



## 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

<b>Resumen</b>	<b>I</b>
<b>1. Introducción general</b>	<b>1</b>
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
<b>2. Introducción específica</b>	<b>7</b>
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	9
<i>Precision</i>	11
<i>Recall</i>	11
<i>F1-Score</i>	11
<i>Accuracy</i>	11
<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	13
2.3.3. Algoritmo de entrenamiento	15
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	18
<b>Bibliografía</b>	<b>19</b>



# Índice de figuras

1.1. Representación esquemática del pulso de presión registrado en la aorta ascendente <sup>1</sup> . . . . .	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 <sup>2</sup> . . . . .	10
2.3. Matriz de confusión. . . . .	10
2.4. Representación gráfica de la curva ROC y las distribuciones de TP y TN <sup>3</sup> . . . . .	12
2.5. Representación de un perceptrón. . . . .	14
2.6. Representación de un perceptrón multicapa <i>fully connected</i> . . . . .	14
2.7. Representación esquemática de <i>k-fold cross validation</i> con $k = 5$ <sup>4</sup> . . .	17





# Índice de tablas

1.1. Clasificación de la presión arterial por niveles . . . . .	2
1.2. Valores de referencia para definir HTA por MAPA . . . . .	3



# 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:

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

---

<sup>1</sup>Imagen 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 ascendente<sup>1</sup>.

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 niveles<sup>2</sup>.

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

<sup>2</sup>Tabla 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 MAPA<sup>3</sup>.

	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].

<sup>3</sup>Tabla realizada con información de Cuffaro, P. E., Morales, M. S. y Waisman, G. D. (2013). MONITOREO AMBULATORIO DE PRESIÓN ARTERIAL. En SAHA (Ed.), Hipertensión 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

ESCRIBIR PARRAFO INTRODUCTORIO

### 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.

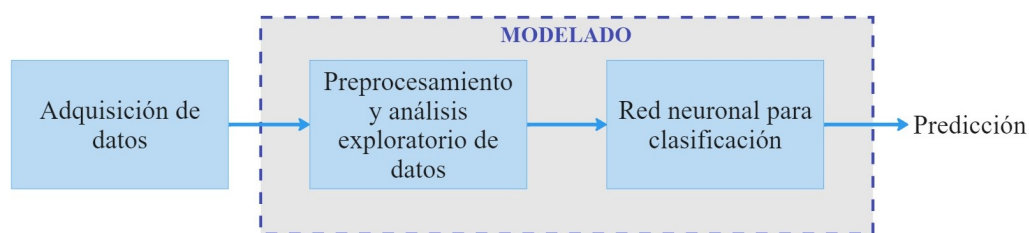


FIGURA 2.1. Representación esquemática de las etapas del trabajo.

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 modelo se enriqueció al incluir también 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 contribuyó a mejorar las métricas de desempeño en la predicción de MACE, y si podría tener 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. 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 implicar réplicas de ejemplos de la clase minoritaria, lo cual puede generar un sobreajuste. En concreto, el modelo 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].

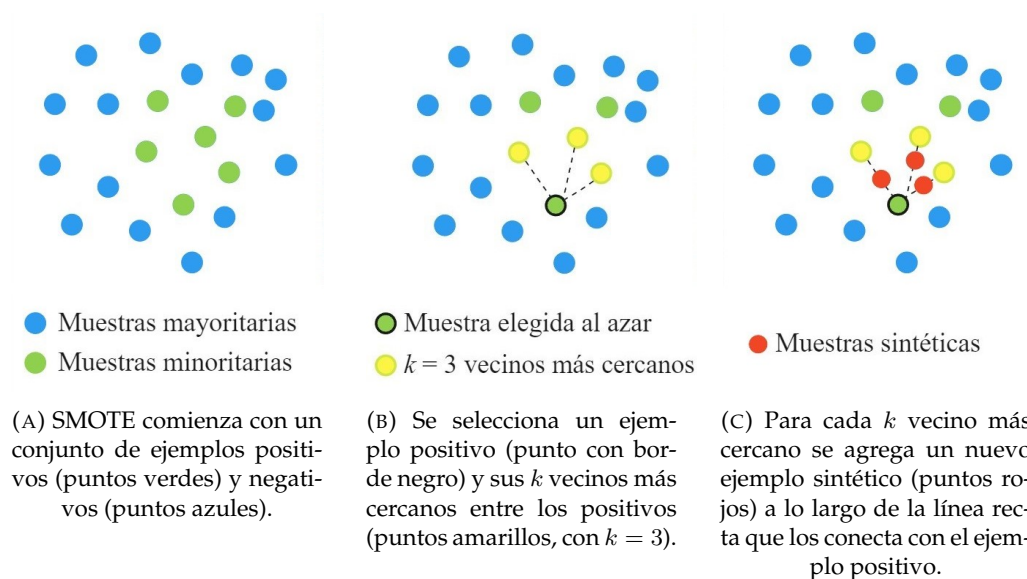
En suma, SMOTE permite generar ejemplos sintéticos que sean relativamente cercanos en el espacio de características con respecto a las muestras existentes de la clase minoritaria. Además, las muestras pueden crearse en la cantidad que sea requerida. Por lo tanto, en escenarios donde el conjunto de datos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

### 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

---

<sup>1</sup>Imagen 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.

FIGURA 2.2. Representación esquemática del algoritmo SMOTE<sup>1</sup>.

columnas representan la clase predicha y las filas corresponden a las clases reales. 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.

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

FIGURA 2.3. Matriz de confusión.

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 [38].

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 explican algunas en detalle.

### **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 siguiente ecuación matemática describe este comportamiento:

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

### **F1-Score**

La métrica *F1-Score* es 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:

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

### **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.

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.

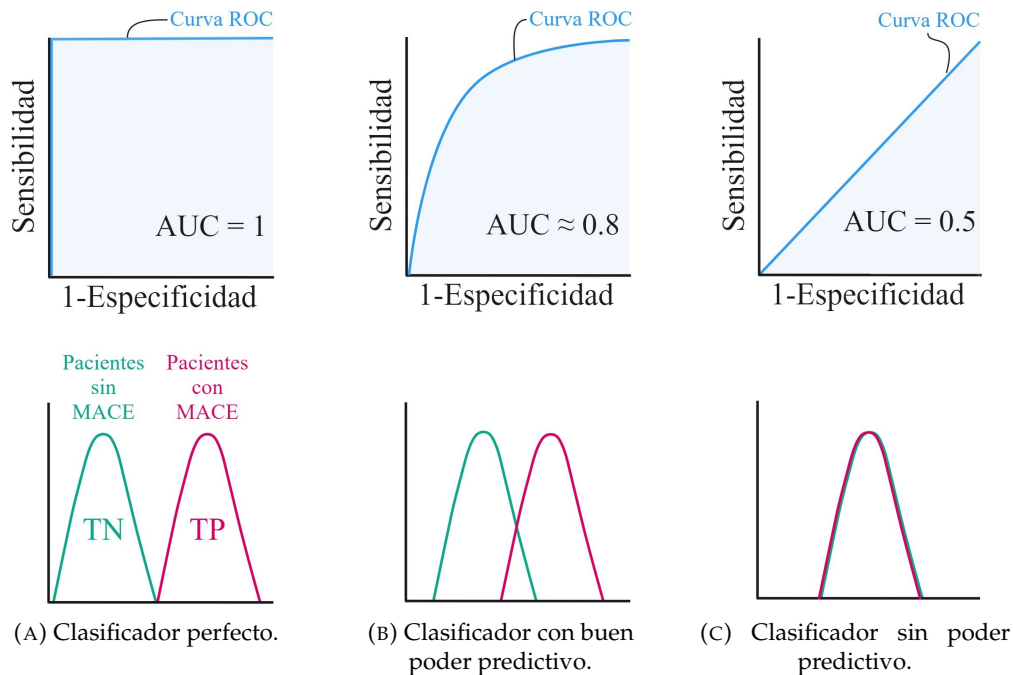


FIGURA 2.4. Representación gráfica de la curva ROC y las distribuciones de TP y TN<sup>2</sup>.

Para el presente trabajo, se evaluaron todas las métricas mencionadas anteriormente. Sin embargo, se seleccionó el AUC como la métrica principal para medir el desempeño de los modelos. Esto se debe a que en el ámbito clínico resulta de gran relevancia analizar la relación de compromiso entre la sensibilidad y especificidad. En definitiva, se buscó encontrar un equilibrio óptimo entre la detección temprana de casos de MACE positivos (alta sensibilidad) y la minimización de falsas alarmas (alta especificidad).

## 2.3. Perceptrón multicapa

Las redes neuronales artificiales (RNA), propuestas inicialmente por McCulloch y Pitts en 1943 [39], 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 [40]. 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 ( $\sum_{i=1}^n W_i^T \cdot X_i$ ) 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] [41]. La descripción matemática de este comportamiento se expresa de la siguiente manera:

$$\begin{aligned}\hat{y} &= g(z) = g\left(\sum_{i=1}^n W_i^T \cdot X + b\right) \\ \hat{y} &= g(x_1 \cdot w_1 + x_2 \cdot w_2 + \dots + x_n \cdot w_n + b)\end{aligned}\tag{2.6}$$

### 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*

<sup>2</sup>Imagen adaptada de Glen, S. (2019). ROC Curve Explained in One Picture. Data Science Central. <https://www.datasciencecentral.com/roc-curve-explained-in-one-picture/>



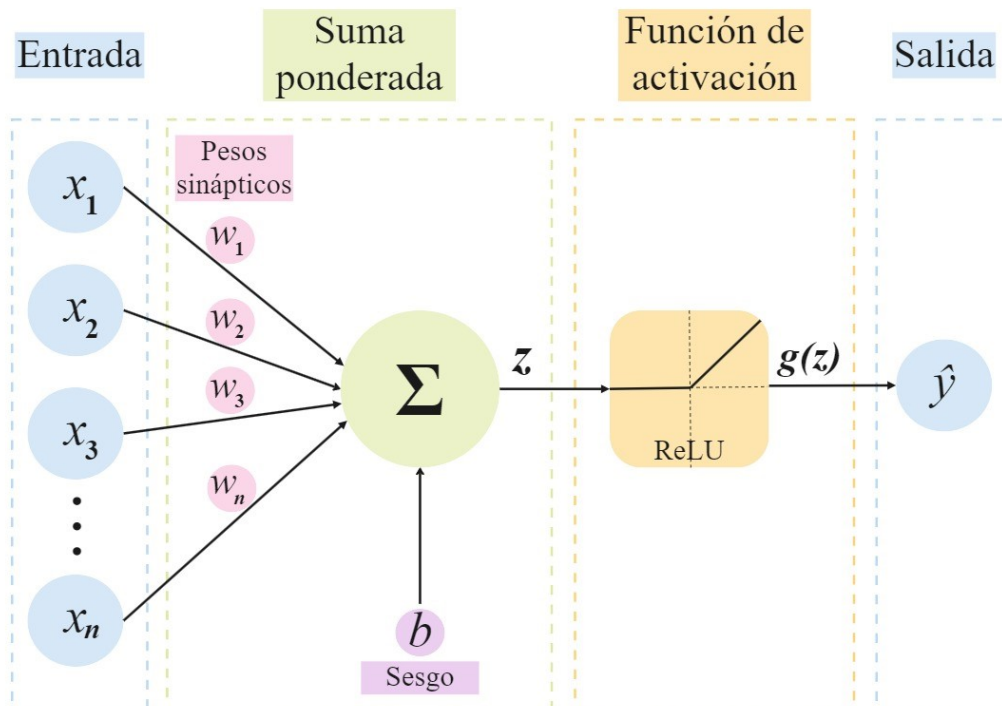
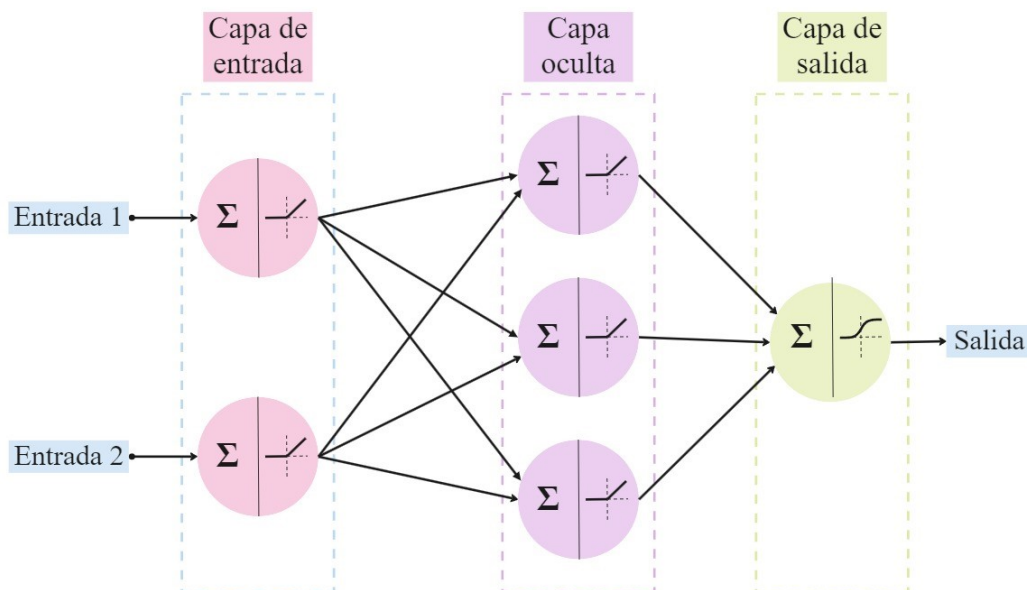


FIGURA 2.5. Representación de un perceptrón.

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] [42].

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



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[41].

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 [42] [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  $z$  [42].

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

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 1. Así, la función sigmoide es útil para predecir la probabilidad de una variable binaria [42].

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

### 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] [42].

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

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  $\lambda\Omega(w)$ .

---

**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.

---

### 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] [41] [42].

**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 y 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 [41]. 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.

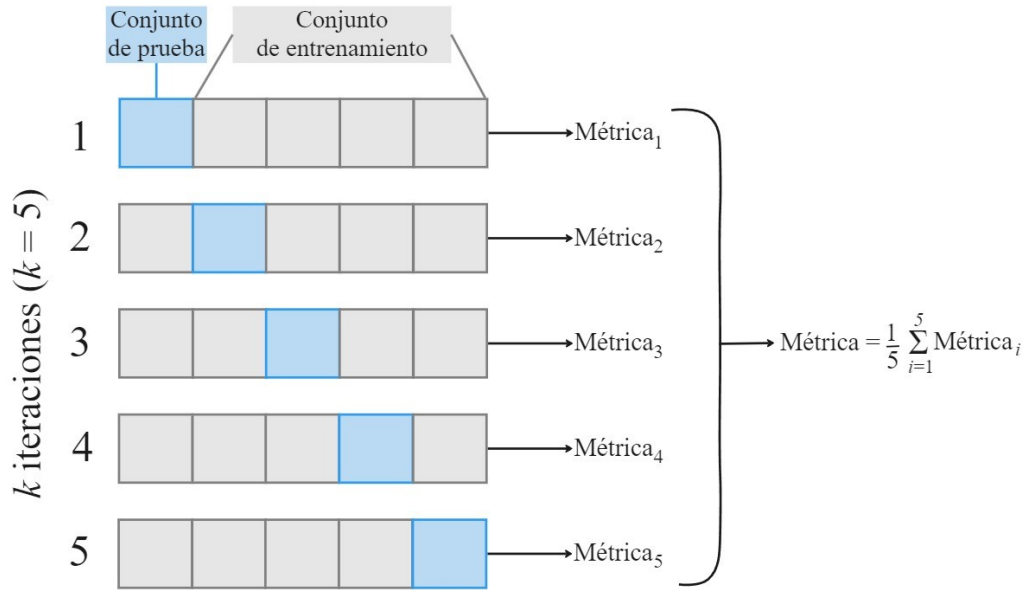


FIGURA 2.7. Representación esquemática de *k-fold cross validation* con  $k = 5$ <sup>3</sup>.

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

<sup>3</sup>Imagen adaptada de Ashfaq Maat, Johar y Iqbal, Amer. (2019). Introduction to Support Vector Machines and Kernel Methods.

el conjunto de entrenamiento. Al finalizar las  $k$  iteraciones, se calcula el promedio de las métricas y error obtenidos para cada subconjunto de prueba [46].

### 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 un 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 [42].

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.
- 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 [42].

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] [41] [42].

# 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](https://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 Reys 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] 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>.
- [39] 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).
- [40] 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).
- [41] 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).
- [42] Aurélien Geron. *Hands-On Machine Learning with Scikit-Learn & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly, 2017.
- [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).