



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CIENCIAS

MODELOS MATEMÁTICOS DE CLASIFICACIÓN AUTOMÁTICA DE
PACIENTES DE PARKINSON USANDO ANÁLISIS DE DATOS
ADQUIRIDOS MEDIANTE TOMOGRAFÍA POR EMISIÓN DE
POSITRONES

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADA EN MATEMÁTICAS

P R E S E N T A

NANANCI ESTELA MARTÍNEZ ZALDÍVAR



DIRECTOR DE TESIS:

DR. ARTURO AVENDAÑO ESTRADA

CIUDAD UNIVERSITARIA, CD. MX., 2019



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

1. Datos del alumno Apellido paterno Apellido materno Nombre(s) Teléfono Universidad Nacional Autónoma de México Facultad Carrera Número de cuenta	Martínez Zaldívar Nananci Estela 55 29 71 45 45 Ciencias Matemáticas 30105322-8
2. Datos del tutor Grado Nombre(s) Apellido paterno Apellido materno	Dr. Arturo Avendaño Estrada
3. Datos del sinodal 1 Grado Nombre(s) Apellido paterno Apellido materno	Dr. Miguel Ángel Ávila Rodríguez
4. Datos del sinodal 2 Grado Nombre(s) Apellido paterno Apellido materno	Dr. Pablo Barrera Sánchez
5. Datos del sinodal 3 Grado Nombre(s) Apellido paterno Apellido materno	Dr. Fernando Brambila Paz
6. Datos del sinodal 4 Grado Nombre(s) Apellido paterno Apellido materno	M.C. Belén Rivera Bravo
7. Datos del trabajo escrito Título Número de páginas Año	Modelos matemáticos de clasificación automática de pacientes de Parkinson usando análisis de datos adquiridos mediante tomografía por emisión de positrones 98 2019

AGRADECIMIENTOS:

A la Universidad Nacional Autónoma de México, casa abierta al mundo y forjadora de hombres útiles a la humanidad. Mi casa.

A la Facultad de Ciencias, germinadora de semillas curiosas e inquietas, por enseñarme la obligación de preguntar “¿Por qué?” y recompensarme con la respuesta.

Al Doctor Pablito Barrera por todo lo que significa para cada alumno que ha pasado por sus aulas, especialmente para mí.

Al Doctor Fernando Brambila, ejemplo de tenacidad y humildad con sus educandos.

A todos aquellos profesores que con su gusto por la docencia me mantuvieron en el camino aun cuando fuese sinuoso y escarpado.

Al Dr. Arturo Avendaño, energía y guía en este proyecto, por su incalculable generosidad y paciencia en esta aventura, mostrándome con confianza y visión, un camino para mí desconocido.

A mis muy estimados sinodales por el apoyo que me brindaron en la culminación de este trabajo.

A Kevin Markham por compartir su pasión por la enseñanza con el mundo.

Al Dr. David Kershenobich, Director del INCMNSZ, quien con humanidad es puente entre ciencia y medicina.

A las personas que más quiero: mi familia. A mis hermanos Luis y Bolívar, porque aprendimos juntos de la manera más divertida. Gracias a mis incondicionales padres María Estela y Luis Andrés por el amor infinito, por darnos sangre compatible con la ciencia y enseñarnos a usarla. Gracias por la feliz vida que nos han regalado, por el increíble esfuerzo de educarnos, impulsarnos y enseñarnos que todo es posible.

A ti Emi, que con tu alegría me haces revivir la feliz infancia. Crecerás, es inevitable.

A mi adorado Rorri por todo el amor y la paciencia, por el apoyo y la confianza inquebrantable.

ÍNDICE

RESUMEN	6
GLOSARIO	8
1 INTRODUCCIÓN	9
2 ANTECEDENTES	13
2.1 TOMOGRAFÍA POR EMISIÓN DE POSITRONES (PET)	14
2.2 INTELIGENCIA ARTIFICIAL	15
2.2.1 ANTECEDENTES HISTÓRICOS.....	15
2.2.2 ESTRUCTURA NATURAL Y ARTIFICIAL DE LA INTELIGENCIA	17
2.3 APRENDIZAJE AUTOMÁTICO.....	19
2.3.1 CONCEPTOS BÁSICOS	19
2.3.2 USOS Y APLICACIONES.....	21
2.3.3 APRENDIZAJE AUTOMÁTICO EN MEDICINA.....	25
2.3.4 TÉCNICAS DE APRENDIZAJE AUTOMÁTICO	26
2.3.4.2 APRENDIZAJE NO SUPERVISADO.....	31
2.3.5 MODELOS DE APRENDIZAJE AUTOMÁTICO EN MEDICINA	31
2.3.6 DESARROLLO DE UN MODELO DE APRENDIZAJE AUTOMÁTICO	39
2.3.7 ANÁLISIS DE DATOS Y EXTRACCIÓN DE CARACTERÍSTICAS.....	40
2.3.8 EVALUACIÓN DE UN CLASIFICADOR.....	44
2.4 MODELOS LINEALES GENERALIZADOS.....	49
2.5 REDES NEURONALES ARTIFICIALES.....	55
3 OBJETIVO	65
4 MATERIALES Y MÉTODOS.....	66
4.1 BASE DE DATOS.....	67
4.2 DISCRIMINACIÓN DE DATOS Y CLASIFICADORES EMPLEADOS.....	70
4.3 EVALUACIÓN Y OPTIMIZACIÓN DEL ALGORITMO	75
5 RESULTADOS	77
5.1.1 Clasificador por Redes Neuronales Artificiales (RNA) para el Radiofármaco ¹¹ C-DTBZ.....	80
5.1.2 Clasificador por Redes Neuronales Artificiales (RNA) para el Radiofármaco ¹¹ C-RAC.....	82

5.1.3	Clasificador por Redes Neuronales Artificiales (MLP) para los datos combinados de ^{11}C -DTBZ y ^{11}C -RAC	84
5.1.4	Clasificador por Regresión logística de Modelos Lineales Generalizados (GLM) para el Radiofármaco ^{11}C -DTBZ	86
5.1.5	Clasificador por Regresión logística de Modelos Lineales Generalizados (GLM) para el Radiofármaco ^{11}C -RAC.....	88
5.1.6	Clasificador por Regresión logística de Modelos Lineales Generalizados (GLM) para los datos combinados de ^{11}C -DTBZ/ ^{11}C -RAC	90
6	DISCUSIÓN Y CONCLUSIONES	92
6.1	TRABAJO FUTURO	94
7	BIBLIOGRAFÍA.....	95

RESUMEN

La enfermedad de Parkinson (EP) es diagnosticada mediante pruebas clínicas que incluyen la evaluación del comportamiento, estado de ánimo, de actividades de la vida diaria y evaluación de problemas motores, se evalúa a través de escalas como la “Unified Parkinson’s Disease Rating Scale” (UPDRS). Aunque las pruebas clínicas son las más usadas a nivel mundial, es altamente susceptible a errores que se ven reflejados en un mal diagnóstico. Este trabajo propone un método auxiliar de diagnóstico de la EP con el uso de Inteligencia Artificial y aprendizaje automático supervisado, empleando un algoritmo desarrollado en el lenguaje de programación Python basado en Redes Neuronales Artificiales para el análisis de datos obtenidos de imágenes adquiridas mediante tomografías por emisión de positrones (PET, por sus siglas en inglés). Los resultados obtenidos de la clasificación mediante el análisis cuantitativo de los datos usando el algoritmo de aprendizaje automático, muestran un alto porcentaje de precisión de clasificación, siendo el mejor desempeño 90% de precisión en predicción al emplear los radiofármacos ^{11}C -DTBZ y ^{11}C -RAC analizados como parámetros complementarios.

ABSTRACT

Parkinson's disease (PD) is diagnosed through clinical test that includes evaluation of mentation, behavior and mood, evaluation of daily life activities and evaluation of motor problems, is evaluated through scales such as "Unified Parkinson's Disease Classification Scale" (UPDRS). Although this scale is the most used worldwide, it is also highly susceptible to errors that are reflected in misdiagnosis. This work proposes an auxiliary method of clasification of PD through Artificial Intelligence and supervised Machine Learning. The algorithm developed in Python and Artificial Neural Networks analyze data obtained from images acquired by positron emission tomography (PET). The results of the classification by quantitative analysis of data shows that the algorithm reaches 90% of accuracy in the prediction using ^{11}C -DTBZ and ^{11}C -RAC as complemetary feeding parameters.

PET	Tomografía por Emisión de Positrones
AA	Aprendizaje Automático (Machine Learning)
RNA / ANN	Redes Neuronales Artificiales
MLP	Perceptrón Multicapa
DTBZ	Radiofármaco pre-sináptico
RAC	Radiofármaco post-sináptico
SE	Sistemas Expertos. Programas informáticos desarrollados mediante procesos de razonamiento de humanos expertos en la materia
Deep Learning	Subcampo del aprendizaje automático en redes neuronales artificiales caracterizado por la extensa cantidad de capas y el poder de análisis de grandes datos
Aprendizaje Supervisado	Método de aprendizaje en donde se entrena un conjunto con salida o respuesta conocida
GML	Modelos Lineales Generalizados
AUC	Área bajo la Curva
Descenso por gradiente	Algoritmo de optimización iterativo de primer orden para hallar el mínimo de una función
SVM	Máquinas de Soporte Vectorial. Modelo de clasificación que inducen separadores lineal o hiperplanos
Sensibilidad	Capacidad de detección de verdaderos positivos
Especificidad	Capacidad de detección de verdaderos negativos
Precisión	Valor predictivo positivo
Prevalencia	Proporción de una determinada población que sufre una afección
Matriz de Confusión	Método de descripción del rendimiento de un clasificador. Valores reales y valores predichos.

1 INTRODUCCIÓN

La enfermedad de Parkinson es la segunda enfermedad neurodegenerativa más frecuente después de la enfermedad de Alzheimer [1]. Afecta a cerca del tres por ciento de la población mayor de 65 años, e incide en mayor medida en los varones con 1.5 veces más posibilidades de desarrollar la enfermedad que las mujeres [2]. Aunque existe un subdiagnóstico importante a nivel mundial, la Organización Mundial de la Salud (OMS) ha determinado que la prevalencia global de 1990 al 2015 se duplicó hasta alcanzar más de 6 millones de enfermos, y que se duplicará o triplicará para el año 2040 debido al envejecimiento de la población y otros factores de riesgo [3]. En México se estima una prevalencia de 40 a 50 casos por 100 mil habitantes, se calcula que en total existen 500 mil habitantes con la EP [4].

Esta enfermedad se caracteriza por pérdida neuronal en la sustancia *nigra* del mesencéfalo y una reducción de dopamina del cuerpo estriado que se manifiesta como una desregulación principalmente en el control motriz, afectaciones en el habla y la vista; además, incrementa el riesgo de depresión, ansiedad, ataques de pánico, y trastornos en el sueño [5]. La EP es afectada también por la desregulación de diversos neurotransmisores como la serotonina, noradrenalina y acetilcolina por lo que, con frecuencia, se presentarán otros síntomas no motrices de la enfermedad como la demencia y la disautonomía (desbalance simpático-vagal) especialmente en etapas avanzadas de la enfermedad.

Se ha calculado que existe una reducción del 70 al 90% de dopamina en el cuerpo estriado y una pérdida del 60 al 70% de las neuronas de la sustancia *nigra* antes de que se presenten los síntomas clínicos característicos de la enfermedad [6]. Los cambios neuropatológicos en el individuo pueden encontrarse en una fase pre-sintomática. Aproximadamente el 90% de los casos de EP se consideran esporádicos o idiopáticos, mientras que el resto tienen un origen genético [7].

Se estima que la precisión en el diagnóstico clínico reportado por diversos artículos publicados entre 1988 y 2014 de la enfermedad del Parkinson fue de 80.6% de precisión combinada de los distintos artículos, 73.8% en diagnósticos realizados por médicos no expertos en EP, 79.6% de precisión en diagnósticos realizados por expertos en trastornos de movimiento, aumentando hasta un 83.9% de la evaluación inicial al refinar el diagnóstico post seguimiento. Al utilizar los criterios del Centro de Investigación del Banco de Cerebro de la Sociedad de la Enfermedad del Parkinson del Reino Unido, la precisión combinada fue de 82.7% [8].

No hay duda que la evaluación tradicional de los datos obtenidos de los pacientes sigue siendo factor fundamental en el diagnóstico de una enfermedad, sin embargo, en los últimos años se han diseñado y mejorado herramientas para mejorar la toma de decisiones sobre el diagnóstico.

La imagenología médica es de fundamental importancia en la medicina ya que se encarga de dar un respaldo visual y/o cuantitativo para el diagnóstico, la planificación del tratamiento o el seguimiento de una dada patología [9].

El PET es una técnica de imagen funcional de suma utilidad para el diagnóstico de enfermedades neurodegenerativas, ofrece una alta sensibilidad para identificar tempranamente procesos o alteraciones metabólicas, ya que los trastornos de este tipo suceden primero a nivel bioquímico y metabólico para luego manifestarse en daños estructurales evidentes[9].

Una tendencia creciente en el área del análisis de imágenes funcionales es el análisis estadístico de estas imágenes, que permite contrastar el valor de cada región cerebral del paciente con los datos estadísticos normalizados de la misma región cerebral con una población de referencia construida por sujetos control. De esta manera se puede analizar las variaciones de cada región y establecer márgenes [10] para considerar una región como sospechosa o patológica.

El uso de sistemas de clasificación en el diagnóstico médico en esta área aumenta gradualmente, los recientes avances en el campo de la inteligencia artificial han dado lugar a la aparición de sistemas expertos y sistemas de apoyo de decisiones para aplicaciones médicas.

El Aprendizaje Automático y las diferentes técnicas de inteligencia artificial tienen el potencial de ser una buena herramienta de apoyo para los especialistas incrementando la precisión y fiabilidad de sus diagnósticos, minimizando posibles errores, así como la reducción del periodo comprendido entre el inicio de la enfermedad y su diagnóstico, especialmente en estadios tempranos de la enfermedad sin signos sintomáticos o anatómicos, lo que potencialmente proporcionará al paciente y su familia una mejor calidad de vida al iniciar el tratamiento y retardar los daños inherentes a la EP.

El uso y aplicación de las nuevas tecnologías relacionadas con Inteligencia Artificial en el área médica, están teniendo mayor relevancia por el enorme potencial predictivo que

aportan en el diagnóstico y pronóstico de enfermedades, a pesar de que en éste campo sólo ha habido muy pocas publicaciones, como se muestra en la *Figura 1* generada de datos extraídos de la página de publicaciones médicas internacionales NCBI, National Center for Biotechnology Information (<https://www.ncbi.nlm.nih.gov/pubmed>), en donde se observa la cantidad de publicaciones médicas relacionadas con Machine Learning con datos de imágenes PET a lo largo de nueve años, en donde la mayor cantidad de publicaciones no superan las 75 publicaciones del año 2018.

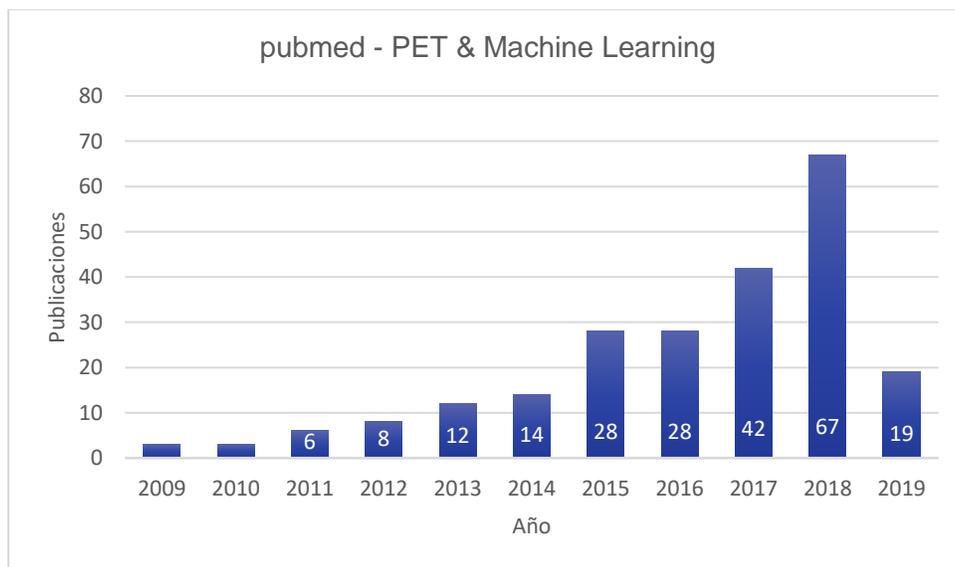


Figura 1. Publicaciones médicas por año de Machine Learning y PET (hasta febrero de 2019).

2 ANTECEDENTES

2.1 TOMOGRAFÍA POR EMISIÓN DE POSITRONES (PET)

La tomografía por emisión de positrones (PET) es una técnica no invasiva de imagen funcional que utiliza radionúclidos, mismos que son producidos en un ciclotrón y son usados para marcar compuestos de interés biológico. El compuesto marcado se introduce en el cuerpo, generalmente por inyección intravenosa, y se biodistribuye en los tejidos dependiendo de sus propiedades bioquímicas.

El positrón es la antipartícula del electrón con la misma masa, pero con carga eléctrica opuesta que tiene la característica que, al encontrarse con un electrón, se aniquila transformando su masa y la del electrón en dos fotones gama de 511 KeV, emitidos a 180 grados entre sí, propiedad que permite localizar los sitios donde se están llevando a cabo dichos procesos, utilizando una cámara de detección PET, como se ilustra en la *Figura 2*.

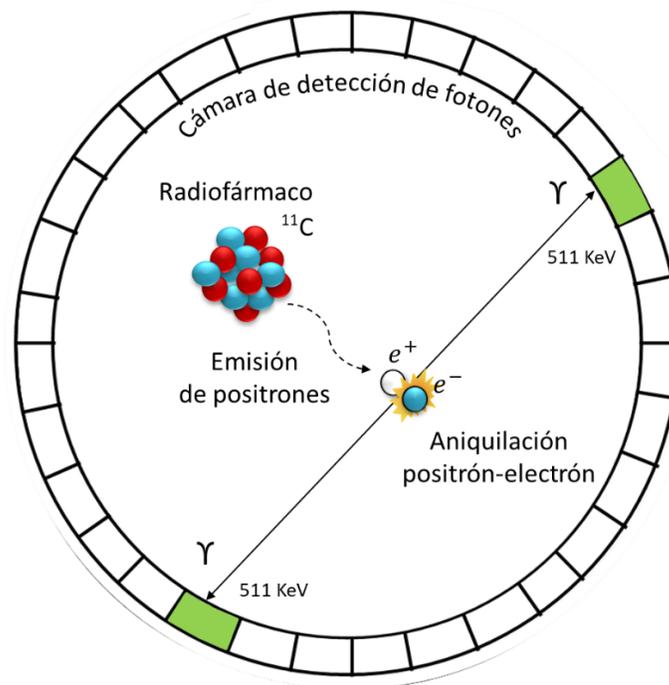


Figura 2. Cámara PET y aniquilación positrón electrón.

En este trabajo, se evaluaron dos radiofármacos específicos de la vía dopaminérgica ^{11}C -DTBZ y ^{11}C -RACLOPRIDE, para comparar la captación en el cerebro de sujetos con la enfermedad de Parkinson y sujetos control.

2.2 INTELIGENCIA ARTIFICIAL

2.2.1 ANTECEDENTES HISTÓRICOS

¿Pueden las máquinas hacer, lo que nosotros (como entidades pensantes) hacemos? Ésta pregunta fue planteada por el matemático, criptógrafo y científico de la computación Alan Turing en la década de los años 50 en su artículo *Computing Machinery and intelligence*.

La Inteligencia Artificial es la disciplina científica y tecnológica que se ocupa del entendimiento computacional, tiene como fin modelar algunas de las facultades intelectuales humanas mediante programas de cómputo que exhiban su comportamiento.

El concepto de Inteligencia Artificial (IA) surge en 1956 durante la conferencia “*Dartmouth Summer Research Project on Artificial Intelligence*” celebrada en el Dartmouth College en Estados Unidos. En dicha conferencia coincidieron John McCarthy (quien acuñó el término), Allen Newell, J.C. Shaw y Herbert Simon, entre otros personajes, quienes presentaron el LT (Logic Theorist), mismo que fue considerado el primer programa de IA de la historia. Este sistema era capaz de demostrar gran parte de los teoremas sobre lógica matemática que se presentaban en los tres volúmenes de *Principia Mathematica* de Alfred N. Whitehead y Bertrand Russell, algunos de los cuales se probaron de forma más elegante que los propios autores.

Más tarde, McCarthy y Minsky fundaron el laboratorio de Inteligencia Artificial en el Massachusetts Institute of Technology (MIT). En el transcurso de los años cincuenta, el desarrollo de la investigación corrió por cuenta de Charles Babbage (Máquina analítica, 1842), Kurt Gödel (Teorema de incompletitud, 1930), Alan Turing (Máquina universal, 1936), Norbert Wiener (Cibernética, 1943) y John von Neumann (Arquitectura del computador, 1950). La arquitectura de Neumann cuenta con una unidad central de procesamiento y un sistema de almacenamiento de datos, CPU y memoria, fue utilizada para construir una de las primeras computadoras en las que más tarde se implementaron sistemas de IA como el Logic Theorist de Newell y Simon (1954). También en 1954

emergió el IBM704, la primera computadora de producción en cadena, y con ella se desarrollaron numerosos lenguajes de programación específicamente diseñados para implementar sistemas de IA como LISP.

Con estos avances en la computación, surgió la necesidad de determinar la presencia de comportamiento inteligente en una máquina. El más destacado, desde el punto de vista histórico, fue el propuesto por Alan Turing en su artículo "*Computing Machinery and Intelligence*" publicado en la revista Mind (1950), en el cual reseña un método para que se puedan probar programas de IA. El concepto introducido por Turing ha tenido a lo largo de los años innumerables adaptaciones, correcciones, críticas y controversias. En este contexto, es de especial relevancia el *Teorema de incompletitud de Gödel*, que establece un conjunto de teoremas de lógica matemática exhibitorias de las limitaciones innatas a un sistema basado en reglas y procedimientos lógicos, como lo son todos los sistemas de IA.

Para los años setenta como parte de una respuesta al test de Turing, surgió el área conocida como procesamiento del lenguaje natural (NLP, por sus siglas en inglés), disciplina especializada en sistemas artificiales capaces de generar frases inteligentes y de mantener conversaciones con humanos. El NLP ha dado lugar a diversas áreas de investigación en el campo de la lingüística computacional, incluyendo aspectos como la desambigüedad semántica o la comunicación con datos incompletos o erróneos. A pesar de los grandes avances en este ámbito, sigue sin existir una máquina que pueda pasar el test de Turing tal como se planteó en el artículo original. Como es natural, a lo largo del tiempo se van ajustando los intereses en diversas áreas y la investigación en el campo de la IA no es la excepción. Así, en 1990 Hugh Loebner en alianza con el Cambridge Center for Behavioral Studies, establecieron un concurso anual que otorga el premio *Loebner* al sistema artificial que mantenga una conversación indistinguible a la de un humano, demostrando así el impacto sociológico que sigue teniendo la IA en la sociedad actual.

Las primeras aplicaciones comerciales de IA comenzaron a desarrollarse en los años 80, fundamentalmente dirigidas a la resolución de problemas de producción, control de contabilidad y procesos. Con el avance de estas aplicaciones nacieron los primeros sistemas expertos, que permitían realizar tareas de diagnóstico y tomar decisiones a partir de información aportada por expertos profesionales.

2.2.2 ESTRUCTURA NATURAL Y ARTIFICIAL DE LA INTELIGENCIA

El cerebro humano es un dispositivo biológico de cómputo que posee el sistema de reconocimiento de patrones más complejo y eficiente conocido. Sin mucho esfuerzo, es capaz de identificar el rostro de un conocido entre una multitud o reconocer la ejecución de su violinista favorito en un concierto para violín. En el cerebro humano, las funciones cognitivas se realizan mediante la activación de miles de millones de neuronas interconectadas mediante enlaces sinápticos, capaces de activarse en paralelo [13]. La activación neuronal sigue procesos biofísicos complejos que garantiza un funcionamiento sólido y adaptativo que le permite realizar funciones como el procesamiento de información sensorial, regulación fisiológica de los órganos, el lenguaje o la abstracción matemática, simultáneamente.

Actualmente la neurociencia todavía no aporta una descripción detallada sobre cómo la actividad individual de las neuronas da lugar a la formación de representaciones simbólicas abstractas. Lo que sí es claro es la importancia de la arquitectura funcional del cerebro en los procesos selectivos a nivel neuronal para la percepción, la memoria y la formación de conceptos, en la mayoría de los procesos cognitivos existe una separación de escalas entre la dinámica a nivel neuronal y la actividad mental abstracta. Esta separación de escalas supone la ruptura del lazo existente entre el hardware (neuronas) y el software de nuestro cerebro (operaciones abstractas, estados mentales) y constituye la hipótesis de partida para que los símbolos abstractos puedan ser manipulados por sistemas artificiales que no requieran un sustrato fisiológico natural.

Una de las cuestiones de mayor relevancia es saber si existen procesos mentales que estén exclusivamente ligados a la realidad biofísica del sistema nervioso humano y sean por tanto inaccesibles a un sistema artificial. Científicos como Roger Penrose consideran que existen intuiciones de la mente que no se pueden expresar algorítmicamente, y apoyado en el teorema de Gödel, dice que la mente no se puede simular con la ciencia que ahora conocemos [14].

Un aspecto fundamental del funcionamiento del cerebro humano es el papel que tiene la experiencia y el aprendizaje. El cerebro humano actual no es solo consecuencia de la evolución biológica basada en alteraciones y cambios genéticos, sino también del conjunto de conocimientos y técnicas acumulados a través del tiempo. Aspectos como el

lenguaje o la cultura transferidos de generación en generación también determinan la forma en la que procesamos información, por lo que contribuyen a germinar procesos de abstracción en los que se basan áreas como las matemáticas o la literatura.

Al modelo de aprendizaje basado en experiencias y entrenamiento se le conoce como *aprendizaje reforzado*, ya que la actividad sináptica se fortifica en función del número de veces que se establece una conexión entre neuronas. Esta regla que relaciona la actividad neuronal con la función cognitiva es llamada “Teoría de la asamblea celular”, comúnmente conocida como *regla de Hebb*, introducida por Donald Hebb en 1949 que dice “Si un elemento de procesado simple recibe una entrada de otro elemento de procesado y ambos están activos, el peso correspondiente a su conexión debería reforzarse” [15]. De esta forma, cuantas más veces se active un cierto canal de activación neuronal, más fácil resultará activarlo en el futuro e integrarlo a nuevos procesos cognitivos de mayor complejidad.

Aun cuando la neurociencia ha avanzado a pasos agigantados desde el descubrimiento de la neurona a finales del siglo XIX, las tecnologías desarrolladas actuales están muy lejos de poder diseñar, fabricar y simular sistemas artificiales con la complejidad del cerebro humano. En definitiva, una computadora no es más que una máquina que procesa representaciones abstractas siguiendo unas reglas predefinidas.

Un sistema basado en IA requiere de una secuencia finita de instrucciones que especifiquen las diferentes secciones a ejecutar por la computadora para resolver un problema determinado. Esta secuencia de instrucciones compone la *estructura algorítmica* de un sistema de IA [16].

Si es posible resolver un problema mediante sistemas naturales (cerebro) o sistemas artificiales (computadora), los algoritmos de cada implementación son enteramente distintos, dado que el conjunto de instrucciones elementales de cada sistema también es diferente.

En un sistema IA las instrucciones básicas son propias de una computadora, es decir, las operaciones aritméticas, lógicas, de lectura, escritura, de registros y de control de flujo secuencial. En ocasiones, los sistemas de IA resuelven problemas de forma heurística mediante un procedimiento de ensayo y error que incorpora información relevante basada en conocimientos previos.

2.3 APRENDIZAJE AUTOMÁTICO

2.3.1 CONCEPTOS BÁSICOS

La naturaleza flexible de la Inteligencia Artificial ha permitido que sea aplicada en múltiples campos de estudio, cada campo de especialización se ramifica en diversas áreas. Mencionaremos algunas de las más importantes sin dejar de lado que cada día surgen nuevas áreas de aplicación y de mayor innovación. La *Figura 3* muestra dichas ramas.

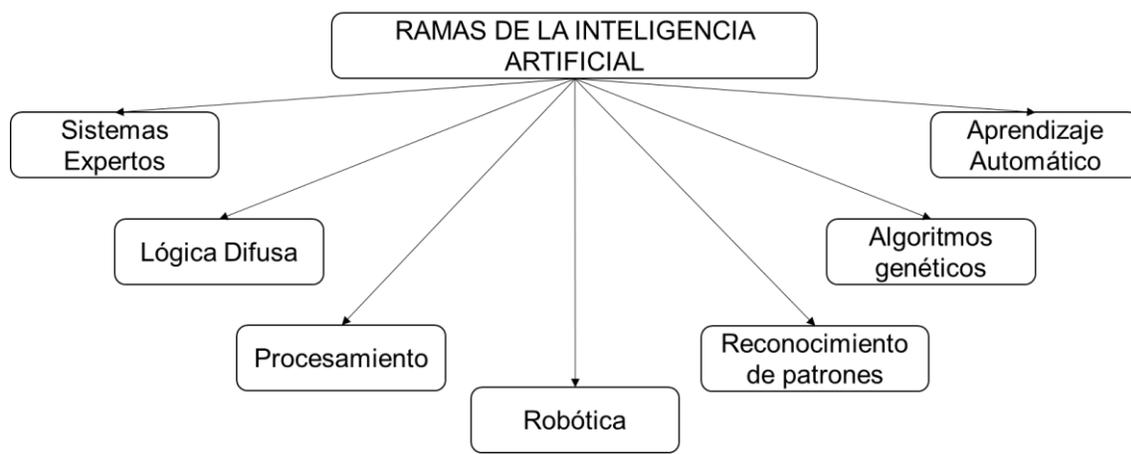


Figura 3. Ramas de la Inteligencia Artificial.

Arthur Samuel (1959) define el Aprendizaje Automático (AA) como “*el campo de estudio que brinda a las computadoras la capacidad de aprender sin ser programadas explícitamente*” [17], más tarde Tom Mitchel (1998) ofrece una definición más precisa: “*se dice que un programa de computadora aprende de la experiencia E con respecto a alguna tarea T y alguna medida de rendimiento P, si su rendimiento en T, medido por P, mejora con la experiencia E*” [18].

Esta rama de la IA surge en la intersección de la estadística, que busca aprender relaciones a partir de datos y la computación, con énfasis en algoritmos de computación eficientes. Este matrimonio entre las matemáticas y la informática está impulsado por los

desafíos computacionales únicos de construir modelos estadísticos a partir de conjuntos de datos masivos.

El aprendizaje automático se ocupa de un aspecto fundamental: dado un problema de IA que se puede describir en términos discretos (por ejemplo, dado un conjunto de posibles acciones para un determinado objetivo, cuál es la correcta), y disponiéndose de una gran cantidad de información sobre el mundo, se determina cual es la acción "correcta", sin que el mecanismo de elección se encuentre previamente programado. Es decir, el sistema aprende de forma autónoma a tomar las decisiones. De forma práctica esto se traduce en una función en la que a partir de una entrada se obtiene una salida, con lo que el problema radica en construir un modelo de esta función matemática de forma automática. Por lo tanto, la principal diferencia entre un programa inteligente que tiene un comportamiento similar al humano y uno de aprendizaje automático es que no será de aprendizaje automático, a menos que sus parámetros se aprendan automáticamente a partir de los datos.

Ahora bien, el aprendizaje automático es un subproducto de la calidad y análisis de los datos aportados al sistema y está determinado por varios factores: **algoritmo**, **datos** y **análisis**. En general es un método de análisis de datos que automatiza la construcción de modelos analíticos, aprende de cálculos previos para producir decisiones y resultados confiables y repetibles, a partir de estas conclusiones y la retroalimentación obtenida se puede utilizar el conocimiento generado en un nuevo conjunto de datos, creando un círculo de aprendizaje que mejora al algoritmo y su precisión.

A pesar de no ser una ciencia nueva, ha cobrado nuevo impulso, muchos de los métodos usados en AA y modelización estadística existen desde hace varias décadas. Algoritmos como las redes neuronales o máquinas de soporte vectorial fueron ideadas hace ya bastante tiempo, incluso algunos de ellos cayeron en desuso.

Unos de los principales factores del esplendor de estas técnicas son, por un lado, que ha incrementado notablemente la capacidad computacional de las máquinas al punto de hacer posible tratar problemas que antes no se podían tratar. Este incremento de la capacidad computacional ha sido tanto individual como al trabajar paralelamente con varias computadoras en simultáneo (usando el paradigma del Big Data). Por otro lado, la revolución de los datos motivada por la digitalización, ha supuesto un aumento colosal de datos que pueden ser procesados y modelados para obtener conocimiento de ellos.

2.3.2 USOS Y APLICACIONES

El AA es adecuado para resolver problemas que implican clasificación (dividir datos en dos o más clases), regresión (descubrir relaciones entre variables) y agrupación (agrupar datos con características similares).

Al identificar a qué clase pertenecen los datos, es posible identificar correos no deseados (spam), tumores benignos o malignos, o clasificar flores según las características obtenidas. La predicción es similar, pero con valores continuos, permite predecir qué valor se obtendrá dado un conjunto de datos de entrada con resultado desconocido. Se utiliza para predecir precios de inmuebles, alquileres, coches o la probabilidad de que ocurra algún evento utilizado con frecuencia en estadística con Regresión Lineal. Para la Biología computacional es una herramienta poderosa que ha auxiliado en el descubrimiento de fármacos y la secuenciación del ADN.

Los algoritmos de aprendizaje automático encuentran patrones naturales en los datos que generan conocimiento y contribuyen a tomar mejores decisiones y a realizar mejores predicciones. Se hace uso del aprendizaje automático cuando se tiene una tarea o un problema complejo que impliquen una gran cantidad de datos y muchas variables, pero no se disponga de ninguna fórmula o ecuación. Es muy útil cuando la naturaleza de los datos cambia sin cesar y el programa debe adaptarse.

De acuerdo a la revista Forbes, las diez aplicaciones y usos más habituales que tiene esta disciplina son [19]:

1. **Seguridad de datos.** El software malicioso (malware) es un problema creciente y pese a que diariamente surgen nuevos malware estos suelen tener un código similar que sus versiones anteriores, de esta manera, el aprendizaje automático puede predecir qué archivos son malware con gran precisión. Es más, los algoritmos de buscan patrones en cómo se accede a los datos e informan de anomalías que podrían predecir infracciones de seguridad.

2. Seguridad personal. El aprendizaje automático también es una ventaja en este sentido. Por ejemplo, es utilizado para aligerar los controles de seguridad en aeropuertos, ya que puede ayudar a eliminar falsas alarmas y detectar anomalías en las proyecciones de seguridad que el ojo humano puede ver con claridad, acelera el proceso a la vez que garantiza una mayor seguridad.

3. Trading financiero (Negociación bursátil). Anteriormente, el sector financiero utilizaba regresiones logísticas para la predicción del riesgo. Es decir, los bancos usaban estadística avanzada a la hora del análisis y concesión de préstamos. Los sistemas avanzados del aprendizaje automático posibilitan una mayor optimización de la concesión de créditos y préstamos al adaptarse con nuevos algoritmos a la hora de predecir los riesgos. Estas son algunas de las ventajas que aportan en la concesión de financiación:

- Posibilita tener en cuenta un gran número de variables, muchas de las cuales antes no se consideraban por la complejidad que implica, pero no dejan de ser importantes para el análisis de riesgo.
- Son sistemas se anticipan rápidamente y calibran el riesgo haciendo reestimaciones de los riesgos. Realizan mejoras constantes en los métodos de estimación del riesgo.
- Los algoritmos creados por el aprendizaje automático mejoran entre un 20-25% los modelos predicción convencionales.

Por otra parte, el aprendizaje automático tiene otras aplicaciones dentro de las financiaciones y análisis del riesgo, auxilian en la detección de morosidad, optimizan la selección de clientes para préstamos y créditos pre concedidos, secuencias inusuales de transacciones financieras, patrones inusuales de lecturas de sensores, valor futuro de las acciones o tipos de cambio y trading algorítmico

4. Reconocimiento de Imágenes. Se utilizan diversos algoritmos para poder comprender imágenes, encontrar algo en particular o agrupar zonas. Sabemos que las imágenes son conjuntos de píxeles contiguos y cada uno contiene información del color que tiene que “iluminar” (por ejemplo, RGB). Los ejemplos comunes son los de Reconocimiento de Caracteres Ópticos (OCR, del inglés Optical Character

Recognition) es decir, encontrar letras, agruparlas, encontrar espacios y poder descifrar textos; detectar personas en imágenes, presencia humana en cámaras de seguridad o más recientemente se utilizan redes neuronales para detectar rostros.

5. Procesamiento de lenguaje natural (NPL, del inglés Natural Programming Language). Una aplicación menos común pero que ya se está utilizando con mayor frecuencia en diferentes disciplinas. Los algoritmos de NPL pueden sustituir a los empleados de atención al cliente y dirigir a los usuarios más rápidamente a la información que necesitan. Se está utilizando, por ejemplo, para traducir la jerga legal poco clara en los contratos a un lenguaje sencillo y para ayudar a los abogados a ordenar grandes volúmenes de información para prepararse para un caso. A partir de las ondas de sonido sintetizadas, algoritmos especializados son capaces de limpiar ruido, intuir los silencios entre palabras y comprender diversos idiomas para interpretar o traducir.

6. Marketing personalizado. Se basa en un sencillo principio: mientras más se sabe sobre el cliente y su comportamiento mejor será la atención que se le ofrece lo que directamente incrementa las ventas. Por ejemplo, es cada vez más común que tras visualizar un producto en una tienda en línea, posteriormente se presente un anuncio de ese mismo producto para que se concrete la compra. Y no solo eso, las empresas pueden personalizar qué correos electrónicos recibe un cliente, ofertas o cupones, visualizar las ofertas que ven, qué productos se muestran como “recomendados”, etc., todos diseñados para conducir al consumidor efectivo. Directamente relacionado con este campo están los algoritmos que realizan recomendaciones, permiten analizar la actividad de un usuario y compararla con la del resto de usuarios para determinar qué le gustaría ver o comprar en próximas ocasiones. En este campo los avances son increíbles, por ejemplo, gracias a esos algoritmos la plataforma puede detectar que un usuario está comprando un artículo para él o para regalar.

7. Detección de fraudes. Los algoritmos de aprendizaje automático son cada vez más eficaces en la detección de posibles casos de fraude en diferentes campos. La compañía PayPal utiliza estos algoritmos para combatir el lavado de dinero; la compañía tiene herramientas que comparan millones de transacciones y pueden

distinguir con precisión entre transacciones legítimas y fraudulentas, entre compradores y vendedores.

- 8. Búsqueda online.** Este quizás sea el uso más famoso. *Google* y sus competidores mejoran constantemente lo que entiende el motor de búsqueda. Cada vez que se ejecuta una búsqueda en *Google*, el programa observa cómo responde a los resultados. Si un usuario hace clic en el resultado superior y permanece en esa página web, podemos suponer que obtuvo la información que estaba buscando y que la búsqueda fue un éxito. Si, por otro lado, hace clic en la segunda página de resultados, o escribe una nueva cadena de búsqueda sin hacer clic en ninguno de los resultados, podemos deducir que el motor de búsqueda no proporcionó los resultados que deseaba, y el programa puede aprender de ese error para ofrecer un mejor resultado en el futuro.
- 9. Coches inteligentes.** IBM encuestó recientemente a los principales ejecutivos de automóviles, y el 74% esperaba autos inteligentes en la carretera para 2025. Un automóvil inteligente no solo se integraría en el Internet de las cosas, sino que también aprendería sobre su propietario y su entorno. Puede ajustar la configuración interna (temperatura, audio, posición del asiento, etc.) de forma automática en función del conductor, informar e incluso solucionar los problemas, conducir y ofrecer consejos en tiempo real sobre el tráfico y las condiciones de la carretera.
- 10. Cuidado de la salud.** Se emplea para el diagnóstico asistido por computadora (CAD), por ejemplo, para revisar las mamografías iniciales de mujeres que desarrollaron cáncer de mama en el que posteriormente el algoritmo detectó el 52% de los cánceres hasta un año antes de que las mujeres fueran diagnosticadas oficialmente [20]. Esto demuestra que los algoritmos de aprendizaje automático pueden procesar más información y detectar más patrones que una mente humana. Además, se puede utilizar para advertir los factores de riesgo de enfermedad en poblaciones grandes.

2.3.3 APRENDIZAJE AUTOMÁTICO EN MEDICINA

En la actualidad, el AA junto con otras tecnologías se está implementando para brindar una atención al paciente más adecuado a un costo asequible. Los datos se recopilan de muchas fuentes, como datos de laboratorio, datos clínicos, datos actuariales, datos fisiológicos y datos del consumidor entre otros.

Mediante el uso del AA y dispositivos portátiles (wearables) cada vez más pacientes pueden ser supervisados de manera remota sin necesidad de hospitalización. Un dispositivo médico portátil puede detectar rápidamente cualquier variación en el cuerpo del paciente, este **monitoreo remoto** por medio de dispositivos va en aumento, impulsando a grandes empresas a desarrollar dispositivos orientados a recopilar diversos tipos de datos.

Una de las áreas de la Medicina que está obteniendo grandes beneficios del AA es la **imagenología**, pues se manejan grandes conjuntos de datos derivados de imágenes médicas. El AA puede ser entrenado para analizar imágenes, identificar anomalías y señalar áreas que necesitan atención, mejorando así la precisión de los procesos, puede ofrecer una opinión objetiva para mejorar la eficiencia, la confiabilidad y la precisión. Puede usarse para predecir enfermedades o una afección médica antes de la aparición de síntomas, optimizar el diagnóstico, el tratamiento, mejorar el pronóstico y encontrar nuevas curas a enfermedades. Los algoritmos pueden proporcionar un beneficio inmediato a las disciplinas con procesos reproducibles o estandarizados.

Concretamente en el ámbito médico, la toma de decisiones es un punto crítico ya que está en juego la salud o la vida de los pacientes, es por ello que, a la hora de emplear un modelo automático para apoyar los diagnósticos médicos, debe considerarse detenidamente las ventajas y desventajas, así como los inconvenientes y los riesgos que conlleva.

Al aplicar un modelo que auxilie en el campo de la medicina, es deseable que mejore el rendimiento de la tarea en la que se implementa, de modo que si se conoce o estima el rendimiento que ofrece el diagnóstico realizado por un médico, se espera que el modelo a implementar lo supere o cuando menos sea similar de modo que también ofrezca alguna otra ventaja en la práctica como ahorro de tiempo o reducción de costos.

Independientemente de la automatización de procesos, es importante que un especialista de la salud supervise los resultados arrojados por el sistema pues es una herramienta de apoyo y no de sustitución, es por ello que es deseable que el modelo sea capaz de ofrecer el conocimiento obtenido sobre el problema de manera comprensible para el médico, de este modo el médico puede estudiarlo y tener acceso a una “segunda opinión” que ayude a complementar sus conocimientos o descubrir relaciones entre los datos que han pasado desapercibidas hasta el momento. Estos detalles consolidan una base robusta en esta área y en cualquier otra.

Diversos académicos e investigadores alrededor del mundo en el área de la Inteligencia Artificial y específicamente en el campo de la medicina, coinciden en que los modelos de aprendizaje automático que se han aplicado con mayor frecuencia en biomedicina por cumplir con los requisitos y características principales de los modelos más exitosos en dicho campo son los modelos de árboles de decisión, Naive Bayes, vecinos más cercanos, redes neuronales artificiales y máquinas de vectores soporte, siendo estos los principales mas no los únicos.

2.3.4 TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Las técnicas de AA resultan muy útiles para problemas en distintos campos, el identificar qué tipo de problema se quiere resolver es el primer paso ya que de esa manera se entenderá cómo abordarlo, qué técnica emplear y qué tipo de algoritmo resulta ideal para ello. Estos problemas se dividen en tres ramas: **Clasificación**, **Regresión** y **Agrupamiento**, asociadas a su vez a los dos tipos de aprendizaje automático: **Supervisado** y **No supervisado**. La *Figura 4* es un esquema en donde se exhiben los métodos y técnicas más representativos y usados.

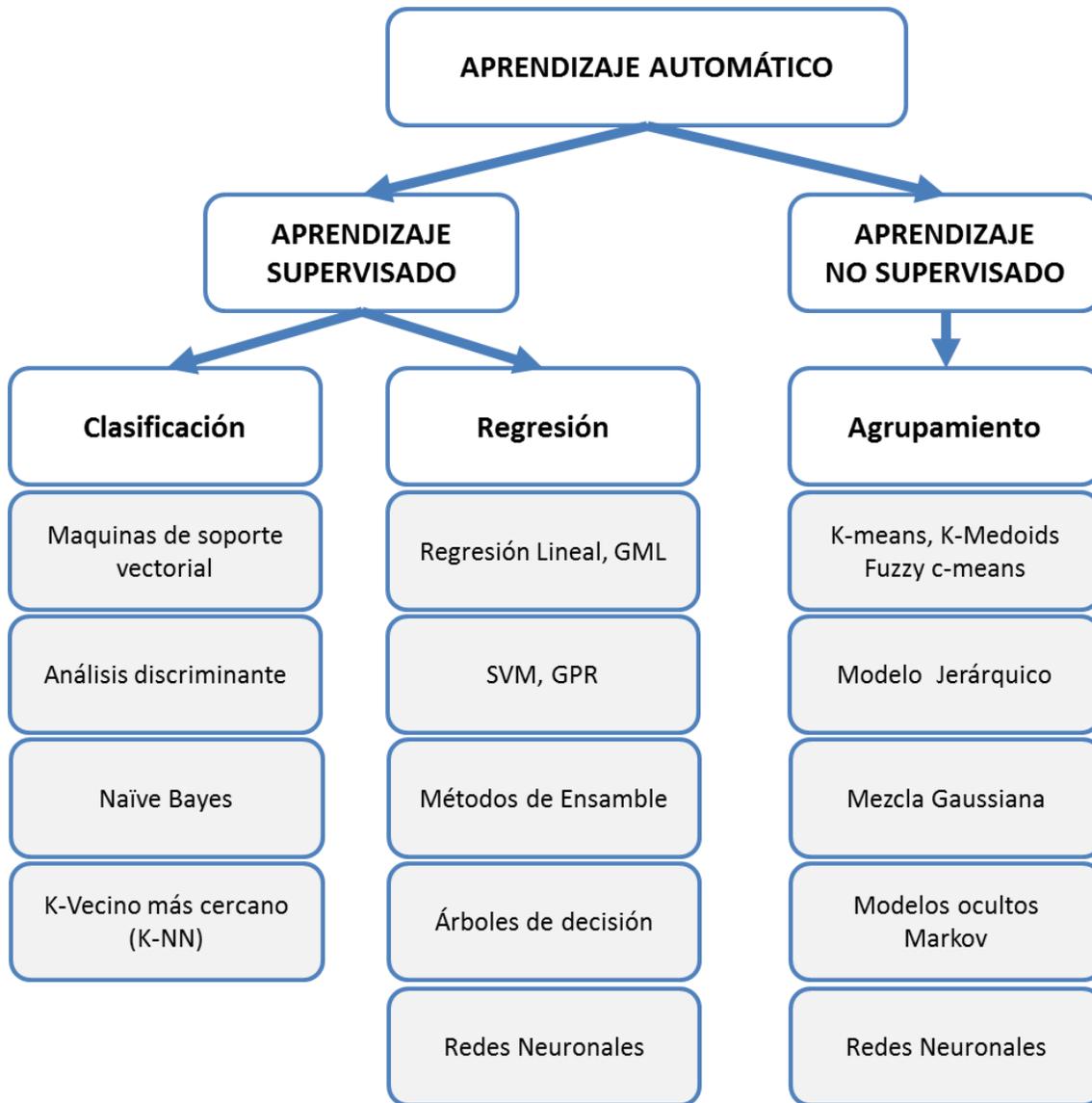


Figura 4. Ramas del Aprendizaje Automático.

Para elegir un modelo que se adapte a las necesidades específicas del usuario, es indispensable conocer el tipo de datos con los que se cuenta y decidir qué tipo de problema se necesita resolver.

En el caso específico de este trabajo, la base de datos descrita corresponde a un modelo de Clasificación con Aprendizaje Supervisado, ya que se requiere asignar la etiqueta sano o enfermo, y se cuenta con información de la entrada y salida esperada.

2.3.4.1 APRENDIZAJE SUPERVISADO

El Aprendizaje Supervisado toma un conjunto conocido de datos de entrada y respuestas conocidas para los datos (salidas) y entrena un modelo con objeto de generar predicciones razonables en función de las pruebas en presencia de incertidumbre como respuesta a datos nuevos.

Dado dos espacios de objetos X y Y , se pretende aprender una función $h: X \rightarrow Y$, frecuentemente llamada aproximación, clasificador o hipótesis que resulte en un objeto $y \in Y$ dado $x \in X$. Para tal objetivo se tiene a disposición un conjunto de entrenamiento $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$, donde $x_i \in X$ es una entrada y $y_i \in Y$ es la correspondiente respuesta que se busca obtener de $h(x_i)$. La componente x de cada ejemplo es un vector $x = (x_1, \dots, x_n)$ de atributos con valores continuos o discretos que describen la información relevante o propiedades del ejemplo. Los valores del espacio de salida Y asociados a cada ejemplo son las clases del problema. Así, cada ejemplo de entrenamiento queda totalmente caracterizado por un conjunto de pares (*atributo, valor*) y una etiqueta de clase.

Para ponerlo en términos más formales, supongamos que existe una probabilidad de distribución conjunta $P(x, y)$ sobre X y Y , y el conjunto de entrenamiento S con variables independientes idénticamente distribuidas de $P(x, y)$. Dado que se supuso que existe la probabilidad de distribución conjunta, es posible modelar la incertidumbre en las predicciones (por ejemplo, a partir del ruido en los datos) porque y no es una función determinista de x , sino una variable aleatoria con distribución condicional $P(y|x)$ para un x fijo.

Sea $L(\hat{y}, y)$ real y no negativa, *función costo* que mide qué tan diferente es la predicción \hat{y} de una hipótesis $h(x)$ del valor verdadero de la salida. El **riesgo** asociado a la hipótesis se define entonces como la **expectativa** de la función de costo:

$$R(h) = E[L(h(x), y)] = \int L(h(x), y) dP(x, y) \quad (1)$$

Los diferentes algoritmos de aprendizaje difieren en el espacio de hipótesis que tienen en cuenta; en el lenguaje de representación que utilizan y en el sesgo que usan al escoger la

hipótesis óptima entre las que son compatibles con el conjunto entrenamiento. La medida del error de los nuevos ejemplos se llama *error de generalización*. El objetivo final de un algoritmo de aprendizaje es encontrar una hipótesis h^* entre una clase de funciones \mathcal{H} para lo cual es riesgo $R(h)$ es mínimo:

$$h^* = \arg_{h \in \mathcal{H}} \min R(h) \quad (2)$$

En general, el error de generalización no se puede minimizar sin conocer previamente la función o la distribución $P(x, y)$. La manera más usual de proceder consiste en intentar minimizar el error sobre el conjunto de entrenamiento (o error empírico). Este principio se conoce como **minimización empírica del riesgo** y da una buena estimación del error de generalización en presencia de muchos ejemplos de entrenamiento.

$$\hat{h} = \arg_{h \in \mathcal{H}} \min R_{emp}(h) \quad (3)$$

Existen dos tipos de modelos dependientes de las posibles salidas que brindan. Si los valores de la salida se restringen a un conjunto finito, entonces se trata de un problema de *clasificación*. Si las salidas pueden tomar valores dentro de un conjunto infinito de la recta de números reales, entonces es un problema de regresión [24].

Se emplea el aprendizaje supervisado cuando se tienen datos conocidos para la salida que está intentando predecir.

Técnicas de clasificación

Predicen respuestas discretas; por ejemplo, si un correo electrónico es legítimo o correo no deseado (spam), o bien si un tumor es maligno o benigno. Los modelos de clasificación organizan los datos de entrada en categorías. Los problemas de clasificación son fundamentales en el análisis de datos ambientales. Las aplicaciones más habituales son las imágenes médicas, el reconocimiento de voz y la calificación crediticia.

Se utiliza la clasificación en datos que se pueden etiquetar, categorizar o dividir en grupos o clases concretos. Es importante analizar el número de clases y el de patrones dentro de cada clase. Por ejemplo, las aplicaciones para el reconocimiento de la escritura emplean

la clasificación para reconocer letras y números. En el procesamiento de imágenes y la visión artificial, se emplean técnicas de reconocimiento de patrones sin supervisión para la detección de objetos y la segmentación de imágenes.

Los algoritmos frecuentes para realizar la clasificación son: Redes Neuronales, máquina de soporte vectorial, análisis discriminante, clasificadores bayesianos (Naïve Bayes), k-vecino más cercano.

Técnicas de regresión

Las *técnicas de regresión* predicen respuestas continuas; por ejemplo, cambios de temperatura o fluctuaciones en la demanda energética. Las aplicaciones más habituales son la predicción de la carga eléctrica y el trading algorítmico. Se utilizan las técnicas de regresión cuando se trabaja con un intervalo de datos o si la naturaleza de la respuesta es un número real, como la temperatura o el tiempo que tardará una pieza de equipamiento en fallar. Algunos algoritmos habituales de regresión son: modelo lineal, modelo no lineal, GLM, SVR, GPR, métodos de ensamble, árboles de decisión, redes neuronales.

La *Figura 5* muestra una técnica de clasificación y regresión.

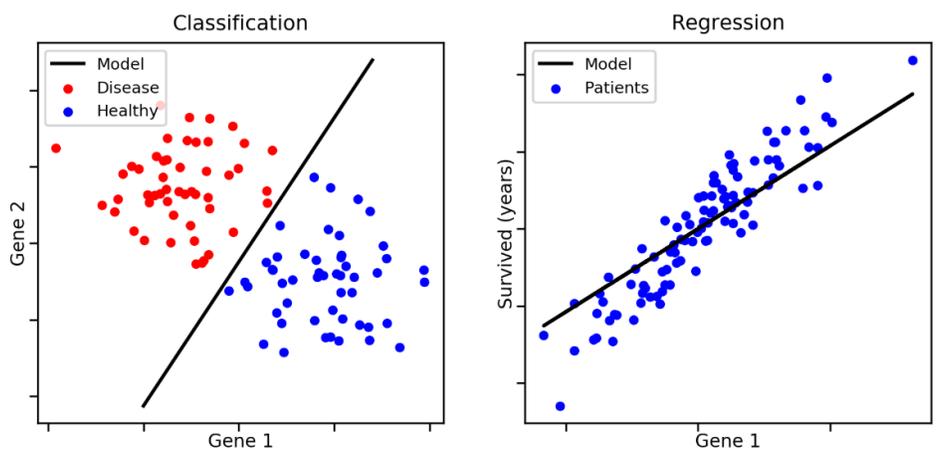


Figura 5. Ejemplo de gráficas de Clasificación y Regresión tomada de los ejemplos ilustrativos de la librería ScikitLearn.

2.3.4.2 APRENDIZAJE NO SUPERVISADO

El Aprendizaje No Supervisado consiste en agrupar (clustering) los datos en grupos cuando no hay categorías o clases disponibles por lo que requiere instancias, pero no etiquetas. Se basa en encontrar instancias en particiones de modo que sean similares entre sí y diferentes a las de otras particiones. En este método de aprendizaje, el modelo es ajustado a las observaciones. Se distingue del aprendizaje supervisado por el hecho de que no hay un conocimiento de antemano. Así, el aprendizaje no supervisado típicamente trata los objetos de entrada como un conjunto de variables aleatorias, siendo construido un modelo de densidad para el conjunto de datos. Típicamente se divide en particional y jerárquico. Uno de los problemas del agrupamiento es que la similitud de grupos es cuestión de subjetividad, por lo que las agrupaciones pueden suponerse ambiguas, de modo que es importante definir una buena medida de similitud en base a los atributos adecuados de los conjuntos.

Algunos algoritmos se adaptan para su uso tanto en aprendizaje supervisado como no supervisado.

2.3.5 MODELOS DE APRENDIZAJE AUTOMÁTICO EN MEDICINA

Son diversos los modelos que se emplean en el área médica, la selección de un modelo depende fundamentalmente de los datos con los que se cuenten y del objetivo que se desea alcanzar, a continuación se mencionan los más representativos.

2.3.5.1 ÁRBOLES DE DECISIÓN

Los árboles de decisión son diagramas de flujo en forma de árbol que usan ramificaciones para ilustrar cualquier posible resultado de una decisión. Muchos de estos diagramas de árbol usan ramificación binaria (dos opciones) basados en valores actuales o atributos de

los datos. Para grandes volúmenes de datos, se puede crear diversos árboles de decisión múltiple, los cuales, juntos forman una decisión consensuada sobre los resultados. Los árboles de decisión se pueden usar para problemas de clasificación y también de regresión. Este método es similar a los sistemas de predicción basados en reglas que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva. Los clasificadores de árboles binarios estructurados son construidos por repetidas particiones de un conjunto en dos subconjuntos disjuntos de manera que las particiones se hacen mediante una expresión booleana, hasta llegar a subconjuntos en los que todos los datos se encuentren en alguna clase.

Una vez construido el árbol, para clasificar un objeto dado su vector de características, se parte del nodo raíz y se evalúan las condiciones sobre el objeto en cuestión, desplazándose por los nodos hasta llegar a un nodo terminal que nos dará la clase del objeto. Inicialmente todos los pesos son iguales y con cada iteración se añade un clasificador al conjunto utilizando algún método de aprendizaje y teniendo en cuenta la distribución de los pesos, a continuación, el peso se reajusta y si el clasificador añadido le asigna la clase correcta, disminuye su peso, si no, aumenta. Así se consigue que los siguientes clasificadores que se añadan se centren en datos mal clasificados por otros clasificadores del conjunto.

2.3.5.2 CLASIFICADOR BAYESIANO O NAÏVE BAYES

El Clasificador bayesiano o Naïve Bayes es un método importante no sólo porque ofrece un análisis cualitativo de los atributos y valores que pueden intervenir en el problema, sino porque da cuenta también de la importancia cuantitativa de esos atributos. En el aspecto cualitativo representa cómo se relacionan esos atributos ya sea en una forma causal o señalando simplemente de la correlación que existe entre esas variables o atributos. Cuantitativamente da una medida probabilística de la importancia de esas variables en el problema y por lo tanto una probabilidad explícita de las hipótesis que se formulan. Entre las características que poseen los métodos bayesianos en tareas de aprendizaje destacan que con cada ejemplo observado modifica la probabilidad de que la hipótesis formulada sea correcta, aumentándola o disminuyéndola. Es decir, una hipótesis que no concuerda con un conjunto de ejemplos más o menos grande no es desechada por completo sino

disminuirá esa probabilidad estimada para la hipótesis. Estos métodos son robustos al posible ruido presente en los ejemplos de entrenamiento y a la posibilidad de tener entre esos ejemplos de entrenamiento datos incompletos o posiblemente erróneos. En el aprendizaje automático interesa seleccionar la mejor hipótesis dados los datos. En un problema de clasificación, la hipótesis puede ser la clase a asignar para una nueva instancia de datos. Una de las formas más sencillas de seleccionar la hipótesis más probable es la información dada como conocimiento previo sobre el problema. El **teorema de Bayes** proporciona una manera de calcular la probabilidad de una hipótesis dado un conocimiento previo. Se describe el caso general en el que puede haber más de 2 clases:

Sean $C_n, n = 1, \dots, N$ las etiquetas para las N clases en las que se distingue el clasificador y x_i el vector de características correspondiente a una imagen funcional, la función de *probabilidad a posteriori* de C_n dado x es definida como

$$P(C_n | x_i) = \frac{P(x_i | C_n)P(C_n)}{P(x_i)} \quad (4)$$

Donde

- $P(C_n)$ es la probabilidad de los datos dada la hipótesis verdadera (*probabilidad a priori*)
- $P(x_i | C_n)$ es la función de densidad de la probabilidad condicional de x_i dado C_n
- $P(x_i)$ es la densidad de la mezcla (independientemente de la hipótesis).

El máximo de la regla de decisión a posteriori para el clasificador de Bayes está definido como:

$$p(x_i | C_n)P(C_n) = \max_j \{p(x_i | C_j)P(C_j)\}, x_i \in C_n \quad (5)$$

El vector de características x_i es clasificado como perteneciente a la clase C_n cuya probabilidad a posteriori dado x_i es mayor. Las densidades dentro de cada clase son normalmente modeladas como distribuciones normales

$$P(C_n | x_i) = \left(\frac{1}{(2\pi)^{\frac{m}{2}} |\Sigma_n|^{\frac{1}{2}}} \right) \exp \left\{ -\frac{1}{2} (x_i - M_n)^t \Sigma_n^{-1} (x_i - M_n) \right\} \quad (6)$$

Donde M_n y Σ_n son la media y la matriz de covarianza de la clase C_n respectivamente y sus expresiones son

$$M_n = \frac{1}{N_n} \sum_{j=1}^{N_n} x_j^{(n)} \quad (7)$$

Con $x_j^{(n)}$, $j = 1, 2, \dots, N_n$ representa los vectores de características pertenecientes a la clase C_n usados como ejemplo.

2.3.5.3 K-VECINOS MAS CERCANOS

El k-vecino más cercano (K-NN del inglés k Nearest Neighbor) fue desarrollado en 1951 por Evelyn Fix y Joseph Hodges Jr., es uno de los algoritmos de clasificación más simples que hay. El algoritmo clasifica cada dato nuevo en el grupo que corresponda, según tenga K vecinos más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al que pertenecer. Este grupo será, por tanto, el de mayor frecuencia con menores distancias. Este algoritmo se fundamenta en una idea muy simple e intuitiva, el nuevo caso a clasificar se asignará a la clase más frecuente a la que pertenecen sus K vecinos más cercanos. Cabe destacar que el K-NN es muy sensible pues se pueden obtener resultados muy distintos al variar el valor de K , este valor suele fijarse tras un proceso de pruebas con varias instancias. La métrica de similitud utilizada también influye fuertemente en las relaciones de cercanía que se irán estableciendo en el proceso de construcción del algoritmo. La métrica de distancia puede contener pesos que ayudarán a calibrar el algoritmo de clasificación, convirtiéndola, de hecho, en una métrica personalizada.

La *Figura 6* exhibe cómo influye el valor de K :

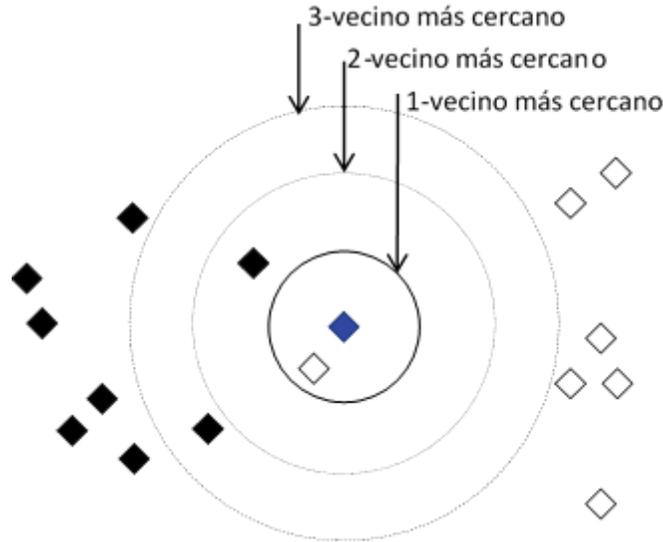


Figura 6. Diagrama K-NN. Clasificación del rombo azul tomando $k = \{1, 2, 3\}$.

- Para $k = 1$ el algoritmo clasificará el rombo azul como blanco.
- Para $k = 2$ el algoritmo no tiene criterio para clasificar el rombo azul.
- Para $k \geq 3$ el algoritmo clasificará el rombo azul como negro.

Su mayor debilidad es la dificultad para elegir el parámetro k adecuado, valores grandes de k crean límites entre clases parecidas mientras que valores pequeños de k , el algoritmo se ve muy afectado por el ruido. Otro inconveniente es la lentitud en el proceso de clasificación puesto que su objetivo no es obtener un modelo optimizado, sino que cada instancia de prueba es comparada contra todo el juego de datos de entrenamiento y, será la bondad de los resultados lo que determinará el ajuste de aspectos del algoritmo como el propio valor k , el criterio de selección de instancias para formar parte del juego de datos de entrenamiento o la propia métrica de medida de similitud. Matemáticamente el clasificador se define como:

$$\hat{f}(x') = \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^k \delta(v, f(x_i)) \quad (8)$$

Donde x' es la entrada que queremos clasificar y cuya clase se desconoce, x_1, x_2, \dots, x_k son los k vecinos más cercanos a x' y V es el conjunto de todas las etiquetas. La función $\delta(x, y)$ está dada por:

$$\delta(x, y) = \begin{cases} 1, & \text{si } x = y \\ 0, & \text{si } x \neq y \end{cases} \quad (9)$$

Existen variantes de este algoritmo que asignan pesos a los atributos que componen los vectores de características para dar más importancia a aquellos atributos más útiles para clasificar, evitando así que los atributos irrelevantes determinen la clasificación.

2.3.5.4 MAQUINAS DE SOPORTE VECTORIAL

Las Máquina de vectores soporte (SVM, del inglés Support Vector Machines) tienen su origen en los trabajos sobre la teoría del aprendizaje estadístico y fueron introducidas en los años 90 por Vapnik y sus colaboradores. Aunque originariamente las SVMs fueron pensadas para resolver problemas de clasificación binaria, actualmente se utilizan para resolver otros tipos de problemas (regresión, agrupamiento, multclasificación). También son diversos los campos en los que han sido utilizadas con éxito, tales como visión artificial, reconocimiento de caracteres, categorización de texto e hipertexto, clasificación de proteínas, procesamiento de lenguaje natural, análisis de series temporales. De hecho, desde su introducción, han ido ganando un merecido reconocimiento gracias a sus sólidos fundamentos teóricos.

Dentro de la tarea de clasificación, las SVMs pertenecen a la categoría de los clasificadores lineales, distinguiéndose tres casos: *SVM lineal con clases linealmente separables*, *SVM lineal con clases no separables* y *SVM no lineal*, puesto que inducen separadores lineales o hiperplanos, ya sea en el espacio original de los ejemplos de entrada, si éstos son separables o semi-separables (ruido), o en un espacio transformado (espacio de características), si los ejemplos no son separables linealmente en el espacio original. La búsqueda del hiperplano de separación en estos espacios transformados, normalmente de muy alta dimensión, se hace de forma implícita utilizando las denominadas *funciones kernel*. Mientras la mayoría de los métodos de aprendizaje se

centran en minimizar los errores cometidos por el modelo generado a partir de los ejemplos de entrenamiento (error empírico), el sesgo inductivo asociado a las SVMs radica en la minimización del denominado riesgo estructural. La idea es seleccionar un hiperplano de separación que equidista de los ejemplos más cercanos de cada clase para conseguir lo que se denomina un **margen máximo** a cada lado del hiperplano. Además, a la hora de definir el hiperplano, sólo se consideran los ejemplos de entrenamiento de cada clase que caen justo en la frontera de dichos márgenes. Estos ejemplos reciben el nombre de **vectores soporte**.

Desde un punto de vista práctico, el hiperplano separador de margen máximo ha demostrado tener una buena capacidad de generalización, evitando en gran medida el problema del sobreajuste a los ejemplos de entrenamiento. La propiedad de convexidad exigida para su resolución garantiza una solución única, en contraste con la no unicidad de la solución producida por una red neuronal artificial entrenada con un mismo conjunto de ejemplos. Este modelo es parecido a las redes neuronales en cuanto al objetivo de ajustar un contenido de parámetros que permite establecer fronteras en el espacio y separar patrones en diferentes regiones del espacio de atributos. La diferencia radica en el método de entrenamiento para ajustar los parámetros.

Las máquinas de vectores soporte presentan un rendimiento muy bueno pero el hecho de que el entrenamiento consista en el ajuste de parámetros ocultos y la predicción consista únicamente en el resultado numérico, hace que la claridad, así como la capacidad de explicación sean muy pobres.

Dado que el problema de diagnóstico de sujetos sanos y enfermos es un problema de clasificación de dos clases, lo ideal es tratar de estimar una función $f: \mathbb{R}^N \rightarrow \{\pm 1\}$.

Las *Figura 7 y 8* muestran ejemplos de gráficas de clasificación usando Maquinas de Soporte Vectorial para clases linealmente separables y no linealmente separables.

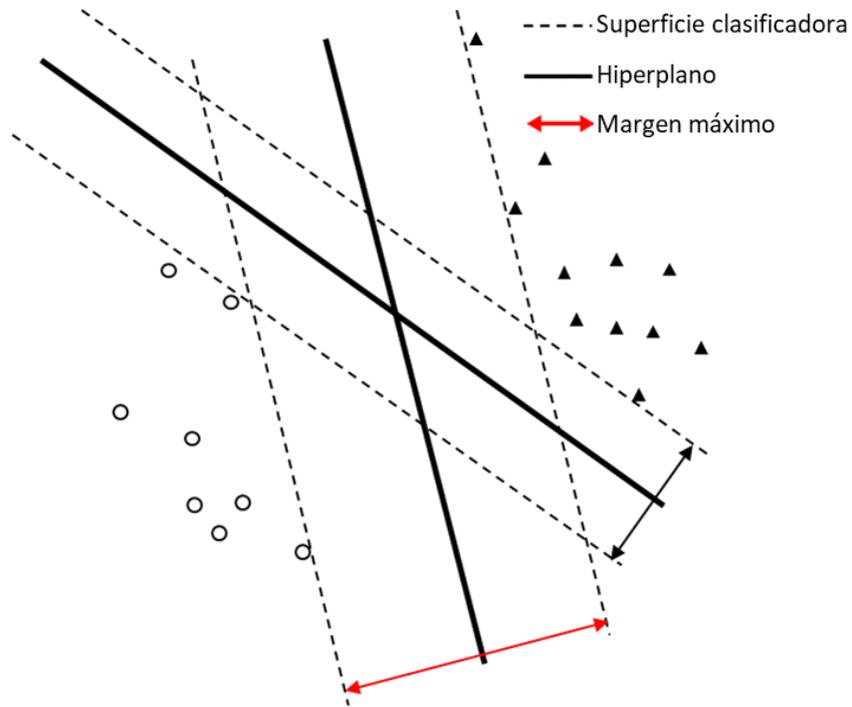


Figura 7. Gráfica de clasificación por SVM lineal con clases linealmente separables. El hiperplano de clasificación óptimo es el que muestra el margen máximo de superficie clasificadora (delimitada por las flechas rojas)

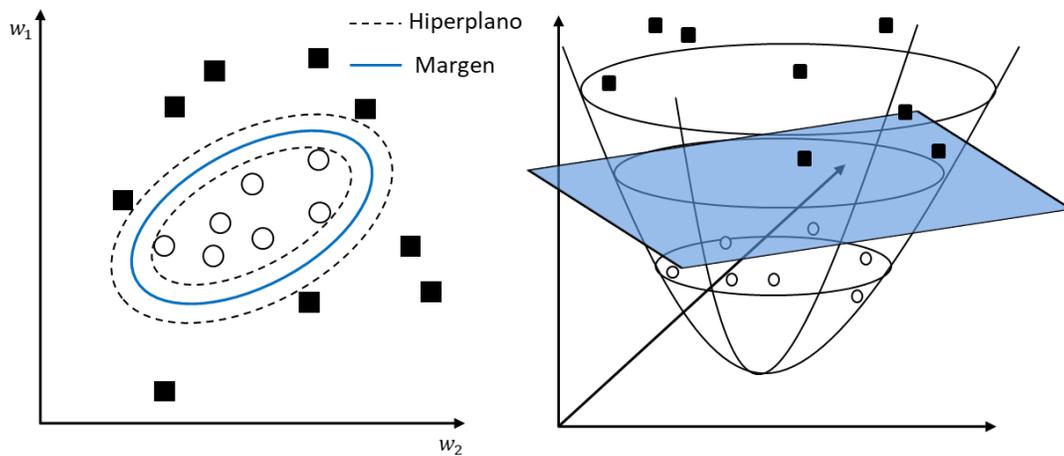


Figura 8. Espacio de características linealmente no separable (izquierda) que, al aplicar la función Kernel se transforma en un espacio linealmente separable (derecha).

2.3.6 DESARROLLO DE UN MODELO DE APRENDIZAJE AUTOMÁTICO

La construcción de un sistema de aprendizaje automatizado comprende varias etapas: **pre-procesamiento** de datos, **entrenamiento** del o los modelos y **post-procesamiento** de los resultados.

Pre-procesamiento: Los datos que recibe el sistema pueden estar incompletos, contener información redundante o irrelevante, sufrir perturbaciones a causa del ruido o presentar cualquier otra dificultad que afecte el rendimiento, es por ello en esta etapa se deben considerar y resolver todos esos inconvenientes.

Entrenamiento: En esta fase se recibe la información ya depurada y se construye el modelo.

Post-procesamiento: Se evalúa la capacidad de las respuestas proporcionadas por los modelos para determinar su rendimiento. De este modo, los elementos fundamentales de un sistema son: entradas, salidas y el modelo. Cada posible entrada al sistema es un vector de atributos que se designa *instancia*. Los atributos pueden ser nominales, si toman valores dentro de un conjunto finito cualquiera, o numéricos si toman valores reales, es decir que el conjunto de posibles valores puede ser finito, infinito numerable o no numerable.

En la *Figura 9* se muestra el desarrollo de un modelo de Aprendizaje Automático.

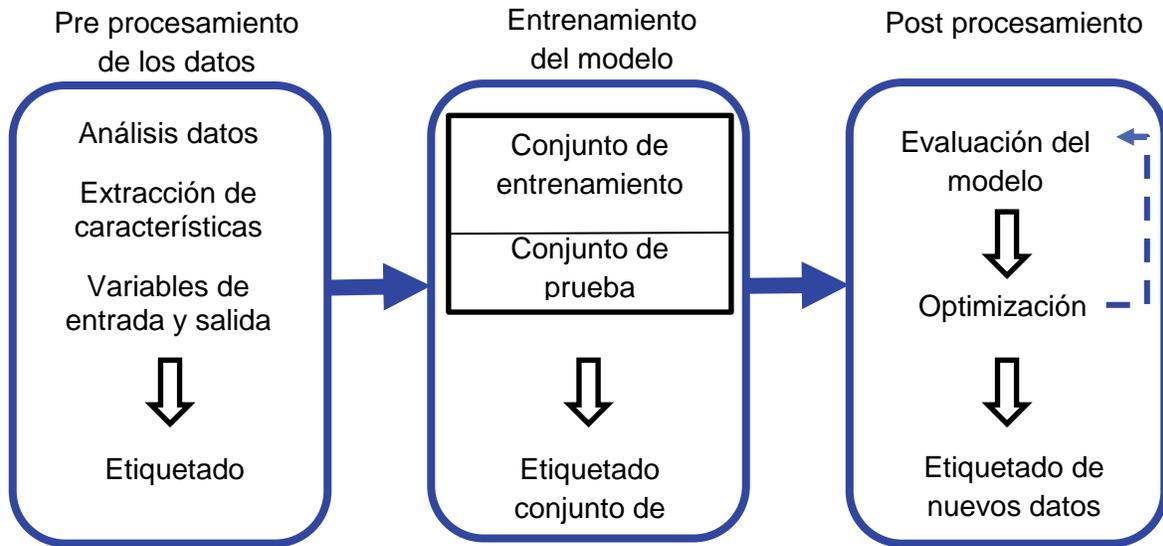


Figura 9. Modelo de desarrollo de Aprendizaje Automático.

2.3.7 ANÁLISIS DE DATOS Y EXTRACCIÓN DE CARACTERÍSTICAS

Sucede en la etapa de pre-procesamiento. Dentro de la tarea de clasificación y separación objetos en clases, es tan importante el método de clasificación que se escoge como la forma la que se construyen los vectores de características que definen cada dato, tienen una relación directa con el comportamiento general del clasificador que se construye; es tan o más importante que el algoritmo de aprendizaje que se utilice. Una buena representación de datos resulta fundamental ya que por lo general los datos son redundantes o contienen información voluble. Antes de proceder al análisis, es recomendable obtener un conjunto reducido de variables que permita describir los datos de forma simplificada. La simplificación de datos es conocida como *reducción de dimensionalidad* y permite eliminar aspectos superfluos como correlaciones entre variables o fluctuaciones estocásticas (ruido), es decir, todos aquellos factores que no representen información relevante del proceso que se estudie. Por otro lado, se denomina señal a aquellos factores que contribuyen de forma relevante a los datos, de manera general se representa con la siguiente ecuación:

$$Datos = señal + ruido \quad (10)$$

La dimensión de los vectores deberá ser lo suficientemente grande como para agrupar todas las características y propiedades de los datos considerando que, para conservar el rendimiento, el número de muestras usadas en el entrenamiento crecerá proporcionalmente al número de características.

Consideremos n puntos en un espacio de características m dimensional. Se asume que los puntos están *bien distribuidos*, si no existe ningún subconjunto de $n - 1$ puntos que se sitúen en un hiperplano de menor dimensión que $m - 1$. El número $O(n, m)$ de agrupaciones puede ser formadas por hiperplanos $(m - 1)$ -dimensionales para separar los n puntos en dos clases está dado por:

$$O(n, m) = 2 \sum_{i=0}^m \binom{n-1}{i} \quad (11)$$

Donde

$$\binom{n-1}{i} = \frac{(n-1)!}{(n-1-i)! i!} \quad (12)$$

De modo que, la probabilidad de agrupar n puntos en un espacio de características m -dimensional en dos clases linealmente separables será:

$$P_n^m = \frac{O(n, m)}{2^n} = \begin{cases} \frac{1}{2^{n-1}} \sum_{i=0}^m \binom{n-1}{i}, & n > m + 1 \\ 1 & n \leq m + 1 \end{cases} \quad (13)$$

La probabilidad P_n^m se muestra como una función m/n . Para espacios de características de baja dimensión ($m/n < 0.3$), la probabilidad de separación de clases P_n^m es casi cero, esto es, los clasificadores lineales tienen un bajo rendimiento discerniendo entre dos clases. Sin embargo, cuando se aumenta la dimensión del espacio de características, la probabilidad de que el conjunto de n puntos sea separable se aproxima a la unidad. Así,

como intuitivamente parece claro, añadir información al vector de características mejora la separabilidad de clases para el caso de un clasificador lineal.

Por otro lado, el hecho de que un cociente m/n pequeño produce bajos valores para probabilidad de separabilidad puede resolverse para un número fijo n de puntos a clasificar mediante un mapeo a un espacio de dimensión mayor. Esto puede conseguirse a través del uso de *kernels* que involucran el uso de un producto interno no lineal, convirtiendo el espacio de características en otro de dimensión mayor donde P_n^m aumente, y un clasificador lineal pueda operar satisfactoriamente.

Gráficamente, la probabilidad de agrupar n puntos en un espacio m -dimensional de características en dos clases linealmente separables se ve como en la *Figura 10*:

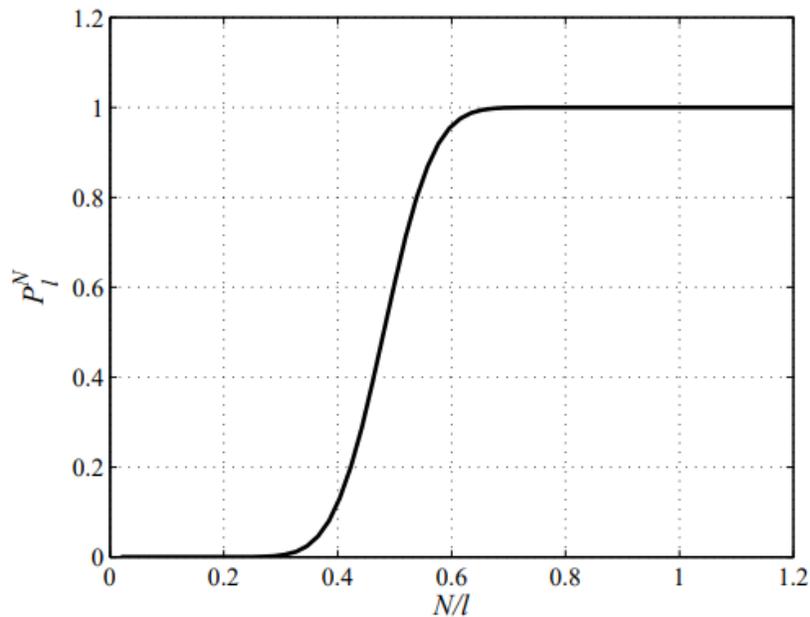


Figura 10. Probabilidad de agrupar n puntos en un espacio m -dimensional de características en dos clases linealmente separables.

De este modo es natural deducir que, al incrementar la dimensión del espacio de características, resultará beneficioso para discriminar entre dos clases. Sin embargo, ocurre exactamente lo contrario, dando lugar al problema del pequeño tamaño muestral, conocido como el fenómeno del máximo. Este problema se describe como la reducción de la eficacia de un clasificador al añadir nuevas características a los vectores de

entrenamiento cuando el número de estos es relativamente pequeño en comparación con el número de características. El problema reside en que para definir un clasificador en un espacio de características de alta dimensionalidad es necesario valorar un número de parámetros comparable a la dimensión del espacio, por ejemplo, en el caso de un clasificador lineal, será necesario estimar $m + 1$ parámetros en un espacio de características m -dimensional, de manera que, aunque el clasificador separe los datos de entrenamiento satisfactoriamente, la fiabilidad en la estimación de los parámetros del clasificador será baja, ya que se estimarán cuantiosos parámetros con un número reducido de vectores de entrenamiento.

El clasificador construido con esta limitación tendrá, por resultado, una baja capacidad de generalización. La *Tabla 1* muestra los resultados posibles en función de la etiqueta.

Tabla 1. Resultados posibles de la prueba en función de la etiqueta.

		Etiqueta		
		Positiva	Negativa	
Test	Positivo	VP	FP	→ Valor predictivo positivo
	Negativo	FN	VN	→ Valor predictivo negativo
		↓	↓	
		Sensibilidad	Especificidad	

El problema del pequeño tamaño muestral justifica el uso de técnicas de reducción de la dimensionalidad del espacio de características, cuando el número de características usadas para diseñar el clasificador es mucho mayor que el número de vectores de entrenamiento disponibles. Además, la reducción del número de características elimina la correlación entre las mismas y disminuye el coste computacional de la clasificación por lo que, aunque el problema del pequeño tamaño muestral no exista, es recomendable usar un número reducido de características.

Posterior al entrenamiento del modelo sigue la evaluación del rendimiento. En el ambiente médico, es de especial interés conocer los detalles concernientes a cada predicción

individual cuando se trata de apoyar en la toma de decisiones sobre el tratamiento o diagnóstico de un paciente. Es por ello que se debe desarrollar un *Análisis de predicciones* en el que se aborden dos problemas principales, la estimación de la fiabilidad de una predicción y la capacidad de proporcionar una explicación basada en los valores de los atributos. Para evaluar la fiabilidad de una predicción, se aplica un *análisis de sensibilidad temporal* orientado a los modelos de predicción, generando perturbaciones en el conjunto de aprendizaje. Lo interesante es que no se requiere conocimiento previo sobre el valor de la variable objetivo de la instancia que se analiza.

2.3.8 EVALUACIÓN DE UN CLASIFICADOR

El propósito de todo clasificador es asignar correctamente una etiqueta a un objeto definido por un vector de atributos. En el caso de clasificación binaria, en la que los datos están divididos entre etiquetas positivas o negativas, existen dos posibles errores de clasificación que se puede cometer: clasificar como positivo un patrón que en realidad era negativo o viceversa. Estas posibilidades de donde se compara el resultado de la prueba con etiqueta original, dando lugar a VP (Verdadero Positivo) cuando los dos coinciden en valor positivo, FP (Falso Negativo) cuando la prueba de negativo mientras en la etiqueta original era positiva y VN (Verdadero Negativo) cuando los dos coinciden en valor negativo.

Establezcamos algunas medidas de rendimiento frecuentemente usadas en esta área:

SENSIBILIDAD

La sensibilidad se define y como la capacidad de un clasificador para detectar los verdaderos positivo y se expresa como:

$$\text{Sensibilidad} = \frac{\text{número de VP}}{\text{número de VP} + \text{FN}} \quad (14)$$

De manera que la sensibilidad del 100% corresponderá a un clasificador que es capaz de clasificar correctamente todos los objetos etiquetados como positivos. Por tanto, si un clasificador con alta sensibilidad da un resultado negativo, este será muy fiable, lo que, en el caso del diagnóstico asistido por computadora, puede ser usado para descartar la enfermedad. La sensibilidad está relacionada con el error de tipo I en la inferencia estadística, que consiste en rechazar la hipótesis nula cuando en realidad es cierta.

ESPECIFICIDAD

Por otro lado, la especificidad se define como la capacidad de un clasificador para detectar verdaderos negativos y se expresa como:

$$\text{Especificidad} = \frac{\text{número de VN}}{\text{número de VN} + \text{FP}} \quad (15)$$

De manera que una especificidad del 100% corresponderá a un clasificador que es capaz de clasificar correctamente todos los objetos etiquetados como negativos. En el caso del diagnóstico asistido por computadora, un clasificador con alta especificidad es muy útil para confirmar la enfermedad, ya que raramente producirá un resultado positivo que en realidad sea falso. La especificidad está relacionada con el error de tipo II donde se acepta la hipótesis nula cuando en realidad es falsa.

PRECISIÓN

Sin embargo, tanto altos valores de la sensibilidad como de la especificidad no tienen por qué corresponder a un clasificador preciso. Se define la **precisión** como:

$$\text{Precisión} = \frac{\text{número de VP} + \text{VN}}{\text{número de VP} + \text{FP} + \text{FN} + \text{VN}} \quad (16)$$

Puede ocurrir que un clasificador tenga valores cercanos al 100% de sensibilidad y cercanos al 0% de especificidad. Este clasificador no tendrá capacidad de discernir entre las clases, ya que será un clasificador que tome cualquier patrón como positivo. Esto es equivalente a una clasificación al azar, ya que su precisión rondará el 50% para una muestra sin predominio de ninguna de las dos clases. El clasificador deseable será aquel que tenga valores altos de sensibilidad, especificidad y precisión simultáneamente, y no sólo de alguno de ellos por separado.

Otros parámetros que pueden resultar interesantes son los valores predictivos. Éstos hacen referencia a la validez de un resultado de clasificación positivo/negativo (valor predictivo positivo/negativo). Se podrá confiar más un resultado positivo de un clasificador con un valor predictivo positivo alto que uno con un valor predictivo positivo menor.

PREVALENCIA

Los valores predictivos dependen de la superioridad de las clases, denominada **prevalencia**, en términos de epidemiología determina la proporción de individuos de una población que padece la enfermedad. Si el conjunto de prueba no tiene igual número de positivos que de negativos, habrán de usarse las fracciones de probabilidad positiva (f_{pp}) o negativa (f_{pn}) que no dependen de la prevalencia:

$$f_{pp} = \frac{\text{sensibilidad}}{1 - \text{especificidad}} \quad (17)$$

$$f_{pn} = \frac{1 - \text{sensibilidad}}{\text{especificidad}} \quad (18)$$

CURVA ROC

Con frecuencia será muy útil valorar cómo se modifica el rendimiento de un clasificador al modificar algún parámetro, ya sea del clasificador o en la definición de algún paso previo. Para este análisis será útil la representación en el espacio ROC (del inglés, Receiver Operating Characteristic), que no es más que una representación bidimensional de la tasa de verdaderos positivos (sensibilidad) frente a la tasa de falsos positivos (1-especificidad). La *Figura 11* muestra la gráfica Roc ideal.

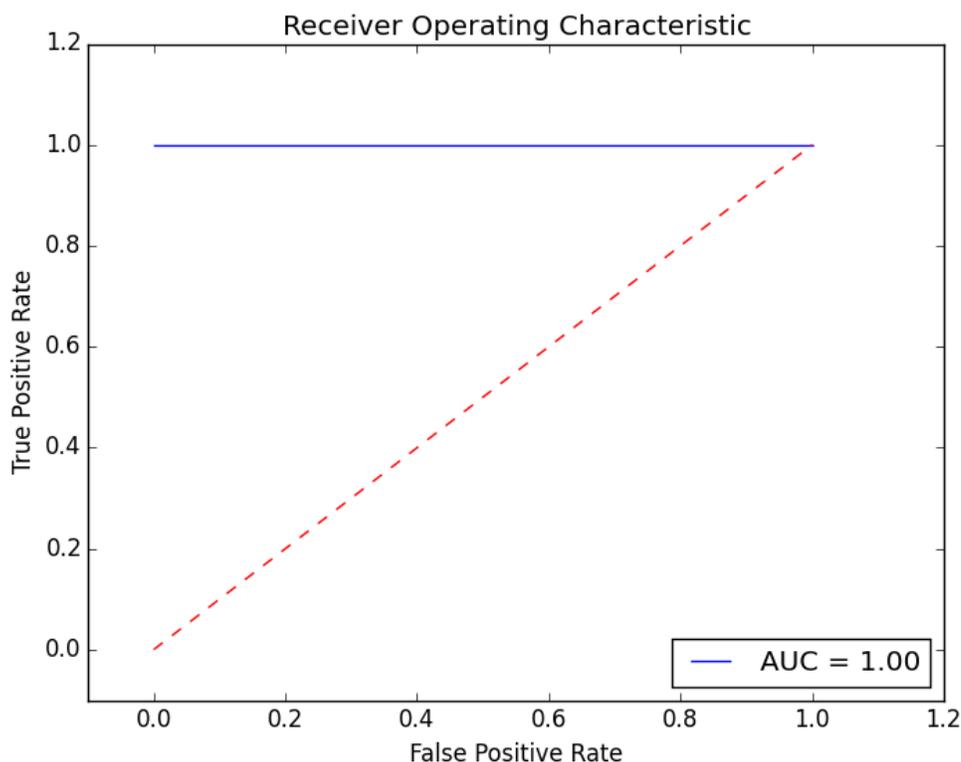


Figura 11. Gráfica “Espacio ROC”, clasificación perfecta. Proporciona una idea gráfica del desempeño del modelo de clasificación en relación Especificidad / Sensibilidad.

En este espacio, la mejor predicción corresponderá con un punto cercano a la esquina superior izquierda, correspondiente al punto (0,1), que representa la sensibilidad del 100% (ningún falso negativo) y especificidad 100% (ningún falso positivo), que producirá

también una precisión del 100%. El punto que cumple con esas características es llamado *clasificación perfecta*. Una clasificación aleatoria se encontraría en la línea diagonal (llamada la línea de *no-discriminación*), por debajo de esa línea se ubican los resultados peores que los de estimaciones azarosas.

Cuando el Área bajo la curva (AUC, por sus siglas en inglés) es máxima (valor 1.0), encontramos una clasificación perfecta. Se utiliza una regla empírica para evaluar un clasificador basado en este valor del siguiente modo: 0.90–1=Muy buena(A), 0.80-0.90=Buena(B), 0.70-0.80=No tan buena(C), 0.60-0.70=Pobre(D), 0.50-0.60=Falla(F).

Documentación disponible en: https://scikit-learn.org/stable/modules/model_evaluation.html#roc-metrics.

2.4 MODELOS LINEALES GENERALIZADOS

Un modelo lineal generalizado tiene tres componentes básicos:

Componente aleatoria: Identifica la variable respuesta y su distribución de probabilidad.

Componente sistemática: Especifica las variables explicativas (independientes o predictoras) utilizadas en la función predictora lineal.

Función enlace: Es una función del valor esperado de $Y, E(Y)$, como una combinación lineal de las variables predictoras.

Componente aleatoria.

La componente aleatoria de un GLM consiste en una variable aleatoria Y con observaciones independientes (y_1, \dots, y_N) .

En muchas aplicaciones, las observaciones de Y son binarias y se identifican como éxito y fracaso. Aunque de modo más general, cada Y_i indica el número de éxitos de entre un número fijo de ensayos, y se modela como una distribución binomial. En otras ocasiones cada observación es un recuento, con lo que se puede asignar a Y una distribución de Poisson o una distribución binomial negativa. Finalmente, si las observaciones son continuas se puede asumir para Y una distribución normal.

Todos estos modelos se pueden incluir dentro de la llamada *familia exponencial de distribuciones* definida como sigue

$$f(y_i | \theta_i) = a(\theta_i) \cdot b(y_i) \cdot \exp[y_i Q(\theta_i)] \quad (19)$$

con $Q(\theta)$ llamado parámetro natural.

Componente Sistemática

La componente sistemática de un GLM especifica las variables explicativas que entran en forma de efectos fijos en un modelo lineal, es decir, las variables x_j se relacionan mediante

$$\alpha + \beta_1 x_1 + \cdots + \beta_k x_k \quad (20)$$

Esta combinación lineal de variables explicativas se denomina **predictor lineal**. Alternativamente, se puede expresar como un vector (v_1, \dots, v_N) tal que

$$v_i = \sum_j \beta_j x_{ij} \quad (21)$$

donde x_{ij} es el valor del j -ésimo predictor en el i -ésimo individuo, e $i = 1, \dots, N$. El término independiente α se obtendría con esta notación haciendo que todos los x_{ij} sean igual a 1 para todos los i .

En cualquier caso, se pueden considerar variables que estén basadas en otras variables como $x_3 = x_1 x_2$ o $x_3 = x_2^2$, para modelar interacciones entre variables o efectos curvilíneos de x_2 .

Función enlace

Se denota el valor esperado de Y como $\mu = E(Y)$, entonces la función enlace especifica una función $g(\cdot)$ que relaciona μ con el predictor lineal como:

$$g(\mu) = \alpha + \beta_1 x_1 + \cdots + \beta_k x_k \quad (22)$$

Así, la función enlace $g(\cdot)$ relaciona las componentes aleatoria y sistemática. De este modo, para $i = 1, \dots, N$

$$\begin{aligned} \mu_i &= E(Y_i) \\ v_i &= g(\mu_i) = \sum_j \beta_j x_{ij} \end{aligned} \quad (23)$$

La función g más simple es la función identidad $g(\mu) = \mu$, que da lugar al modelo de regresión lineal clásico

$$\mu = E(Y) = \alpha + \beta_1 x_1 + \dots + \beta_k x_k \quad (24)$$

Los modelos de regresión lineal típicos para respuestas continuas son un caso particular de los GLM. Estos modelos generalizan la regresión ordinaria de dos modos: permitiendo que Y tenga distribuciones diferentes a la normal y, por otro lado, incluyendo distintas funciones enlace de la media. Esto resulta bastante útil para datos categóricos.

Modelos lineales Generalizados para datos binarios

En muchos casos las respuestas tienen solo dos categorías del tipo sí/no, positivo/negativo, 1 (éxito)/0 (fracaso), sano/enfermo, de modo que se puede definir una variable Y que tome dos posibles valores. En este caso

$$\begin{aligned} f(y|\pi) &= \pi^y (1 - \pi)^{1-y} \\ &= (1 - \pi) \left(\frac{\pi}{1 - \pi} \right)^y \\ &= (1 - \pi) \exp \left[y \log \left(\frac{\pi}{1 - \pi} \right) \right] \end{aligned} \quad (25)$$

con $y = 0, 1$.

El parámetro natural es:

$$Q(\pi) = \log \left(\frac{\pi}{1 - \pi} \right) = \text{logit}(\pi) \quad (26)$$

En este caso

$$E(Y) = P(Y = 1) = \pi(x) \quad (27)$$

dependiente de p variables explicativas $x = (x_1, \dots, x_p)$ y

$$\text{Var}(Y) = \pi(x)(1 - \pi(x)) \quad (28)$$

En respuestas binarias, un modelo análogo al de regresión lineal es:

$$\pi(x) = \alpha + \beta x \quad (29)$$

que se denomina *modelo de probabilidad lineal*, ya que la probabilidad de éxito cambia linealmente respecto a x . El parámetro β representa el cambio en probabilidad por unidad de x .

Este modelo es un GLM con un componente aleatorio binomial y con función enlace igual a la identidad. Sin embargo, tiene el problema de que, aunque las probabilidades deben estar entre 0 y 1, el modelo puede predecir a veces valores $\pi(x) > 1$ y $\pi(x) < 0$.

Regresión Logística

Normalmente las relaciones entre $\pi(x)$ y x son no lineales, de modo que el cambio en x tiene menor impacto cuando π está cerca de 0 o de 1 que cuando π está más cerca de la mitad del rango.

La relación habitualmente tiene forma de curva en forma sigmoideal, como puede verse en la *Figura 12*.

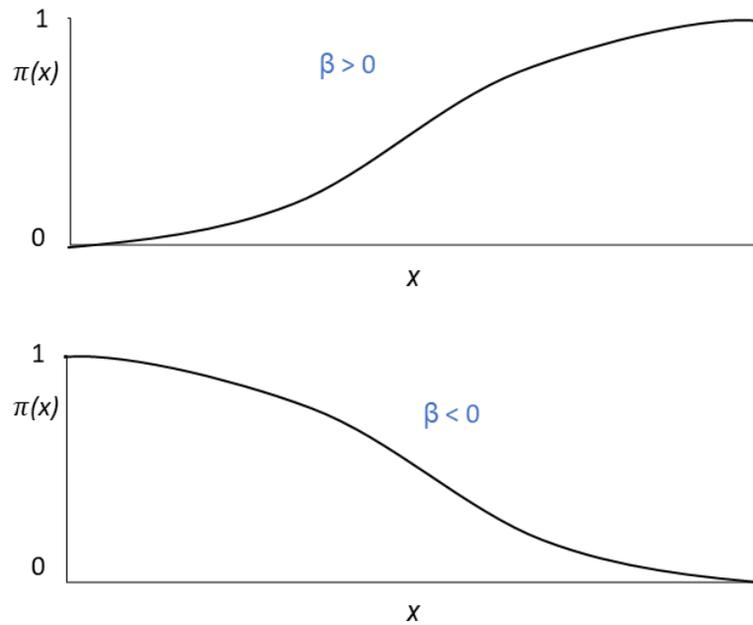


Figura 12. Gráfica de relación entre $\pi(x)$ y x .

La función matemática que modela esta forma es:

$$\pi(x) = \frac{\exp(\alpha + \beta x)}{1 + \exp(\alpha + \beta x)} \quad (30)$$

que se denomina la función logística de la que se derivan los modelos de regresión logística:

$$1 - \pi(x) = 1 - \frac{\exp(\alpha + \beta x)}{1 + \exp(\alpha + \beta x)} = \frac{1}{1 + \exp(\alpha + \beta x)} \quad (31)$$

Por lo que

$$\frac{\pi(x)}{1 - \pi(x)} = \exp(\alpha + \beta x) \Rightarrow \log\left(\frac{\pi(x)}{1 - \pi(x)}\right) = \alpha + \beta x \quad (32)$$

La función de enlace $\log\left(\frac{\pi}{1-\pi}\right)$ de π se denomina función *logit*, de modo que así se asegura que no exista ningún problema estructural respecto al posible rango de valores de π .

El parámetro β determina el rango y la velocidad de incremento o decremento de la curva.

2.5 REDES NEURONALES ARTIFICIALES

ANN, del inglés Artificial Neural Network. Este concepto está motivado por la manera que funciona el sistema nervioso y el cerebro para procesar información. En 1943, el neurofisiólogo Warren McCulloch y Walter Pitts, un apasionado autodidacta de las matemáticas y la lógica, elaboraron el primer modelo artificial de una neurona que simulaba el procesamiento de información que tenía lugar en una neurona real, siendo la base de construcción de redes neuronales mucho más complejas.

Alrededor de 1949, Donald Hebb crea una hipótesis de aprendizaje basado en investigaciones psicofisiológicas acerca del mecanismo de plasticidad neuronal, cuyas principales observaciones son que el aprendizaje se localiza en las sinapsis o conexiones entre las neuronas y que la información se representa en el cerebro mediante un conjunto de neuronas activas o inactivas, a esto se le conoce como *aprendizaje de Hebb*, considerado como el primer modelo de aprendizaje no supervisado, éstas hipótesis se sintetizan en la *regla de aprendizaje de Hebb* que sigue siendo usada en los actuales modelos y que son fueron los primeros modelos de potenciación a largo plazo.

Con el paso del tiempo surgieron modelos inspirados en el trabajo de MacCulloch y Pitts que resultan muy útiles para abordar determinadas tareas, uno de los primeros modelos fue el **perceptrón** de Frank Rosenblatt (1958), un algoritmo de reconocimiento de patrones basado en una red de aprendizaje de computadora de dos capas que utiliza solo adición y sustracción simples, se empleó de inicio como un identificador de patrones ópticos binarios, y salida binaria.

La primera aplicación práctica de las redes neuronales surge hacia 1959 con la *Teoría sobre la adaptación neuronal, Adaline* (del inglés Adaptive Linear Neuron) y Madaline (Multiple Adaline), fueron usados en los filtros adaptativos para eliminar ecos en las líneas telefónicas, desarrollada por Bernard Widrow.

Es importante hacer mención que la investigación de redes neuronales se estancó después de la publicación de la investigación de aprendizaje automático publicado por Marvin Minsky y Seymour Papert de 1969, la cual expone las limitaciones de la naturaleza lineal de los perceptrones básicos incapaces de procesar modelos que implican el operador lógico “disyunción exclusiva”, que solamente es verdadera cuando ambas frases

tienen valores diferentes y es falsa si las dos frases son ambas verdaderas o ambas falsas. Sus valores de verdad se muestran en la *Tabla 2*.

$$\leftrightarrow : P \times P \rightarrow P$$

$$(a, b) \mapsto c = a \leftrightarrow b$$

Tabla 2. Tabla de verdad de la disyunción exclusiva.

<i>a</i>	<i>b</i>	<i>a</i> \leftrightarrow <i>b</i>
<i>V</i>	<i>V</i>	<i>F</i>
<i>V</i>	<i>F</i>	<i>V</i>
<i>F</i>	<i>V</i>	<i>V</i>
<i>F</i>	<i>F</i>	<i>F</i>

Hacia 1987 surge el modelo ART (de las siglas en inglés Adaptive Resonance Theory) de Stephen Grossberg y Gail Carpenter que estudia los principios de la estabilidad de redes neuronales.

Típicamente una Red Neuronal Artificial tiene tres parámetros: el patrón de interconexión entre las capas de las neuronas, el proceso de aprendizaje para la actualización de pesos o costos en las sinapsis y la función de activación. El conocimiento se almacena en los pesos sinápticos. Matemáticamente, la función de red de una neurona se define como la composición de otras funciones, esto se representa como una estructura de red, de modo que las flechas comúnmente usadas representan las dependencias entre variables. Un tipo ampliamente utilizado de la composición es la *suma ponderada no lineal*. La característica más importante de la función de activación es que proporciona una transición suave como valores de entrada de cambio, es decir, un pequeño cambio en la entrada produce un pequeño cambio en la salida.

El modelo de **Perceptrón** es la forma más simple de redes neuronales artificiales, consiste en una sola neurona. Un perceptrón funciona de manera sencilla, toma varias entradas binarias y produce una única salida binaria calculada mediante el ajuste de sus parámetros, también llamados pesos sinápticos.

La *Figura 13* muestra un ejemplo de un modelo neuronal con n entradas y muestra el funcionamiento comparativo entre una neurona biológica y una neurona artificial:

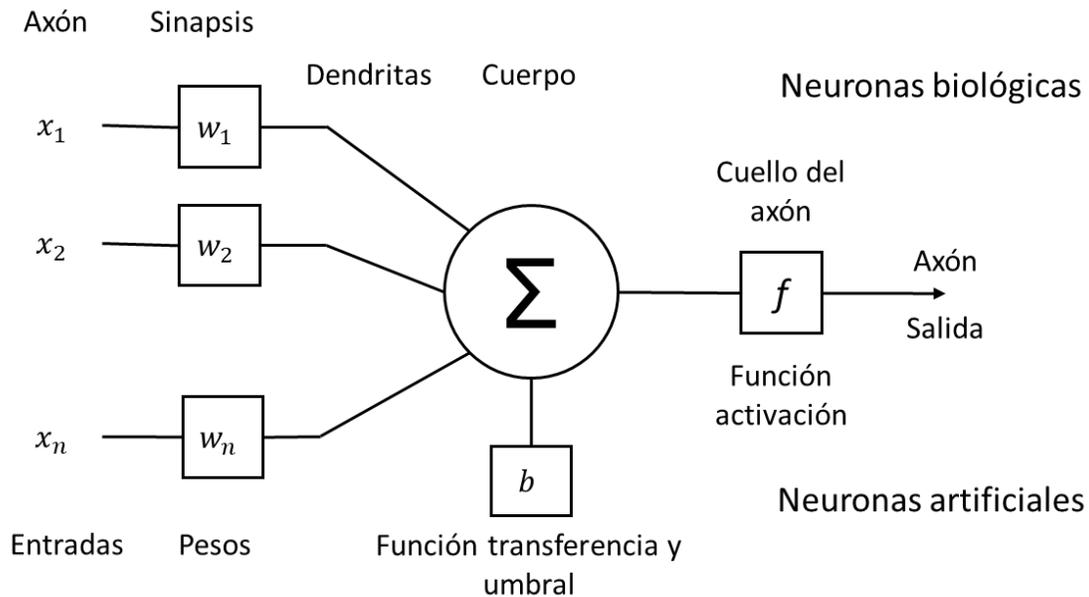


Figura 13. Modelo neuronal con n entradas. Funcionamiento comparativo entre neurona biológica y neurona artificial.

En donde $\{x_1, x_2 \dots x_n\}$ es el conjunto de entradas y $\{w_1, w_2 \dots w_n\}$ pesos correspondientes a cada entrada, Σ es la función transferencia, b el umbral y f la función activación.

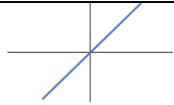
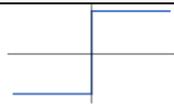
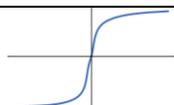
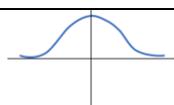
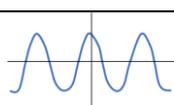
Las entradas son el estímulo que la neurona artificial recibe del entorno que la rodea, y la salida es la respuesta a tal estímulo. La neurona puede adaptarse al medio circundante y aprender de él modificando el valor de sus pesos sinápticos, y por ello son conocidos como los parámetros libres del modelo, ya que pueden ser modificados y adaptados para realizar una tarea determinada.

En este modelo, la salida neuronal Y está dada por:

$$Y = f\left(\sum_{i=1}^n w_i x_i\right) \quad (33)$$

La función de activación se elige de acuerdo a la tarea. La *Tabla 3* muestra las más comunes dentro del campo de las RNAs:

Tabla 3. Función activación de mayor uso en RNA

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, \infty]$	
Escalonada	$y = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -1 & \text{si } x < 0 \end{cases}$	$[-1,1]$	
Sigmoide	$y = \frac{1}{1 + e^{-x}}$	$[0,1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0,1]$	
Sinusoidal	$y = A \text{sen}(\omega x + \varphi)$	$[-1,1]$	

Si la función de activación es la función sigmoide, que es una de las más comúnmente usadas debido a que su derivada está siempre acotada entre 0 y 0.25, cuando tenemos una red con muchas capas, el valor del gradiente cada vez es más cercano a 0. Debido a este problema, las primeras capas de una red neuronal son las más lentas y difíciles de entrenar ya que el valor del gradiente que se usa para actualizar en cada iteración del entrenamiento es muy pequeño. Esto causa un problema adicional, si las primeras capas no están bien entrenadas, el problema propaga el error de una capa a otra, a este problema se le conoce como “problema del desvanecimiento del gradiente” mismo que se subsana adoptando un modelo jerárquico multicapa de redes pre entrenados, una capa a la vez, por aprendizaje no supervisado y refinado por retropropagación, esta adaptación fue desarrollada en 1992 por el alemán Jürgen Schmidhuber. Con el paso de los años este modelo jerárquico ha evolucionado en lo que hoy se conoce como **Aprendizaje Profundo** (Deep Learning, en inglés).

En el modelo de Perceptrón usado para realizar tareas de clasificación en el plano se consideran solo dos entradas, x_1 y x_2 . Para ello, se supone que existe una entrada de peso constante 1 y valor de entrada b y como función de activación a la función signo definida por:

$$\Gamma(s) = \begin{cases} 1, & \text{si } s \geq 0 \\ -1, & \text{si } s < 0 \end{cases} \quad (34)$$

De modo que la salida neuronal Y está dada de la siguiente manera:

$$Y = \begin{cases} 1, & \text{si } w_1x_1 + w_2x_2 + b \geq 0 \\ -1, & \text{si } w_1x_1 + w_2x_2 + b < 0 \end{cases} \quad (35)$$

Supóngase que se tienen dos clases en el plano: la clase C_1 y la clase C_2 como se muestra en la *Figura 14*, representadas por puntos rojos y azules respectivamente, donde cada elemento de estas clases está representado por un punto (x,y) en el plano. Considérese además que tales clases son linealmente separables, es decir, es posible trazar una recta que separe estrictamente ambas clases.

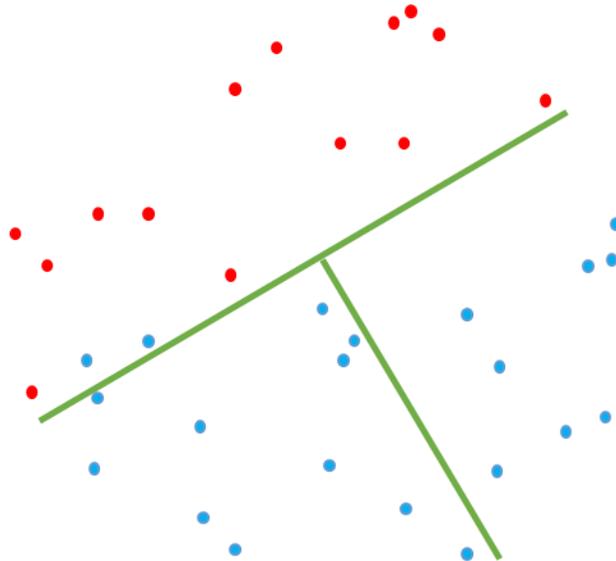


Figura 14. Representación en el plano de dos clases linealmente separables.

Se dirá que el modelo clasifica correctamente las clases C_1 y C_2 si dados los pesos sinápticos w_1 , w_2 , y el término aditivo b , la recta que separa las dos clases y tiene la ecuación:

$$y = -\frac{w_1}{w_2}x - \frac{b}{w_2} \quad (36)$$

La ecuación implícita de la recta es:

$$w_1x + w_2y + b = 0 \quad (37)$$

Obsérvese que si el punto $(x_0, y_0) \in C_1$, entonces $w_1x_0 + w_2y_0 + b < 0$ y, si $(x_0, y_0) \in C_2$, entonces $w_1x_0 + w_2y_0 + b > 0$. Por lo tanto, dado el par $(x_0, y_0) \in C_1 \cup C_2$, la neurona clasifica de la siguiente manera:

$$\begin{aligned} (x_0, y_0) \in C_1 &\leftrightarrow Y = -1 \\ (x_0, y_0) \in C_2 &\leftrightarrow Y = 1 \end{aligned} \quad (38)$$

Si ahora se toman dos clases nueva y distintas a las anteriores C^*_1 y C^*_2 linealmente separables, entonces el modelo podrá no clasificar correctamente estas clases pues la recta anterior puede no ser una recta válida para separarlas. Sin embargo, es posible modificar los parámetros anteriores y obtener nuevos parámetros w^*_1 , w^*_2 y b^* de forma que la recta $y = -\frac{w^*_1}{w^*_2}x - \frac{b^*}{w^*_2}$ sí sirva como recta de separación entre las clases nuevas.

El proceso por el cual la neurona pasa de los parámetros w_1 , w_2 , y b a los parámetros w^*_1 , w^*_2 y b^* se conoce como **aprendizaje**. Este proceso es el que permite modificar los parámetros libres con el fin de que la neurona se adapte y sea capaz de realizar diversas tareas.

El método de corrección de error que se utiliza para adaptar los parámetros libres a fin de clasificar correctamente las clases C_1 y C_2 requiere de un conjunto de entrenamiento definido como $D = C_1 \cup C_2$, un instructor y valores iniciales w'_1 , w'_2 y b' arbitrarios de los parámetros libres.

El entrenamiento consiste en lo siguiente:

- El instructor toma un elemento (x_0, y_0) aleatorio y lo presenta al modelo de la neurona
- Si el modelo clasifica mal este elemento, es decir, si la salida es $Y = 1$ cuando $(x_0, y_0) \in C_1$ o $Y = -1$ cuando $(x_0, y_0) \in C_2$, entonces se aplica la siguiente corrección a los parámetros libres iniciales:

$$\begin{aligned}w_1 &= w'_1 + dx_0 \\w_2 &= w'_2 + dy_0 \\b &= b' + d\end{aligned}\tag{39}$$

- Con d definido como sigue:

$$d = \begin{cases} 1 & \text{si } Y = -1 \text{ y } (x_0, y_0) \in C_1 \\ -1 & \text{si } Y = 1 \text{ y } (x_0, y_0) \in C_2 \end{cases}\tag{40}$$

- Si la neurona clasifica bien el punto (x_0, y_0) , entonces no se realiza ninguna corrección.
- El proceso se repite pasando a la neurona otro punto del conjunto D y usando los últimos parámetros corregidos y no los iniciales. Nuevamente, si la neurona clasifica mal el punto, entonces se aplica una corrección similar a la anterior.
- Esta tarea se repite con todos los puntos del conjunto D . Si en el proceso hubo correcciones, entonces el procedimiento es repetido nuevamente con todos los puntos de D .
- El entrenamiento termina cuando la neurona clasifica correctamente todos los elementos del conjunto de entrenamiento.

Este procedimiento converge, es decir, en un número finito de pasos es posible obtener los parámetros finales que separan entre sí los conjuntos.

Con este modelo se pueden resolver problemas lineales, sin embargo, la mayoría de los problemas reales no son lineales, desarrollándose entonces un modelo que incluía varias capas de neuronas artificiales y generando un nuevo algoritmo de aprendizaje llamado **Perceptrón Multicapa (MLP, del inglés Multi Layer Perceptron)**, su arquitectura se muestra a continuación en la *Figura 15*:

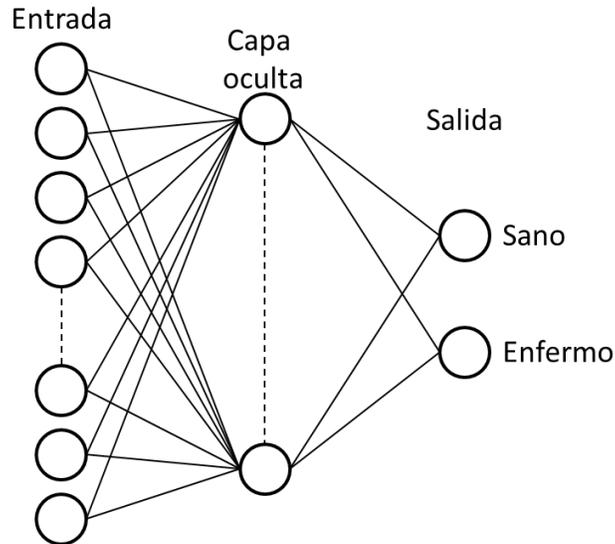


Figura 15. Estructura de una red de perceptrón multicapa [21].

Una sola neurona en la red del Perceptrón multicapa es capaz de separar linealmente su espacio de entrada en dos subespacios por un hiperplano definido por los pesos y el umbral. Los pesos definen la dirección de este hiperplano [22].

El modelo de Perceptrón multicapa puede aprender un aproximador de función no lineal para clasificación o regresión. Es diferente de la regresión logística, ya que entre la capa de entrada y la capa de salida puede haber una o más capas no lineales.

Este tipo de sistemas aprenden y se forman a sí mismos, en vez de ser programados de manera explícita y sobresalen en áreas donde la detección de soluciones o características es difícil de expresar con la programación convencional.

La red del Perceptrón multicapa usa el algoritmo de retropropagación, método para la adaptación de los pesos. El funcionamiento del algoritmo se puede resumir brevemente como sigue:

- Se asignan los pesos sinápticos, generalmente elegidos al azar.
- Se introducen los datos de entrada elegidos al azar en la capa de entrada entre el conjunto de datos de entrada que se usará para el entrenamiento.
- La red genera un vector de datos de salida (propagación hacia adelante).
- Se compara la salida generada por la red con la salida deseada.

- La diferencia obtenida entre la salida generada y la deseada (denominada error) se usa para ajustar los pesos sinápticos de las neuronas de la capa de salidas.
- El error se propaga hacia atrás (back propagation), hacia la capa de neuronas anterior y se usa para ajustar los pesos sinápticos en dicha capa.
- Se continúa propagando el error hacia atrás y ajustando los pesos hasta que se alcance la capa de entradas.
- Este proceso se repetirá con los diferentes datos de entrenamiento.

El algoritmo de retropropagación adapta los pesos y los umbrales de las neuronas de modo que minimicen la **función de error E** definida como sigue:

$$E = \frac{1}{2} \sum_{p=1}^n (d_p - y_p)^2 \quad (41)$$

Donde y_p es la salida real y d_p es la salida deseada para el patrón de entrada p . La minimización de E se puede lograr mediante la función gradiente que realiza sobre los parámetros iteraciones proporcionales al valor negativo del gradiente en el punto actual, es decir el ajuste de los pesos para cambiar el valor de E en dirección de su gradiente negativo (descenso por gradiente).

Para una red neuronal simplificada con n capas ocultas y con una neurona por capa como se muestra en la *Figura 16*, la salida de cada neurona se calcula multiplicando la salida de la neurona previa por el parámetro w_n y aplicando la función de activación $f(t)$. La función de error E al final de la red devuelve el error del sistema y se utiliza para modificar el resto de parámetros de la red a través del descenso por gradiente.

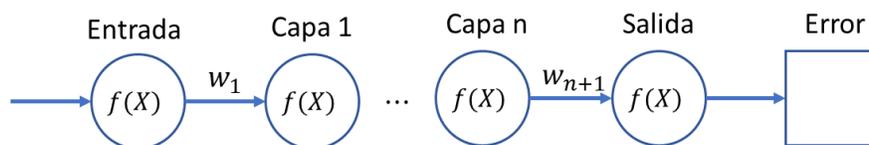


Figura 16. Red neuronal simple de n capas ocultas y una neurona por capa.

De modo que la derivada respecto al parámetro de la primera capa empleando la regla de la cadena, queda como sigue:

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial salida} \frac{\partial salida}{\partial capa_n} \dots \frac{\partial capa_2}{\partial capa_1} \frac{\partial capa_1}{\partial w_1} \quad (42)$$

En donde

$$\frac{\partial salida}{\partial capa_n} = \frac{\partial f(z_1)}{\partial z_1} w_{n+1} \text{ con } z_1 = capa_n w_{n+1} \quad (43)$$

Una ventaja a destacar de este modelo es su flexibilidad ya que se puede ser tan sencillo o elaborado como se requiera, empleando una red desde unos miles a unos cuantos millones de unidades neuronales o conexiones que, si bien parecieran muchas, apenas se asemejan a la potencia de cálculo de un gusano [23].

3 OBJETIVO

El objetivo de este trabajo es desarrollar un modelo de clasificación automática con Aprendizaje Automático supervisado basado en Redes Neuronales Artificiales (RNA) en el modelo de Perceptrón Multicapa y Modelos Lineales Generalizados con regresión lineal (MLG), evaluar la viabilidad de su uso como método auxiliar de clasificación para mejorar la precisión en el diagnóstico de pacientes de Parkinson y compararlo con la precisión combinada del diagnóstico clínico de la EP reportado por diversos artículos. Estará enfocado en el problema de clasificación binaria, es decir, clasificar a un sujeto como sano o enfermo. Se emplearán técnicas de AA aplicadas a una base de datos obtenidos de imágenes adquiridas mediante tomografías por emisión de positrones en sujetos control y pacientes diagnosticados con la EP.

4 MATERIALES Y MÉTODOS

4.1 BASE DE DATOS

En este trabajo, se utilizó una base de datos previamente adquirida de estudios cerebrales PET como prueba de concepto de 4 pacientes con Parkinson y dos controles, adquiridos con dos radiofármacos por sujeto: (+)- α -[^{11}C]Dihidotetabenazina (^{11}C -DTBZ) y ^{11}C -Raclopride (^{11}C -RAC), utilizados para marcar neuronas pre-sinápticas y post-sinápticas respectivamente. Un ejemplo de las imágenes adquiridas se ilustra en la *Figura 17*. Todos los sujetos firmaron una carta de consentimiento informado y el protocolo del cual fueron tomados los datos fue aprobado por comités de ética.

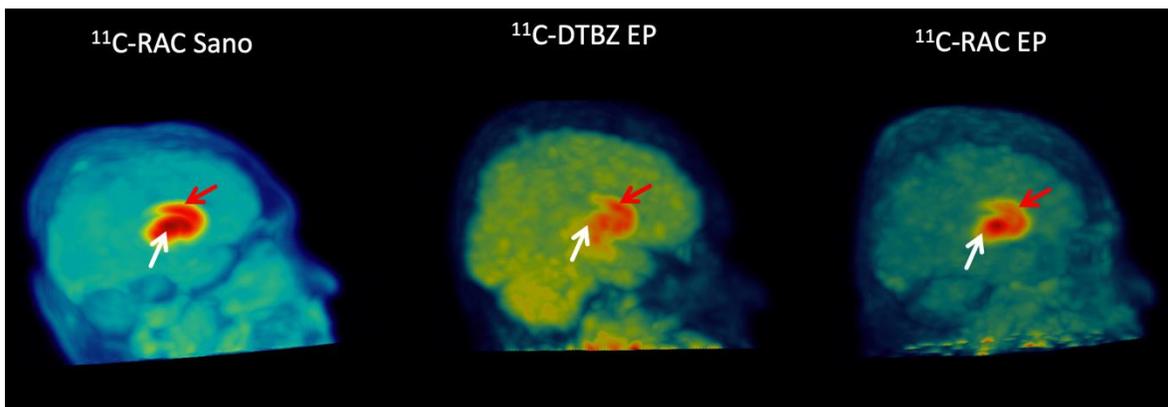


Figura 17. Ejemplo de captación de los radiofármacos evaluados en cerebro de un sujeto sano comparado con uno con la enfermedad de Parkinson (EP). Las flechas blancas indican el putamen y las rojas el caudado. Notar la disminución en la captación de ^{11}C -DTBZ en el paciente en putamen y el aumento con ^{11}C -RAC.

El radiofármaco pre-sináptico en enfermos de Parkinson tendrá una menor captación en el putamen con respecto a sanos cuando existen alteraciones en la vía dopaminérgica mientras que la captación del radiofármaco postsináptico aumentará su captación en la misma región debido a la disminución de dopamina endógena [11].

Las imágenes se adquirieron en el equipo Siemens Biograph 64 de la Unidad PET/CT de la Facultad de Medicina de la UNAM. Se reconstruyeron las imágenes usando un algoritmo 2D-OSEM en una matriz de 128 x 128 píxeles con un tamaño de píxel de 4mm

por lado, las imágenes fueron corregidas por tiempo muerto de los detectores, decaimiento, eventos dispersados, aleatorios y por atenuación, usando una tomografía computarizada adquirida antes de cada estudio.

Las imágenes fueron normalizadas anatómicamente y analizadas para obtener valores de captación a nivel pixel en regiones estriales, usando la corteza occipital y el cerebelo como región de referencia para ^{11}C -DTBZ y ^{11}C -RAC respectivamente. La normalización anatómica de las imágenes es importante para comparar la captación a nivel pixel entre sujetos, pues los cerebros difieren en tamaño y forma. La normalización se realizó usando el software Statistical Parametric Mapping (SPM) en Matlab (R2014_b) con un tamaño de pixel de 2 mm. Se utilizó cada pixel del cuerpo estriado como región de interés usando un atlas anatómico [12]. En la *Figura 18* se muestra el ejemplo de una máscara usada para obtener las regiones de interés y de referencia para realizar el cociente.

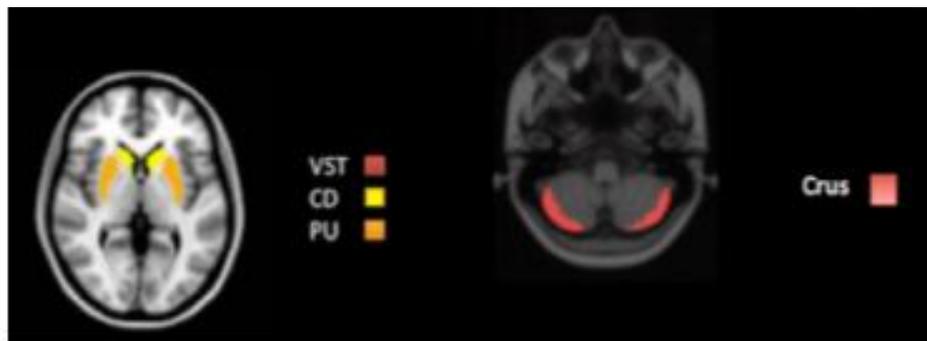


Figura 18. Ejemplo de máscara usada para obtener el cociente de la región de interés (izquierda) y la región de referencia (derecha).

Para realizar un análisis semicuantitativo, se dividió la captación específica en el estriado entre la región de referencia (promedio de cerebelo o corteza occipital) estriado/corteza occipital para ^{11}C -DTBZ y estriado/cerebelo para ^{11}C -RAC.

La base de datos usadas en este trabajo para evaluar los algoritmos de clasificación automática está compuesta por aproximadamente 2700 valores por radiofármaco empleado por sujeto, provenientes del valor semicuantitativo de cada pixel dentro del estriado (ya normalizado a la región de referencia). Entendemos por base de datos al conjunto de datos obtenidos por sujeto, en total se analizaron seis conjuntos de datos pertenecientes a 6 sujetos (cuatro enfermos y 2 controles). Las etiquetas que han sido

asignadas a los sujetos permiten distinguir entre bases de datos de pacientes con Parkinson y de sujetos control.

Un ejemplo en el espacio euclidiano de la relación de la captación de los dos radiofármacos de un sujeto control y de un paciente en se ilustra en la *Figura 19*.

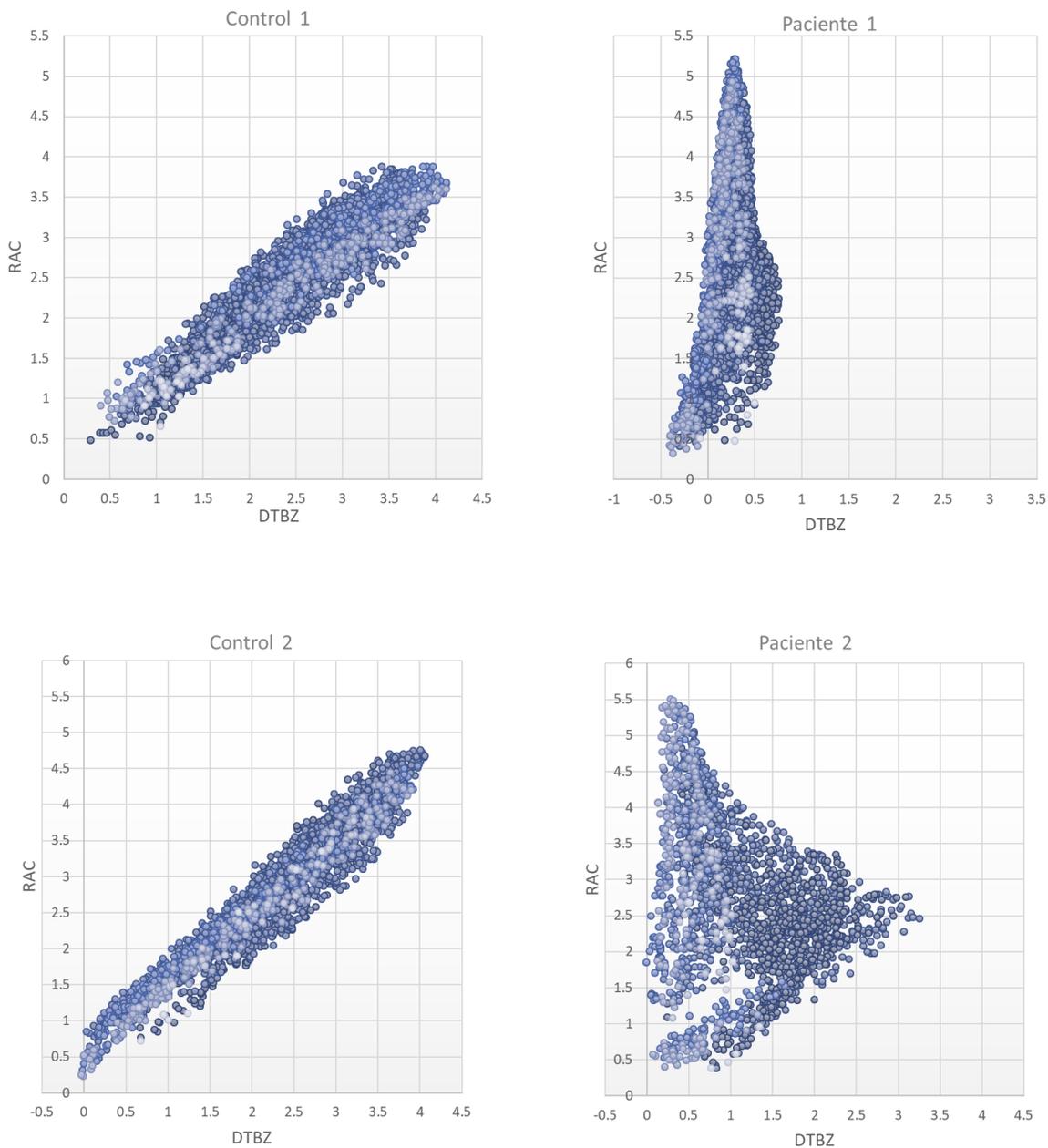


Figura 19. Ejemplos de la relación de la captación de ^{11}C -DTBZ y ^{11}C -RAC en sujeto control (izquierda) y paciente con Parkinson (derecha). Valor de cada pixel en el estriado entre la región de referencia.

4.2 DISCRIMINACIÓN DE DATOS Y CLASIFICADORES EMPLEADOS

Se desarrollaron dos algoritmos de clasificación para el análisis de los datos, el primero en base a Redes Neuronales Artificiales (MLP, del inglés Multi Layer Perceptron) y el segundo, a modo comparativo, fue desarrollado con Regresión logística de Modelos Lineales Generalizados (MLG). Dado que se contaba con datos de dos radiofármacos, se alimentó el algoritmo con la información separada de cada radiofármaco y posteriormente con ambos radiofármacos en conjunto.

Se definen como variables de entrada los valores correspondientes a los radiofármacos y los valores de salida correspondientes al diagnóstico: 0-Sano, 1-Enfermo.

La base de datos fue dividida en conjuntos de entrenamiento y prueba a fin de que el algoritmo aprenda de manera supervisada la clasificación binaria requerida. Con el uso de la librería Scikitlearn en la versión 0.19.1 (<https://scikit-learn.org/stable/index.html>) de Python 3.6.5 (<https://www.python.org/downloads/release/python-365/>), se seccionan los datos y se realizan los ajustes necesarios de los parámetros como número y tamaño de capas ocultas, función activación, solucionador para la optimización de los pesos, índice de aprendizaje, número de iteraciones, entre otros, a fin de optimizar la sensibilidad y especificidad del clasificador que implican una mejora en el porcentaje de predicción.

El código del algoritmo desarrollado en base a Redes Neuronales Artificiales empleando los valores de los dos radiofármacos como parámetros de entrada se muestra a continuación.

File Edit View Insert Cell Kernel Help

Trusted Python 3

Code

```
In [1]: import pandas as pd
import numpy as np
df = pd.read_csv('data/DATOS_TODOS.csv')
```

```
In [2]: # Muestra primeros 5 registros
df.head()
```

```
In [3]: #Se definen variables de entrada 'X' y salida 'y'
feature_cols = ['DTBZ', 'RAC']
X = df[feature_cols]
y = df.ESTADO
```

```
In [4]: # Divide 'X' y 'y' en conjuntos de entrenamiento y prueba
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

```
In [5]: # Entrena el modelo de clasificación de Redes Neuronales en el conjunto de entrenamiento
from sklearn.neural_network import MLPClassifier
#mlp=MLPClassifier(hidden_layer_sizes=(12,12,12), max_iter=500, alpha=0.0001, solver='adam', random_state=1, tol=0.00000001)
mlp = MLPClassifier(hidden_layer_sizes=(6,6,6,6), solver='adam', max_iter=6000)

# Aprende la relación entre el conjunto de entrada y salida del conjunto de entrenamiento
mlp.fit(X_train, y_train)
```

```
In [6]: #Realiza predicciones de clase para el conjunto de pruebas
y_pred_class=mlp.predict(X_test)

#Muestra métricas del clasificador
from sklearn.metrics import classification_report
print (classification_report(y_test, y_pred_class))
```

Modulo de predicción pixel a pixel

```
In [7]: #Realiza predicciones de clase para el conjunto de pruebas
#new_pred = y_pred_class=mlp.predict([[1, 1], [2,2], [3,3], [4,4]])
#print(new_pred)
```

```
In [8]: # Calcula valor de predicción
from sklearn import metrics
print(metrics.accuracy_score(y_test, y_pred_class))
```

```
In [9]: # Examina la distribución de clases del conjunto de pruebas
y_test.value_counts()
```

...

```
In [10]: # Calcula el porcentaje de 'unos' presentes en el conjunto
y_test.mean()
```

...

```
In [11]: # Calcula el porcentaje de 'ceros' presentes en el conjunto prueba
1 - y_test.mean()
```

...

```
In [12]: # Calcula 'Null accuracy' (En problemas binarios)
max(y_test.mean(), 1 - y_test.mean())
```

...

```
In [13]: # Calcula 'Null accuracy' (Para problemas de clasificación multi-clase)
y_test.value_counts().head(1) / len(y_test)
```

...

```
In [14]: # Muestra las primeras 25 respuestas verdaderas y predichas
print('True:', y_test.values[0:25])
print('Pred:', y_pred_class[0:25])
```

Matriz de Confusión

Verdadero Positivo (VP)	Falso Negativo (FN)
Falso Positivo (FP)	Verdadero Negativo (VN)

Terminología

- **Verdadero Positivo (VP):** Predice *correctamente* paciente con Parkinson
- **Verdadero Negativo (VN):** Predice *correctamente* paciente sin Parkinson
- **Falso Positivo (FP):** Predice *incorrectamente* paciente con Parkinson
- **Falso Negativo (FN):** Predice *incorrectamente* paciente sin Parkinson

```
In [15]: print(metrics.confusion_matrix(y_test, y_pred_class))
```

```
In [16]: # Guarda la matriz de confusión y la secciona en 4 partes
confusion = metrics.confusion_matrix(y_test, y_pred_class)
VP = confusion[1, 1]
VN = confusion[0, 0]
FP = confusion[0, 1]
FN = confusion[1, 0]
```

```
In [17]: print((VP + VN) / float(VP + VN + FP + FN))
print(metrics.accuracy_score(y_test, y_pred_class))
```

...

```
In [18]: #Error de Clasificación (Tasa de clasificación errónea)
print((FP + FN) / float(VP + VN + FP + FN))
print(1 - metrics.accuracy_score(y_test, y_pred_class))
```

...

Sensibilidad

```
In [19]: #Sensibilidad (Detección de casos positivos)
print(VP / float(VP + FN))
print(metrics.recall_score(y_test, y_pred_class))
```

...

Especificidad

```
In [20]: #Especificidad: Cuando el valor real es negativo, ¿con qué frecuencia es correcta la predicción?
print(VN / float(VN + FP))
```

```

In [21]: #Tasa Falso Positivo
print(FP / float(VN + FP))
...

In [22]: #Precisión (En casos positivos)
print(VP / float(VP + FP))
#print(metrics.precision_score(y_test, y_pred_class))
...

In [23]: # Muestra la predicción de las primeras 10 respuestas
mlp.predict(X_test)[0:10]
...

In [24]: # Muestra las primeras 10 probabilidades predichas de pertenencia a clases
mlp.predict_proba(X_test)[0:10, :]
...

In [25]: # Muestra las primeras 10 probabilidades predichas para la clase 1
mlp.predict_proba(X_test)[0:10, 1]
...

In [26]: y_pred_prob = mlp.predict_proba(X_test)[: , 1]

In [27]: %matplotlib inline
import matplotlib.pyplot as plt

In [28]: # histograma de probabilidades predichas
plt.hist(y_pred_prob, bins=8)
plt.xlim(0, 1)
plt.title('Histograma de probabilidades predichas RNA para DTBZ y RAC')
plt.xlabel('Probabilidades predichas de Parkinson')
plt.ylabel('Frecuencia')
#plt.savefig('Histo_nn_todo.png')
...

In [29]: # predice Parkinson si la probabilidad pronosticada es mayor a 0.3
from sklearn.preprocessing import binarize
y_pred_class = binarize([y_pred_prob], 0.7)[0]

In [30]: # Muestra las primeras 10 probabilidades predichas
y_pred_prob[0:10]
...

In [31]: # Matriz de confusión previa (umbral por defecto 0.5)
print(confusion)
...

In [32]: # Nueva matriz de confusión (Umbral de 0.3)
print(metrics.confusion_matrix(y_test, y_pred_class))

In [33]: #Guarda nueva matriz de confusión (umbral 0.3)
nueva_conf = metrics.confusion_matrix(y_test, y_pred_class)
VP2 = nueva_conf[1, 1]
VN2 = nueva_conf[0, 0]
FP2 = nueva_conf[0, 1]
FN2 = nueva_conf[1, 0]

In [34]: # Calcula nueva Sensibilidad
print(VP2 / float(VP2 + FN2))
print(metrics.recall_score(y_test, y_pred_class))
...

In [35]: # Calcula nueva Especificidad
print(VN2 / float(VN2 + FP2))
...

In [42]: #Calcula Precisión con valores de la nueva matrix de confusión
print((VP2 + VN2) / float(VP2 + VN2 + FP2 + FN2))
print(metrics.accuracy_score(y_test, y_pred_class))

```

```
In [36]: # IMPORTANTE: Pareja ordenada (valores reales, probabilidades predichas)
fpr, tpr, thresholds = metrics.roc_curve(y_test, y_pred_prob)
plt.plot(fpr, tpr)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.0])
plt.title('Curva ROC RNA para DTBZ y RAC')
plt.xlabel('Tasa Falso Positivo (1 - Especificidad)')
plt.ylabel('Tasa Verdadero (Sensibilidad)')
plt.grid(True)
#plt.savefig('ROC_nn_todo')
```

...

```
In [37]: def evaluate_threshold(threshold):
print('Sensitividad:', tpr[thresholds > threshold][-1])
print('Especificidad:', 1 - fpr[thresholds > threshold][-1])
```

```
In [38]: evaluate_threshold(0.3)
```

...

```
In [43]: evaluate_threshold(0.5)
```

...

```
In [40]: print(metrics.roc_auc_score(y_test, y_pred_prob))
```

...

```
In [41]: from sklearn.model_selection import cross_val_score
cross_val_score(mlp, X, y, cv=10, scoring='roc_auc').mean()
```

4.3 EVALUACIÓN Y OPTIMIZACIÓN DEL ALGORITMO

Considerando el propósito de clasificación sano-enfermo de este trabajo, el problema de clasificación se reduce a sólo dos clases.

Para la evaluación y optimización del algoritmo, se emplea la **Matriz de Confusión** o **Matriz de Error** que es una matriz de 2x2 que se usa a menudo para describir el rendimiento de un modelo de clasificación (o "clasificador") en un conjunto de datos de prueba para los cuales se conocen los valores reales (modelo supervisado), el uso de esta matriz en una clasificación binaria de sólo dos clases resulta ideal.

Dado un clasificador y una entrada, hay cuatro salidas posibles. Si la entrada es positiva y es etiquetada como positiva, se toma como *Verdadero positivo*, si es clasificado como negativo, se considera como un *Falso Negativo*. Si la entrada es negativa y se etiqueta como negativo, toma el valor de *Verdadero Negativo*, en cambio, si se clasifica como positivo, se cuenta como *Falso Positivo*. La *Figura 20* muestra la Matriz de Confusión.

		Clase Real	
		Verdadero Positivo (VP)	Falso Positivo (FP)
Clase predicha	Verdadero Positivo (VP)	Verdadero Positivo (VP)	Falso Positivo (FP)
	Falso Negativo (FN)	Falso Negativo (FN)	Verdadero Negativo (VN)

Figura 20. Matriz de Confusión de los valores reales y predichos del modelo de clasificación binaria.

$$\text{Sensibilidad} = \frac{VP}{VP + FN}$$

$$\text{Especificidad} = \frac{VN}{VN + FP}$$

En la práctica, un clasificador binario puede cometer dos tipos de errores, asignar incorrectamente a un sujeto que por defecto está en la categoría no predeterminada, o puede asignar incorrectamente a un sujeto que no está predeterminada a la categoría predeterminada. Emplear una matriz de confusión da una mejor idea de lo que el modelo de clasificación está haciendo en cuanto a aciertos y errores, para ello se emplean diversas métricas antes mencionadas en la sección de Evaluación del Clasificador.

La precisión de la clasificación puede resultar engañosa si se tiene un número desigual de observaciones en cada clase, en el caso de la base de datos que se cuenta, se tiene el mismo número de observaciones con ambos radiofármacos para cada conjunto de datos.

El algoritmo de clasificación por Redes Neuronales Artificiales (MLP) implementa un algoritmo de perceptrón multicapa que se entrena con Retropropagación inicialmente establecido en 0.5. Se entrenan dos matrices: una matriz X de tamaño $(n_{\text{muestras}}, n_{\text{parámetros}})$, que contiene las muestras de entrenamiento representadas como vectores de características de punto flotante; y una matriz y de tamaño (n_{muestras}) , que contiene los valores objetivo (etiquetas de clase) para las muestras de entrenamiento. El conjunto de los datos de entrenamiento que toma el algoritmo es aleatorio, por lo que cambia cada vez que se ejecuta el programa lo que resulta en una mínima variación de los porcentajes de predicción.

5 RESULTADOS

Se desarrollaron en Python dos los dos algoritmos basados en modelos de RNA y MLG alimentados con distintas entradas, por un lado, se empleó como entrada los valores correspondientes a los datos de los radiofármacos de manera independiente y posteriormente empleando los datos de los dos radiofármacos de manera conjunta como entradas complementarias. Los resultados de los algoritmos de clasificación se muestran en la *Tabla 4* y en la *Figura 21* se muestra la curva ROC comparando los resultados.

Tabla 4. Resultados de Clasificadores por parámetros

	RNA			RL		
Umbral 0.3						
Radiofármaco	¹¹ C-DTBZ	¹¹ C-RAC	¹¹ C-DTBZ ¹¹ C-RAC	¹¹ C-DTBZ	¹¹ C-RAC	¹¹ C-DTBZ ¹¹ C-RAC
Falso Positivo	24 %	27 %	10 %	24 %	22 %	19 %
Precisión (0)	83 %	86 %	81 %	83 %	83 %	81 %
Precisión (1)	89 %	87 %	94 %	89 %	89 %	91 %
Margen de error inicial	13 %	13 %	10 %	13 %	13 %	13 %
Sensibilidad	97 %	88 %	98 %	96 %	86 %	95 %
Especificidad	62 %	91 %	61 %	63 %	85 %	68 %
Precisión	85 %	86 %	87 %	84 %	87 %	86 %
Umbral 0.5						
Sensibilidad	92 %	97 %	87 %	92 %	92 %	90 %
Especificidad	76 %	62 %	94 %	76 %	75 %	81 %
Predicción	87 %	85 %	90 %	86 %	86 %	87 %

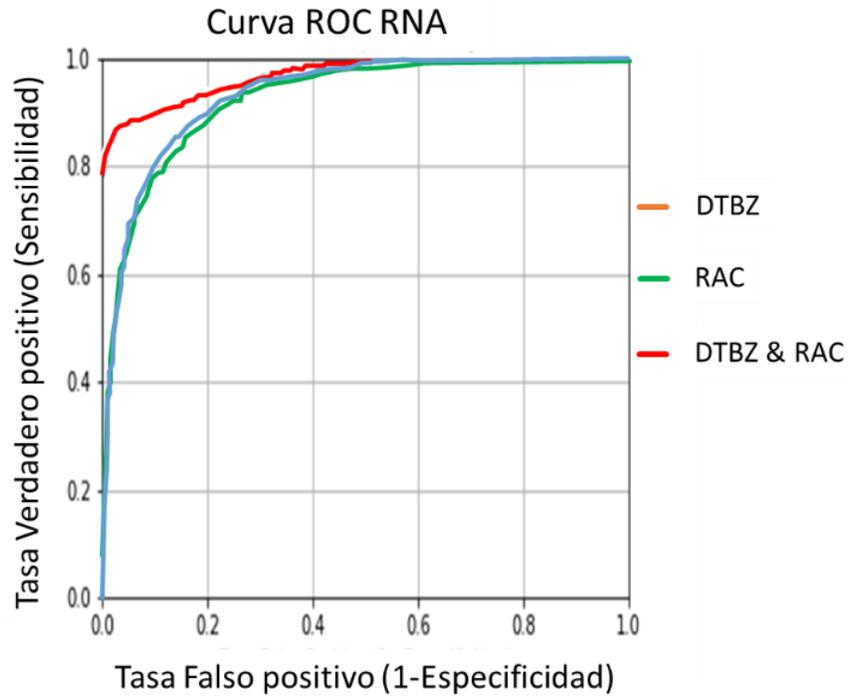


Figura 21. Curva ROC comparativa del desempeño de RNA con los distintos parámetros de alimentación.

5.1.1 Clasificador por Redes Neuronales Artificiales (RNA) para el Radiofármaco ^{11}C -DTBZ

En el análisis realizado para el radiofármaco pre-sináptico se obtuvieron los siguientes valores:

Para el valor de umbral establecido en 0.3, se obtuvo: Precisión conjunta del 85%, por clase de 83% para 0-Sano y 89% para 1-Enfermo, margen de error de clasificación inicial de 13%, Sensibilidad de 97% para la detección de casos positivos y Especificidad de 62%, así como una tasa de detección de Falsos Positivos de 24%. Para el valor del umbral establecido en 0.5, se logra una **Precisión** del 87%, con **Sensibilidad** a 92% y **Especificidad** a 76%. Las siguientes gráficas muestran la frecuencia de las probabilidades predichas y la curva ROC que auxilia en la elección de un umbral que equilibre sensibilidad y especificidad de manera que optimice el valor de la predicción y minimice el margen de error.

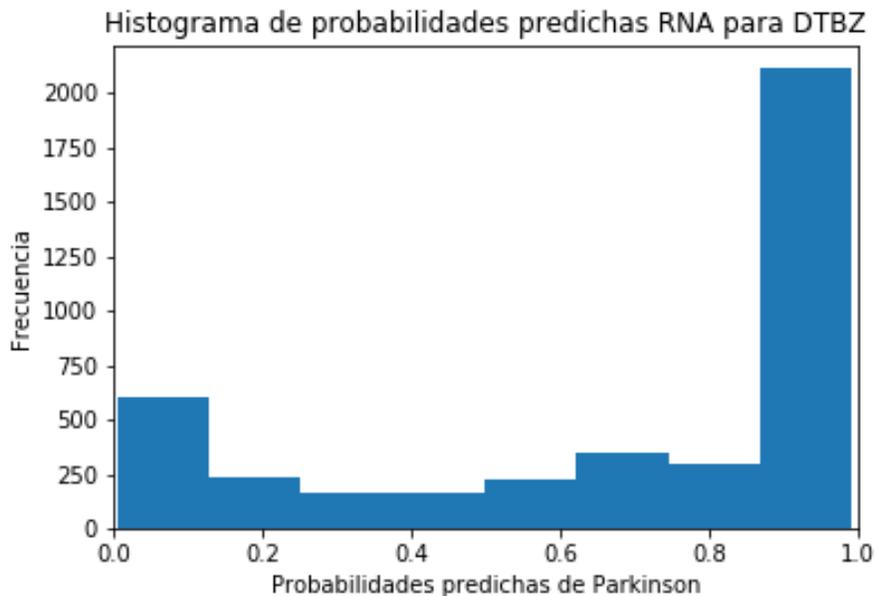


Figura 22. Histograma de predicción de probabilidades RNA con base ^{11}C -DTBZ

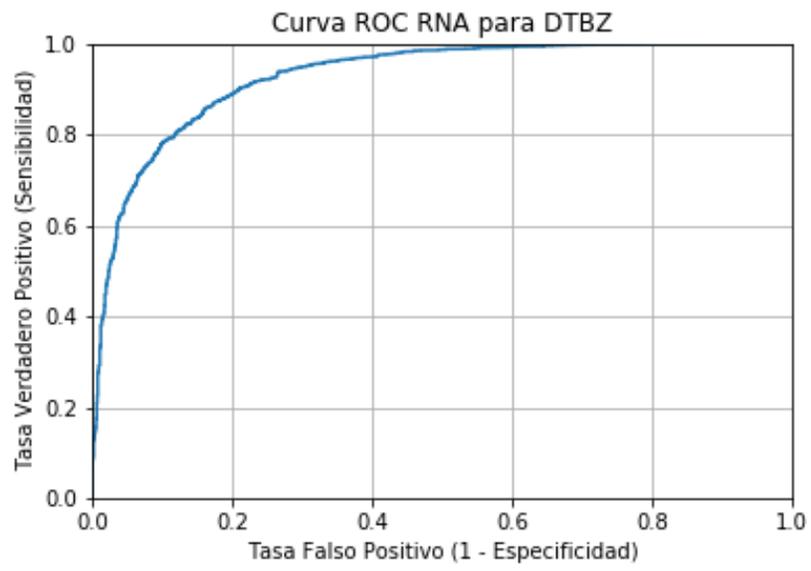


Figura 23. Curva ROC RNA con base ^{11}C -DTBZ. La curva ROC del clasificador indica una buena precisión dada la relación especificidad/sensibilidad.

5.1.2 Clasificador por Redes Neuronales Artificiales (RNA) para el Radiofármaco ^{11}C -RAC

En el análisis realizado para el radiofármaco post-sináptico se obtuvieron los siguientes valores:

Para el valor de umbral establecido en 0.3, se obtuvo: Precisión conjunta del 86%, por clase de 86% para 0-Sano y 87% para 1-Enfermo, margen de error de clasificación inicial de 13%, Sensibilidad de 88% para la detección de casos positivos y Especificidad de 91%, así como una tasa de detección de Falsos Positivos de 27%. Para el valor del umbral establecido en 0.5, se logra una **Precisión** del 85%, con **Sensibilidad** a 97% y **Especificidad** a 62%.

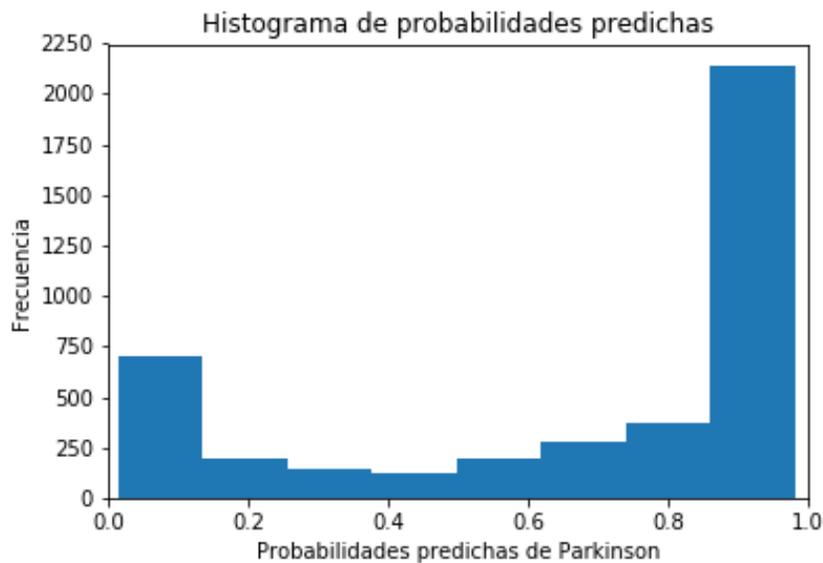


Figura 24. Histograma de predicción de probabilidades RNA con base ^{11}C -RAC.

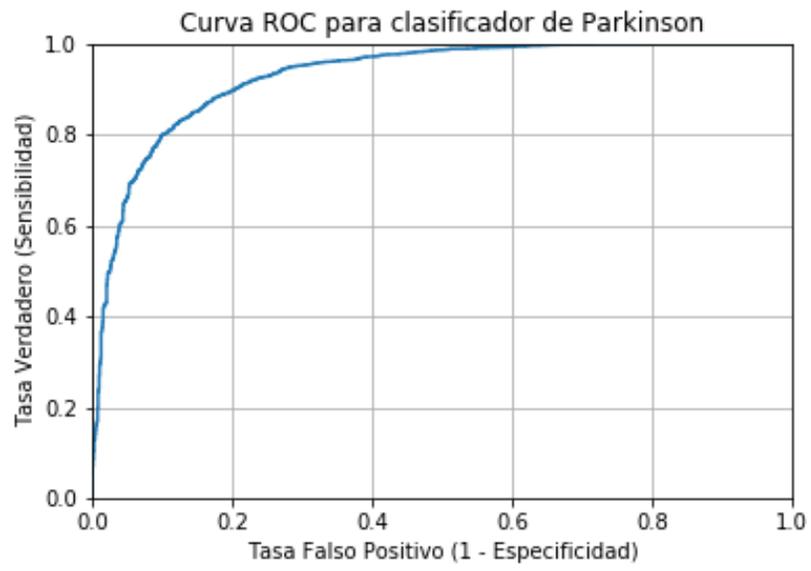


Figura 25. Curva ROC RNA con base ^{11}C -RAC. La curva ROC del clasificador indica la precisión dada la relación especificidad/sensibilidad.

5.1.3 Clasificador por Redes Neuronales Artificiales (MLP) para los datos combinados de ^{11}C -DTBZ y ^{11}C -RAC

Se realizó el análisis de los datos en conjunto para evaluar los mismos parámetros evaluados de forma independiente, obteniendo los siguientes resultados:

Para el valor de umbral establecido en 0.3, se obtuvo: Precisión conjunta del 87%, por clase de 81% para 0-Sano y 94% para 1-Enfermo, margen de error de clasificación inicial de 10%, Sensibilidad de 98% para la detección de casos positivos y Especificidad de 61%, así como una tasa de detección de Falsos Positivos de 10%. Para el valor del umbral establecido en 0.5, se logra una **Precisión** del 90%, con **Sensibilidad** a 87% y **Especificidad** a 94%, siendo la mejor precisión lograda.

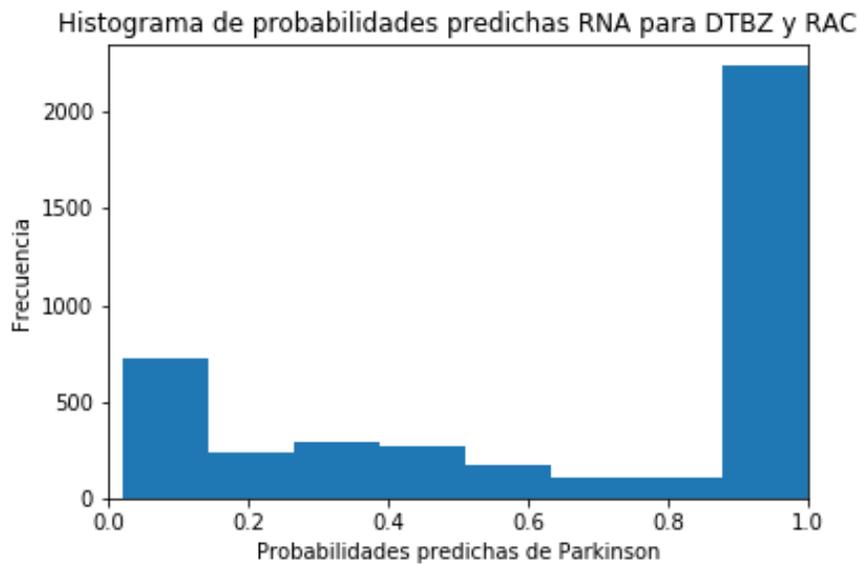


Figura 26. Histograma de predicción de probabilidades MLP con base ^{11}C -DTBZ y ^{11}C -RAC.

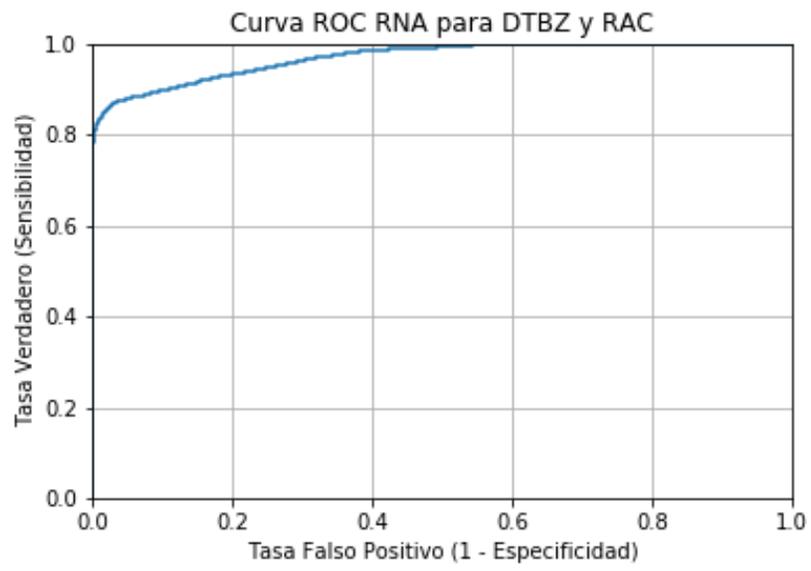


Figura 27. Curva ROC RNA con base ^{11}C -DTBZ y ^{11}C -RAC de especificidad/sensibilidad. El Área bajo la curva (AUC) es la mayor obtenida en comparación con los otros clasificadores y los distintos parámetros, resultando éste el mejor clasificador.

5.1.4 Clasificador por Regresión logística de Modelos Lineales Generalizados (GLM) para el Radiofármaco ^{11}C -DTBZ

A modo comparativo, se desarrolló un algoritmo de clasificación basado en GLM dado que es otro modelo comúnmente usado en el área médica. En el análisis realizado para el radiofármaco pre-sináptico se obtuvieron los siguientes valores:

Para el valor de umbral establecido en 0.3, se obtuvo: Precisión conjunta del 84%, por clase de 81% para 0-Sano y 94% para 1-Enfermo, margen de error de clasificación inicial de 13%, Sensibilidad de 96% para la detección de casos positivos y Especificidad de 63%, así como una tasa de detección de Falsos Positivos de 24%. Para el valor del umbral establecido en 0.5, se logra una **Precisión** del 86%, con **Sensibilidad** a 92% y **Especificidad** a 76%.

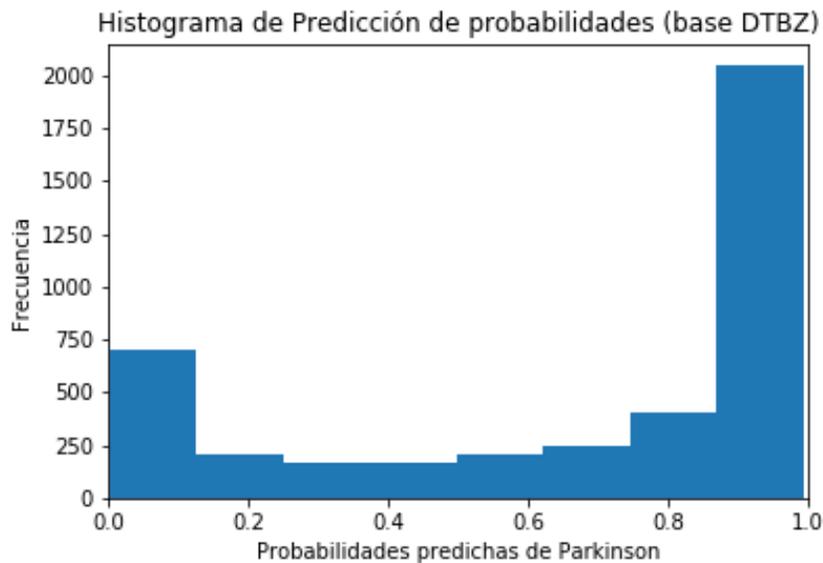


Figura 28. Histograma de predicción de probabilidades GML con base ^{11}C -DTBZ.

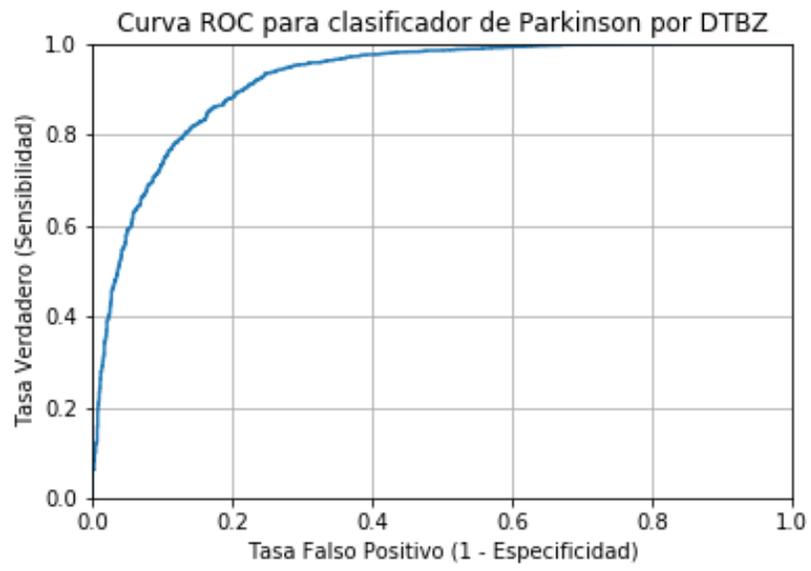


Figura 29. Curva ROC GML con base ^{11}C -DTBZ.

5.1.5 Clasificador por Regresión logística de Modelos Lineales Generalizados (GLM) para el Radiofármaco ¹¹C-RAC

En el análisis realizado para el radiofármaco post-sináptico se obtuvieron los siguientes valores:

Para el valor de umbral establecido en 0.3, se obtuvo: Precisión conjunta del 87%, por clase de 83% para 0-Sano y 89% para 1-Enfermo, margen de error de clasificación inicial de 13%, Sensibilidad de 86% para la detección de casos positivos y Especificidad de 86%, así como una tasa de detección de Falsos Positivos de 22%. Para el valor del umbral establecido en 0.5, se logra una **Precisión** del 86%, con **Sensibilidad** a 92% y **Especificidad** a 75%.

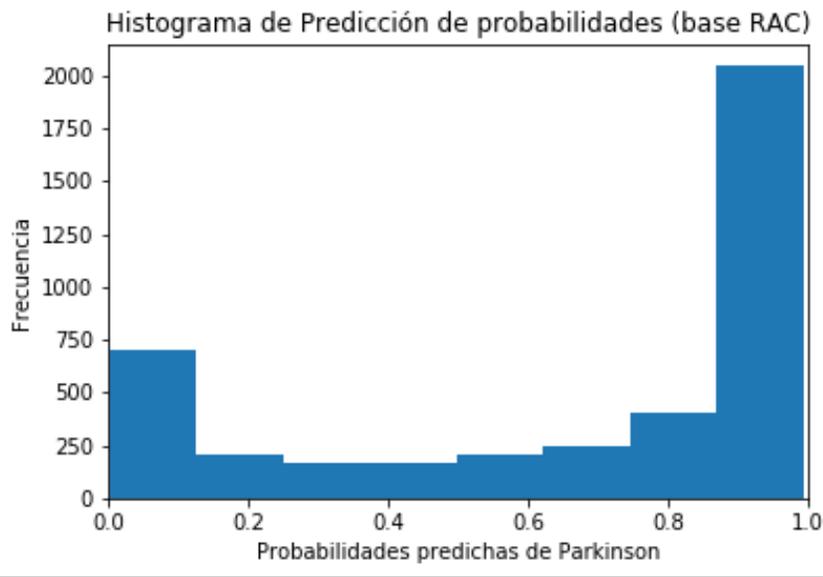


Figura 30. Histograma de predicción de probabilidades GML con base ¹¹C-RAC.

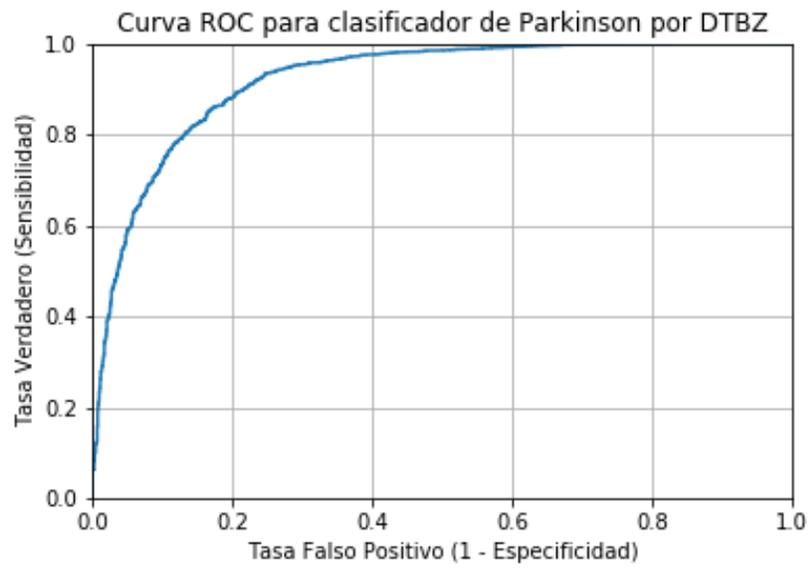


Figura 31. Curva ROC GML con base ^{11}C -RAC.

5.1.6 Clasificador por Regresión logística de Modelos Lineales Generalizados (GLM) para los datos combinados de ¹¹C-DTBZ/¹¹C-RAC

Finalmente, se realizó el análisis de los datos de ambos radiofármacos en conjunto, obteniendo los siguientes resultados:

Para el valor de umbral establecido en 0.3, se obtuvo: Precisión conjunta del 86%, por clase de 81% para 0-Sano y 91% para 1-Enfermo, margen de error de clasificación inicial de 13%, Sensibilidad de 95% para la detección de casos positivos y Especificidad de 68%, así como una tasa de detección de Falsos Positivos de 19%. Para el valor del umbral establecido en 0.5, se logra una **Precisión** del 87%, con **Sensibilidad** a 90% y **Especificidad** a 81%.

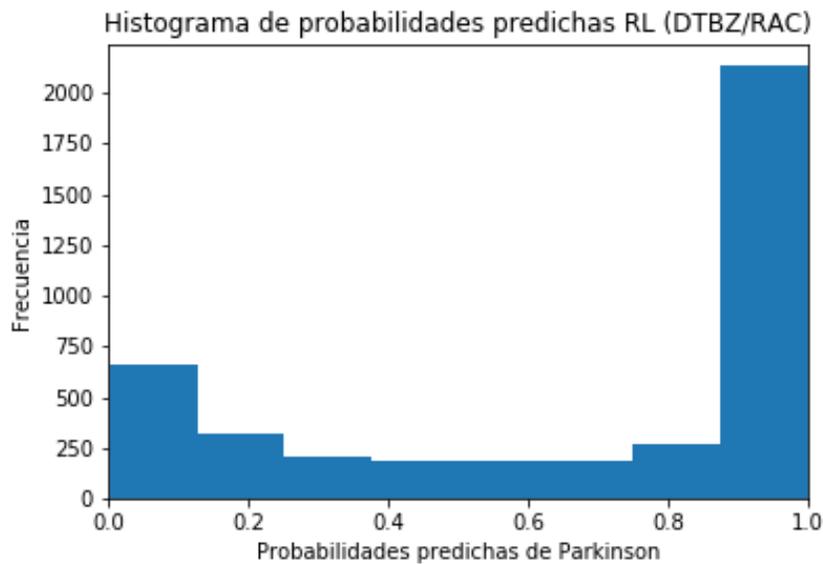


Figura 32. Histograma de predicción de probabilidades GML con base ¹¹C-DTBZ y ¹¹C-RAC.

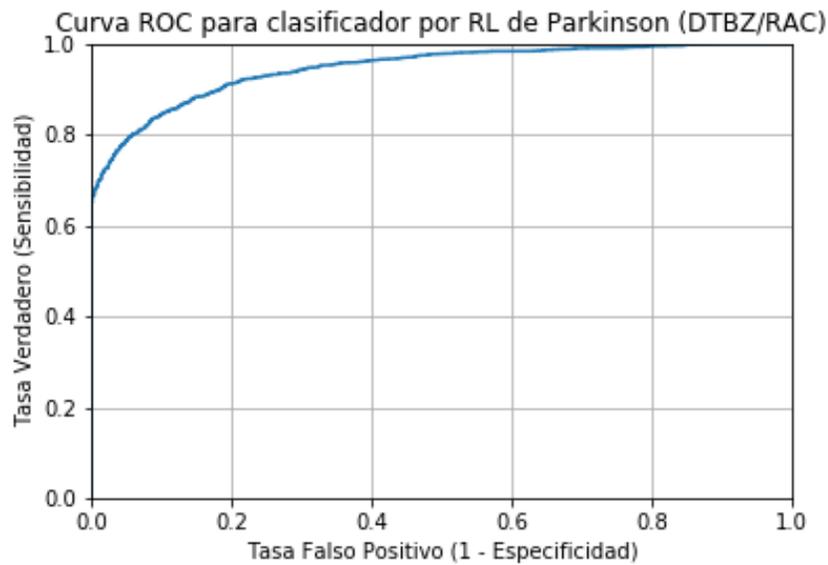


Figura 33. Curva ROC GML con base ^{11}C -DTBZ y ^{11}C -RAC. Se observa que el área bajo la curva (AUC) es mayor al emplear los dos radiofármacos.

Después de analizar los dos algoritmos empleados, RNA y GML individualmente en cada radiofármaco y en conjunción, se obtuvo un mejor desempeño al emplear los dos radiofármacos juntos como parámetros para ambos algoritmos, siendo el de mejor precisión el de Redes Neuronales Artificiales.

6 DISCUSIÓN Y CONCLUSIONES

En este trabajo, evaluado como prueba de principio, se demostró que es posible usar las herramientas computacionales para clasificar de manera exitosa, de forma usuario independiente, para clasificar datos provenientes de imágenes PET de pacientes y controles sanos, que pueden ser usados para mejorar el diagnóstico clínico.

Basados en los resultados, podemos destacar dos observaciones principales: los algoritmos clasifican con mayor precisión a los pacientes con la enfermedad del Parkinson y los porcentajes de precisión incrementan cuando se emplean los datos de los dos radiofármacos ^{11}C -DTBZ y ^{11}C -RAC como parámetros complementarios. El uso complementario de los datos de los dos radiofármacos muestra mejores resultados puesto que uno funciona como biomarcador presináptico y el otro postsináptico, permitiéndonos comparar las dos poblaciones, considerando que también son mayor cantidad de datos.

Los resultados empleando sólo un radiofármaco fueron mejores para ^{11}C -DTBZ que para ^{11}C -RAC en cuanto a precisión y equilibrio entre sensibilidad y especificidad.

Es entendible la mayor precisión en la clasificación de enfermos con Parkinson dado que se alimentó el algoritmo con una base de datos de 4 pacientes y 2 sujetos control, de modo que se cuenta con mayor información en cuanto a los datos que corresponden a pacientes, cabe mencionar que conforme se incrementa el número de datos y la diversidad de ellos, el algoritmo mejorará la precisión dado que la característica más poderosa del Machine Learning es precisamente su capacidad de mejorar con un mayor número de datos.

Al comparar los resultados del algoritmo de RNA y el de GML, notamos mayor precisión en el primero, aunque la diferencia no es significativa.

Se desarrolló un algoritmo empleando las últimas herramientas y librerías especializadas en el área de la inteligencia artificial.

Cada vez es más frecuente la medicina personalizada, aumentando la demanda de pruebas especializadas de diagnóstico para la óptima selección de tratamientos, basados en los datos obtenidos de los distintos métodos que ayuden a tomar decisiones y tratamientos ajustados a las necesidades específicas de cada paciente.

Si bien es cierto que en los últimos años las nuevas técnicas IA en análisis de imágenes médicas como auxiliares en el diagnóstico y prevención de enfermedades han cobrado mayor relevancia, también lo es la carencia de la colaboración multidisciplinaria en la investigación y desarrollo de estas técnicas, a efecto de lograr que se detecten enfermedades en los primeros estadios para que los pacientes afectados por enfermedades neurodegenerativas reciban la atención antes de que se presenten los primeros síntomas para que tengan una mejor calidad de vida, ya que el diagnóstico temprano beneficia tanto al paciente como a su familia.

Aunque los resultados de esta investigación son alentadores, la limitante de este trabajo se hallan en el número de imágenes evaluadas, sin embargo, se sientan las bases para retroalimentar el algoritmo con más datos, pudiendo ser de utilidad para clasificar Parkinson de otras enfermedades que se le parecen en síntomas pero que no son Parkinson e incluso podría ser usado para predecir la enfermedad antes de que el paciente presente los primeros síntomas, pero se necesita evaluar el algoritmo en una población más grande y variada.

Otra de las aportaciones del trabajo es que el código se puede alimentar con datos provenientes de otras regiones o de otros radiofármacos, abriendo las puertas al estudio de otras enfermedades, usando el mismo código desarrollado en este trabajo.

6.1 TRABAJO FUTURO

La investigación realizada en este trabajo abre las puertas a nuevas líneas de investigación que completen y mejoren el mismo.

- **Optimización del algoritmo.** Para mejorar el porcentaje de predicción es necesario analizar más a fondo las métricas y parámetros que se han empleado, así como explorar otras opciones de clasificación y evaluación de clasificadores en el vasto mar de opciones.
- **Mejorar la herramienta.** El algoritmo realiza predicciones en el conjunto de datos, elemento a elemento, siendo de mayor utilidad entrenarlo y probarlo para conjuntos de datos, esto es, que realice las predicciones en conjuntos de datos y no sólo por elementos, facilitando más adelante el análisis por regiones funcionales del cerebro.
- **Proveer al algoritmo con mayor cantidad de datos.** Se cuenta con una base de datos de más de 30 mil elementos, aunque aún carece de diversidad y distribución. Es recomendable e ideal alimentar la base de datos con más información para que el algoritmo pueda aprender la sutileza de los pequeños cambios.

Una vez establecida la base del análisis de datos obtenidos de imágenes médicas, el abanico de posibilidades es muy amplio y diverso en cuanto a aplicaciones de IA que auxilien en el diagnóstico de múltiples enfermedades.

7 BIBLIOGRAFÍA

- [1] L. M. de Lau and M. M. Breteler, "Epidemiology of Parkinson's disease," *Lancet Neurol.*, vol. 5, no. 6, pp. 525–535, Jun. 2006.
- [2] F. Moisan, S. Kab, and F. Mohamed, "Parkinson disease male-to-female ratios increase with age: French nationwide study and meta-analysis," *J Neurol Neurosurg Psychiatry*, vol. 87, pp. 952–957, 2016.
- [3] E. R. Dorsey, T. Sherer, M. S. Okun, and B. R. Bloem, "The Emerging Evidence of the Parkinson Pandemic," *J. Parkinsons. Dis.*, vol. 8, no. s1, pp. S3–S8, Dec. 2018.
- [4] A. Cervantes Arriaga *et al.*, "Caracterización de la enfermedad de Parkinson en México: estudio ReMePARK," *Gac. Med. Mex.*, vol. 149, pp. 497–501, 2013.
- [5] A. Caliskan, H. Badem, A. Bastürk, and M. E. Yüksel, "Diagnosis of the parkinson disease by using deep neural network classifier," Erciyes University, Kayseri. Turkey, 2017.
- [6] L. del C. Hernández Ramírez, J. Querol Vinagre, A. L. de la Garza Barroso, and A. Garfias Escarpulli, "Enfermedad de Parkinson," *Boletín Epidemiológico*, vol. 51 (32), 2015.
- [7] A. C. Belin and M. Westerlund, "Parkinson's disease: A genetic perspective," *FEBS J.*, vol. 275, no. 7, pp. 1377–1383, Apr. 2008.
- [8] G. Rizzo, M. Copetti, S. Arcuti, D. Martino, A. Fontana, and G. Logroscino, "Accuracy of clinical diagnosis of Parkinson disease: A systematic review and meta-analysis.," *Neurology*, vol. 86, no. 6, pp. 566–76, Feb. 2016.
- [9] C. S. Caldart, "ANÁLISIS DE IMÁGENES CEREBRALES CON FDG-PET EN RELACIÓN A UNA PLANTILLA DE ESTUDIOS NORMALES," Universidad Nacional de Cuyo, 2012.
- [10] X. Hua *et al.*, "3D characterization of brain atrophy in Alzheimer's disease and mild cognitive impairment using tensor-based morphometry.," *Neuroimage*, vol. 41, no. 1, pp. 19–34, May 2008.

- [11] A. Avendaño Estrada, "Uso de métodos matemáticos para el análisis cuantitativo de imágenes moleculares basadas en tomografía por emisión de positrones," Universidad Nacional Autónoma de México, 2018.
- [12] A. C. Tziortzi *et al.*, "Imaging dopamine receptors in humans with [11C]-(+)-PHNO: Dissection of D3 signal and anatomy," *Neuroimage*, vol. 54, no. 1, pp. 264–277, Jan. 2011.
- [13] P. Varona Martínez, "Escalas, modelos y técnicas de simulación en neurociencia computacional," Universidad Autónoma de Madrid, 1997.
- [14] C. Blank, "Penrose y la Inteligencia Artificial," *EPISTEME NS*, vol. 20–1, pp. 29–49, 2000.
- [15] D. O. Hebb, *Organización de la conducta*. Debate, 1985.
- [16] R. Benitez, G. Escudero, and S. Kanaan, *Inteligencia artificial avanzada*. Barcelona, 2013.
- [17] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM J. Res. Dev.*, vol. 44, no. 1.2, pp. 206–226, Jan. 2000.
- [18] T. M. Mitchell, *Machine Learning*. McGraw-Hill Science, 1997.
- [19] B. Marr, "The Top 10 AI And Machine Learning Use Cases Everyone Should Know About," 2016. [Online]. Available: <https://www.forbes.com/sites/bernardmarr/2016/09/30/what-are-the-top-10-use-cases-for-machine-learning-and-ai/#5830d64694c9>.
- [20] "Computer Technology Helps Radiologists Spot Overlooked Small Breast Cancers | Cancer Network," *Mod. Med. Netw.*, vol. 14, no. 10, 2000.
- [21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.
- [23] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, "The Structure of the Nervous System of the Nematode *Caenorhabditis elegans*," *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 314, no. 1165, pp. 1–340, Nov. 1986.

- [24] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," 2007.
- [25] Acton, P., & Newberg, A. (2006). *Artificial neural network classifier for the diagnosis of Parkinson's disease using [99mTc]TRODAT-1 and SPECT*. *Physics In Medicine And Biology*, 51(12), 3057-3066. doi: 10.1088/0031-9155/51/12/004
- [26] Belin AC, Westerlund M. *Parkinson's disease: a genetic perspective*. *FEBS J* 2008; 275:1377-83.
- [27] Benítez, R., Escudero, G., Kanaan, S. & Masip Rodó, D. (2013). *Inteligencia artificial avanzada*. Barcelona: UOC.
- [28] Bermudo Ávila, J. (1983). *La filosofía moderna y su proyección contemporánea*. Barcelona, España: Barcanova.
- [29] Bishop, C. (2016). *Pattern recognition and machine learning*. [s.l.]: Springer-Serlag New York.
- [30] Breiman, L. (1996). *Bagging Predictors*. *Machine Learning*, 24(3), 123-140.
- [31] Breiman, L., Friedman, J., Stone, C., & Olshen, R. *Classification and Regression Trees*. New York: Chapman & Hall.
- [32] Burges, C. J. C. (1998). *A tutorial on Support Vector Machines for Pattern Recognition*. *Data Mining and Knowledge Discovery*, 2(2), 121-167.
- [33] Caliskan, A., Badem, H., Bastürk, A., Yüksel, M. (2017). *Diagnosis of the parkinson disease by using deep neural network classifier*. Vol 17(2), ppp. 3311-3318.
- [34] Carmona Suárez, E. (2014). *Tutorial sobre Máquinas de Soporte (SVM)*. Universidad Nacional de Educación a Distancia (UNED), España.
- [35] Cervantes-Arriaga A, Rodriguez-Violante M, Lopez-Ruiz M, EstradaBellmann I, Zuñiga-Ramirez C, Otero-Cerdeira E, et al. *Caracterización de la enfermedad de Parkinson en México: estudio ReMePARK*. *Gac Med Mex*. 2013; 149: 497-501.
- [36] Cost, S., & Salzberg, S. (1993). *A weighted nearest neighbor algorithm for learning with symbolic features*. *Machine Learning*, 10(1), 57-78. doi: 10.1007/bf00993481
- [37] D. G. Standaert, M. H. Saint-Hilaire, C. A. Thomas. (2015). *Parkinson's Disease Handbook*, American Parkinson Disease Association, New York, USA.
- [38] Das R. (2010). *A comparison of multiple classification methods for diagnosis of Parkinson disease*. *Expert Systems with Applications*, vol. 37, no 2, pp.1568-1572.
- [39] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861-874. doi: 10.1016/j.patrec.2005.10.010
- [40] Finger, S. (2005). *Minds behind the brain*. Oxford: Oxford University Press.

- [41] Gala García, I. (2013). *Algoritmos SVM para problemas sobre big data* (Máster en Investigación e Innovación en Tecnologías de la Información y las Comunicaciones). Universidad Autónoma de Madrid, España.
- [42] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An introduction to statistical learning*. Springer.
- [43] Katz, Y. (2012). *Noam Chomsky on Where Artificial Intelligence Went Wrong*. Recuperado de https://www.theatlantic.com/technology/archive/2012/11/noam-chomsky-on-where-artificial-intelligence-went-wrong/261637/?single_page=true
- [44] Moisan F, Kab S, Mohamed F, et al *Parkinson disease male-to-female ratios increase with age: French nationwide study and meta-analysis*. J Neurol Neurosurg Psychiatry 2016;87:952-957.
- [45] *Neurological disorders: Public health challenges*. World Health Organization, 2006. ISBN 978 92 4 156336 9.
- [46] N. Singh, V. Pillay, Y. E. Choonara. (2007). *Advances in the treatment of Parkinson's disease*. Progr. Neurobiol., vol. 81, pp. 29–44.
- [47] Pajares Martinsanz, G., & Santos Peñas, M. (2005). *Inteligencia artificial e ingeniería del conocimiento*. Madrid: Ra-Ma.
- [48] Pérez, M. (2010). *Nuevos modelos estadísticos para detección de patrones de hipo/perfusión-metabolismo en imágenes de tomografía funcional cerebral*. Ph.D Tesis, Universidad de Granada, España.
- [49] Ruiz González, J. (2017). *Fuzzy C-Mean Distribuido en la Nube*. (Grado en Ingeniería Informática). Universitat Oberta de Catalunya, España.
- [50] Vrooman, J.R. (1970). *René Descartes: A Biography*. New York: G. P. Putnam's Sons.
- [51] Wang, J., Yen, G., & Polycarpou, M. (2012). *Advances in neural networks--* ISSN 2012. Berlin: Springer.
- [52] Zadeh, L. (1999). *Fuzzy sets as a basis for a theory of possibility*. Fuzzy Sets And Systems, 100, 9-34. doi: 10.1016/s0165-0114(99)80004-9
- [53] Zhang, Y. (2017). *Can a Smartphone Diagnose Parkinson Disease? A Deep Neural Network Method and Telediagnosis System Implementation*. Parkinson's Disease, 1-11. doi: 10.1155/2017/6209703