



universidad
de león



FACULTAD DE CIENCIAS BIOLÓGICAS Y AMBIENTALES

**DIAGNÓSTICO DE CÁNCER
MEDIANTE INTELIGENCIA
ARTIFICIAL**

**CANCER DIAGNOSIS WITH
ARTIFICIAL INTELLIGENCE**

Autor: Diego Matilla Olivera

Tutora: María Montserrat López Cabeceira

GRADO EN BIOTECNOLOGÍA

Julio, 2023

ÍNDICE

1. Introducción.....	1
1.1. ¿Qué es la inteligencia artificial?.....	1
1.2. ¿Qué es el cáncer?.....	2
1.3. Métodos de diagnóstico de cáncer de mama.....	4
1.4. Métodos de diagnóstico de cáncer de piel.....	5
1.5. ¿Por qué usar inteligencia artificial?.....	6
2. Objetivos y procedimiento.....	6
2.1. Objetivos.....	6
2.2. Procedimiento.....	7
3. Modelos de inteligencia artificial.....	8
3.1. Regresión lineal y PCA para cáncer de mama.....	9
3.2. Automatización con <i>sklearn</i>	10
3.3. Aprendizaje profundo de una red perceptrón multicapa.....	11
3.4. Red neuronal convolucional para imágenes de cáncer de piel.....	13
3.5. Método alternativo de lectura de imágenes mediante <i>OpenCV</i>	15
4. Resultados.....	16
4.1. Observaciones generales.....	17
4.2. Regresión lineal con y sin PCA para cáncer de mama.....	17
4.3. Perceptrón multicapa para cáncer de mama.....	18
4.4. Red convolucional para cáncer de piel.....	20
4.5. Método alternativo de lectura de imágenes mediante <i>OpenCV</i>	22
5. Conclusiones.....	23
6. Aplicaciones futuras.....	24
7. Bibliografía.....	25
Anexos.....	28

RESUMEN

El cáncer es la enfermedad más prevalente del mundo. Debido a su dificultad en el tratamiento y a la dependencia de las condiciones genéticas del paciente, el diagnóstico es la principal vía de mejora de la calidad de vida del paciente. Aquí entra en juego la inteligencia artificial, la cual hoy en día parece estar por todas partes. Sin embargo, más allá de sus usos personales, posee un alto potencial para emplearse en el ámbito científico-médico para tareas que impliquen el reconocimiento de patrones como es el diagnóstico de cáncer. A lo largo de este trabajo se realizarán modelos de inteligencia artificial que emplearán diferentes algoritmos para verificar si es posible el diagnóstico de cáncer con una elevada exactitud. Se modelarán desde algoritmos numéricos como la regresión lineal múltiple, hasta complejos algoritmos similares al funcionamiento del cerebro humano, como es el aprendizaje profundo. Se explorarán estas opciones para determinar cuales son capaces de ofrecer un diagnóstico altamente fiable para un tipo específico de cáncer, ya sea mediante parámetros fisiopatológicos o por análisis de imágenes, y sin aplicar ningún tipo de técnica invasiva al paciente.

Palabras clave: clasificación de tumores, *python*, redes neuronales, regresión lineal

ABSTRACT

Cancer is the most prevalent disease in the world. Due to its difficult treatment and its dependance on the patient genes, diagnosis has become the main way to improve patient life quality. This is where artificial intelligence comes in, which seems to be everywhere nowadays. Nevertheless, beyond its personal uses, it possesses a high potential for application in the scientific and medical field, particularly for tasks that involves pattern recognition, such as cancer diagnosis. Throughout this project, artificial intelligence models will be developed based on different algorithms in order to high accuracy cancer diagnosis. It will be modeled from numerical algorithms, such as multiple linear regression, to complex human brain-like algorithms, such as deep learning. These options will be tested aiming the determination of which ones are capable of diagnose a specific type of cancer with a high reliability, using either physiopathological parameters or image analysis, without applying any invasive techniques on the patient.

Keywords: lineal regression, neural networks, *python*, tumor classification

1. INTRODUCCIÓN

En este apartado se realizará una introducción abordando los conocimientos previos que se requerirán a lo largo del trabajo para comprender con una mayor facilidad lo que se ha realizado.

1.1. ¿Qué es la inteligencia artificial?

A lo largo de este año han aparecido diferentes modelos de inteligencia artificial (IA) como son el *chat Generative Pre-trained Transformers* (chatGPT), google bard o la revolución de los *deepfakes*. Las empresas dedicadas a su creación mismas han hecho su uso de dominio público a través de internet. Sin embargo, la IA lleva existiendo desde hace mucho tiempo, aunque a escalas mucho más pequeñas y de forma privada. Algunos ejemplos serían los denominados "algoritmos de las redes sociales" o los asistentes de voz como *Siri* o *Alexa*. Este trabajo se centrará en su utilización para el diagnóstico de cáncer *in silico* a una menor escala.

La IA se define como una rama de la ciencia de la computación que se dedica a la creación de algoritmos tales que sean capaces de imitar el comportamiento del cerebro humano para tomar decisiones en base a la experiencia (Shimizu y Nakayama, 2020). Además, se especializa en la búsqueda de patrones complejos dentro de cantidades masivas de datos, por lo que su uso en medicina es inmediato debido a que el diagnóstico se basa principalmente en la comparación con otros pacientes anteriores (Gubatan *et al.*, 2021). Existen diferentes maneras de clasificar la IA según las propiedades que presentan o las funciones que realizan. Atendiendo al tipo y complejidad de algoritmo empleado, se ha definido la siguiente clasificación representada en la **figura 1** (Shimizu y Nakayama, 2020).

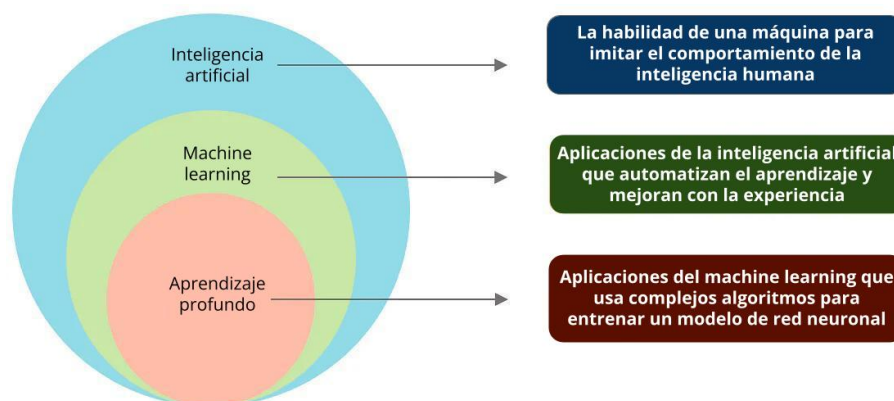


Figura 1: representación de las diferentes subclases de IAs (modificado de Shimizu y Nakayama, 2020).

Se puede diagnosticar cáncer mediante cualquiera de los tipos presentes en la **figura 1**, dado que cada uno de ellos posee sus ventajas y sus inconvenientes. Concretamente, se comenzará usando una regresión lineal, la cual se clasifica dentro del primer tipo de IA y posee la ventaja de ser el algoritmo más rápido en tiempo de ejecución y de poseer una alta reproducibilidad (Chang *et al.*, 2021). Posteriormente, se pasará al uso de algoritmos de aprendizaje profundo mediante complejos diseños de redes neuronales que toman unos datos de entrada, los procesan y emiten una salida en función de sus parámetros internos y de un umbral de excitación. Una red neuronal está compuesta por capas. Las principales son la de entrada, que toma los datos y la de salida, que emite el diagnóstico. Las capas intermedias se denominan capas ocultas, una red neuronal puede poseer tantas capas ocultas o tantas neuronas por capa como quiera el usuario. Es más sencillo el uso de este tipo de algoritmos de aprendizaje profundo para la búsqueda de patrones complejos, además de ser capaz de lograrlo a una velocidad mucho mayor que una persona. Sin embargo, este tipo de algoritmos poseen una baja reproducibilidad respecto al entrenamiento, por lo que es importante guardar los datos cuando la red finalice, para evitar entrenar de nuevo y conseguir así resultados reproducibles. El diagnóstico a partir de una red neuronal entrenada sí es reproducible; es decir, una vez que el entrenamiento ha sido guardado en un archivo, se puede cargar de nuevo al modelo y continuar desde donde lo dejó, por lo que el diagnóstico será el mismo para los mismos datos de entrada todas las veces y siempre que no se manipule la red de forma externa (Looney, 1997).

1.2. ¿Qué es el cáncer?

El cáncer es una enfermedad de base celular caracterizada por la desregulación tanto de las cascadas de señalización como del metabolismo, lo que causa un crecimiento descontrolado de las células (Upadhyay, 2021). Se trata de una de las enfermedades más prevalentes del mundo y se estima que un tercio de la población sufrirá cáncer en algún momento de su vida. Según la Organización Mundial de la Salud, la incidencia de cáncer en el año 2020 fue de 201 afectados por cada 100.000 personas, siendo el cáncer de mama el más prevalente con 47'8 afectadas por cada 100.000 personas, seguido por el cáncer de próstata con 30'7, pulmón con 22'4 o colorrectal con 19'5; superando entre los 4 la mitad de los tumores diagnosticados durante ese año (*Cancer today*, sin fecha). Asimismo, se trata de una de las enfermedades más letales que puede sufrir una persona, caracterizándose también por la dificultad de su tratamiento. El hecho de que esta enfermedad afecte de forma distinta a cada persona y de la heterogeneidad de los tipos de cáncer hace que existan numerosos tratamientos (Greaves, 2015). Principalmente se utilizan la inmunoterapia (Morrison *et al.*, 2018), la radioterapia o la quimioterapia (Petit *et al.*, 2021) y la cirugía, la cual intenta eliminar completamente el tejido infectado del organismo (Kaufman, 2019).

Estos tratamientos dependen del tipo de cáncer y de las condiciones del paciente. Por consiguiente, no siempre son eficaces en la eliminación completa de la enfermedad incluso bajo condiciones semejantes de dos pacientes.

Existen diferentes maneras de clasificar al cáncer, la más común es según el tejido afectado. Dicha clasificación comprende tres tipos (Kindt *et al.*, 2007):

1. Carcinoma: son aquellos tumores donde las células transformadas son células del tejido endodérmico o ectodérmico, tejidos como la piel o el revestimiento de los órganos. Es el tipo más prevalente, superando el 80% de los casos, sus variantes más comunes son el cáncer de colon, de mama, de próstata o de pulmón.
2. Leucemias y linfomas: también llamados cáncer líquido, afectan a las células sanguíneas o a las células madre hematopoyéticas de la médula ósea roja. Las leucemias proliferan independientemente y los linfomas como masas tumorales. Es el segundo tipo en prevalencia, alcanzando unos valores del 9% de los casos.
3. Sarcoma: son aquellos tumores específicos de tejidos mesodérmicos como son el tejido muscular, el óseo, el adiposo o el cartilaginoso. Este último tipo es el menos prevalente de todos, se estima que posee una incidencia del 1% (Kindt *et al.*, 2007).

Este trabajo se centrará principalmente en cáncer del tipo carcinoma, concretamente en el cáncer de mama y de piel. Si bien para el estudio de cáncer mediante IA se podría realizar empleando cualquier tipo de cáncer, la elección fue de estos dos tipos en concreto debido a la disponibilidad de una cantidad suficiente de datos de buena calidad y ya preparados para entrenar IA. Es decir, los datos habían sido preprocesados de antemano de tal manera que fueran todos homogéneos, sin datos perdidos ni incertidumbre.

Por otra parte, el cáncer también se pueden clasificar en función a su estado de desarrollo mediante la clasificación Tumor-Nódulo-Metástasis (TNM) atendiendo a su extensión y capacidad invasiva (Rosen y Sapra, 2023):

1. Tumor primario (T): en este estadio la proliferación celular es local. Es decir, la extensión es dentro de un órgano o tejido concreto pudiendo alcanzar únicamente los órganos contiguos. Existen subclases, separados desde T0, como la ausencia de tumor, hasta una clasificación T3 donde se observa invasión subserosa. Su clasificación se realiza en función al tamaño, extensión y capacidad de invasión del tumor. Cuando no se conoce con exactitud la distribución del tumor, pero se ha comprobado que es a nivel local, se utiliza la nomenclatura Tx.
2. Nodular (N): en este segundo estadio, el tumor ha invadido ganglios linfáticos locales. Los ganglios linfáticos actúan como filtro de la sangre, de forma que las células en división son absorbidas por los mismos. De esta forma la clasificación N

comprende clasificaciones inferiores al igual que T, N0 se corresponde a aquellos tumores que no poseen distribución nodular, hasta N3 cuando más de 7 nódulos están involucrados. Cuando no se conoce el estado de invasión nodular, se emplea la nomenclatura Nx.

3. Metástasis (M): en este último estadio el tumor se extiende más allá del dominio linfático. Se denominan M0 aquellos tumores sin extensión distante y M1 aquellos con distribución sistémica. Para este último tipo de tumores la esperanza de vida es muy baja (Rosen y Sapra, 2023).

Este trabajo se centrará en discernir entre un bulto o mancha que podría ser un tumor, el cual se denominará benigno y aquel que posee distribución local avanzada o mayor, el cual se denominará maligno. Es de especial interés ya que si se diagnostica en un estado poco avanzado, como es la subclasificación T, puede que el tumor sea operable o bien ser tratado mediante uno de los métodos descritos anteriormente. Por el contrario, si el tumor se clasifica como M, las probabilidades de supervivencia del paciente serán bajas. Por eso es importante diagnosticar la naturaleza y extensión del tumor lo más rápido posible y de forma correcta.

1.3. Métodos de diagnóstico de cáncer de mama

El cáncer de mama se diagnostica mediante diferentes métodos partiendo de la aparición de un bulto en la zona del pecho. Aunque la mayoría de los bultos son benignos, puede darse el caso de que empeoren y se extienda. Los métodos principales de diagnóstico de cáncer de mama que se utilizan actualmente son los siguientes:

1. Mamografía: es una técnica ligeramente invasiva que consiste en la toma de radiografías de las mamas y su posterior análisis. Este método ofrece una fiabilidad de entre el 68 y 79% de exactitud. Es una técnica poco invasiva, sin embargo, puede hacer que ciertas personas no se sometan a la misma por no realizar la radiografía, ya que en este caso, puede llegar a ser doloroso o molesto (Wolberg *et al.*, 1995).
2. Aspiración con aguja fina con visualización interpretada: consiste en la extracción de fluidos internos o parte de tejido del bulto mediante una aguja fina, tal y como dice su nombre. Esta técnica posee una alta exactitud, siendo de entre el 65% y 98%, pero con una alta variabilidad. En el mejor de los casos, la exactitud es elevada debido al complemento de la técnica con una red neuronal encargada del diagnóstico en base a los parámetros fisiológicos extraídos. Sin embargo, posee el inconveniente de que se trata de una técnica altamente invasiva (Shafique *et al.*, 2023).
3. Biopsia: esta última técnica consiste en la extracción quirúrgica de un fragmento

interno del bulto presente en la mama para su diagnóstico a través de un microscopio. Aunque su exactitud en algunos casos alcanza el 100%, se trata de la técnica más invasiva de las tres, además de ser costosa y lenta de realizar (Tagliafico *et al.*, 2019).

Estas tres técnicas son las que se utilizan actualmente para el diagnóstico de cáncer de mama. Como se ha descrito anteriormente, la fiabilidad en el diagnóstico es directamente proporcional a la invasividad y dolor asociados a la realización de la técnica.

1.4. Métodos de diagnóstico de cáncer de piel

Por otro lado, el cáncer de piel también posee diferentes métodos de diagnóstico:

1. Diagnóstico *on-line*: este tipo de diagnóstico utiliza una fotografía tomada con un aparato electrónico (teléfono móvil, cámara...) y permite el diagnóstico de cáncer de piel desde casa. No obstante, las aplicaciones que existen hoy en día con este funcionamiento poseen baja exactitud, de modo que al ser la cantidad de falsos diagnósticos muy elevada, se le puede dar una falsa sensación de seguridad al paciente (Davis *et al.*, 2019).
2. Tomografía computarizada: este método se basa en el mismo principio que el anterior, pero en este caso se emplean sistemas más sofisticados de toma de muestra y del análisis de imagen en ámbito clínico. Sin embargo, aunque este método es más fiable que el anterior, su exactitud también es reducida. Se usa sobre todo para determinar qué pacientes deben ser sometidos a biopsia. Para este método, la fiabilidad toma valores del 79% (Bi *et al.*, 2019).
3. Biopsia: existen diferentes formas de aplicarla para este tipo de cáncer, el fundamento es similar al descrito para cáncer de mama. El proceso comienza con la extracción de parte del tejido sospechoso de tumor y el análisis posterior al microscopio. Esta técnica es de fácil aplicabilidad al no requerir anestesia, además de dar buenos resultados. El mayor inconveniente es su elevada invasividad (Kilic *et al.*, 2020).
4. Inmunohistoquímica: consiste en la utilización de biomarcadores para la detección de cáncer de piel. Se suelen utilizar dos tipos de biomarcadores: los melanocíticos y los proliferativos. Los melanocíticos determinan la biosíntesis proteica, mientras que los proliferativos determinan si existe crecimiento celular. A pesar de que posea una elevada efectividad, de que sus reactivos no sean muy caros y de que sea una técnica altamente reproducible, se necesita personal experto tanto como para llevar a cabo la técnica como para concluir sobre los resultados (Davis *et al.*, 2019).

La desventaja principal de los métodos más fiables es su alta invasividad, al igual que en cáncer de mama. Además de que la fiabilidad de los resultados no es tan elevada, salvando el caso de la biopsia que es el método más eficaz, pero también el más invasivo. Es decir, el problema actual del diagnóstico de cáncer no varía según el tipo de cáncer, ni tampoco según el método utilizado para la obtención del diagnóstico.

1.5. ¿Por qué usar inteligencia artificial?

Como se ha descrito en las técnicas anteriores, la fiabilidad es directamente proporcional a la invasividad. Dicho de otra manera, para que el diagnóstico sea preciso y la probabilidad de error sea menor, la técnica a utilizar debe ser altamente invasiva. Por ello, mediante el uso de IA se pretende alcanzar una fiabilidad lo más alta posible minimizando la invasividad de las técnicas. Entonces, la obtención de muestras se podría realizar mediante métodos de imagen como fotografía, radiografía o midiendo parámetros fisiopatológicos como son el tamaño, la forma, el color, etc.

La IA es muy efectiva encontrando patrones dentro de cantidades masivas de datos, por lo que aprendiendo de casos anteriores del mismo tipo de cáncer, se puede realizar el diagnóstico de nuevos pacientes. Estos algoritmos de aprendizaje profundo pueden solucionar eficientemente problemas que de otra manera serían imposibles de solventar dentro del ámbito de la oncología. Estudios anteriores han demostrado los usos potenciales de estos algoritmos para diagnosticar otros tipos de cáncer, como la retinopatía diabética a partir de imágenes tomadas de la retina, o para la detección de drogas en el tracto gastrointestinal (Shimizu y Nakayama, 2020).

2. OBJETIVOS Y PROCEDIMIENTO

En este apartado se expondrán los objetivos a cumplir y se resumirá el procedimiento que se ha realizado durante el presente trabajo.

2.1. Objetivos

Optimizar un modelo de IA supervisado capaz de diagnosticar un tipo específico de cáncer sustituyendo a los métodos invasivos, consiguiendo también una fiabilidad mayor a las técnicas de imagen que se emplean actualmente, además de que sea posible su mantenimiento y escalado a lo largo del tiempo.

Entender el funcionamiento general de las redes perceptrón multicapa y convolucional, ajustando los parámetros y funciones matemáticas involucradas en ello. Así como de

mejorar el conocimiento en python y aprender a manejar los paquetes de este lenguaje relacionados con el aprendizaje profundo y con las operaciones sobre grandes conjuntos de datos.

Mejorar en el diseño de páginas web programando el presente documento utilizando las tecnologías web HTML, CSS y javascript. Descrito con más detalle en **anexo II**.

Integrar los conocimientos adquiridos en diferentes asignaturas del grado respecto al cáncer, así como entender los métodos principales de diagnóstico y tratamiento empleados en la actualidad. Aplicar dichos conocimientos al diagnóstico mediante IA para muestras obtenidas de bases de datos de dominio público. Aprender a elegir el tipo de red que más se ajuste a las muestras (imágenes, parámetros fisiopatológicos, etc.).

Realizar una pequeña aplicación con estructura cliente-servidor capaz de diagnosticar cáncer *online* desde cualquier dispositivo con acceso a internet. Desplegar dicha aplicación en un servidor de dominio público para que cualquier persona sea capaz de enviar imágenes y recibir un diagnóstico instantáneo mediante una simple foto de la piel.

2.2. Procedimiento

Para la elaboración del presente trabajo se realizaron los siguientes pasos:

1. Obtención de un conjunto de datos (dataset) de cáncer de mama: datos de 665 pacientes obtenidos de la página web de la Universidad de California Irvin (*University of California Irvin, UCI*), ya adaptados para ser utilizados en pruebas de *machine learning* (*UCI machine learning repository*, sin fecha).
2. Separación del dataset en entrenamiento (90%) y test (10%).
3. Creación de dos programas, el primero hecho a mano desde cero y el segundo importado de *scikit-learn* (*sklearn*) (Varoquaux *et al.*, 2015). Ambos debían ser capaces de realizar una regresión multilineal sobre el dataset, aplicando previamente un análisis de componentes principales (*Principal Component Analysis, PCA*) mediante funciones importadas también de *sklearn*. El PCA es una técnica matemática que reduce el número de parámetros basándose en la dependencia mutua que pueden existir entre ellos, intentando minimizar las pérdidas de información sobre lo que representan dichos datos.
4. Realización de una curva característica operativa del receptor (*Receiver Operating Characteristic, ROC*) mediante funciones de *sklearn* para cada uno de las pruebas realizadas.
5. Todos los programas se escribieron utilizando el lenguaje de programación *python* en su versión 3.10 (Van Rossum y Drake, 2009). Las operaciones de álgebra

matricial se importaron del paquete *numpy* (Harris *et al.*, 2020). Este paquete contiene funciones relacionadas con las operaciones con matrices multidimensionales y el manejo de cantidades masivas de datos.

6. Repetición del mismo proceso anterior cambiando la regresión hecha desde cero por una importada del módulo *sklearn*.
7. Todas las figuras que representan datos en forma de gráficos se realizaron con el paquete *matplotlib* (Hunter, 2007) para *python*. Este paquete contiene herramientas para la visualización de datos.
8. Búsqueda de un nuevo dataset, en este segundo caso de imágenes. Se encontró un dataset de 3297 imágenes de cáncer de piel en la base de datos *online Kaggle* (Fanconi, 2019). Este conjunto de datos ya venía separado en dos carpetas de entrenamiento y validación.
9. Realización y entrenamiento de una red neuronal del tipo perceptrón multicapa sobre este mismo dataset mediante el paquete *tensorflow* (Blokdyk, 2018). Este paquete está disponible tanto para *python* como para *javascript*, se utilizó la versión para *python* de dicho paquete. Posee las herramientas y funciones necesarias para personalizar y crear modelos de aprendizaje profundo con facilidad.
10. Realización y entrenamiento de una red neuronal convolucional para el diagnóstico de cáncer de piel mediante las imágenes de este último dataset con el paquete *tensorflow*.
11. Adaptación de la red para tomar los datos mediante funciones de *OpenCV* (Bradski y Kaehler, 2008), un paquete de *python* que posee funciones para trabajar con archivos multimedia.
12. Creación de una aplicación propia con estructura cliente-servidor capaz de almacenar y aplicar la red neuronal *OpenCV* para el diagnóstico *online* de pacientes.

3. MODELOS DE INTELIGENCIA ARTIFICIAL

En este apartado se abordará la metodología seguida para el ajuste de modelos de IA para el diagnóstico de cáncer de mama y de piel. Para cada una de las pruebas se realizó una curva ROC, ya que fue necesario analizar estadísticamente cada una para evaluar la fiabilidad del diagnóstico que ofrecía cada modelo. El código general para la obtención y representación gráfica del ROC se puede ver en la **figura 2**. Los parámetros medidos de mayor interés fueron la exactitud (*accuracy*, ACC), la sensibilidad (*True Positive Rate*, TPR) y la especificidad (*specificity*, SPC), los cuales se definen a continuación:

Exactitud: se define como el número de resultados exitosos predichos respecto al número total de resultados. Cuanto mayor sea este valor, mejor habrán sido los resultados de la prueba:

$$ACC = \frac{V}{P+N} = \frac{VP+VN}{P+N}.$$

Sensibilidad: se define como el número de verdaderos positivos (VP) respecto al número total de positivos (P), es decir, el parámetro es comparable a la exactitud pero solo teniendo en cuenta aquellos valores que han sido positivos:

$$TPR = \frac{VP}{P} = \frac{VP}{VP+FN}.$$

Especificidad: se define como el número de verdaderos negativos (VN) respecto al número total de negativos de la prueba (N). Este parámetro es similar a la sensibilidad teniendo en cuenta los negativos en lugar de los positivos:

$$SPC = \frac{VN}{N} = \frac{VN}{VN+FP}.$$

```

from sklearn.metrics import roc_curve
from matplotlib import pyplot as plt
from numpy import linspace

FPR, TPR, thr = roc_curve(test['y'], [round(i[0]) for i in res])
fig, ax = plt.subplots(figsize = (8, 8))

plt.plot(FPR, TPR, linewidth = 2)
plt.plot([0, 1], [0, 1], 'k--')

ax.set_xlabel("Ratio de Falsos Positivos", fontsize = 15)
ax.set_ylabel("Ratio de Verdaderos Postivos", fontsize = 15)
ax.set_xticks(linspace(0, 1, 11))
ax.set_yticks(linspace(0, 1, 11))

plt.show()

```

Figura 2: código para la obtención y representación gráfica del ROC. El proceso y el código fue el mismo para todas las pruebas posteriores, cambiando la variable de entrada.

3.1. Regresión lineal y PCA para cáncer de mama

La primera prueba de diagnóstico de cáncer consistió en la realización de regresión lineal a un conjunto de $s = 513$ muestras teniendo cada una de ellas $n = 30$ atributos como el tamaño, la forma, el color, etc. (*UCI machine learning repository*, sin fecha). Un ejemplo son los parámetros fisiopatológicos de la paciente 10, que poseía los atributos 857810, B, 13'05, 19'31, 82'61, 527'2, 0'0806, 0'03789, 0'000692, 0'004167, 0'1819, 0'05501, 0'404, 1'214, 2'595, 32'96, 0'007491, 0'008593, 0'000692, 0'004167, 0'0219, 0'00299, 14'23, 22'25, 90'24, 624'1, 0'1021, 0'06191, 0'001845, 0'01111, 0'2439, 0'06289. El primer atributo corresponde al número de identificación, el cual no se utilizó para el diagnóstico. Posteriormente, el segundo atributo se corresponde al diagnóstico, siendo B benigno y M maligno. La paciente número 10 recibió un diagnóstico benigno. El resto se corresponden a los parámetros fisiopatológicos de la paciente, los cuales se utilizaron para entrenar la red neuronal. El conjunto de datos se truncó en dos partes: una parte contenía el 90% de los

casos benignos y el 90% de los casos malignos, mientras que el 10% restante de cada diagnóstico se encontraba en otro archivo separado. La separación se realizó al azar y el objetivo consistía en la separación de dos subconjuntos, uno de entrenamiento y otro de validación. Se programó una regresión lineal propia en *python* (Van Rossum y Drake, 2009) sustituyendo en la ecuación matricial $\underline{a} = (B^t \cdot B)^{-1} \cdot B^t \cdot \underline{y}$. Siendo \underline{a} el vector de s coeficientes que queremos obtener para aplicar a posteriori en otros casos, siendo B la matriz $s \times (n + 1)$ de los atributos de cada muestra, junto a una columna a mayores con 1's. Y siendo \underline{y} un vector de los s resultados ya conocidos para el diagnóstico, y utilizados durante el entrenamiento. Se utilizó el paquete *numpy* (Harris *et al.*, 2020) para la realización de las operaciones de álgebra matricial. Un resumen del código se puede ver en la **figura 3**.

```
funciones = [lambda i: 1] + [lambda i, j=val: i[j] for val, _ in enumerate(train['x'][0])]
b = [[j(i) for j in funciones] for i in train['x']]
resultados = matmul(inv(matmul(transpose(b), b)), matmul(transpose(b), transpose(train['y'])))

def g(x):
    res = 0
    for i, j in zip(funciones, resultados):
        res += j * i(x)
    return res
```

Figura 3: código propio de la regresión lineal. La variable "funciones" se corresponde a la lista de funciones que extraen los parámetros de las muestras, la variable "b" es la matriz de aplicación (B), la variable "resultados" es el vector de resultados, la función g(x) se encargará de predecir los nuevos casos tomando las funciones y los parámetros del paciente.

Al ser la regresión una prueba reproducible, no fue necesario realizar un guardado del entrenamiento; ya que cada vez que se ejecute el programa, se obtendrá el mismo resultado.

3.2. Automatización con *sklearn*

La segunda prueba consistió en repetir el mismo proceso descrito anteriormente, pero automatizado utilizando funciones importadas del módulo *sklearn* (Varoquaux *et al.*, 2015). Adicionalmente, se comprobó si se lograba el mismo resultado en la predicción de casos utilizando un método diferente y sobre el mismo dataset.

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(train['x'], train['y'])

predict = model.predict(test['x'])
```

Figura 4: código de la regresión realizada con el paquete *sklearn* sin la realización del PCA. En el código se ve la creación de un modelo de regresión lineal que almacenará el entrenamiento, y que posteriormente se someterá a validación.

Todas las operaciones estadísticas se importaron de este módulo, el cual posee funciones estadísticas del nivel más general de IA, como son la regresión lineal o el PCA. La funcionalidad que aporta este módulo es altamente reproducible, ya que dichas funciones pueden ser utilizadas para la predicción de casos de la misma manera que el modelo

anterior. Un fragmento del código de la regresión sin PCA se puede ver en la **figura 4**.

Tras la realización de esta prueba, se quiso comprobar el efecto de aplicar un PCA sobre el dataset. Se realizaron tantos PCA como número de parámetros poseían las muestras, en este caso reducciones entre 1 y 30, incluídos. El mismo programa de regresión lineal fue usado tras la aplicación del PCA, evaluándose también una curva ROC para concluir si el PCA era efectivo, en cuántos parámetros y si dichos resultados eran más fiables que el experimento control. Un resumen del código de la realización del PCA se puede leer en la **figura 5**.

```
from sklearn.decomposition import PCA
from sklearn.linear_model import LinearRegression

analysis = PCA(15)
reg = LinearRegression()

pca_train = analysis.fit_transform(train['x'])
pca_test = analysis.fit_transform(test['x'])
reg.fit(pca_train, train['y'])

predict = reg.predict(pca_test)
```

Figura 5: código de la regresión con PCA. El procedimiento es el mismo que el anterior, pero realizando el PCA sobre las muestras de entrenamiento.

Debido a la naturaleza reproducible de la regresión lineal, se esperaban resultados iguales o muy similares al modelo anterior. La única ventaja de este método fue la escritura de una menor cantidad de código y de una mejor legibilidad del mismo.

3.3. Aprendizaje profundo de una red perceptrón multicapa

Una vez se comprobó la fiabilidad de la regresión para el diagnóstico de cáncer de mama, se procedió a la creación de un red perceptrón multicapa utilizando el paquete *tensorflow* (Blokdyk, 2018). Se construyó la red partiendo de una estructura de perceptrón multicapa, un ejemplo de este tipo de red está representado en la **figura 7**.

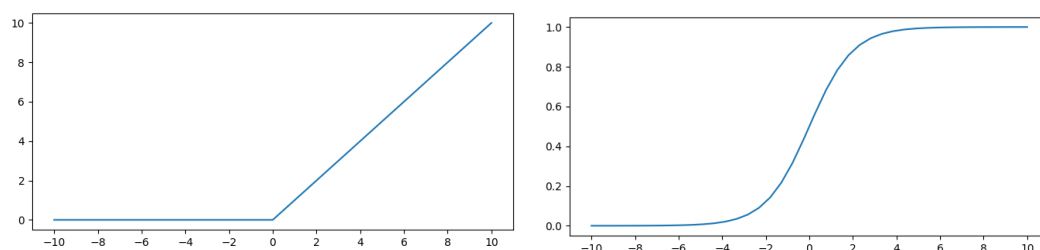


Figura 6: representación gráfica de las funciones ReLU (6a) y sigmoidea (6b). Ambas son funciones unidad.

En este caso se ajustó con los siguientes parámetros: cuatro capas, la primera consistía en una capa de entrada con tantas neuronas como parámetros había por muestra, 30 en este caso. Dado que esta capa se genera de forma automática, no aparece en el código

representado en la **figura 8**. Posteriormente seguían dos capas densas consecutivas de 20 neuronas cada una, a las que se le asignó la función lineal unidad rectificada (*Rectified Linear Unit*, ReLU) como función de activación. Esta función se define como 0 para valores de dominio menores o iguales a 0 y x para los demás, se puede ver su representación gráfica en la **figura 6a**. La función de activación determina cuánto vale la salida que emite la neurona tras haber sometido a la entrada a su función característica y a superar el umbral de excitación. Finalmente la red presentaba una capa densa de salida que consistió en una única neurona que recogía y emitía la salida binaria esperada, la salida consistía en el valor 0 para un tumor maligno y un valor de 1 para un tumor benigno. Aunque la salida real es un número decimal comprendido en dicho rango. La última capa poseía como función de activación la función sigmoidea, la cual se define como $\text{sigm}(x) = \frac{1}{1+e^{-x}}$, y se puede ver gráficamente en la **figura 6b**.

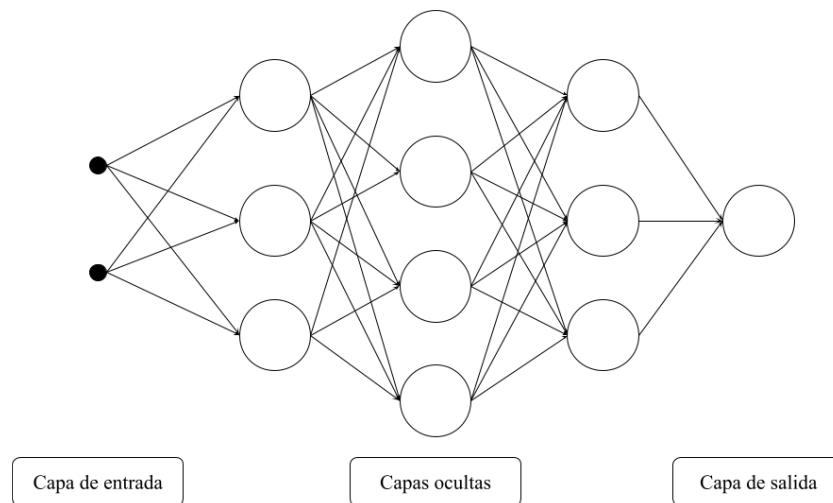


Figura 7: representación genérica de una red perceptrón multicapa con los círculos como neuronas. La capa de entrada debe poseer tantas neuronas como atributos posee cada muestra y la capa de salida tantas neuronas como parámetros esperados de salida (1 en el caso de salida binaria).

La función de *loss* utilizada en este caso fue la de *binary crossentropy*, una función de *loss* se encarga de calcular el error que existe al someter a test la red neuronal frente al conjunto de entrenamiento, y es utilizado para aprender según su valor durante la siguiente iteración. Se utilizó *binary crossentropy* debido a que se deseaba obtener una salida binaria para el diagnóstico. El modelo se entrenó utilizando el optimizador *ADaptive Moment stimation* (ADAM), el término optimizador en el ámbito de las redes neuronales es el algoritmo que sigue la red para entrenar. El entrenamiento constó de 1000 pasos distribuidos en 20 etapas cada uno. El término "etapa" se refiere a cada uno de los procedimientos que realiza la red sobre los datos; mientras que el término "paso" se refiere a una iteración del conjunto de etapas que se realizan sobre los datos, por lo que un entrenamiento de 20 pasos con 20 etapas realizaría 400 etapas donde cada 20 pasos se repite el mismo procedimiento desde el inicio pero con la experiencia de los casos anteriores. El código de esta red neuronal se

puede observar en la **figura 8**.

```
import tensorflow as tf
model = tf.keras.Sequential([
    tf.keras.layers.Dense(20, activation='relu', input_dim=len(train['x'][0])),
    tf.keras.layers.Dense(20, activation="relu"),
    tf.keras.layers.Dense(1, activation="sigmoid")])
```

Figura 8: código del perceptrón multicapa. Se pueden observar las tres capas densas, la primera posee el input esperado de entrada con tantos valores como cantidad de parámetros posee cada muestra de entrenamiento. Se comprobó previamente que todas las muestras poseían el mismo número de parámetros y no hubiera valores perdidos.

Tras el entrenamiento, la red neuronal se sometió a una evaluación de resultados a partir del 10% de datos restante que no fueron utilizados previamente para entrenar. Con el objetivo de optimizar el diagnóstico, se ajustó la cantidad de neuronas y el tiempo de entrenamiento. Se llegó a un óptimo local cuando se asignaron 20 neuronas a cada una de las dos capas ocultas intermedias, junto con un aumento del tiempo de entrenamiento a 2000 pasos con 30 etapas.

3.4. Red neuronal convolucional para imágenes de cáncer de piel

La siguiente prueba se realizó sobre el dataset de imágenes de cáncer de piel (Fanconi, 2019). Para ello se creó un nuevo modelo de red neuronal, un modelo convolucional (*Convolutional Neural Network*, CNN) capaz de leer, interpretar y reconocer imágenes para discernir entre aquellos tumores benignos o malignos. Dos ejemplos de este dataset se pueden observar en las **figuras 9a y 9b**.

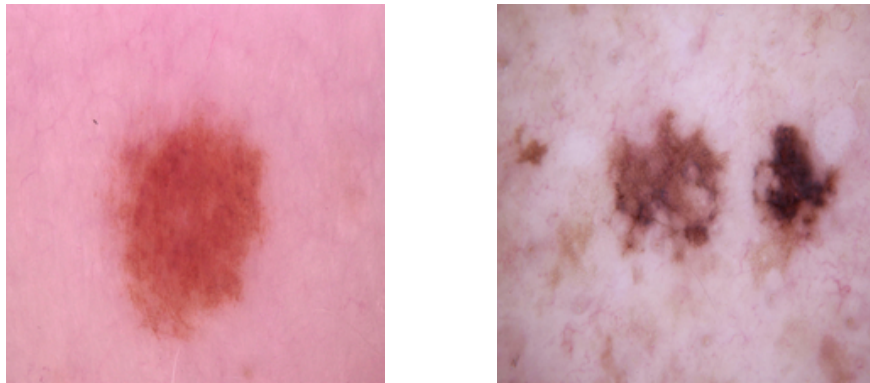


Figura 9: imágenes de cáncer de piel del conjunto de entrenamiento. Muestra 258 benigna (9a) y muestra 1305 maligna (9b) (Fanconi, 2019).

Para lograr dicho objetivo, la capa de entrada de la red debía ser capaz de leer las imágenes como números para así poder entrenar. Se puede observar la estructura general de una red neuronal convolucional en la **figura 10**. La estructura de la red en este caso fue de tres parejas de capas convolucionales con una capa *maxpooling2d* de 32 grupos de neuronas cada una. La capa convolucional se encarga de leer cada imagen y realizar un resumen de la misma alterando sus dimensiones, mientras que la capa *maxpooling2d* reduce a la mitad los

valores de la imagen que emitió la capa convolucional anterior, descartando los valores no informativos. Cada pareja de capas posee tantas neuronas como atributos posee la imagen recibida de la capa anterior 32 veces. Extendiendo la definición de atributo a imágenes desde el dataset de cáncer de mama, una imagen poseería $224 \times 224 \times 3$ atributos, correspondiéndose a las dimensiones de cada imagen, 224×224 píxeles y cada píxel posee 3 valores de rojo, verde y azul (*red, green and blue*, RGB) como atributos. Entonces, una capa convolucional de esta red posee 4816896 ($224 \times 224 \times 3 \times 32$) neuronas. Las capas convolucionales se construyeron con un tamaño de kernel de (3, 3); el kernel es la matriz de píxeles que entran a la vez a la capa convolucional para realizar el resumen de la imagen. Los píxeles entrarían en matrices de 3×3 cada vez. El salto del kernel fue de 1, por lo que se recorrió toda la imagen píxel por píxel, arrastrando los contiguos.

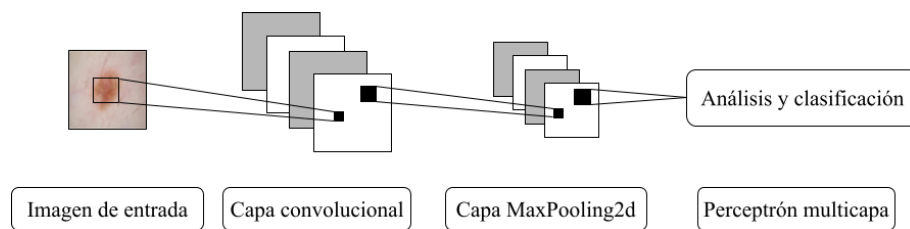


Figura 10: Esquema general de una red neuronal convolucional de una sola capa. Se observa que la capa convolucional siempre se acompaña por una capa MaxPooling. La imagen de esta figura es la imagen 98 del dataset de entrenamiento.

Posteriormente se construyó una capa *flatten*, la cual se encarga de convertir la imagen de 3 dimensiones en una lista de valores de una sola dimensión alineando los números. Finalmente, se construyeron dos capas densas, la primera con 128 neuronas con función de activación ReLU (véase **figura 6a**) para captar y procesar la imagen modificada que emitió la capa *flatten*. La capa final poseía una única neurona que procesaba la salida y emitía el valor binario esperado, 0 para los tumores maligno y 1 para los benignos. Se le asignó la función de activación sigmoidea (véase **figura 6b**) a esta última capa. La distribución de capas de la red con sus respectivos parámetros se puede apreciar en la **figura 11**.

```
import tensorflow as tf
model = tf.keras.Sequential([
    tf.keras.layers.Rescaling(1/255),
    tf.keras.layers.Conv2D(32, (3, 3), 1, activation="relu", input_shape=(224, 224, 3)),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(32, (3, 3), 1, activation="relu"),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(32, (3, 3), 1, activation="relu"),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation="relu"),
    tf.keras.layers.Dense(1, activation="sigmoid")
])
```

Figura 11: código de la red neuronal convolucional. Se muestra cada una de las capas secuenciales con el número de neuronas de cada una. (3, 3) se corresponde al tamaño de kernel y el 1 siguiente al paso. A esta red se le añadió una capa extra inicial de reescalado.

Esta red se creó con el mismo paquete que la de cáncer de mama, mediante el uso de

tensorflow (Blokdyk, 2018). Las imágenes de entrada poseían una resolución de 224×224 píxeles y se leían consecutivamente mediante un generador, una función del paquete que extraía las imágenes de las carpetas. Los valores de las imágenes debían ser reescalados al intervalo [0-1] para su mayor facilidad de comparación por parte de la red, ya que los valores enteros RGB del intervalo [0-255] poseen una mayor amplitud. Para ello se dividieron entre 255, normalizando así las muestras al intervalo deseado. Como optimizador de esta red se volvió a utilizar ADAM, y como función de *loss* también se volvió a utilizar *binary crossentropy*, por el mismo objetivo que la red anterior.

3.5. Método alternativo de lectura de imágenes mediante *OpenCV*

Finalmente, se encontró y probó un método alternativo de lectura de imágenes para complementar las carencias de la red anterior. El método anterior leía las imágenes mediante un generador que atravesaba los directorios recursivamente y clasificaba las imágenes según la carpeta en la que estaban, las iba introduciendo en la red de una en una. Debido a la naturaleza del paquete, no se sabe con exactitud de qué manera se leían las imágenes. En consecuencia, se cambió el generador por la función *imread* presente en la librería *OpenCV* (Bradski y Kaehler, 2008), paquete disponible para una multitud de lenguajes de programación; en este caso se utilizó la versión para *python*. Esta librería posee funciones para manipular archivos multimedia como fotos y vídeos.

```
from cv2 import imread, normalize, NORM_MINMAX
from os import scandir
from numpy import array
def read_from_dir(dir: str) -> tuple:
    imgs, outp = [], []

    print(f"Leyendo imágenes benignas . . .")
    for i in scandir(f"{dir}/benign"):
        imgs.append(normalize(imread(i.path).astype('float'), None, 0, 1.0, NORM_MINMAX))
        outp.append(0)
    print(f"Leyendo imágenes malignas . . .")
    for i in scandir(f"{dir}/malignant"):
        imgs.append(normalize(imread(i.path).astype('float'), None, 0, 1.0, NORM_MINMAX))
        outp.append(1)
    return array(imgs), array(outp)
```

Figura 12: función encargada de leer todas las imágenes benignas y malignas del directorio especificado. Las imágenes son leídas como un array de numpy de tres dimensiones mediante *OpenCV* y reescaladas al intervalo normalizado. Mediante este método se introducen todas las imágenes a la vez en la memoria del ordenador para el entrenamiento. La función devuelve por una parte un array de numpy con las imágenes, y como segunda respuesta, sus diagnósticos en caso de que los haya.

El código de la red es el mismo que aparece en la **figura 11** sin la capa de reescalado, ya que se implementó manualmente. La función creada se muestra en **figura 12** y se basó en la función *imread* de este paquete, que convierte una imagen a una matriz tridimensional de números. Es decir, la función cargaba en la memoria del ordenador (*Random Access Memory*, RAM) la imagen representada como una lista tridimensional de los valores que se

correspondían a filas×columnas×valores RGB (rojo, verde y azul). Todos estos valores RGB eran un número entre 0 y 255. Cada imagen individual se acomodaba a una nueva matriz generando una matriz final de cuatro dimensiones. Una vez leídas todas las imágenes, se introducía la matriz resultante como conjunto de entrenamiento en la red. La principal ventaja de esta red fue que se podían introducir imágenes de forma individual para el diagnóstico, independientemente de la estructura de directorios. En este segundo caso, se necesitaba que la imagen tuviera las mismas dimensiones que las imágenes de entrenamiento (224×224) e introducirle a la red la ruta de acceso de la imagen para que fuera capaz de encontrarla, leerla y procesarla. La facilidad de diagnóstico y la mayor flexibilidad del método hacen que esta red sea más apta para su escalado y utilización en un entorno menos controlado.

Todos los parámetros de la red neuronal de esta prueba fueron los mismos que en el caso anterior, salvo la capa de reescalado, la cual había sido eliminada. Esta capa no fue necesaria ya que las imágenes entraban reescaladas con anterioridad a la red gracias a la función *normalize* de *OpenCV*. Las funciones de optimización y *loss* se mantuvieron invariables respecto a la red anterior.

Finalmente, se desarrolló una aplicación web con estructura cliente-servidor donde la máquina servidor almacenaba y aplicaba la red neuronal de esta prueba. La aplicación conectaba la red neuronal *OpenCV* con el usuario, consiguiendo diagnosticar casos a partir de imágenes enviadas por el usuario a través de internet mediante el protocolo de transferencia de hipertexto (*hypertext transfer protocol*, http) mediante cualquier navegador de internet. Por otro lado, la máquina cliente, quién enviaba la imagen de cáncer y recibía el diagnóstico utilizando una interfaz de usuario de fácil uso. El *backend* o la parte de la aplicación que se ejecuta en el servidor, fue programada en *python* utilizando el módulo *flask* (Grinberg, 2018); el cual posee las herramientas necesarias para crear un servidor web y las utilidades básicas para responder peticiones http. Por otra parte, el *frontend* o la parte de la aplicación que se ejecuta en la máquina cliente, fue programado utilizando las tecnologías web HTML, CSS y javascript (para una descripción de estos lenguajes ver **anexo II**).

4. RESULTADOS

En esta sección se describirán los resultados que se han obtenido para cada prueba respecto a sus conjuntos de validación, así como de sus respectivos tests estadísticos que determinan su fiabilidad.

4.1. Observaciones generales

Tabla 1: tabla resumen de los resultados para cada prueba en este experimento. Las siglas VP se refieren a los verdaderos positivos, FP a los falsos positivos, VN a los verdaderos negativos y FN a los falsos negativos.

Test	Dataset	VP	FP	VN	FN	Exactitud	Sensibilidad	Especificidad
Regresión	Mama	20	0	35	1	95'24%	98'21%	100%
Perceptrón	Mama	34	1	20	1	96'43%	97'14%	95'24%
1° Keras CNN	Piel	106	85	215	254	48'64%	55'50%	45'84%
2° Keras CNN	Piel	285	47	253	75	81'52%	85'84%	77'13%
Opencv CNN	Piel	223	21	279	137	76'06%	91'39%	67'07%

Para las pruebas realizadas sobre el conjunto de datos de diagnóstico de cáncer de mama se disponía de un menor número de muestras (569), de las cuales 56 fueron utilizadas para el test (10%). Según los resultados presentes en **tabla 1**, las pruebas realizadas sobre el conjunto de datos de cáncer de mama presentan unos buenos resultados. La disposición de un menor número de parámetros (30) para el diagnóstico hace que las muestras sean estadísticamente más parecidas entre sí, además de facilitar el procesamiento de los datos a la red. Sin embargo, que hayan dado resultados tan buenos para este conjunto de datos no significa que dicha exactitud vaya a ser la misma en la vida real, ya que en cada persona el cáncer se manifiesta de una forma diferente.

Las pruebas siguientes se realizaron sobre el conjunto de datos de cáncer de piel, el cual poseía 3297 imágenes distribuidas en 660 para validación y las 2637 imágenes restantes para entrenamiento. Al tratarse de imágenes, el procesado era más complejo; fue necesario introducir las imágenes en la red con unas dimensiones de 224×224 y reescaladas.

4.2. Regresión lineal con y sin PCA para cáncer de mama

La regresión lineal fue la primera prueba de IA realizada sobre el conjunto de datos de cáncer de mama. Los resultados obtenidos para cada PCA se sometieron a una curva ROC para comparar la fiabilidad de los resultados en función del número de parámetros reducidos, ya que el algoritmo requería introducción manual del número de parámetros a reducir, por lo que se realizaron tantas pruebas como parámetros, junto una prueba control sin PCA, a partir de ahora prueba NP (31 pruebas en total). El procesamiento del PCA fue muy ligero en recursos, ya que el tiempo de entrenamiento no llegó a 1 segundo y la cantidad de RAM utilizada rondaba los 60 MB; lo que otorga una alta reproducibilidad a la prueba y se esperen siempre los mismos resultados independientemente de la máquina que lo ejecute. Los parámetros principales medidos de todas las pruebas se observan en la **figura 13**.

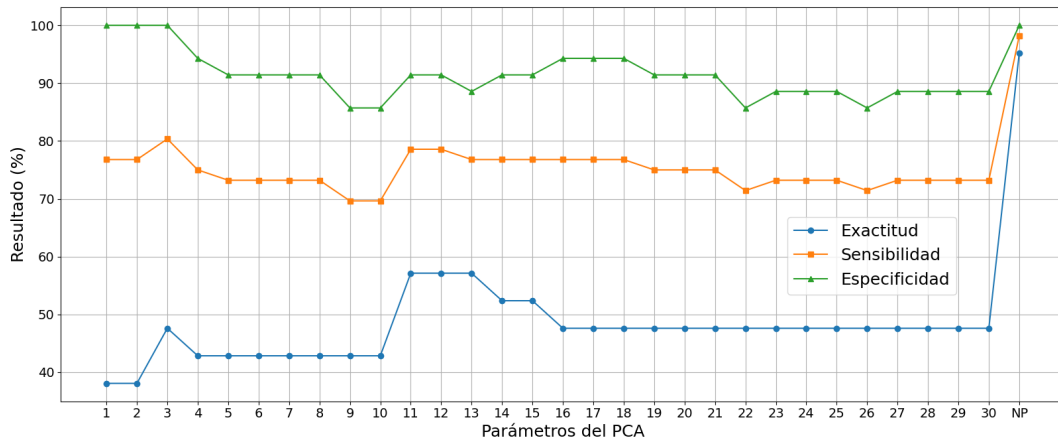


Figura 13: gráfico de los resultados de la curva ROC para las pruebas de PCA. El eje x indica el test PCA realizado para cada número entre 1 y el número máximo de parámetros, además de la prueba NP, al cual no se le aplicó PCA. Se observan 3 líneas: exactitud como círculos, sensibilidad como cuadrados y exactitud como triángulos.

La realización del PCA fue contraproducente de acuerdo a los resultados obtenidos, debido a que en ningún caso se superaron los valores de la prueba control. Se pueden observar los resultados completos de todos los PCA realizados en el **anexo I**. Todos los valores se mantienen aproximadamente estables sin importar el número de parámetros que se hayan reducido para el PCA. La mayor diferencia se observa en la prueba NP, por ser aquella que ha obtenido los mejores resultados. Se observa que para dicha prueba los valores de exactitud y sensibilidad son los mayores de todos, además de la especificidad.

4.3. Perceptrón multicapa para cáncer de mama

Aunque los resultados de la regresión lineal fueron más que aceptables, se probó un modelo de aprendizaje profundo para conseguir la misma exactitud en el diagnóstico del modelo anterior o intentar mejorarla. La razón de esto fue la escalabilidad de vista al futuro, puesto que una red neuronal sería teóricamente capaz de ofrecer resultados mejores a mayor número de muestras se utilicen para entrenar. Es decir, la red neuronal aprende a una tasa mayor que la regresión lineal para altas cantidades de muestras de entrenamiento. Como se esperaba, se obtuvieron peores predicciones para este modelo respecto al modelo de regresión lineal con el número actual de muestras.

El entrenamiento de este modelo fue más pesado que el anterior, ya que tardaba aproximadamente 2 s por etapa y el consumo de RAM fue de 240 MB. En este caso, al tratarse de un modelo de aprendizaje profundo, el consumo del procesador (*Central Processor Unit*, CPU) fue del 100%, ya que se quiso entrenar con la mayor capacidad de recursos del sistema posibles, al contrario que con el método anterior, en el que el modelo utilizaba un único núcleo de la CPU.

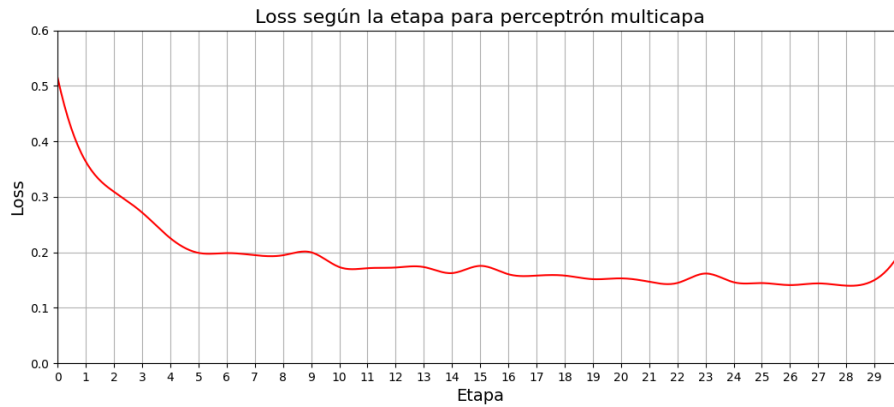


figura 14: gráfica del loss respecto a la etapa para la red neuronal perceptrón de cáncer de mama.

Los resultados del entrenamiento fueron bastante buenos, observándose el descenso del *loss* en la **figura 14**. Según dichos resultados, se puede ver un decrecimiento exponencial del *loss* hasta aproximadamente la 6ª etapa, donde se estabiliza en 0'2 durante varias etapas y decrece lentamente hasta alcanzar un valor final cercano a 0'1. Dado que el *loss* es un parámetro que resume el error de la red, se requiere que el valor sea lo más pequeño posible, por lo que un resultado de *loss* de 0'1 se considera un buen resultado. Comparando los resultados presentados en la **figura 15** y la prueba NP descrita en la **figura 12**, los resultados de esta prueba fueron muy similares al modelo anterior de regresión lineal para este número de muestras.

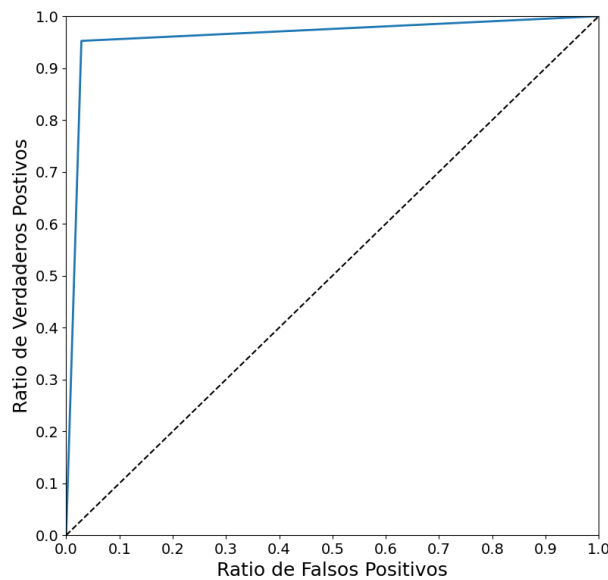


Figura 15: curva ROC para la red neuronal perceptrón sobre el dataset de cáncer de mama.

Sin embargo, según se puede observar en la **figura 15** este modelo posee mayor capacidad de aprendizaje que el anterior, convirtiéndose en una buena opción para diagnosticar cáncer de mama mediante este tipo de variables. Cabe destacar que para obtener la mayor fiabilidad posible en el diagnóstico, han de introducirse los datos en la red siguiendo la misma estructura que el conjunto de entrenamiento. Es decir, los datos para el diagnóstico

deben ser los mismos 30 parámetros fisiopatológicos y en el mismo orden, ya que sino la red tomará mal los datos y ofrecerá un diagnóstico completamente erróneo.

4.4. Red convolucional para cáncer de piel

Después de obtener buenos resultados en el modelo anterior, se probó la realización de una red neuronal convolucional para el nuevo dataset sobre carcinoma. Una red convolucional es un modelo de aprendizaje profundo orientado a trabajar con imágenes, ya que reduce rápidamente la dimensión de los datos.



Figura 16: gráfica del loss respecto a la etapa de entrenamiento para la red neuronal convolucional por el método de generador de imágenes. Esta gráfica de loss se corresponde con el ROC de la **figura 18**.

Como se esperaba, el entrenamiento de esta red consumió más recursos que las anteriores; concretamente, 2 GB de memoria RAM durante el entrenamiento, un consumo de CPU del 100% y un tiempo de entrenamiento de 70 segundos por etapa. Según los resultados presentados en **figura 16**, el *loss* descendía de forma homogénea, perdiendo tendencia en la última etapa de una forma similar al modelo anterior.

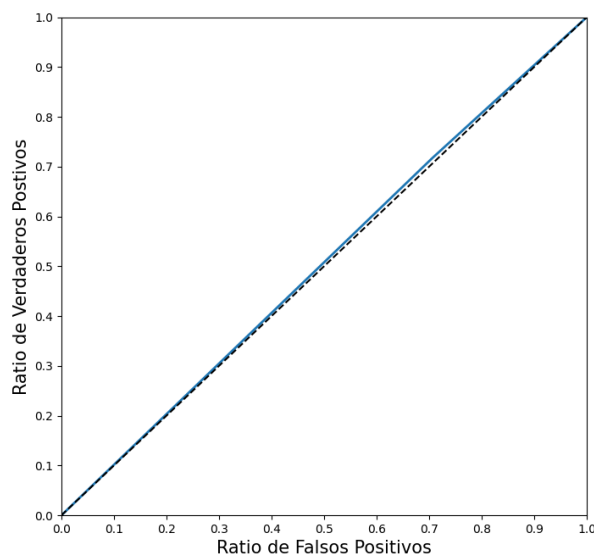


Figura 17: curva ROC de los resultados de diagnóstico de carcinoma por el método del generador de imágenes. Esta curva ROC fue extraída del primer entrenamiento de esta red.

El *loss* final fue de 0'117 aproximadamente, un valor bastante bueno dada la complejidad del dataset. Podría disminuir en mayor medida si el entrenamiento constase de más etapas, entonces podría mejorarse la fiabilidad del diagnóstico. No obstante, aumentarían los costes asociados a los recursos de computación. Y en principio, no se permite realizar diagnósticos mientras el entrenamiento está en curso.

Como se puede observar en la **figura 17**, la curva ROC posee una tendencia de función lineal. Esto significa que existen tantos resultados verdaderos como falsos, pudiéndose considerar el diagnóstico como aleatorio. Por este motivo, la red se sometió a un segundo entrenamiento desde cero, eliminando el modelo guardado entrenado para reiniciarse.

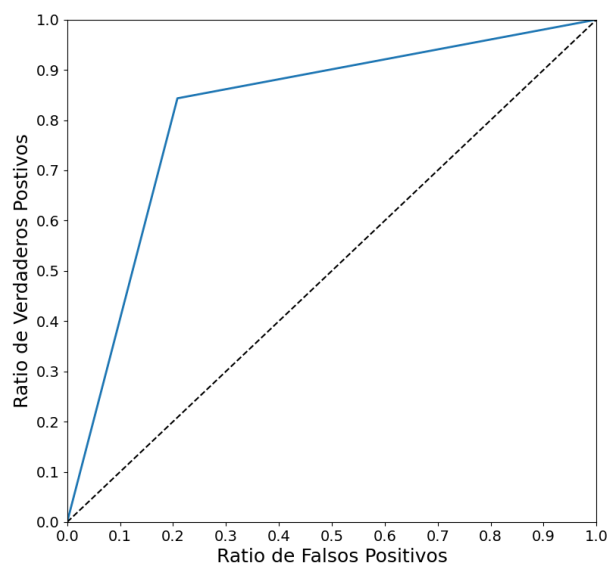


Figura 18: segunda curva ROC para la red neuronal anterior. Esta segunda curva ROC procede del re-entrenamiento de la misma red con las mismas muestras para optimizar el resultado.

Como se puede ver en la **figura 18**, el ROC ha mejorado considerablemente, ya que ha pasado de ofrecer un diagnóstico aparentemente aleatorio a obtener una precisión del 77'15%. Esto demuestra la baja reproducibilidad del método, puesto que se puede observar que repitiendo el mismo experimento bajo las mismas condiciones han salido resultados completamente diferentes. Esto quiere decir que la misma red neuronal entrenada en la misma máquina y con las mismas muestras en dos momentos diferentes pueden entrenarse de manera diferente y dar resultados diferentes. Es debido que el modelo inicializa de manera aleatoria, contribuyendo a la variabilidad del método. Esta variabilidad puede provocar que el entrenamiento sea más efectivo, o por el contrario, que se obtengan unos resultados incoherentes. En el caso de que fuese un método estandarizado con inicialización fija, daría siempre los mismos resultados. De todos modos, una vez se obtenga una red bien entrenada, se puede guardar para que la fiabilidad de los resultados se mantenga, descartando las redes cuyo entrenamiento fue infructuoso.

4.5. Método alternativo de lectura de imágenes mediante *OpenCV*

Finalmente, el objetivo de esta última red era el de suplir la baja eficiencia y alta variabilidad de la red anterior, así como de facilitar la manipulación de muestras individuales.

El entrenamiento de este modelo fue el más pesado de todos, ya que debían cargarse todas las imágenes al mismo tiempo en la RAM, por lo que el consumo de recursos fue el mayor de todos los modelos, así como de necesitar más tiempo para acceder a ellos. Por otro lado, el tiempo de entrenamiento de cada etapa fue aproximadamente de 3 minutos, el consumo de RAM superó los 8'5 GB en todo momento y el consumo de CPU se mantuvo en el 100% durante todo el entrenamiento.

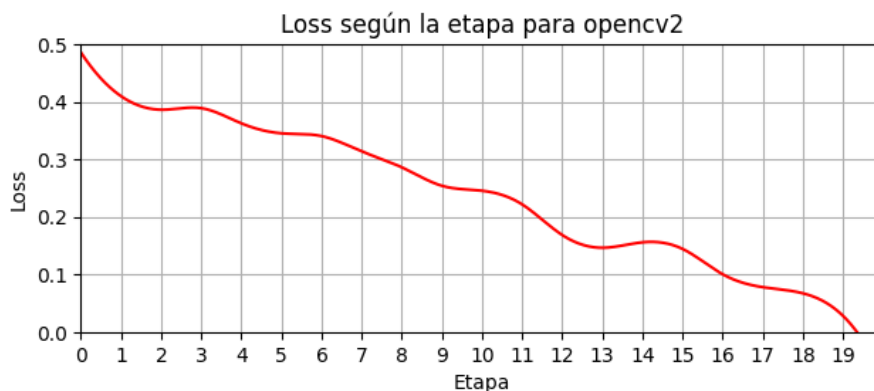


Figura 19: gráfico del *loss* respecto a cada etapa para la red neuronal convolucional que utiliza el método de *OpenCV*.

Atendiendo a los resultados presentes en la **figura 19**, el entrenamiento de la red logró hacer que el *loss* disminuyera siguiendo aproximadamente un modelo lineal. Donde, en la 19ª etapa el *loss* alcanzó el valor de 0'029 (un valor cercano a cero); convirtiendo a esta red neuronal en la red con menor *loss* tras el entrenamiento de todo este trabajo. Posteriormente, la red se sometió a los tests estadísticos para comprobar si el entrenamiento realmente se ajusta a la vida real.

Según los resultados del ROC presentados en la **figura 20**, la curva ROC se asemeja a las redes neuronales anteriores. Sin embargo, se puede observar que la curva se acerca más al borde superior que al izquierdo, esto quiere decir que la sensibilidad es superior a la especificidad. Aunque la exactitud sea menor, esto hace a esta red perfecta para su uso en un entorno real debido a su alta sensibilidad. El diagnóstico de los tumores benignos es muy fiable, por lo que si la red concluye que es benigno, el resultado será estadísticamente fiable siendo la probabilidad de ser un falso positivo muy baja. No obstante, el caso para los tumores malignos es diferente, ya que al poseer una exactitud menor, un diagnóstico maligno sería menos fiable y por ende, se recomendaría acudir a un profesional que verifique el caso en particular.

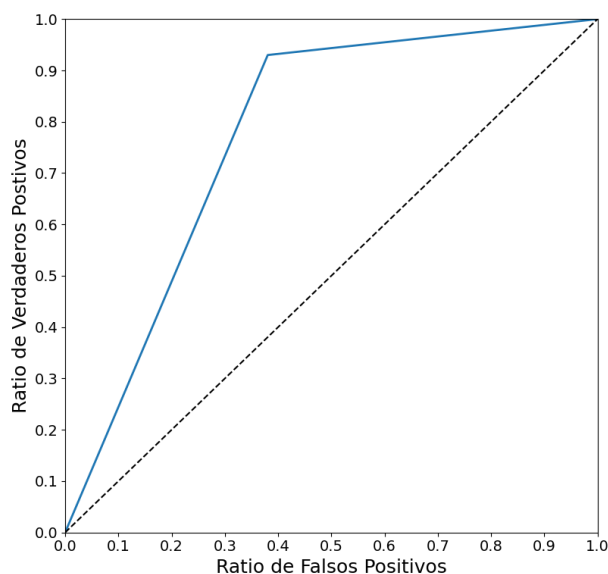


Figura 20: curva ROC para la red neuronal convolucional que utiliza el método de lectura de imágenes OpenCV.

Finalmente, se realizó la prueba de la aplicación web en un entorno de desarrollo local, es decir, de forma que solo se podía acceder al programa desde la misma máquina que lo ejecuta. Para la utilización de este programa, se acopló la red *OpenCV*, la cual poseía una alta sensibilidad. Los resultados del programa fueron muy prometedores, quedando solamente su despliegue en un servidor público para que cualquier persona tuviera acceso a él.

5. CONCLUSIONES

Los objetivos del presente trabajo se han cumplido en su mayoría, descartando el despliegue de la aplicación en un servidor de dominio público. Sin embargo, sí se ha logrado su funcionamiento en dominio local. Asimismo, se ha mejorado en gran medida el manejo en python, sus paquetes específicos y en las tecnologías web durante el desarrollo del presente trabajo.

Como se ha demostrado a partir de los experimentos anteriores, el uso de IA para el diagnóstico de cáncer de mama a partir de parámetros fisiopatológicos y para el diagnóstico de cáncer de piel a partir de imágenes ha sido altamente satisfactorio. La exactitud de estos experimentos se puede considerar aceptable al ser cercana a la de los métodos invasivos (ver apartados 1.3 y 1.4), exceptuando el primer entrenamiento por el método del generador (véase apartado 4.4 y tabla 1).

Cada modelo de IA que existe es útil para una tarea concreta. Conocer qué modelo utilizar en cada caso es primordial, por ejemplo el perceptrón multicapa se especializa en trabajar con números y el modelo convolucional con imágenes.

Los modelos han sido diseñados permitiendo el escalado. Es decir, han sido diseñados de tal forma que se pueda mejorar la exactitud aumentando el número de muestras y el tiempo de entrenamiento sobre lo que ya ha entrenado. Esto podría complementar al entrenamiento mejorando la predicción de nuevos casos. El tiempo de entrenamiento puede aumentar tanto como desee el usuario. Sin embargo, la mejoría es mayor al aumentar la cantidad y variabilidad de muestras, dado que la red explora y aprende de más opciones diferentes. Dichas muestras deben poseer una calidad aceptable y estar acompañadas de un diagnóstico preciso para que no influya negativamente en el aprendizaje.

También se ha observado que ningún parámetro influye más que otro en el diagnóstico de cáncer mediante aprendizaje profundo, pues todos influyen en igual medida. Esto puede ser debido en parte a la heterogeneidad de manifestación de la patología. Entre más información se tenga del tipo de tumor concreto, más fiable será el diagnóstico. Por eso mismo, un PCA no será favorable en la realización de este tipo de pruebas.

Las bases de datos de internet sobre cáncer son un buen punto de inicio para construir y entrenar IA, ya que todos los datos son homogéneos, están supervisados y no existen datos perdidos. Sin embargo, en su mayoría poseen un número muy bajo de muestras, lo que hace que no se puedan utilizar de forma directa en una escala grande.

6. APLICACIONES FUTURAS

Respecto al futuro, para utilizar estas redes en diagnóstico de personas reales deberían mantenerse entrenando continuamente con datos nuevos, para así ser capaces de mejorar y optimizar el diagnóstico. Un modo de hacer esto podría ser el de mantener dos programas ejecutándose a la vez. Uno de los programas sería el que entrena la red, estaría encargado de mejorar la fiabilidad del diagnóstico y de actualizar el modelo guardado cada cierto tiempo; dado que una red neuronal entrenada se trata de un archivo en el ordenador que se puede guardar y cargar de nuevo. El otro programa se encargaría de aplicar la red guardada para diagnosticar un paciente cuando se requiera. Ambos programas podrían comunicarse mediante señales del sistema, o incluso compartir una región de memoria para utilizar siempre la última versión de la red para el diagnóstico.

La red de elección podría utilizarse en diferentes sitios, como por ejemplo para el diagnóstico de tumores en hospitales y centros de salud municipales. Esto permitiría reducir la lista de espera realizando un cribado de aquellos pacientes con tumores benignos. Respecto a este trabajo, se utilizaría la red *OpenCV* para diagnosticar de forma rápida y precisa, dado que sus pruebas estadísticas han demostrado su elevada sensibilidad. El cribado se realizaría respecto a los tumores benignos, mientras que aquellos que han salido malignos serían introducidos a la lista de espera para ser atendidos por el profesional. Esto

podría reducir también el número de pacientes que recibe el profesional, haciendo que dicho profesional se enfoque en los casos malignos o dudosos para ayudar en una mayor medida al paciente.

Otra aplicación interesante a futuro sería la de liberar la red neuronal en internet en un servidor de dominio público para que todo el mundo fuera capaz de diagnosticar cáncer mediante una simple fotografía tomada con cualquier dispositivo electrónico. La fotografía debería ser preprocesada al tamaño de entrada de la red y ser verificada para que la imagen que se envía realmente se trate de una foto sospechosa de cáncer; pudiendo ser ambos pasos automatizados con otra red neuronal complementaria. Además, esta aplicación podría conectarse al sistema de sanidad público para que en caso de malignidad se enviara automáticamente el diagnóstico y la imagen al médico de cabecera del paciente. Esto podría mejorar la calidad de vida de las personas al no necesitar desplazamiento para comprobar el diagnóstico del tumor. Habría que tener siempre en cuenta que la fiabilidad de la red es mucho mayor para resultados benignos.

En el caso de que se desplegara la aplicación en internet y escalara de tal forma que no se pudiera mantener en un servidor normal debido a la falta de recursos, sería necesario migrar al uso de un superordenador como servidor. Una ventaja del uso de un superordenador es que al ser tan elevada la capacidad de procesamiento y RAM disponible, permitiría el diagnóstico simultáneo de un mayor número de personas sin tener que esperar a que el servidor responda.

En base a este tipo de IA para análisis de datos, sería interesante la creación y entrenamiento de más redes neuronales de estos tipos; o bien podría realizarse una prueba que juntase diagnóstico de valores con imágenes y otras posibles entradas. También se podría crear y entrenar redes neuronales para cada tipo de cáncer de los cuales se permita obtener un número de muestras consistente. Se realizaría con el objetivo de diagnosticar lo antes posible y de forma altamente fiable para poder ayudar a que los pacientes obtengan un tratamiento en caso de malignidad. Aún así, esta idea no se podría aplicar a corto plazo para tipos de cáncer poco comunes debido al reducido número de muestras, lo que limitaría el entrenamiento de la red y podría aumentar el riesgo de falsos diagnósticos.

7. BIBLIOGRAFÍA

Bi, W. L., Hosny A., Schabath, M. B., Giger, M. L. *et al.* (2019) “Artificial intelligence in cancer imaging: Clinical challenges and applications”, *CA: a cancer journal for clinicians*, 69(2), pp. 127–157. doi: [10.3322/caac.21552](https://doi.org/10.3322/caac.21552).

Blokdyk, G. (2018) *Tensorflow: A Complete Guide*. 5starcooks. Disponible en: <https://www.tensorflow.org/>. (Accedido: 16 de diciembre de 2022)

Bradski, G. y Kaehler, A. (2008) *Learning OpenCV: Computer vision with the OpenCV library*.

- Cancer today (sin fecha) *Iarc.fr*. Disponible en: <https://gco.iarc.fr/today/home> (Accedido: 29 de mayo de 2023).
- Chang, C. H., Lin, C.-H. y Lane, H.-Y. (2021) “Machine learning and novel biomarkers for the diagnosis of Alzheimer’s disease”, *International journal of molecular sciences*, 22(5), pp. 2761. doi: [10.3390/ijms22052761](https://doi.org/10.3390/ijms22052761).
- Davis, L. E., Shalin, S. C. y Tackett, A. J. (2019) “Current state of melanoma diagnosis and treatment”, *Cancer biology & therapy*, 20(11), pp. 1366–1379. doi: [10.1080/15384047.2019.1640032](https://doi.org/10.1080/15384047.2019.1640032).
- Fanconi, C. (2019) “Skin cancer: Malignant vs. Benign”. Disponible en: <https://www.kaggle.com/datasets/fanconic/skin-cancer-malignant-vs-benign> (Accedido: 12 de diciembre de 2022)
- Greaves, M. (2015) “Evolutionary determinants of cancer”, *Cancer discovery*, 5(8), pp. 806–820. doi: [10.1158/2159-8290.cd-15-0439](https://doi.org/10.1158/2159-8290.cd-15-0439)
- Grinberg, M. (2018) *Flask web development: Developing web applications with python*. 2a ed. O’Reilly Media.
- Gubatan, J., Levitte, S., Patel, A., Balabanis, T. *et al.* (2021) “Artificial intelligence applications in inflammatory bowel disease: Emerging technologies and future directions”, *World journal of gastroenterology*, 27(17), pp. 1920–1935. doi: [10.3748/wjg.v27.i17.1920](https://doi.org/10.3748/wjg.v27.i17.1920).
- Harris, C. R., Millman, K. J., Van der Walt, S. J., Gommers, R. *et al.* (2020) “Array programming with NumPy”, *Nature*, 585(7825), pp. 357–362. doi: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2).
- Hunter, J. D. (2007) “Matplotlib: A 2D Graphics Environment”, *Computing in science & engineering*, 9(3), pp. 90–95. doi: [10.1109/mcse.2007.55](https://doi.org/10.1109/mcse.2007.55).
- Kaufman, C. S. (2019) “Increasing role of oncoplastic surgery for breast cancer”, *Current oncology reports*, 21(12), pp. 111. doi: [10.1007/s11912-019-0860-9](https://doi.org/10.1007/s11912-019-0860-9).
- Kilic, A., Kilic, A., Kivanc, A. E. y Sisik, A. (2020) “Biopsy techniques for skin disease and skin cancer: A new approach”, *Journal of cutaneous and aesthetic surgery*, 13(3), pp. 251–254. doi: [10.4103/JCAS.JCAS_173_19](https://doi.org/10.4103/JCAS.JCAS_173_19).
- Kindt, T. J., Goldsby, R. A. y Osborne, B. A. (2007) *Inmunología de Kuby*.
- Looney, C. G. (1997) *Pattern recognition using neural networks: Theory and algorithms for engineers and scientists*. Nueva York, NY, Estados Unidos de América: Oxford University Press.
- Morrison, A. H., Byrne, K. T. y Vonderheide, R. H. (2018) “Immunotherapy and prevention of pancreatic cancer”, *Trends in cancer*, 4(6), pp. 418–428. doi: [10.1016/j.trecan.2018.04.001](https://doi.org/10.1016/j.trecan.2018.04.001).
- Petit, C., Lacas, B., Pignon, J. P., Le, Q. T. *et al.* (2021) “Chemotherapy and radiotherapy in locally advanced head and neck cancer: an individual patient data network meta-analysis”, *The lancet oncology*, 22(5), pp. 727–736. doi: [10.1016/S1470-2045\(21\)00076-0](https://doi.org/10.1016/S1470-2045(21)00076-0).
- Rosen, R. D. y Sapiro, A. (2023) *TNM Classification*. StatPearls Publishing.
- Shafique, R., Rustam, F., Choi, G. S., Díaz, I. T. *et al.* (2023) “Breast cancer prediction using fine needle aspiration features and upsampling with supervised machine learning”, *Cancers*, 15(3), pp. 681. doi: [10.3390/cancers15030681](https://doi.org/10.3390/cancers15030681).
- Shimizu, H. y Nakayama, K. I. (2020) “Artificial intelligence in oncology”, *Cancer science*, 111(5), pp. 1452–1460. doi: [10.1111/cas.14377](https://doi.org/10.1111/cas.14377).
- Tagliafico, A. S., Piana, M., Schenone, D., Lai, R. *et al.* (2020) “Overview of radiomics in breast cancer diagnosis and prognostication”, *Breast (Edinburgh, Scotland)*, 49, pp. 74–80. doi: [10.1016/j.breast.2019.10.018](https://doi.org/10.1016/j.breast.2019.10.018).
- UCI machine learning repository: Breast cancer Wisconsin (diagnostic) data set* (sin fecha) *Uci.edu*. Disponible en: <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29> (Accedido: 25 de octubre de 2022).
- Upadhyay, A. (2021) “Cancer: An unknown territory; rethinking before going ahead”, *Genes & diseases*, 8(5), pp. 655–661. doi: [10.1016/j.gendis.2020.09.002](https://doi.org/10.1016/j.gendis.2020.09.002).
- Van Rossum, G. y Drake, F. L. (2009) *Python (Versión 3.10.6) [Programa de ordenador]*. Disponible en <https://www.python.org/downloads/>.

Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., *et al.* (2015) “Scikit-learn: Machine learning without learning the machinery”, *GetMobile Mobile Computing and Communications*, 19(1), pp. 29–33. doi: [10.1145/2786984.2786995](https://doi.org/10.1145/2786984.2786995).

Wolberg, W. H., Street, W. N. y Mangasarian, O. L. (1994) “Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates”, *Cancer letters*, 77(2–3), pp. 163–171. doi: [10.1016/0304-3835\(94\)90099-x](https://doi.org/10.1016/0304-3835(94)90099-x).

ANEXO I: RESULTADOS COMPLETOS DE LA PRIMERA PRUEBA, REGRESIÓN LINEAL CON PCA

Tabla 2: resultados del test ROC para cada número de parámetros reducidos en el PCA junto al control (test sin PCA) que aparece con el nombre de NP.

Test	VP	VN	FP	FN	TPR	FPR	ACC	SPC	PPV	NPV	FDR	FV
1	8	35	0	13	38'1%	0%	76'79%	100%	100%	72'92%	0%	55'17%
2	8	35	0	13	38'1%	0%	76'79%	100%	100%	72'92%	0%	55'17%
3	10	35	0	11	47'62%	0%	80'36%	100%	100%	76'09%	0%	64'52%
4	9	33	2	12	42'86%	5'71%	75'0%	94'29%	81'82%	73'33%	18'18%	56'25%
5	9	32	3	12	42'86%	8'57%	73'21%	91'43%	75'0%	72'73%	25'0%	54'55%
6	9	32	3	12	42'86%	8'57%	73'21%	91'43%	75'0%	72'73%	25'0%	54'55%
7	9	32	3	12	42'86%	8'57%	73'21%	91'43%	75'0%	72'73%	25'0%	54'55%
8	9	32	3	12	42'86%	8'57%	73'21%	91'43%	75'0%	72'73%	25'0%	54'55%
9	9	30	5	12	42'86%	14'29%	69'64%	85'71%	64'29%	71'43%	35'71%	51'43%
10	9	30	5	12	42'86%	14'29%	69'64%	85'71%	64'29%	71'43%	35'71%	51'43%
11	12	32	3	9	57'14%	8'57%	78'57%	91'43%	80'0%	78'05%	20'0%	66'67%
12	12	32	3	9	57'14%	8'57%	78'57%	91'43%	80'0%	78'05%	20'0%	66'67%
13	12	31	4	9	57'14%	11'43%	76'79%	88'57%	75'0%	77'5%	25'0%	64'86%
14	11	32	3	10	52'38%	8'57%	76'79%	91'43%	78'57%	76'19%	21'43%	62'86%
15	11	32	3	10	52'38%	8'57%	76'79%	91'43%	78'57%	76'19%	21'43%	62'86%
16	10	33	2	11	47'62%	5'71%	76'79%	94'29%	83'33%	75'0%	16'67%	60'61%
17	10	33	2	11	47'62%	5'71%	76'79%	94'29%	83'33%	75'0%	16'67%	60'61%
18	10	33	2	11	47'62%	5'71%	76'79%	94'29%	83'33%	75'0%	16'67%	60'61%
19	10	32	3	11	47'62%	8'57%	75'0%	91'43%	76'92%	74'42%	23'08%	58'82%
20	10	32	3	11	47'62%	8'57%	75'0%	91'43%	76'92%	74'42%	23'08%	58'82%
21	10	32	3	11	47'62%	8'57%	75'0%	91'43%	76'92%	74'42%	23'08%	58'82%
22	10	30	5	11	47'62%	14'29%	71'43%	85'71%	66'67%	73'17%	33'33%	55'56%
23	10	31	4	11	47'62%	11'43%	73'21%	88'57%	71'43%	73'81%	28'57%	57'14%
24	10	31	4	11	47'62%	11'43%	73'21%	88'57%	71'43%	73'81%	28'57%	57'14%
25	10	31	4	11	47'62%	11'43%	73'21%	88'57%	71'43%	73'81%	28'57%	57'14%
26	10	30	5	11	47'62%	14'29%	71'43%	85'71%	66'67%	73'17%	33'33%	55'56%
27	10	31	4	11	47'62%	11'43%	73'21%	88'57%	71'43%	73'81%	28'57%	57'14%
28	10	31	4	11	47'62%	11'43%	73'21%	88'57%	71'43%	73'81%	28'57%	57'14%
29	10	31	4	11	47'62%	11'43%	73'21%	88'57%	71'43%	73'81%	28'57%	57'14%
30	10	31	4	11	47'62%	11'43%	73'21%	88'57%	71'43%	73'81%	28'57%	57'14%
NP	20	35	0	1	95'24%	0%	98'21%	100%	100%	97'22%	0%	97'56%

En la **tabla 2** se pueden observar todos los resultados de la curva ROC para la primera prueba de regresión lineal. Los diferentes parámetros están representados por sus siglas, que se corresponden a: VP como verdaderos positivos, VN como verdaderos negativos, FP como falsos positivos, FN como falsos negativos, FPR como el ratio de falsos positivos, siendo este el complementario de la especificidad ($1 - \text{SPC}$). PPV como el valor predictivo positivo o precisión, el cual se define como la razón entre los VP y la suma entre VP y FP. NPV como el valor predictivo negativo, el cual se define igual que la precisión pero con los parámetros complementarios. FDR como la razón de falsos descubrimientos, el cual se define como la razón entre FP respecto a la suma entre FP y VP. Finalmente, F representa el valor F, el cual es la media armónica entre la precisión y la sensibilidad.

ANEXO II: REALIZACIÓN DEL ESCRITO UTILIZANDO LAS TECNOLOGÍAS WEB

El presente escrito se programó en su totalidad exceptuando la portada mediante las tecnologías web *hyper-text markup language* (HTML), *cascading style-sheet* (CSS) y *javascript*. El objetivo era doble; por una parte fue el de obtener un documento en formato de página web con estilos para su posible publicación en internet y adaptado a la vista de un navegador, así como el de cumplir con uno de los objetivos de este trabajo, mejorar en la programación de estas tres tecnologías. Dicho documento puede ser fácilmente impreso en formato PDF desde un navegador.

El lenguaje más básico de las tecnologías web es el HTML. Es un estándar de la world wide web (www), se trata de un lenguaje de marcado que conocen e interpretan todos los navegadores. Se utiliza para construir la estructura básica de un documento web, mediante este lenguaje se añaden las estructuras genéricas como los párrafos, los saltos de línea o página, las imágenes, las tablas o las listas. Además de añadir todo el texto de la página. El texto que se dispone entre "<" y ">" se denomina etiqueta y se utilizan para definir los elementos de la página web. Mediante HTML no se pueden añadir estilos, para ello se amplió mediante CSS. Un ejemplo del uso de este lenguaje en este documento se observa en la **figura 21**.

```
<figure>
  <div class="flexfig cancerfig center" class="center">
    
    
  </div>
  <figcaption class="foot"><b>Figura 9</b>: imágenes de
  cáncer de piel del conjunto de entrenamiento. Muestra
  258 benigna (9a) y muestra 1305 maligna (9b) (Fanconi,
  2019).</figcaption>
</figure>
```

Figura 21: código HTML de la **figura 9** con sus respectivos textos alternativos y con la clase encargada de aplicar el estilo para centrarlas, envueltas en una división de contenido (div) y seguido del pie de figura (figcaption).

Por otra parte, el lenguaje CSS permite la adición de estilos a una página escrita en HTML. Permite personalizar el documento modificando el tipo o distribución de la fuente, desplazando los bloques, cambiando colores, manipulando imágenes, etc. También permite crear animaciones o reglas de estilo de página, o discernir entre una versión de escritorio y de dispositivo móvil. El lenguaje CSS utiliza selectores para aplicar los estilos solo a ciertos elementos o bloques que cumplan ciertas características. Un ejemplo de uso de este lenguaje aplicado a este documento se observa en la **figura 22**.

```

*
{
  text-align: justify;
  font-family: 'Times New Roman';
  font-size: 12pt;
  line-height: 150%;
}

@media print
{
  @page
  {
    size: 21cm 29.7cm;
    margin: 25mm 25mm 25mm 25mm;
  }
}

```

Figura 22: código CSS que aplica el estilo a todo los elementos de texto del documento. Aplica una fuente de 12 puntos, justificada y en Times New Roman, con un interlineado de 1'5 respecto al tamaño del texto (22a). Regla arroba CSS para el momento de impresión y regla arroba de página otorgando un tamaño de márgenes y de página de DIN A4 (22b).

Finalmente, el lenguaje de programación javascript otorga dinamismo a la página web, añadiendo la capacidad de los lenguajes de programación comunes. Posee también la capacidad de automatizar la generación de partes del documento escribiendo menos HTML. El ejemplo de cómo automatizar la generación de una tabla, añadiendo filas (tr) y celdas (td) se puede observar en la **figura 23**.

```

function display_table(table, data)
{
  let tr = document.createElement("tr")
  table.appendChild(tr)
  for(let i = 0; i < Object.values(data)[0].length; i++)
  {
    let tr = document.createElement("tr")
    for(let j of Object.values(data))
    {
      let td = document.createElement("td")
      td.innerText = j[i]
      tr.appendChild(td)
    }
    table.appendChild(tr)
  }
}

```

Figura 23: código javascript que automatiza la creación de una tabla mediante la toma de datos en formato de lista y con encabezados. El elemento tr se refiere a una fila de la tabla y td a una celda sobre la fila.