto puede ocasionar una mayor influencia en los pesos de la red y distorsionar el modelo resultante. Además, las funciones de activación utilizadas en las neuronas

pueden funcionar de manera óptima cuando los datos están normalizados en un rango adecuado.

En este trabajo, se decidió utilizar el método *StandardScaler* de *scikit-learn* [48] para normalizar las características de entrada. Este método aplica una transformación que ajusta la media de cada característica a 0 y la desviación estándar a 1. La definición de la transformación se muestra en la ecuación 3.1, donde X representa la matriz de características de entrada, μ a la media y σ al desvío estándar.

$$X_{\text{normalizado}} = \frac{X - \mu}{\sigma} \quad (3.1)$$

Al normalizar los datos de esta manera, se garantiza que todas las características tengan la misma escala, lo que facilita el proceso de aprendizaje de la red neuronal y mejora la convergencia del modelo. En la figura 3.5 se muestran a modo de ejemplo las distribuciones de la PAS, PAD y FC medias durante 24 horas antes y después de ser normalizadas.

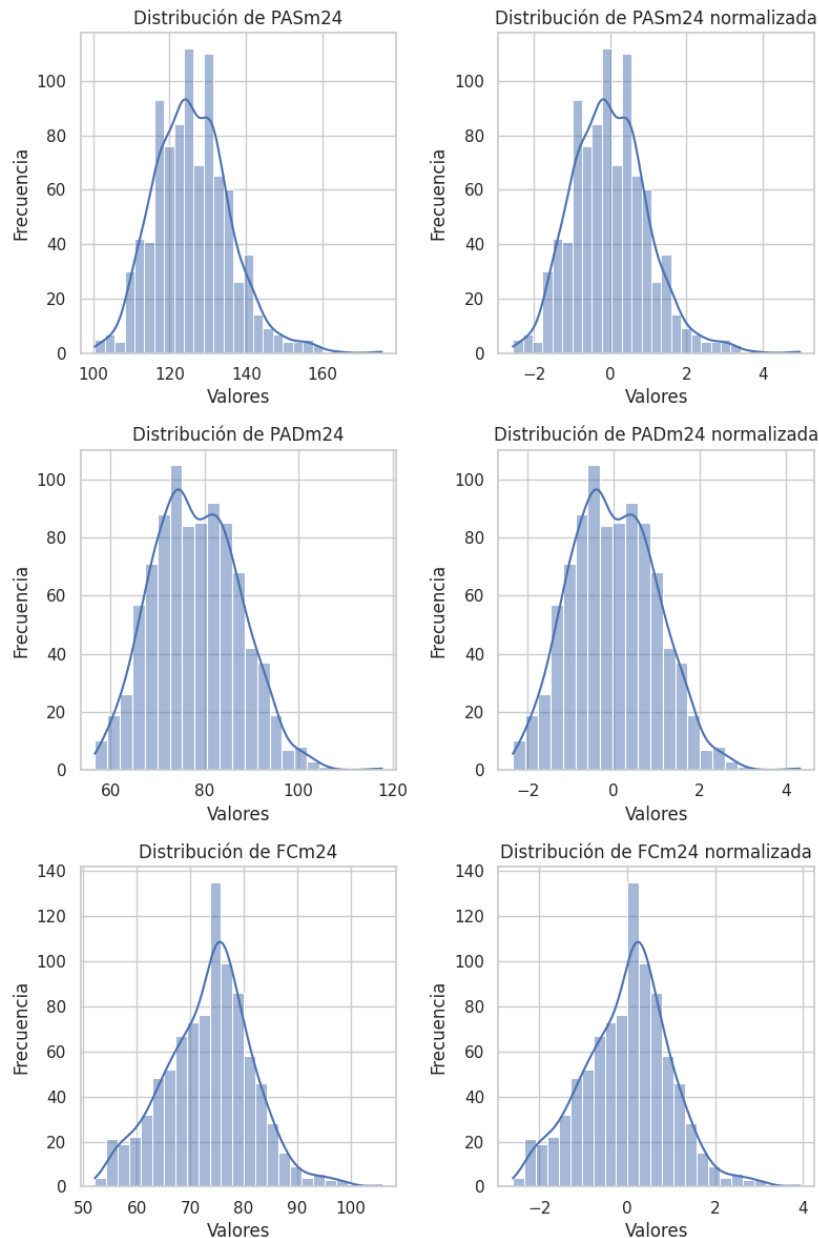


FIGURA 3.5. Distribuciones de la PAS, PAD y FC durante 24 h antes y después de la normalización.

Selección de características

La selección de variables para un modelo de red neuronal es un proceso crítico que tiene como objetivo elegir las variables más relevantes y evitar la inclusión de aquellas que sean redundantes o problemáticas. Durante el desarrollo de este trabajo, se identificaron dos variables que son combinaciones lineales de otras. Específicamente, se encontró que la presión arterial media y la presión de pulso dependen linealmente de la presión arterial sistólica y diastólica. Esta relación se definió matemáticamente en la subsección 1.1.1 y se puede visualizar claramente en la figura 3.6.

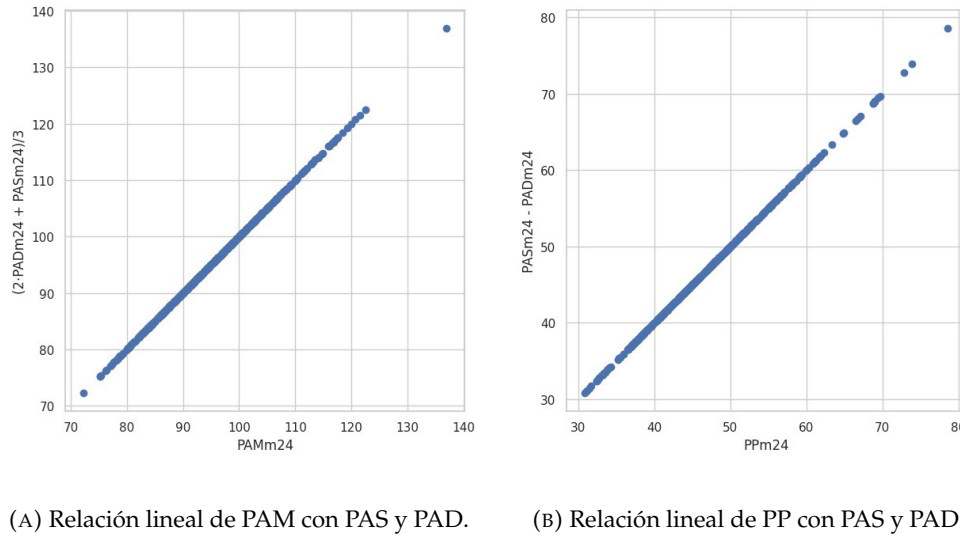


FIGURA 3.6. Combinaciones lineales de la presión arterial media y presión de pulso con las presiones sistólicas y diastólicas.

Teniendo en consideración esta información, se tomó la decisión de no incluir estas variables en la red neuronal por diversos motivos. En primer lugar, se buscó evitar la redundancia de información. Si una variable es una combinación lineal de otras, significa que la información contenida en dicha variable ya está presente en las demás. Por lo tanto, su inclusión solo duplicaría la información en la red neuronal, lo cual genera una sobrecarga y ralentiza el proceso de entrenamiento. Además, es posible que se generen problemas de inestabilidad numérica puesto que incluso pequeñas variaciones en los valores de entrada pueden dar lugar a variaciones significativas en las salidas. Esto podría comprometer la estabilidad y precisión de la red neuronal, afectando negativamente su rendimiento y capacidad de generar resultados confiables. Otro motivo para excluir estas variables es evitar que el modelo se ajuste en exceso a los datos de entrenamiento. Es posible que esto resulte en una pérdida de generalización, lo que significa que la red tendría dificultades para adaptarse a nuevos datos no vistos anteriormente.

En consecuencia, para abordar estos problemas, se decidió eliminar los valores medios y desviaciones estándar de PAM y PP. Así, se preservaron únicamente los valores medidos por el presurómetro: la PAS y PAD. De este modo, se asegura que la red neuronal se enfoque en las variables fundamentales y se evitan los problemas mencionados anteriormente.

Por otro lado, las variables que incluyen fechas son útiles para brindar fiabilidad a la base de datos, ya que proporcionan información sobre el momento en el que se realizó el MAPA y, en caso de que hubiese ocurrido, el momento en el que se produjo un evento. Sin embargo, en el contexto de predecir MACE, estas variables pueden considerarse irrelevantes ya que no aportan información sustancial para predecir el resultado deseado. En otras palabras, la fecha en sí misma no contiene información directa sobre los atributos fisiológicos o clínicos que están más relacionados con la ocurrencia de MACE. Además de su falta de relevancia, mantener la variable de fecha agregaría una complejidad adicional al modelo de red neuronal. La representación y codificación de la fecha como una característica en el modelo requiere un procesamiento adicional. Su transformación en valores numéricos o la creación de variables categóricas adicionales aumentaría

la dimensionalidad de los datos y podría dificultar la interpretación de la red neuronal. Por lo tanto, se decidió eliminar esta variable con el fin de agilizar el análisis y entrenamiento de la red neuronal.

Después de realizar el preprocesamiento a la información proveniente del MAPA, el conjunto de datos final utilizado para el desarrollo del modelo incluye las siguientes variables de entrada:

- PAsm24
- PADm24
- FCm24
- PASsd24
- PADsd24
- FCsd24
- PAsmDIA
- PADmDIA
- FCmDIA
- htloadsbpDIA
- htloaddbpDIA
- HTAdia01
- PAsmNOCHE
- PADmNOCHE
- FCmNOCHE
- htloadsbpNOCHE
- htloaddbpNOCHE
- HTAnoche01

3.2.2. Conjunto de datos de presurometrías y datos clínicos

El segundo conjunto de datos utilizado incorpora los datos clínicos adicionales de los 491 pacientes que se realizaron el MAPA, complementando así la información del conjunto de datos anterior. Dado que muchas de las variables son las mismas, esta subsección se enfoca exclusivamente en el preprocesamiento aplicado a las nuevas variables clínicas. Estas incluyen 4 variables numéricas (talla, peso, IMC y edad) y 5 variables categóricas (sexo, diabetes, tabaquismo, dislipemia e hipertrofia ventricular izquierda).

Distribuciones de las variables numéricas

Para analizar las variables clínicas numéricas del conjunto de datos, se llevaron a cabo pruebas visuales y estadísticas. En la figura 3.7 se exhiben los gráficos Q-Q e histogramas para las variables de interés. En cuanto a la talla, se observó que los datos siguen una distribución aproximadamente normal. Tanto el gráfico Q-Q como el histograma mostraron una forma de campana característica de esta

distribución. En el caso del peso y el IMC, se encontró una distribución asimétrica positiva. Esto indica que hay una mayor concentración de valores más bajos y una dispersión hacia valores más altos. Esto se debe a que la media y la mediana del peso fueron de 76 kg, mientras que la moda se situó en 85 kg. Dado que el IMC se calcula a partir del peso y la talla, su distribución sigue una tendencia similar a la del peso. Por otro lado, en cuanto a la variable de edad, se identificó una distribución asimétrica negativa. Esto sugiere que existe una mayor cantidad de personas mayores en el conjunto de datos. La media de la edad fue de 65 años, con una edad mínima de 19 años y una máxima de 98 años. La moda de la distribución se encontró en 72 años.

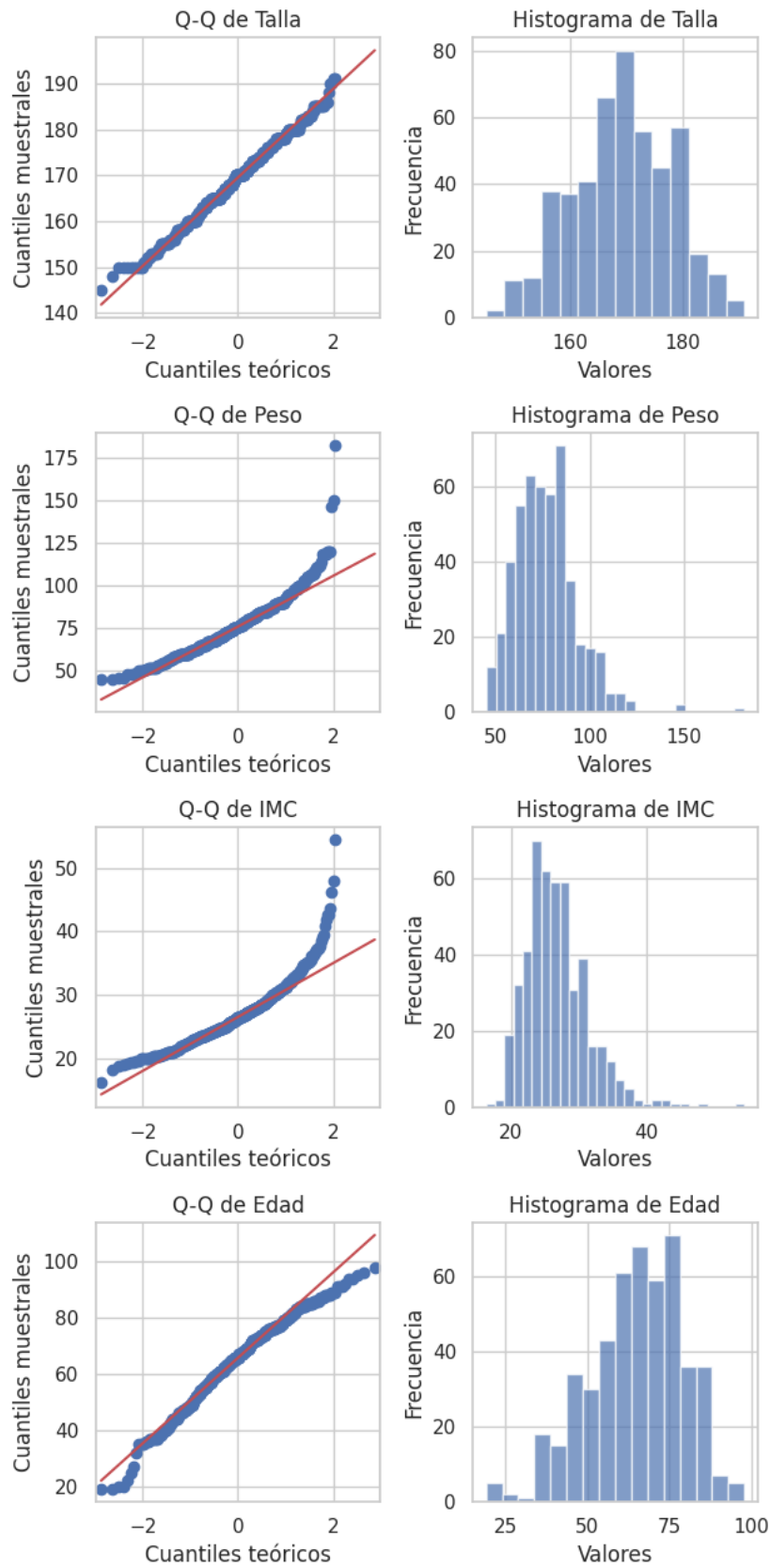


FIGURA 3.7. Gráficos Q-Q e histogramas de la talla, peso, IMC y edad.

Además de los análisis visuales, se realizaron pruebas estadísticas, incluyendo los tests de Shapiro-Wilk y Kolmogorov-Smirnov. Los resultados de ambas pruebas indicaron que la talla, el peso y el IMC siguen una distribución normal, mientras que la edad no presenta una distribución normal.

Imputación de valores faltantes

En el segundo conjunto de datos utilizado, se identificaron diferentes variables con valores faltantes. A continuación, se detalla la cantidad de valores ausentes en cada una de estas variables:

- Sexo: 1 paciente sin información.
- Talla, peso e IMC: 9 pacientes sin datos de talla ni peso, lo cual implica la ausencia del cálculo del IMC.
- Dislipemia: 1 paciente sin información.
- HVI (hipertrofia ventricular izquierda): 2 pacientes sin información.

Ante la presencia de estos valores faltantes, es necesario aplicar técnicas de imputación para completar la información y garantizar la integridad de los datos. En este trabajo, se optó por utilizar el método *KNNImputer* de *scikit-learn* [48]. Esta elección se basa en la capacidad del método para capturar la estructura de los datos y aprovechar la información de las instancias vecinas para la imputación. Además, este método es adecuado para manejar variables numéricas y categóricas, lo cual es relevante en el contexto de este trabajo.

La imputación de datos faltantes utilizando técnicas como el *KNNImputer* desempeña un papel fundamental al minimizar la pérdida de información en conjuntos de datos relativamente pequeños, como es el caso de este trabajo. De esta manera, se maximiza la utilidad de los datos disponibles y se evita la exclusión de observaciones valiosas debido a la falta de información.

Tratamiento de clases desbalanceadas

En relación al tratamiento de desbalance de clases, se aplicó el mismo enfoque que en la subsección 3.2.1 debido a que la variable de salida es la misma. Se utilizó la técnica de sobremuestreo SMOTE para abordar el desequilibrio de clases y mejorar la representación de la clase minoritaria en el conjunto de datos.

Normalización de características

Para asegurar que todas las características tengan una escala uniforme, se llevó a cabo un proceso de normalización de los datos. Al igual que en la subsección 3.2.1, se utilizó el método *StandardScaler* [48]. La figura 3.8 presenta las distribuciones antes y después de la normalización de las variables talla, peso, IMC y edad. Este procedimiento garantiza que todas las variables estén en una escala comparable, lo que facilita la interpretación y el análisis de los datos para la red neuronal.

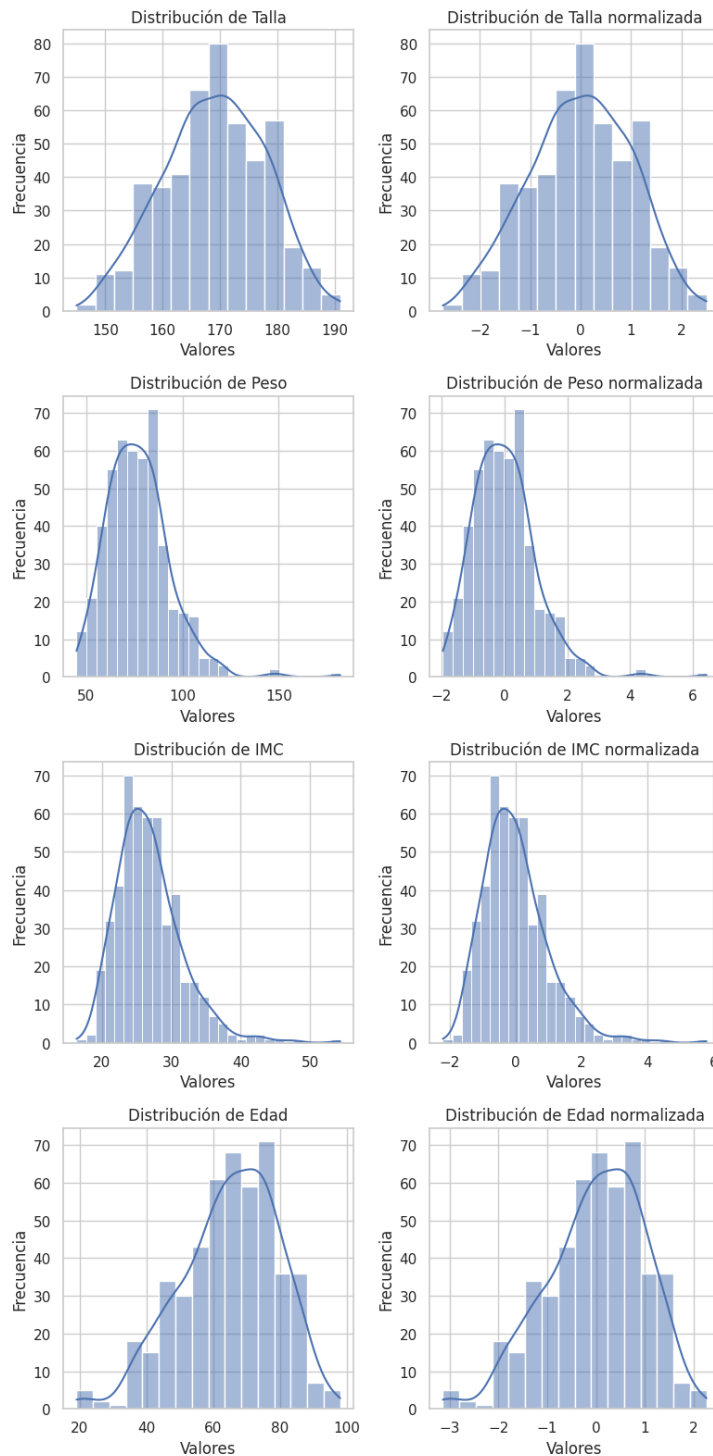


FIGURA 3.8. Distribuciones de la talla, peso, IMC y edad antes y después de la normalización.

Después de realizar el preprocesamiento a la información proveniente del MAPA y los datos clínicos, el segundo conjunto de datos utilizado para el desarrollo del modelo incluye las siguientes variables de entrada:

- PAsm24
- PADm24

- FCm24
- PASsd24
- PADsd24
- FCsd24
- PASmDIA
- PADmDIA
- FCmDIA
- htaloadsbpDIA
- htaloaddbpDIA
- HTAdia01
- PASmNOCHE
- PADmNOCHE
- FCmNOCHE
- htaloadsbpNOCHE
- htaloaddbpNOCHE
- HTAnoche01
- Talla
- Peso
- IMC
- Edad
- Sexo
- DBT
- Tabaquismo
- Dislipemia
- HVI

3.3. Diseño y desarrollo de modelos

El proceso de diseño e implementación de los modelos se llevó a cabo de igual manera para ambos conjuntos de datos. A continuación, se presentan en detalle las métricas utilizadas, la arquitectura de los modelos y el proceso de entrenamiento.

3.3.1. Definición de métricas

Para evaluar el desempeño de los modelos de forma objetiva, se empleó la métrica AUC. La curva ROC y, en particular, el AUC son ampliamente utilizados en la

evaluación de modelos de clasificación para medir la capacidad de distinguir entre clases positivas y negativas. Para este trabajo en particular, la elección se debe a la relevancia clínica de analizar la relación entre la sensibilidad y especificidad. En otras palabras, se buscó un equilibrio óptimo entre la detección temprana de casos positivos de MACE y la minimización de falsas alarmas. Además, la solicitud específica del cliente de lograr un AUC superior al 85 % justificó el uso del AUC como métrica clave.

A su vez, el servicio de cardiología del Hospital Alemán enfatizó la importancia de evitar falsos negativos. Esto implicó que se priorizara la correcta clasificación de casos positivos aunque esto signifique un aumento en la tasa de falsos positivos y una posible reducción en otras métricas de evaluación.

Por otro lado, se calcularon otras métricas además del AUC como *precision*, *recall* y *accuracy* para obtener una visión más completa de los resultados del modelo.

3.3.2. Arquitectura de los modelos

Una vez definido el objetivo del modelo y preprocesados los datos, se seleccionó la arquitectura básica del modelo. Se decidió emplear un perceptron multicapa puesto que permite capturar relaciones y patrones no lineales en los datos. Además, los MLP tienen gran flexibilidad en la representación de características y enorme capacidad para aprender representaciones más abstractas y de mayor nivel. De este modo, esta arquitectura puede llevar a una mejor habilidad de generalización [31] [38].

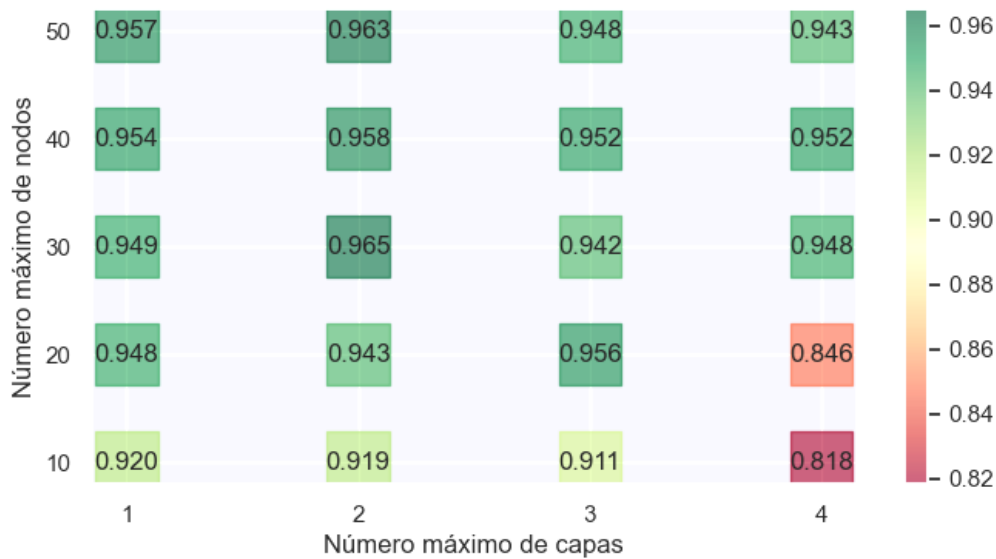
Definición de la estructura de capas

Se llevó a cabo una investigación exhaustiva para determinar la configuración óptima del modelo, incluyendo el número de capas ocultas y la cantidad de neuronas en cada capa. Dado que no se disponía de conocimientos previos sobre la arquitectura ideal, se realizó una búsqueda sistemática de hiperparámetros. Se consideraron diferentes alternativas, como 1, 2, 3 o 4 capas ocultas, y se establecieron múltiples opciones para el número de neuronas en la primera capa oculta, incluyendo 10, 20, 30, 40 o 50 nodos. En caso de tener múltiples capas ocultas, se decidió que la cantidad de neuronas en las capas posteriores fuese la mitad de las de la capa anterior. La tabla 3.1 muestra las diferentes combinaciones de modelos que resultaron de esta exploración.

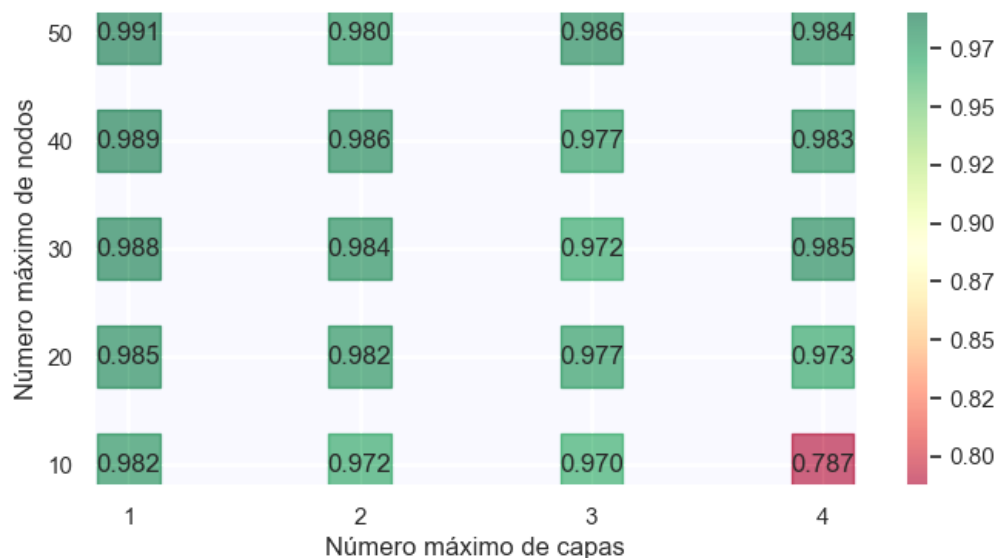
TABLA 3.1. Configuraciones de modelos explorados con la cantidad de neuronas por capa oculta.

Modelo	Capa 1	Capa 2	Capa 3	Capa 4
1 capa oculta				
1	50	-	-	-
2	40	-	-	-
3	30	-	-	-
4	20	-	-	-
5	10	-	-	-
2 capas ocultas				
6	50	25	-	-
7	40	20	-	-
8	30	15	-	-
9	20	10	-	-
10	10	5	-	-
3 capas ocultas				
11	50	25	12	-
12	40	20	10	-
13	30	15	7	-
14	20	10	5	-
15	10	5	2	-
4 capas ocultas				
16	50	25	12	6
17	40	20	10	5
18	30	15	7	3
19	20	10	5	2
20	10	5	2	1

La figura 3.9 muestra los valores promedio de AUC obtenidos mediante $k = 5$ iteraciones de validación cruzada para los modelos presentados en la tabla 3.1. Se observa que, en el caso del conjunto de datos de presurometrías, los tres modelos con el mejor rendimiento en términos de AUC presentan una arquitectura de dos capas ocultas, donde la primera capa cuenta con 30 o más nodos (modelos 6, 7 y 8 de la tabla 3.1). En contraste, para el conjunto de datos que incluye información clínica, se encontró que el mejor desempeño en términos de AUC se logró con un modelo de una sola capa oculta, con 30 o más nodos en la primera capa (modelos 1, 2 y 3 de la tabla 3.1).



(A) Valores promedio de AUC obtenidos para el conjunto de datos de MAPA.



(B) Valores promedio de AUC obtenidos para el conjunto de datos de MAPA y datos clínicos.

FIGURA 3.9. Valores promedio de AUC obtenidos mediante $k = 5$ iteraciones de validación cruzada para los modelos evaluados.

Con base en los hallazgos anteriores, se llevó a cabo una búsqueda más exhaustiva para determinar la arquitectura óptima del modelo. Para el primer conjunto de datos, se realizaron pruebas con modelos que incluían dos capas ocultas, explorando diferentes combinaciones. La primera capa se configuró con un rango de 20 a 50 nodos, incrementando de uno en uno, mientras que la segunda capa se definió con la mitad de las neuronas de la primera capa, es decir, de 10 a 25 nodos, también en incrementos de uno. Los resultados obtenidos para cada configuración se registraron en la tabla 3.2. Por otro lado, para el conjunto de datos que incluía variables clínicas adicionales, se realizó una búsqueda de la cantidad de nodos óptima para la capa oculta, dentro de un rango de 30 a 55, con un incremento de un nodo. Los resultados de esta exploración se presentan en la tabla

3.3.

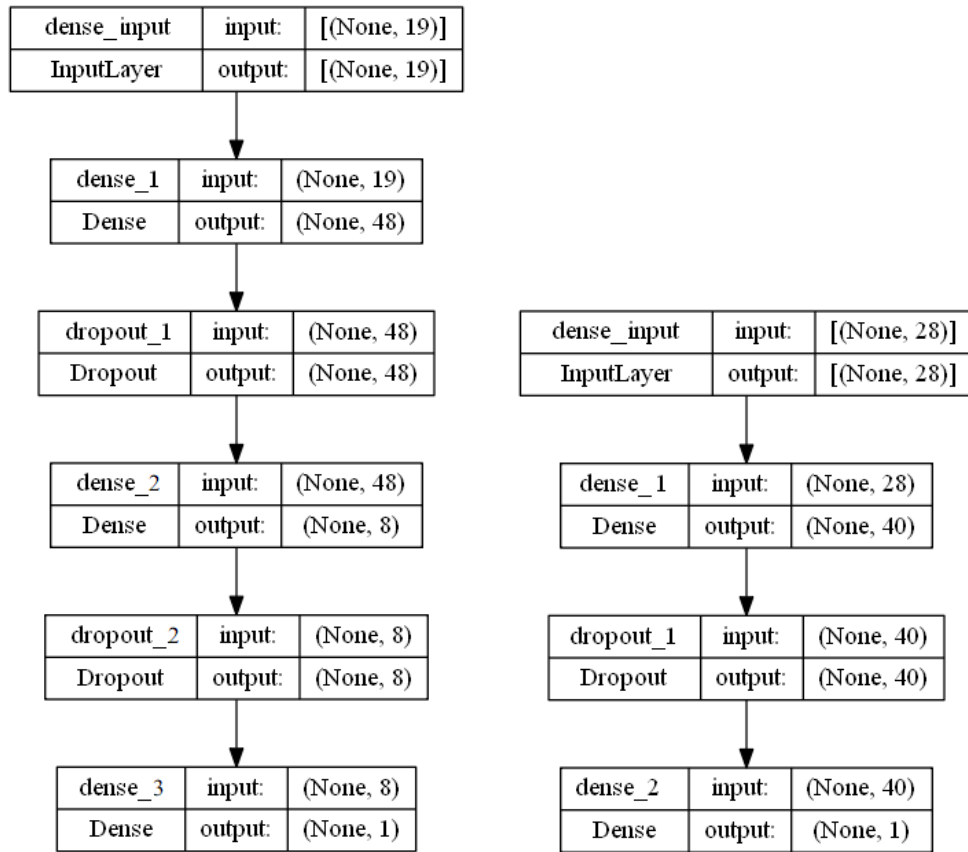
TABLA 3.2. Configuraciones de modelos explorados con la cantidad de neuronas por capa oculta para el conjunto de datos de MAPA.

Capa 1	Capa 2	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	AUC
50	25	0.935 ± 0.021	0.902 ± 0.037	0.977 ± 0.014	0.968 ± 0.016
49	24	0.930 ± 0.026	0.900 ± 0.044	0.971 ± 0.005	0.964 ± 0.010
48	24	0.931 ± 0.022	0.901 ± 0.033	0.971 ± 0.011	0.967 ± 0.009
47	23	0.924 ± 0.017	0.890 ± 0.028	0.971 ± 0.015	0.963 ± 0.008
46	23	0.922 ± 0.025	0.888 ± 0.040	0.968 ± 0.011	0.953 ± 0.011
45	22	0.924 ± 0.024	0.889 ± 0.041	0.973 ± 0.009	0.961 ± 0.011
44	22	0.921 ± 0.019	0.877 ± 0.034	0.982 ± 0.015	0.958 ± 0.016
43	21	0.933 ± 0.017	0.902 ± 0.025	0.973 ± 0.009	0.963 ± 0.012
42	21	0.922 ± 0.015	0.887 ± 0.021	0.968 ± 0.004	0.962 ± 0.010
41	20	0.920 ± 0.021	0.881 ± 0.034	0.973 ± 0.011	0.961 ± 0.011
40	20	0.920 ± 0.028	0.880 ± 0.049	0.977 ± 0.014	0.958 ± 0.017
39	19	0.923 ± 0.024	0.891 ± 0.033	0.966 ± 0.014	0.961 ± 0.015
38	19	0.923 ± 0.022	0.888 ± 0.036	0.971 ± 0.011	0.956 ± 0.009
37	18	0.929 ± 0.012	0.898 ± 0.024	0.968 ± 0.013	0.964 ± 0.012
36	18	0.928 ± 0.016	0.899 ± 0.032	0.966 ± 0.012	0.960 ± 0.006
35	17	0.925 ± 0.023	0.899 ± 0.041	0.962 ± 0.006	0.960 ± 0.012
34	17	0.910 ± 0.008	0.866 ± 0.018	0.971 ± 0.017	0.940 ± 0.011
33	16	0.920 ± 0.014	0.885 ± 0.022	0.966 ± 0.007	0.965 ± 0.013
32	16	0.921 ± 0.012	0.885 ± 0.021	0.968 ± 0.013	0.959 ± 0.011
31	15	0.920 ± 0.020	0.881 ± 0.030	0.973 ± 0.013	0.959 ± 0.009
30	15	0.921 ± 0.017	0.889 ± 0.028	0.964 ± 0.011	0.959 ± 0.012
29	14	0.923 ± 0.020	0.891 ± 0.031	0.966 ± 0.007	0.961 ± 0.016
28	14	0.930 ± 0.034	0.901 ± 0.053	0.971 ± 0.014	0.965 ± 0.021
27	13	0.920 ± 0.027	0.887 ± 0.046	0.966 ± 0.016	0.957 ± 0.012
26	13	0.909 ± 0.023	0.868 ± 0.037	0.966 ± 0.016	0.953 ± 0.019
25	12	0.922 ± 0.014	0.887 ± 0.025	0.969 ± 0.027	0.957 ± 0.012
24	12	0.916 ± 0.013	0.878 ± 0.021	0.968 ± 0.008	0.952 ± 0.010
23	11	0.919 ± 0.017	0.884 ± 0.028	0.966 ± 0.012	0.950 ± 0.015
22	11	0.920 ± 0.019	0.885 ± 0.027	0.966 ± 0.007	0.953 ± 0.013
21	10	0.916 ± 0.011	0.878 ± 0.022	0.968 ± 0.013	0.954 ± 0.013
20	10	0.920 ± 0.023	0.884 ± 0.035	0.968 ± 0.017	0.945 ± 0.026

TABLA 3.3. Configuraciones de modelos explorados con la cantidad de neuronas en la capa oculta para el conjunto de datos de MAPA y datos clínicos.

Capa 1	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	AUC
50	0.896 ± 0.029	0.851 ± 0.032	0.931 ± 0.048	0.960 ± 0.016
49	0.909 ± 0.045	0.864 ± 0.052	0.947 ± 0.050	0.963 ± 0.021
48	0.868 ± 0.014	0.839 ± 0.031	0.874 ± 0.028	0.942 ± 0.023
47	0.913 ± 0.025	0.877 ± 0.026	0.937 ± 0.048	0.963 ± 0.020
46	0.878 ± 0.036	0.836 ± 0.047	0.905 ± 0.045	0.953 ± 0.020
45	0.921 ± 0.028	0.879 ± 0.032	0.956 ± 0.036	0.968 ± 0.007
44	0.952 ± 0.013	0.911 ± 0.024	0.991 ± 0.012	0.977 ± 0.014
43	0.950 ± 0.019	0.899 ± 0.033	0.990 ± 0.016	0.985 ± 0.005
42	0.961 ± 0.023	0.924 ± 0.034	0.994 ± 0.013	0.982 ± 0.005
41	0.956 ± 0.026	0.915 ± 0.044	0.997 ± 0.023	0.985 ± 0.003
40	0.971 ± 0.016	0.939 ± 0.032	0.991 ± 0.004	0.990 ± 0.007
39	0.935 ± 0.039	0.897 ± 0.034	0.966 ± 0.061	0.974 ± 0.018
38	0.943 ± 0.012	0.897 ± 0.014	0.984 ± 0.031	0.979 ± 0.005
37	0.939 ± 0.036	0.900 ± 0.033	0.969 ± 0.062	0.974 ± 0.013
36	0.931 ± 0.037	0.887 ± 0.035	0.969 ± 0.055	0.973 ± 0.013
35	0.889 ± 0.039	0.846 ± 0.037	0.918 ± 0.062	0.952 ± 0.023
34	0.900 ± 0.031	0.847 ± 0.039	0.950 ± 0.033	0.953 ± 0.023
33	0.860 ± 0.042	0.821 ± 0.054	0.880 ± 0.068	0.945 ± 0.022
32	0.871 ± 0.034	0.824 ± 0.048	0.901 ± 0.076	0.944 ± 0.023
31	0.909 ± 0.048	0.869 ± 0.032	0.934 ± 0.087	0.958 ± 0.024
30	0.909 ± 0.048	0.869 ± 0.032	0.934 ± 0.087	0.958 ± 0.024

La figura 3.10 muestra la arquitectura final utilizada para cada conjunto de datos. Se observa que se optó por una estructura basada en capas densas, también conocidas como capas completamente conectadas, complementadas con capas de *dropout*. La incorporación de las capas densas permitió capturar relaciones complejas presentes en los datos, mientras que las capas de *dropout* desempeñaron un papel fundamental en la regularización del modelo y en la prevención del sobreajuste. Como resultado, se logró obtener un modelo con una mayor capacidad de generalización y un mejor desempeño en la predicción de datos desconocidos.



(A) Arquitectura del modelo para el conjunto de datos de MAPA.

(B) Arquitectura del modelo para el conjunto de datos de MAPA y datos clínicos.

FIGURA 3.10. Arquitectura elegida para los modelos.

Selección de funciones de activación, función de pérdida y algoritmo de optimización

Se empleó la función de activación ReLU en las capas ocultas de la red neuronal. Esta función es comúnmente elegida debido a su capacidad para introducir no linealidad en el modelo y capturar patrones complejos en los datos. Además, se utilizó la función de activación sigmoide en la capa de salida, ya que el problema abordado se trata de una clasificación binaria. La función sigmoide es adecuada en este contexto, considerando que mapea los valores de salida a un rango entre 0 y 1, lo que se interpreta como la probabilidad de pertenecer a una de las dos clases posibles.

Por otro lado, se optó por utilizar la función de pérdida *binary cross entropy* debido a que el problema en cuestión involucra una clasificación binaria. Esta función de pérdida es adecuada para este tipo de problemas, ya que penaliza de manera efectiva las discrepancias entre las predicciones del modelo y los valores reales. En cuanto al algoritmo de optimización, se eligió Adam debido a su eficiencia en la convergencia y adaptabilidad a diferentes tasas de aprendizaje, como se explicó en la subsección 2.3.5.

Ajuste de hiperparámetros

Una vez definida la arquitectura inicial de la red neuronal, se realizó una búsqueda de hiperparámetros adicionales utilizando el enfoque de búsqueda aleatoria, también conocido como *RandomSearch* [48]. Esto se combinó con una validación cruzada de $k = 5$ iteraciones para evaluar el rendimiento del modelo en diferentes conjuntos de datos y asegurar que los hiperparámetros ajustados generalicen bien. Este ajuste tuvo como objetivo encontrar el equilibrio adecuado entre el rendimiento del modelo y la capacidad de generalización. La tabla 3.4 muestra los hiperparámetros que fueron ajustados y sus mejores valores para cada conjuntos de datos.

TABLA 3.4. Hiperparámetros elegidos para el *dataset 1* (datos MAPA) y *dataset 2* (datos MAPA y datos clínicos).

Hiperparámetros	Valor elegido <i>dataset 1</i>	Valor elegido <i>dataset 2</i>
Tasa de aprendizaje	$1e - 03$	$1e - 2$
Adam ϵ	$1e - 08$	$1e - 7$
Adam β_1	0,9	0,9
Adam β_2	0,999	0,999
Épocas	300	250
Tamaño de lote	32	32
Porcentaje de <i>dropout</i>	20 %	50 %

3.3.3. Estrategias implementadas para evitar el *overfitting*

El *overfitting* es un fenómeno común que se refiere a la situación en la cual un modelo se ajusta excesivamente a los datos de entrenamiento y pierde su capacidad de generalización en datos no vistos. En el contexto de este trabajo, se adoptaron varias estrategias con el objetivo de mitigar el *overfitting* y garantizar la robustez de los modelos.

En primer lugar, se buscó priorizar la selección de modelos más simples, evitando la incorporación de una complejidad excesiva. Esto implicó mantener reducido el número de capas y/o neuronas en cada capa del modelo, con el objetivo de lograr un equilibrio entre la capacidad de aprendizaje y la generalización. Además, como se mencionó en la subsección 3.3.2, se incorporaron capas de *dropout* en el modelo. Esta técnica consistió en apagar de forma aleatoria un porcentaje de las neuronas durante el entrenamiento, para evitar que se ajusten excesivamente a los datos de entrenamiento. Al no ser entrenadas, estas neuronas contribuyen a la regularización del modelo y ayudan a prevenir el *overfitting*. Asimismo, otra estrategia empleada fue el uso de *early stopping*. Esta técnica implica detener el proceso de entrenamiento del modelo en el momento en que se observa un aumento en el valor del error de validación. De este modo, se buscó evitar el ajuste excesivo y lograr un modelo con una precisión óptima en datos no utilizados durante el entrenamiento [38].

Bibliografía

- [1] OpenCourseWare. *Tema 6. Circulación arterial. Presión arterial*.
<https://ocw.unican.es/mod/page/view.php?id=510>. Jun. de 2017.
(Visitado 09-05-2023).
- [2] Antonio López Farré y Carlos Macaya Miguel. «Libro de la salud cardiovascular del hospital clínico San Carlos y la fundación BBVA». En: Editorial Nerea, S.A., 2009, págs. 121-129.
- [3] John E. Hall. «Guyton y Hall: Tratado De Fisiologia Medica». En: ELSEVIER ESPAÑA, S.A., 2011, págs. 157-165.
- [4] Gustavo Blanco. «MEDICIÓN DE LA PRESIÓN ARTERIAL». En: *Hipertension Arterial, Epidemiología, Fisiología, Fisiopatología, Diagnóstico y Tratamiento*. Ed. por SAHA. Buenos Aires, 2013. Cap. 70, págs. 341-344.
- [5] Gabriel Darío Waisman Paula Edit Cuffaro Margarita Susana Morales. «MONITOREO AMBULATORIO DE PRESIÓN ARTERIAL». En: *Hipertension Arterial, Epidemiología, Fisiología, Fisiopatología, Diagnóstico y Tratamiento*. Ed. por SAHA. Buenos Aires, 2013. Cap. 82, págs. 391-395.
- [6] Geoffrey A. Head et al. «Ambulatory blood pressure monitoring». En: *Australian family physician* (2011).
- [7] Yihong Zhao, Eric P. Wood et al. «Social determinants in machine learning cardiovascular disease prediction models: A systematic review». En: *American Journal of Preventive Medicine* 61.4 (2021), págs. 596-605. DOI: [10.1016/j.amepre.2021.04.016](https://doi.org/10.1016/j.amepre.2021.04.016).
- [8] Hospital Alemán Asociación Civil. *Institucional. Acerca del HA*.
hospitalaleman.org.ar. Jun. de 2017. (Visitado 01-11-2022).
- [9] Kevin E. Kip, Kimberly Hollabaugh et al. «The problem with composite end points in cardiovascular studies: the story of major adverse cardiac events and percutaneous coronary intervention». En: *Journal of the American College of Cardiology* 51.7 (2008), págs. 701-707. DOI: [10.1016/j.jacc.2007.10.034](https://doi.org/10.1016/j.jacc.2007.10.034).
- [10] Zhi Huang, Tak-Ming Chan y Wei Dong. «MACE prediction of acute coronary syndrome via boosted resampling classification using electronic medical records». En: *Journal of biomedical informatics* 66 (2017), págs. 161-170. DOI: [10.1016/j.jbi.2017.01.001](https://doi.org/10.1016/j.jbi.2017.01.001).
- [11] Pei-I Zhang, Chien-Chin Hsu et al. «Real-time AI prediction for major adverse cardiac events in emergency department patients with chest pain». En: *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine* 28 (2020). DOI: [10.1186/s13049-020-00786-x](https://doi.org/10.1186/s13049-020-00786-x).
- [12] Jinwan Wang, Shuai Wang, Mark Xuefang Zhu et al. «Risk Prediction of Major Adverse Cardiovascular Events Occurrence Within 6 Months After Coronary Revascularization: Machine Learning Study». En: *JMIR Med Inform* 10.4 (2020). DOI: [10.2196/33395](https://doi.org/10.2196/33395).
- [13] Bobak J. Mortazavi, Nicholas S. Downing et al. «Analysis of machine learning techniques for heart failure readmissions». En: *Circulation*:

- Cardiovascular Quality and Outcomes* 9.6 (2016). DOI: [10.1161/CIRCOUTCOMES.116.003039](https://doi.org/10.1161/CIRCOUTCOMES.116.003039).
- [14] Adriaan A Voors, Wouter Ouwerkerk et al. «Development and validation of multivariable models to predict mortality and hospitalization in patients with heart failure». En: *European Journal of Heart Failure* 19.5 (2017). DOI: [10.1002/ehf.785](https://doi.org/10.1002/ehf.785).
 - [15] Oluwarotimi Williams Samuel, Grace Mojisola Asogbon et al. «An integrated decision support system based on ANN and Fuzzy AHP for heart failure risk prediction». En: *Expert Systems with Applications* 68 (2017). DOI: [10.1016/j.eswa.2016.10.020](https://doi.org/10.1016/j.eswa.2016.10.020).
 - [16] Stephen F Weng, Jenna Reps et al. «Can machine-learning improve cardiovascular risk prediction using routine clinical data?» En: *PloS one* 12.4 (2017). DOI: [10.1371/journal.pone.0174944](https://doi.org/10.1371/journal.pone.0174944).
 - [17] Garrett S Bowen, Michelle S Diop et al. «A Multivariable Prediction Model for Mortality in Individuals Admitted for Heart Failure». En: *Journal of the American Geriatrics Society* 66.5 (2018). DOI: [10.1111/jgs.15319](https://doi.org/10.1111/jgs.15319).
 - [18] Giulia Lorenzoni, Stefano Santo Sabato et al. «Comparison of Machine Learning Techniques for Prediction of Hospitalization in Heart Failure Patients». En: *Journal of Clinical Medicine* 8.9 (2019). DOI: [10.3390/jcm8091298](https://doi.org/10.3390/jcm8091298).
 - [19] Joon Myoung Kwon, Kyung Hee Kim et al. «Development and Validation of Deep-Learning Algorithm for Electrocardiography-Based Heart Failure Identification». En: *Korean circulation journal* 49.7 (2019). DOI: [10.4070/kcj.2018.0446](https://doi.org/10.4070/kcj.2018.0446).
 - [20] Joon Myoung Kwon, Kyung Hee Kim et al. «Artificial intelligence algorithm for predicting mortality of patients with acute heart failure». En: *PLoS One* 14.7 (2019). DOI: [10.1371/journal.pone.0219302](https://doi.org/10.1371/journal.pone.0219302).
 - [21] Joon Myoung Kwon, Kyung Hee Kim et al. «Deep learning for predicting in-hospital mortality among heart disease patients based on echocardiography». En: *Echocardiography* 36.2 (2019). DOI: [10.1111/echo.14220](https://doi.org/10.1111/echo.14220).
 - [22] Na-Kyung Lim, Seung-Eun Lee et al. «Risk prediction for 30-day heart failure-specific readmission or death after discharge: data from the Korean Acute Heart Failure (KorAHF) registry». En: *Journal of Cardiology* 73.2 (2019). DOI: [10.1016/j.jjcc.2018.07.009](https://doi.org/10.1016/j.jjcc.2018.07.009).
 - [23] Rui Chen, Aijia Lu et al. «Using machine learning to predict one-year cardiovascular events in patients with severe dilated cardiomyopathy». En: *European Journal of Radiology* 117 (2019), págs. 178-183. DOI: [10.1016/j.ejrad.2019.06.004](https://doi.org/10.1016/j.ejrad.2019.06.004).
 - [24] Saqib Ejaz Awan, Mohammed Bennamoun et al. «Machine learning-based prediction of heart failure readmission or death: implications of choosing the right model and the right metrics». En: *ESC heart failure* 6.2 (2019). DOI: [10.1002/ehf2.12419](https://doi.org/10.1002/ehf2.12419).
 - [25] Saqib Ejaz Awan, Mohammed Bennamoun et al. «Machine learning-based prediction of heart failure readmission or death: implications of choosing the right model and the right metrics». En: *ESC heart failure* 6.2 (2019). DOI: [10.1002/ehf2.12419](https://doi.org/10.1002/ehf2.12419).
 - [26] Davide Chicco y Giuseppe Jurman. «Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone». En: *BMC Medical Informatics and Decision Making* 20.1 (2020). DOI: [10.1186/s12911-020-1023-5](https://doi.org/10.1186/s12911-020-1023-5).

- [27] Eric D Adler, Adriaan A Voors et al. «Improving risk prediction in heart failure using machine learning». En: *European Journal of Heart Failure* 22.1 (2020). DOI: [10.1002/ehf.1628](https://doi.org/10.1002/ehf.1628).
- [28] Ashir Javeed, Sanam Shahla Rizvi et al. «Heart Risk Failure Prediction Using a Novel Feature Selection Method for Feature Refinement and Neural Network for Classification». En: *Mobile Information Systems* (2020). DOI: [10.1155/2020/8843115](https://doi.org/10.1155/2020/8843115).
- [29] Sarah Cohen, Aihua Liu et al. «Risk prediction models for heart failure admissions in adults with congenital heart disease». En: *International Journal of Cardiology* 322 (2021), págs. 149-157. DOI: [10.1016/j.ijcard.2020.08.039](https://doi.org/10.1016/j.ijcard.2020.08.039).
- [30] Shishir Rao, Yikuan Li et al. «An Explainable Transformer-Based Deep Learning Model for the Prediction of Incident Heart Failure». En: *IEEE Journal of Biomedical and Health Informatics* 26.7 (2022). DOI: [10.1109/JBHI.2022.3148820](https://doi.org/10.1109/JBHI.2022.3148820).
- [31] Yoshua Bengio y Aaron Courville Ian Goodfellow. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [32] Andrew Ng. *Machine Learning Yearning: Technical Strategy for AI Engineers, In the Era of Deep Learning*. Kindle Edition, 2019.
- [33] Robert Tibshirani y Jerome Friedman Trevor Hastie. *Machine Learning Yearning: Technical Strategy for AI Engineers, In the Era of Deep Learning*. Springer, 2009.
- [34] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [35] E. DeRouin y J. Brown. «Neural Network Training on Unequally Represented Classes». En: *Intelligent Engineering Systems Through Artificial Neural Networks*. ASME Press, 1991, págs. 135-140.
- [36] Nitesh V. Chawla, Kevin W. Bowyer et al. «SMOTE: Synthetic Minority Over-sampling Technique». En: *Journal Of Artificial Intelligence Research* 16 (2002), págs. 321-357. DOI: [10.1613/jair.953](https://doi.org/10.1613/jair.953).
- [37] Haibo He y Yunqian Ma. *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley-IEEE Press, 2013.
- [38] Aurélien Geron. *Hands-On Machine Learning with Scikit-Learn & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly, 2017.
- [39] Robert Challen, Joshua Denny, Martin Pitt, Luke Gompels, Tom Edwards y Krasimira Tsaneva-Atanasova. «Artificial intelligence, bias and clinical safety». En: *BMJ Quality & Safety* 28.3 (2019), págs. 231-237. DOI: [10.1136/bmjqs-2018-008370](https://doi.org/10.1136/bmjqs-2018-008370). URL: <https://qualitysafety.bmj.com/content/28/3/231>.
- [40] Warren McCulloch y Walter Pitts. «A logical calculus of the ideas imminent in nervous activity». En: *Bulletin of Mathematical Biology* 5 (1943), págs. 115-133. DOI: [10.1136/bmjqs-2018-008370](https://doi.org/10.1136/bmjqs-2018-008370).
- [41] Frank Rosenblatt. «The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain». En: *Psychological Review* 65.6 (1958), págs. 386-408. DOI: [10.1037/h0042519](https://doi.org/10.1037/h0042519).
- [42] Young-Seuk Park y Sovan Lek. «Artificial Neural Networks». En: *Ecological Model Types*. Elsevier, 2016, págs. 123-140. DOI: [10.1016/b978-0-444-63623-2.00007-4](https://doi.org/10.1016/b978-0-444-63623-2.00007-4).
- [43] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever y Ruslan Salakhutdinov. «Dropout: A Simple Way to Prevent Neural Networks from Overfitting». En: *Journal of Machine Learning Research* 15.56

- (2014), págs. 1929-1958. URL:
<http://jmlr.org/papers/v15/srivastava14a.html>.
- [44] Dennis Banga y Peter Wagacha. *Abnormality Detection in Musculoskeletal Radiographs with Convolutional Neural Networks(Ensembles) and Performance Optimization*. Ago. de 2019.
- [45] David E. Rumelhart y James L. McClelland. «Learning Internal Representations by Error Propagation». En: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. MIT Press, 1987, págs. 318-362.
- [46] Leticia Ochoa. «Evaluación de Algoritmos de Clasificación utilizando Validación Cruzada». En: ene. de 2019. DOI:
[10.18687/LACCEI2019.1.1.471](https://doi.org/10.18687/LACCEI2019.1.1.471).
- [47] Diederik P. Kingma y Jimmy Ba. «Adam: A method for stochastic optimization». En: *arXiv preprint arXiv:1412.6980* (2014).
- [48] F. Pedregosa, G. Varoquaux, A. Gramfort et al. «Scikit-learn: Machine Learning in Python». En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.