Diagnóstico automático de casos de riesgo de melanoma basado en imágenes dermatoscópicas

Antonio Carlos Rodríguez Bajo

GRADO DE CIENCIAS DE DATOS APLICADA
Trabajo final de grado 22.536

Tutora del TFG
Teresa Divorra Vallhonrat
Profesor responsable de la asignatura
David Merino Arranz

15 de enero de 2023

Universitat Oberta
de Catalunya

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Diagnóstico automático de casos de riesgo de melanoma basado en imágenes dermatoscópicas*
Nombre del autor:	*Antonio Carlos Rodríguez Bajo*
Nombre del consultor/a:	*Teresa Divorra Vallhonrat*
Nombre del PRA:	*David Merino Arranz*
Fecha de entrega (mm/aaaa):	*01/2023*
Titulación o programa:	*Grado de Ciencia de Datos Aplicada*
Área del Trabajo Final:	*Ciencia de Datos aplicada al ámbito de la salud*
Idioma del trabajo:	*Castellano*
Palabras clave:	*melanoma, aprendizaje automático, visión por computadora*

Resumen del Trabajo

El melanoma es un cáncer de la piel que puede ser letal si no es tratado convenientemente. La supervivencia de los pacientes depende en gran medida de una atención temprana por parte de profesionales médicos.

El objetivo principal de este Trabajo es demostrar el uso aplicado de la Ciencia de Datos y de la Inteligencia Artificial para crear un sistema de apoyo capaz de emitir una predicción de riesgo de melanoma a partir de imágenes dermatoscópicas.

Siguiendo una planificación del Trabajo por etapas, se han realizado las siguientes fases:

1. Estudio del arte sobre el tratamiento de imágenes mediante redes neuronales convolucionales, seleccionando EfficientNet.
2. Análisis de los conjuntos de datos abiertos de imágenes ISIC, seleccionando los datos del desafío del año 2019.
3. Experimentación y entrenamiento de modelos en Google Cloud para obtener el modelo óptimo.
4. Evaluación de las métricas de rendimiento y equidad.
5. Implementación del modelo en la nube de AWS, acompañado de una aplicación *web* para realizar diagnósticos sobre nuevas imágenes.

Los resultados de la implementación se consideran satisfactorios, con la recomendación de su uso en pacientes adultos, tanto mujeres como hombres, mayores de 30 años, con tonos de piel clara o ligeramente morena.

Mejoras en el sistema derivadas del incremento de la calidad de los datos, el aseguramiento de su interpretabilidad en un contexto clínico y la implementación de una práctica de MLOps para gestionar nuevas versiones podrían llevar a una implantación en un entorno real en producción al servicio de la comunidad médica.

Abstract

Melanoma is a skin cancer that can be life-threatening if not properly treated. Patient survival largely depends on early care by medical professionals.

The main objective of this Final Project is to demonstrate the applied use of Data Science and Artificial Intelligence to create a support system capable of producing melanoma risk predictions from dermoscopic images.

Following the planning of the Project in stages, these phases have been carried out:

1. Study of the state of the art in image processing with convolutional neural networks, selecting EfficientNet.
2. Analysis of open image datasets by ISIC, selecting the data from the challenge of the year 2019.
3. Experimentation and training of models in Google Cloud to obtain the optimal model.
4. Evaluation of performance and fairness metrics.
5. Implementation of the model in AWS cloud, complemented by a web application to perform diagnostics on new images.

The outcome of the implementation is considered satisfactory, with the recommendation for its use in adult patients, both women and men, over 30 years of age, with light skin tones.

Enhancements in the system derived from the improvement of the quality of the data, the assurance of its interpretability in clinical contexts and the application of MLOps to create and deploy new versions could lead to an implementation in a real-world production environment at the service of the medical community.

Índice

Lista de figuras

Lista de Tablas

1. Introducción

Gracias al gran avance en los últimos siglos de la medicina, la calidad en la vida de las personas ha mejorado de manera significativa, mejora que puede acelerarse de una manera importante gracias a la utilización de datos masivos que pueden procesarse mediante técnicas de Ciencia de Datos que ayudan de una manera decisiva a los profesionales de la salud en su lucha contra las enfermedades.

Según datos de la Organización Mundial de la Salud [1], el cáncer es la principal causa de muerte en el mundo. En el año 2020 se contabilizaron alrededor de 10 millones de muertes debidas a esta enfermedad, lo que supone casi una de cada seis fallecimientos en todo el mundo. El diagnóstico precoz del cáncer es fundamental, ya que si es detectado en una fase temprana es más probable que los pacientes respondan al tratamiento, con lo que aumentan las probabilidades de supervivencia y se mejora de una manera significativa su calidad de vida.

La American Cancer Society [2] revela que "el cáncer de piel es el más común entre todos los tipos de cáncer. El melanoma, aunque solo representa el 1% de los casos, es el tipo más letal de cánceres de piel y su incidencia ha aumentado significativamente en las últimas décadas". Si un melanoma se diagnostica mientras aún está confinado a las capas externas de la piel, la escisión simple en general produce su curación con una supervivencia muy alta. A pesar de que el melanoma puede ser diagnosticado en una etapa temprana a través de una simple inspección visual, en muchos pacientes se diagnostica en un estado ya avanzado, disminuyendo la probabilidad de supervivencia.

La dermatoscopia es una técnica diagnóstica no invasiva que permite observar estructuras de la epidermis y de la dermis mediante un sistema de amplificación de la imagen y un sistema de iluminación que elimina la distorsión producida por la reflexión y refracción de la luz en la superficie cutánea, de manera que se muestran patrones de pigmento y de vascularización no visibles a simple vista. Esta técnica es "utilizada de manera sistemática en el diagnóstico del melanoma, ya que se ha estimado que la fiabilidad del diagnóstico de un dermatólogo experto ronda el 70-85% con la visualización a simple vista, mientras que puede aumentar hasta un 92% mediante el uso de la dermatoscopia" [3].

Hay que tener en cuenta que las diferencias visuales entre un melanoma y una lesión benigna de la piel en las imágenes dermatoscópicas pueden ser muy sutiles [4], incluso para dermatólogos expertos en la materia. Por esta razón, un sistema de diagnóstico automático de casos de riesgo de melanoma basado en este tipo de imágenes sería una herramienta valiosa a la hora de ayudar a expertos o para entrenar a dermatólogos en formación a la hora de diagnosticar lesiones cutáneas como melanomas.

1.1. Contexto y justificación del Trabajo

Una vez establecida la utilidad de un sistema de diagnóstico automático de casos de riesgo de melanoma basado en imágenes dermatoscópicas, el siguiente paso es plantear cómo se puede realizar el mismo utilizando métodos de Ciencia de Datos Aplicada. Dada la naturaleza de un Trabajo de Fin de Grado, las actividades desarrolladas se encuentran acotadas por el tiempo (un semestre de curso universitario), los recursos humanos (un único estudiante con apoyo de su tutora) y los recursos económicos (utilización de recursos computacionales de bajo coste).

Considerando las restricciones mencionadas, se puede considerar este Trabajo como una "prueba de concepto", es decir, un proyecto con recursos limitados que demuestre la viabilidad de una idea que posteriormente, en caso de éxito, se tome como punto de partida para el desarrollo en un proyecto con más recursos, para ser puesto en producción en un entorno de altas prestaciones, al servicio de la comunidad médica en general.

Aplicando el principio de prueba de concepto, se ha decidido realizar el desarrollo de este proyecto en una plataforma con la capacidad de ser extendida de manera prácticamente ilimitada en términos de rendimiento, escalabilidad y disponibilidad, garantizando al mismo tiempo la seguridad del sistema. Con este fin, se han utilizado los recursos de dos de los proveedores mundiales más importantes de servicios informáticos en la nube: Amazon AWS y Google Cloud.

Aplicando Ciencia de Datos, ha sido factible preparar un conjunto de datos de gran volumen, incluyendo imágenes dermatoscópicas, para que se pueda emplear por un modelo de diagnóstico creado mediante técnicas de Inteligencia Artificial de aprendizaje automático. De esta manera, el modelo ha sido capaz de predecir automáticamente el riesgo de melanoma de una imagen nueva introducida en el sistema. Aunque en sistemas avanzados de este tipo la capacidad predictiva puede ser comparable a la de expertos humanos, no se pretende que estos sistemas sustituyan en un futuro próximo a los profesionales médicos, sino que sirvan de apoyo al diagnóstico.

Un aspecto fundamental en los sistemas de Inteligencia Artificial que realizan predicciones sobre las personas, en este caso diagnósticos médicos, es asegurar la equidad en el tratamiento de todos los grupos o colectivos afectados. A menudo, los datos que se utilizan en el entrenamiento de un modelo pertenecen a un grupo mayoritario, ya sea por razones demográficas, económicas, políticas o culturales, por lo que el modelo puede encontrar dificultades a la hora de efectuar una predicción sobre una persona de un grupo que estaba subrepresentado cuando fue entrenado. Por esta causa, además de realizar la habitual evaluación de la capacidad predictiva del modelo, es imprescindible evaluar también la equidad de este.

1.2. Objetivos del Trabajo

El objetivo principal de este Trabajo es la presentación de un caso de aplicación de Ciencia de Datos mediante la realización de un producto. Este producto se trata de un sistema informático capaz de realizar el diagnóstico automático de casos de riesgo de melanoma basado en imágenes dermatoscópicas.

El sistema consta de dos componentes principales:

- Modelo de diagnóstico, generado mediante métodos de Ciencia de Datos e Inteligencia Artificial: el modelo realiza de manera automática la tarea de clasificación binaria entre melanoma y no melanoma, con la correspondiente probabilidad. Una vez preparados los datos e imágenes, provenientes de conjuntos de datos abiertos y anonimizados con diagnósticos realizados por dermatólogos expertos en la materia, se aplican técnicas de aprendizaje automático supervisado para generar un modelo de diagnóstico.
- Aplicación *web* de diagnóstico automático de nuevas imágenes dermatoscópicas: aplicación disponible en Internet donde los usuarios proporcionan al sistema imágenes nuevas sin diagnosticar para que el sistema produzca un resultado de riesgo de melanoma.

Para la consecución del objetivo principal de Trabajo se consideran los siguientes objetivos específicos:

- Estudio del estado del arte: el tratamiento automatizado de imágenes médicas es un ámbito que ha sido profusamente tratado por prestigiosos equipos multidisciplinares formados por profesionales médicos y de ciencias de la computación. Se trata, pues, de analizar propuestas ya realizadas con sus beneficios e inconvenientes con el fin de extraer conocimiento que oriente en la elección de las mejores técnicas de Ciencia de Datos a la hora de desarrollar el objetivo principal del Trabajo.
- Diseño de la propuesta de técnicas específicas de Ciencia de Datos e Inteligencia Artifical a utilizar: partiendo del estudio del estado del arte y de las reflexiones efectuadas, se diseña una propuesta con métodos basados en aprendizaje automático supervisado para alcanzar el objetivo del Trabajo. El diseño debe seleccionar las técnicas más adecuadas, teniendo en cuenta las restricciones temporales y de recursos del Trabajo.
- Desarrollo de las técnicas seleccionadas y realización de experimentos: una vez determinados las técnicas a emplear, se generan una serie de modelos experimentales que puedan ser parametrizados con el fin de obtener unos resultados óptimos. Se deben definir qué métricas se necesitan emplear para evaluar como óptimo un modelo y parametrización determinados frente a otras opciones.
- Realización e implementación del sistema en un entorno real de producción: mediante esta implantación se demuestra el uso aplicado de la Ciencia de Datos en la resolución del problema planteado.

1.3. Impacto en sostenibilidad, ético-social y de diversidad

Se debe considerar que los modelos predictivos basados en aprendizaje automático, Inteligencia Artificial y Ciencia de Datos se utilizan cada vez más en problemas que pueden tener un impacto importante en la vida de las personas, como puede ser la detección temprana del cáncer de piel del tipo melanoma.

Si se analiza el impacto positivo de un proyecto de diagnóstico de riesgo de melanoma en aspectos de sostenibilidad medioambiental, es importante considerar que una detección temprana mejora de una manera significativa la calidad de vida de los pacientes, al tiempo que disminuye también de forma importante el consumo de medicamentos y la realización de intervenciones médicas, con el consiguiente ahorro energético. Además, se debe considerar que, en los episodios más avanzados de esta enfermedad, los tratamientos son en general cada vez más agresivos y producen mayor cantidad de residuos biosanitarios contaminados que deben ser tratados de manera adecuada.

Por lo que respecta al impacto ético-social, un proyecto de este tipo debe considerar el reto que supone para cualquier proyecto de Ciencia de Datos el tratamiento de información personal, como son las imágenes dermatoscópicas y los metadatos asociados a ellas. Es importante asegurarse de que estos datos no son utilizados para fines que no sean estrictamente los especificados a los pacientes a la hora de recopilar dicha información.

Aunque las fuentes de datos que se utilizan en este proyecto están anonimizadas, cabe siempre el peligro de realizar generalizaciones en función a resultados con riesgo de sesgo que ofrezca el sistema. Así, por ejemplo, existe la posibilidad de realizar un análisis de los diagnósticos efectuados y llegar a la conclusión de que ciertos segmentos de la población tienen un mayor riesgo de desarrollar un cáncer de tipo melanoma, con lo cual, se podría aplicar un factor de penalización que incremente una póliza de seguro médico. Esta penalización sería un ejemplo de impacto negativo en el aspecto ético-social.

Aun considerando como importante el impacto de este tipo de proyecto en los aspectos de sostenibilidad y ético-social, probablemente sea el impacto en diversidad el aspecto más significativo para considerar. Diversas investigaciones han suscitado preocupaciones sobre el riesgo de sesgos no intencionados en estos modelos, que pueden afectar de forma injusta a individuos de ciertos grupos.

Como ejemplo de posible trato discriminatorio en el ámbito de aplicación del sistema de detección de melanoma, se puede considerar la información proporcionada por la Academia Americana de Dermatología [5], donde se indica que, *"*si bien las personas de piel clara corren mayor riesgo de contraer cáncer de piel, la tasa de mortalidad de las personas afroamericanas es considerablemente más alta, ya que su tasa de supervivencia a cinco años es del 73%, en comparación con el 90% de los estadounidenses blancos*".*

Hay que tener en cuenta que, si un modelo basa la mayor parte de su conocimiento en cómo aparecen las lesiones cutáneas en segmentos mayoritarios de la población, al ser la gran mayoría de datos de entrenamiento pertenecientes a personas de ese grupo, entonces se corre el riesgo de no predecir de manera correcta lesiones en pacientes de segmentos minoritarios. Por consiguiente, se deben evaluar los sesgos en los datos de entrenamiento y en el modelo que puedan afectar al color de la piel, sexo, edad u otros factores demográficos.

Así pues, es importante considerar las características sensibles, es decir *"*atributos como raza, edad o sexo cuya representación insuficiente o excesiva pueda dar lugar a discriminación en la respuesta del sistema. Eliminando o ignorando las características sensibles no evita el aprendizaje de modelos sesgados, porque otras características correlacionadas se pueden usar como intermediarias para las características sensibles. Si bien se han propuesto muchas métricas de medición de sesgo y definiciones de equidad, no hay consenso sobre qué definiciones y métricas deben usarse en la práctica para evaluar y auditar este tipo de proyectos de Ciencia de Datos*"* [6].

Una propuesta que puede ser empleada para evaluar la equidad en los resultados es el "marco para la equidad algorítmica" de la iniciativa Aequitas [7]. Esta iniciativa dispone de un conjunto de herramientas de código abierto para la auditoría de sesgos. Mediante estos instrumentos, se pueden auditar las predicciones de un sistema de diagnóstico automático y detectar diferentes tipos de sesgos. En el *Anexo I: Métricas de equidad de Aequitas* de este documento se encuentra información detallada sobre estas métricas y sus correspondientes fórmulas de cálculo.

1.4. Enfoque y método seguido

Para la realización de este Trabajo se ha seguido una metodología propia de un proyecto. El Project Management Institute define un proyecto como "un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único". En este caso, se ha creado un producto nuevo que es único en lo que se refiere a sus características específicas, el cual ha necesitado de un tiempo determinado para su consecución.

La metodología empleada para la gestión de un proyecto depende fundamentalmente de la naturaleza del propio proyecto. Los dos tipos de metodología más utilizados son:

- Metodología ágil (*agile*): eficiente y flexible con suficiente ambigüedad para hacer cambios en un proyecto sin importar lo avanzado que esté el proyecto. La gestión ágil de proyectos trabaja en iteraciones o *sprints*. Cada iteración es un elemento entregable pequeño con una fecha límite corta, por lo general de 1 a 3 semanas, en la que un equipo del proyecto se centra y completa. Después de cada entrega, se reciben comentarios del cliente. A partir de los comentarios recibidos, el equipo adapta el proyecto en consecuencia a medida que trabaja en cada *sprint*.
- Metodología en cascada (*waterfall*): completamente estructurada, define cada paso de principio a fin. El concepto se basa en la idea de que los proyectos se componen de etapas que se suceden una tras otra, como si fuese una cascada, con una fecha de inicio, fin y resultados conocidos y consensuados entre el equipo del proyecto y el cliente.

La metodología en cascada es la que mejor se adapta a las necesidades de este tipo de proyecto, al ser este un proyecto que se ha desarrollado con una restricción de principio y fin bien determinada, y con un resultado final definido entre el estudiante y la tutora del Trabajo de Fin de Grado.

Una metodología en cascada muy empleada para la gestión de proyectos es la creada por el Project Management Institute a través de su Guía de los Fundamentos para la Dirección de Proyectos PMBOOK (*Project Management Body of Knowledge*) [8]. En esta metodología se define un ciclo de vida del proyecto con cinco etapas:

- Iniciación: en esta fase la dirección de la organización identifica un problema o necesidad, la conceptualiza en forma de proyecto, analiza su viabilidad técnica y económica y los riesgos y, en su caso, la aprueba.
- Planificación: en esta fase se obtiene un acuerdo acerca de los temas del proyecto. Se debe producir al menos un conjunto de documentos que servirán como elementos fundamentales de orientación en las siguientes fases:
    - Alcance: definición de lo que va a hacer el proyecto.
    - Plan estratégico: se descompone el trabajo establecido en el alcance en partes o paquetes más pequeños o EDT (estructura de distribución del trabajo), que pueden constituir entregables parciales o generales.
    - Plan operativo: se descompone cada EDT en actividades, se ponen en secuencia, con los recursos necesarios y se establece un calendario. Con estos datos, se estiman los costes y se elabora un presupuesto.
- Ejecución: siguiendo la planificación establecida se lleva a cabo el alcance del proyecto, realizando las actividades definidas en el plan operativo para completar los paquetes de trabajo definidos en el EDT.
- Seguimiento y control: son procesos permanentes y en paralelo a lo largo del ciclo de vida del proyecto. Todos los aspectos contenidos en los diferentes planes deben ser evaluados y, en su caso, reajustados. Los procesos más críticos en esta fase son los de control de cambios (cualquier petición o incidencia que afecta a la planificación inicial) y los de gestión de riesgos.
- Cierre: incluye todas las actividades necesarias para finalizar la gestión del proyecto y completar las obligaciones contenidas en el acuerdo con el cliente.

Es importante considerar que la etapa de ejecución es específica a cada tipo de proyecto. Para el caso de proyectos de Ciencia de Datos, se suele considerar la metodología CRISP-DM (*CRoss-Industry Standard Process for Data Mining*), publicada en 1999 para estandarizar los procesos de minería de datos en todas las industrias y que desde entonces se ha convertido en la metodología más común para proyectos de minería de datos, análisis y Ciencia de Datos.

El siguiente diagrama muestra las seis fases de esta metodología:

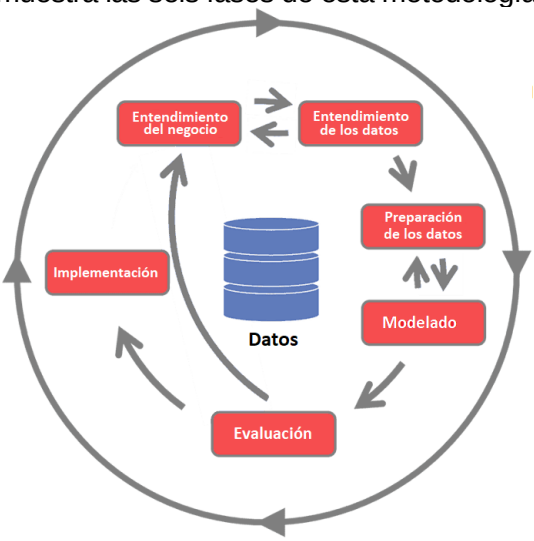


Figura 1: Diagrama CRISP-DM.
Fuente: [9]

Según la Data Science Process Alliance [9], cada una de estas fases tiene la finalidad de responder a unas preguntas concretas:

- Entendimiento del negocio: ¿Qué necesita el negocio, en este caso el proyecto para el Trabajo?
- Entendimiento de los datos: ¿Qué datos se tienen y qué datos se necesitan? ¿Tienen los datos la calidad suficiente o se necesita hacer una limpieza de datos previa?
- Preparación de los datos: ¿Cómo se organizan los datos para el modelado? ¿Es necesario realizar extracción de características, normalización de datos, etc.?
- Modelado: ¿Qué técnicas de modelado, en este caso de aprendizaje automático supervisado, se deben aplicar?
- Evaluación: ¿Qué modelo cumple mejor con el objetivo planteado en el proyecto, es decir, el diagnóstico de casos de riesgo de melanoma?
- Implementación: ¿Cómo se implementa el sistema para que se acceda al mismo e introducir nuevas observaciones y obtener los diagnósticos?

Siguiendo este modelo para la etapa de ejecución, las respuestas a las preguntas que se plantean se desarrollan en los siguientes apartados de este documento:

Fase	Apartado
Entendimiento del negocio	*1.1. Contexto y justificación del Trabajo* *1.3. Impacto en sostenibilidad, ético-social y de diversidad*
Entendimiento de los datos	*4.3.1. Análisis de los conjuntos de datos de la base de datos de ISIC* *4.3.2. Análisis del conjunto de datos del desafío ISIC del año 2019* *4.3.2.1. Análisis exploratorio de los metadatos*
Preparación de los datos	*4.3.3. Preparación de los datos de entrenamiento, validación y prueba*
Modelado	*4.3. Modelo de diagnóstico*
Evaluación	*4.3.7. Evaluación del rendimiento del modelo de diagnóstico* *4.3.8. Evaluación de la equidad del modelo de diagnóstico*
Implementación	*5. Implantación del sistema.*

Tabla 1: Fases de CRISP-DM aplicadas en el Trabajo.
Fuente: elaboración propia.

1.5. Planificación del Trabajo

Para la realización del trabajo se han empleado aproximadamente 300 horas de trabajo (una media de unas 18 horas por semana). Teniendo en cuenta el modelo de evaluación continua de la UOC, se ha considerado una planificación temporal con cinco entregas parciales o PEC (prueba de evaluación continua).

La planificación y su correspondencia con el ciclo de vida del proyecto con cinco etapas del PMBOOK es la siguiente:

Núm. PEC	Descripción	Etapa PMBOOK	Fechas estimadas
PEC 1	Definición y plan del proyecto	Iniciación Planificación	28/09/22 a 07/10/22
PEC 2	Desarrollo del trabajo - Fase 1	Ejecución	08/10/22 a 11/11/22
PEC 3	Desarrollo del trabajo - Fase 2	Ejecución	12/11/22 a 23/12/22
PEC 4	Cierre de la memoria y de la presentación	Cierre	27/12/22 a 15/01/23
PEC 5	Defensa pública del TFG	Cierre	23/01/23 a 03/02/23

Tabla 2: Hitos del proyecto.
Fuente: elaboración propia.

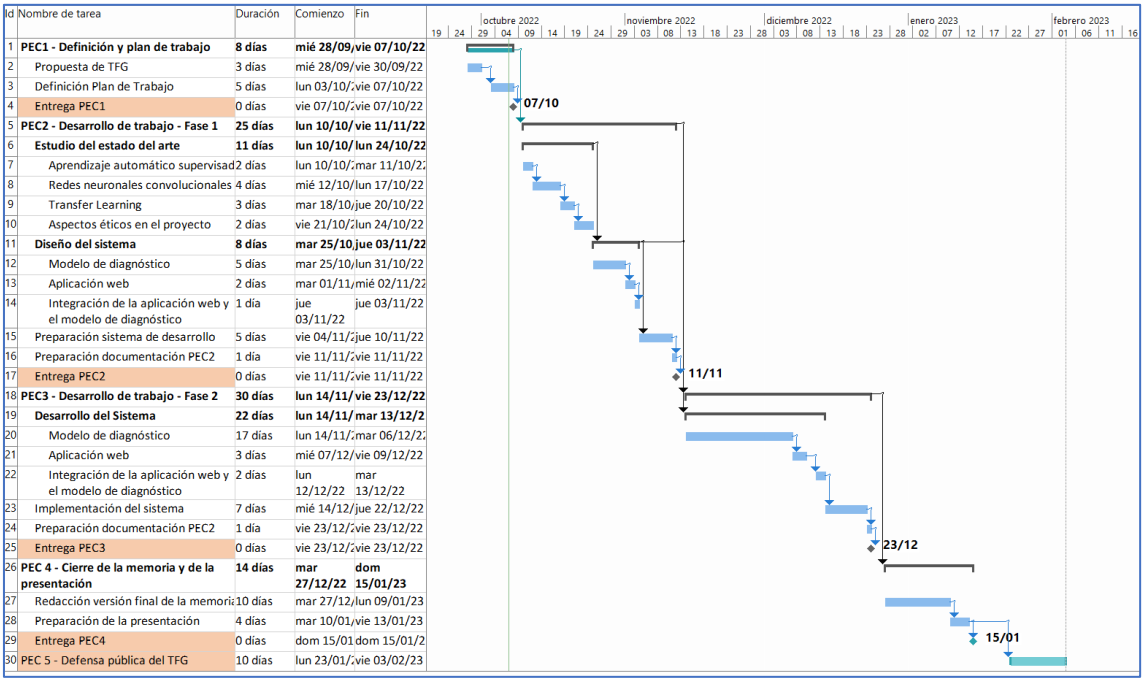
De acuerdo con los objetivos generales y específicos del TFG se han establecido las siguientes tareas:

Objetivo	Tareas	Duración (días laborales)
Estudio del estado del arte	Aprendizaje automático supervisado	2
	Redes neuronales convolucionales	4
	Transfer Learning	3
	Aspectos éticos en el proyecto	2
Diseño del sistema	Modelo de diagnóstico	5
	Aplicación web	2
	Integración de la aplicación web y el modelo de diagnóstico	1
Desarrollo del sistema	Preparación sistema de desarrollo	5
	Modelo de diagnóstico	17
	Aplicación web	3
	Integración de la aplicación web y el modelo de diagnóstico	2
Implementación del sistema	Implementación del sistema	7
Realización versión final de la memoria del TFG	Realización versión final de la memoria del TFG	10

Realización de la presentación del TFG	Realización de la presentación del TFG	4
Defensa pública del TFG	Defensa pública del TFG	10

Tabla 3: Tareas del proyecto.
Fuente: elaboración propia.

El calendario con el que ha realizado el proyecto es el siguiente:



Id	Nombre de tarea	Duración	Comienzo	Fin
1	**PEC1 - Definición y plan de trabajo**	**8 días**	**mié 28/09/**	**vie 07/10/22**
2	Propuesta de TFG	3 días	mié 28/09/	vie 30/09/22
3	Definición Plan de Trabajo	5 días	lun 03/10/	vie 07/10/22
4	Entrega PEC1	0 días	vie 07/10/	vie 07/10/22
5	**PEC2 - Desarrollo de trabajo - Fase 1**	**25 días**	**lun 10/10/**	**vie 11/11/22**
6	**Estudio del estado del arte**	**11 días**	**lun 10/10/**	**lun 24/10/22**
7	Aprendizaje automático supervisad	2 días	lun 10/10/	mar 11/10/2
8	Redes neuronales convolucionales	4 días	mié 12/10/	lun 17/10/22
9	Transfer Learning	3 días	mar 18/10/	jue 20/10/22
10	Aspectos éticos en el proyecto	2 días	vie 21/10/	lun 24/10/22
11	**Diseño del sistema**	**8 días**	**mar 25/10/**	**jue 03/11/22**
12	Modelo de diagnóstico	5 días	mar 25/10/	lun 31/10/22
13	Aplicación web	2 días	mar 01/11/	mié 02/11/22
14	Integración de la aplicación web y el modelo de diagnóstico	1 día	jue 03/11/22	jue 03/11/22
15	Preparación sistema de desarrollo	5 días	vie 04/11/	jue 10/11/22
16	Preparación documentación PEC2	1 día	vie 11/11/	vie 11/11/22
17	Entrega PEC2	0 días	vie 11/11/	vie 11/11/22
18	**PEC3 - Desarrollo de trabajo - Fase 2**	**30 días**	**lun 14/11/**	**vie 23/12/22**
19	**Desarrollo del Sistema**	**22 días**	**lun 14/11/**	**mar 13/12/2**
20	Modelo de diagnóstico	17 días	lun 14/11/	mar 06/12/2
21	Aplicación web	3 días	mié 07/12/	vie 09/12/22
22	Integración de la aplicación web y el modelo de diagnóstico	2 días	lun 12/12/22	mar 13/12/22
23	Implementación del sistema	7 días	mié 14/12/	jue 22/12/22
24	Preparación documentación PEC2	1 día	vie 23/12/	vie 23/12/22
25	Entrega PEC3	0 días	vie 23/12/	vie 23/12/22
26	**PEC 4 - Cierre de la memoria y de la presentación**	**14 días**	**mar 27/12/22**	**dom 15/01/23**
27	Redacción versión final de la memoria	10 días	mar 27/12/	lun 09/01/23
28	Preparación de la presentación	4 días	mar 10/01/	vie 13/01/23
29	Entrega PEC4	0 días	dom 15/01/	dom 15/01/2
30	PEC 5 - Defensa pública del TFG	10 días	lun 23/01/	vie 03/02/23

Figura 2: Cronograma del proyecto.
Fuente: elaboración propia.

1.6. Breve sumario de productos obtenidos

Como producto obtenido por este Trabajo se entrega un sistema de diagnóstico automático de casos de riesgo de melanoma basado en imágenes dermatoscópicas. Sus componentes y características principales son las siguientes:

- Se ha desarrollado un modelo de diagnóstico optimizado para imágenes dermatoscópicas de 512x512 píxeles con 3 canales de color RGB. El modelo es una red neuronal convolucional (CNN) de tipo EfficientNet-B5 entrenada en la plataforma Google Colab con cerca de 20.000 imágenes de la base de datos ISIC, utilizando transferencia de aprendizaje desde ImageNet.

- El modelo de diagnóstico ha sido evaluado con cerca de 3.500 imágenes de prueba que se han reservado y no han sido vistas por el modelo durante la fase de entrenamiento. Las métricas de rendimiento de precisión y sensibilidad obtenidas son superiores a 0.8, mientras que las métricas de equidad son favorables para la mayoría de los grupos de personas en los que se puede segmentar la base de datos de imágenes de prueba en función a sus metadatos.

- Se ha realizado la implantación del modelo de diagnóstico en la nube de AWS empleando la plataforma SageMaker. Se ha optado por utilizar el servicio de inferencia sin servidor, mediante el cual AWS asigna de forma automática recursos de computación para el modelo según sea necesario, sin necesidad de tener una o varias máquinas virtuales asignadas al modelo de manera permanente. Este tipo de despliegue permite escalabilidad y disponibilidad prácticamente ilimitada del servicio, garantizando la seguridad del sistema, aunque el tiempo de respuesta pueda ser superior al de un despliegue con máquinas virtuales dedicadas.

- Se ha desarrollado e implantado una aplicación *web* de diagnóstico, la cual se encuentra disponible en Internet en la dirección:

  https://d39fz5pvr27vh5.cloudfront.net

  Mediante esta aplicación se pueden introducir nuevas imágenes dermatoscópicas y obtener el nivel de riesgo de melanoma emitido por el modelo de diagnóstico. La aplicación se ha desarrollado e implantado con AWS Amplify, que como en el caso del modelo de diagnóstico permite una completa seguridad, escalabilidad y disponibilidad del servicio.

1.7. Breve descripción de los otros capítulos de la memoria

En el **capítulo 2** se describe el estudio del arte que se ha realizado para el Trabajo. Se han explorado las técnicas de Inteligencia Artificial basada en aprendizaje automático mediante las cuales una máquina es capaz de aprender y emitir un diagnóstico de riesgo de melanoma a partir de imágenes médicas.

Se ha profundizado en el análisis de los tipos de modelos que están proporcionando en la actualidad un mejor rendimiento en este tipo de problemas de visión por computadora, como son las redes neuronales convolucionales (CNN). Asimismo, se ha analizado cómo se puede evaluar un sistema de estas características, tanto desde el punto de vista de rendimiento, es decir, precisión y sensibilidad en sus predicciones, como desde el punto de vista de la equidad, de manera que se compruebe que el modelo realiza predicciones que sean independientes al grupo demográfico al que pertenezcan los pacientes y no difiera la calidad del diagnóstico en función del sexo, tono de piel o edad.

Una vez establecido el estado del arte y la estrategia a seguir para evaluar el modelo, en el **capítulo 3** se describe el diseño del sistema de diagnóstico. Este sistema consta de dos elementos principales: el modelo, que se diseña en función al conocimiento adquirido en el estudio del arte, y la aplicación *web*, que proporciona un interfaz para introducir nuevas imágenes para que el modelo emita una predicción de riesgo de melanoma.

A continuación, en el **capítulo 4**, se detalla cómo se ha desarrollado el sistema de diagnóstico. De capital importancia es la selección y preparación de las fuentes de datos, con el correspondiente análisis exploratorio. Se ha concluido con la selección de las imágenes y fuentes de datos correspondientes al desafío ISIC del año 2019, al presentar una distribución más alta de casos de melanoma. Una vez seleccionadas las fuentes de datos, se ha procedido a prepararlos en el formato más apropiado para una CNN de tipo EfficientNet.

Una vez explicada la selección y preparación de datos, en el capítulo 4 se detalla cómo se han realizado una serie de experimentos con distintas configuraciones de hiperparámetros, resoluciones de imágenes y versiones de EfficientNet para obtener un modelo óptimo respecto a las métricas de rendimiento. Este modelo ha sido también evaluado con respecto a las métricas de equidad seleccionadas. Asimismo, también se explica cómo se ha desarrollado la aplicación *web* de diagnóstico y cómo se ha integrado esta aplicación con el modelo de diagnóstico.

Una vez desarrollado el sistema, en el **capítulo 5** se describe cómo se ha implantado el mismo en la plataforma en la nube de AWS, para continuar con el **capítulo 6** donde se detallan los resultados obtenidos. Se finaliza con el **capítulo 7**, dedicado a las conclusiones del Trabajo y líneas de trabajo que se podrían seguir para profundizar y mejorar los resultados de este sistema de visión por computadora especializado en el diagnóstico de casos de melanoma, con el fin de que se pudiese emplear en un entorno real por parte de la comunidad médica.

2. Estudio del estado del arte

2.1. Aprendizaje automático

El aprendizaje automático (*machine learning*) es una disciplina dentro de la Inteligencia Artificial cuyo objetivo es el estudio de algoritmos informáticos que mejoran de manera automática a partir de la experiencia. Una definición clásica sobre el concepto de aprendizaje en el contexto del aprendizaje automático fue dada en 1997 por el profesor T. M. Mitchell [10]:

"Diremos que un programa informático aprende de la experiencia E respecto de alguna tarea T y de alguna medida de rendimiento P, si su rendimiento en T, medido por P, mejora con la experiencia E".

La tarea T se puede definir como realizar la correspondencia entre una entrada, la cual es definida por una serie de características, con una salida determinada. Por ejemplo, en el caso de una imagen dermatoscópica, La entrada serían los pixeles de la imagen o características de la imagen si se hace una extracción de características visuales, y la salida sería una información concreta, por ejemplo, una probabilidad de que el paciente sufra cáncer de melanoma. La mayor fortaleza del aprendizaje automático radica en "su capacidad para descubrir correspondencias entre entradas y salidas que no pueden ser codificados estáticamente mediante una aplicación informática o incluso por expertos humanos en la materia" [11].

Respecto a las medidas de rendimiento P, varían en función de la problemática tratada. En general, los algoritmos de aprendizaje automático son entrenados con un conjunto de datos de entrenamiento para proporcionar una función de salida. Posteriormente a este entrenamiento, se implementa una medida cuantitativa para evaluar cómo funciona el algoritmo sobre un conjunto de datos no vistos con anterioridad por el algoritmo.

La experiencia E a través de la cual un algoritmo de aprendizaje automático puede aprender a realizar la tarea T depende sobre todo de los datos utilizados en su fase de entrenamiento. Se pueden establecer muchos tipos, siendo los principales:

- Aprendizaje supervisado: el conjunto de entrenamiento contiene la salida deseada, para cada uno de sus elementos. Si la salida es continua se suele denominar una tarea de regresión y en el caso de salida discreta, de clasificación. No obstante, hay casos donde a partir de una salida continua como puede ser el caso de probabilidad de riesgo de melanoma se puede producir una clasificación (melanoma sí o melanoma no).
- Aprendizaje no supervisado: en este caso, el conjunto de datos de entrenamiento no contiene una salida específica, es decir se trata de datos sin etiquetar, y el algoritmo debe inferir conocimiento a partir de estos datos, normalmente observando la estructura en la representación en los datos. Un ejemplo sería la realización de segmentación de clientes de una empresa en función a sus preferencias y hábitos de compra para realizar diferentes campañas publicitarias diferenciadas para cada segmento.
- Aprendizaje semisupervisado: el etiquetado de datos suele ser un proceso costoso, por lo que es común en una organización tener muchas instancias sin etiquetar y pocas instancias etiquetadas. Los algoritmos de aprendizaje semisupervisado están diseñados para manejar datos que están parcialmente etiquetados. Un ejemplo son los servicios de redes sociales con fotografías que pueden realizar en una primera fase un proceso no supervisado donde se agrupan las fotografías donde aparece la misma persona y después el sistema

pregunta a los usuarios quién es esa persona, con lo que es capaz de reconocer a esa persona en todas las fotos.

- Aprendizaje reforzado: en este tipo de algoritmo, el sistema solo recibe datos sin etiquetar. A través de interacciones con un entorno, el sistema recibe retroalimentación positiva o negativa acerca de su desempeño, siendo capaz de adaptarse según la retroalimentación recibida. Por ejemplo, los robots utilizan este tipo de algoritmos para aprender a caminar. De esta manera, si un robot planifica una trayectoria y tropieza con un objeto recibirá una penalización, pero en ningún momento hay una persona que le indique una trayectoria alternativa correcta. De igual manera, si la trayectoria planificada por el robot no causa una colisión, el robot recibe una gratificación. La planificación de la trayectoria que el robot haga a partir de este momento tendrá en cuenta las penalizaciones y gratificaciones recibidas con anterioridad.

En el caso de aprendizaje supervisado, que es el que se ha empleado para este proyecto, se realiza el entrenamiento para ajustar los parámetros del algoritmo empleado y obtener la mejor predicción posible. Para ello, en primer lugar, se debe definir cómo se van a comparar las predicciones realizadas por el modelo frente a la salida deseada, lo cual se realiza mediante la llamada función de pérdida (*loss function*), que depende del tipo de problema.

El objetivo del entrenamiento no es otro que minimizar esa función de pérdida a través de múltiples iteraciones (épocas o *epochs*) que ajusten cada vez mejor los parámetros del algoritmo empleado. El cambio que se produce en cada iteración para intentar encontrar ese mínimo es un hiperparámetro que se denomina tasa de aprendizaje (*learning rate*) y suele ser de gran importancia a la hora de conseguir un buen modelo. Por una parte, un *learning rate* elevado puede acelerar el aprendizaje, pero se corre el peligro de no alcanzar el mínimo global de la función de perdida, mientras que un *learning rate* reducido puede provocar que el entrenamiento se quede de manera permanente en un mínimo local y no se acerque al mínimo global.

Otro aspecto importante para considerar durante la fase de entrenamiento es el riesgo de sobreajuste (*overfitting*). Este fenómeno se produce cuando el modelo después de ser entrenado es capaz de dar una predicción mucho mejor sobre los datos de entrenamiento que sobre datos nuevos. En este caso, se puede decir que el modelo ha memorizado los datos con los que ha sido entrenado, pero no ha sido capaz de generalizar la resolución del problema planteado.

Una técnica habitual para paliar el problema de sobreajuste es reservar una porción de los datos de entrenamiento para realizar la validación del modelo al final de cada época, de manera que se compara el rendimiento de las predicciones sobre los datos de entrenamiento frente a los de validación, que no han sido vistos todavía por el modelo, Mediante este mecanismo se pueden ajustar diversos hiperparámetros del algoritmo y reducir el sobreajuste durante la fase de entrenamiento.

2.2. Redes neuronales

Las redes neuronales son "un conjunto de algoritmos que están inspirados en el mecanismo de comunicación de la neurona biológica" [12]. Este tipo de algoritmos son indicados para resolver problemas donde el conocimiento puede ser impreciso o puede variar en el tiempo. Se pueden utilizar tanto para sistemas de aprendizaje automático de clasificación y regresión como para sistemas de aprendizaje no supervisado para tareas de agrupamiento.

Explicado de una manera sencilla, una red neuronal está formada por un conjunto de unidades elementales denominadas neuronas que se interconectan entre sí. Cada neurona existente en la red aplica una función sobre los valores de las entradas que provienen de otras neuronas que están conectadas a ella, y proporciona una salida, que a su vez sirve de entrada para otras neuronas.

El esquema básico de una neurona es el siguiente:

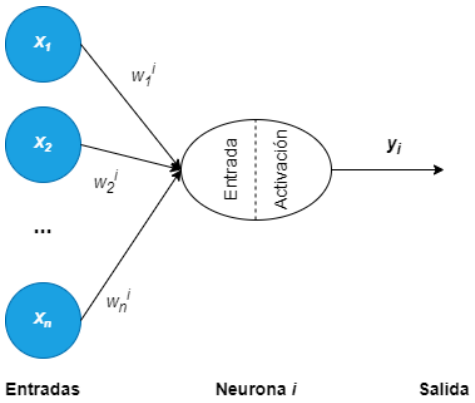


Figura 3: Esquema de una neurona.
Fuente: [12].

En este esquema se representa una neurona *i* que presenta un conjunto de entradas $X = \{x_1, x_2, \dots, x_n\}$. Cada elemento del conjunto de entradas se encuentra ponderada por un conjunto de valores $W^i = \{w_1^i, w_2^i, \dots, w_n^i\}$. Estos valores w_j^i representan la importancia o peso que se asigna al valor de entrada x_j que llega a la neurona *i* desde una neurona *j* conectada a ella.

La neurona toma el conjunto de entradas X que llegan a ella y le aplica una función de entrada a las mismas. El valor obtenido sirve de entrada de una función de activación que genera un valor de salida y_i. Este valor, a su vez, sirve de entrada a otras neuronas conectadas con ella o puede ser el valor de salida de toda la red si no hubiera más neuronas conectadas a continuación.

Las funciones de entrada más habituales son las siguientes [12]:

Suma ponderada: $z(x) = \sum_{j=1}^{n} x_j w_j^i$.
Máximo: $z(x) = \max(x_1 w_1^i, \dots, x_n w_n^i)$.
Mínimo: $z(x) = \min(x_1 w_1^i, \dots, x_n w_n^i)$.
AND lógico (solo en entradas binarias): $z(x) = (x_1 w_1^i \wedge \dots \wedge x_n w_n^i)$.
OR lógico (solo en entradas binarias): $z(x) = (x_1 w_1^i \vee \dots \vee x_n w_n^i)$.

En cuanto a las funciones de activación, hay que considerar que para problemas que se puedan resolver por aproximaciones lineales hay algoritmos más apropiados, como el algoritmo de regresión. En el caso de las redes neuronales, se intentan resolver otros tipos de problemas, donde el conocimiento es más impreciso y de los cuales no se conoce una ecuación matemática que represente un modelo de predicción. Por esta razón, es conveniente emplear combinaciones de funciones lineales y no lineales en las funciones de activación.

Las funciones de activación que se usan con más frecuencia son [12]:

Lineal: crea combinaciones lineales a partir de las entradas.

$$y(x) = \beta x$$

Escalón: se activa a partir de un umbral α, dando una salida con los valores $\{-1, 1\}$ *o* $\{0, 1\}$

$$y(x) = \begin{cases} -1, & x < \alpha \\ 1, & x \geq \alpha \end{cases}$$

Sigmoide o logística: curva acotada entre los valores $\{0, 1\}$ cuyo parámetro ρ determina la suavidad de la curva. Es no lineal, diferenciable y monótona y el cambio es mayor para valores intermedios y menor para valores extremos. Es muy empleada en problemas de clasificación binaria.

$$y(x) = \frac{1}{1 + e^{\frac{-x}{\rho}}}$$

Tangente hiperbólica: con propiedades similares a la función sigmoide, acotada entre $\{-1, 1\}$.

$$y = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

Rectificadora o rampa: es muy utilizada en redes neuronales profundas.

$$y(x) = \max(0, x)$$

Se muestran a continuación las representaciones gráficas de las funciones de activación más frecuentes.

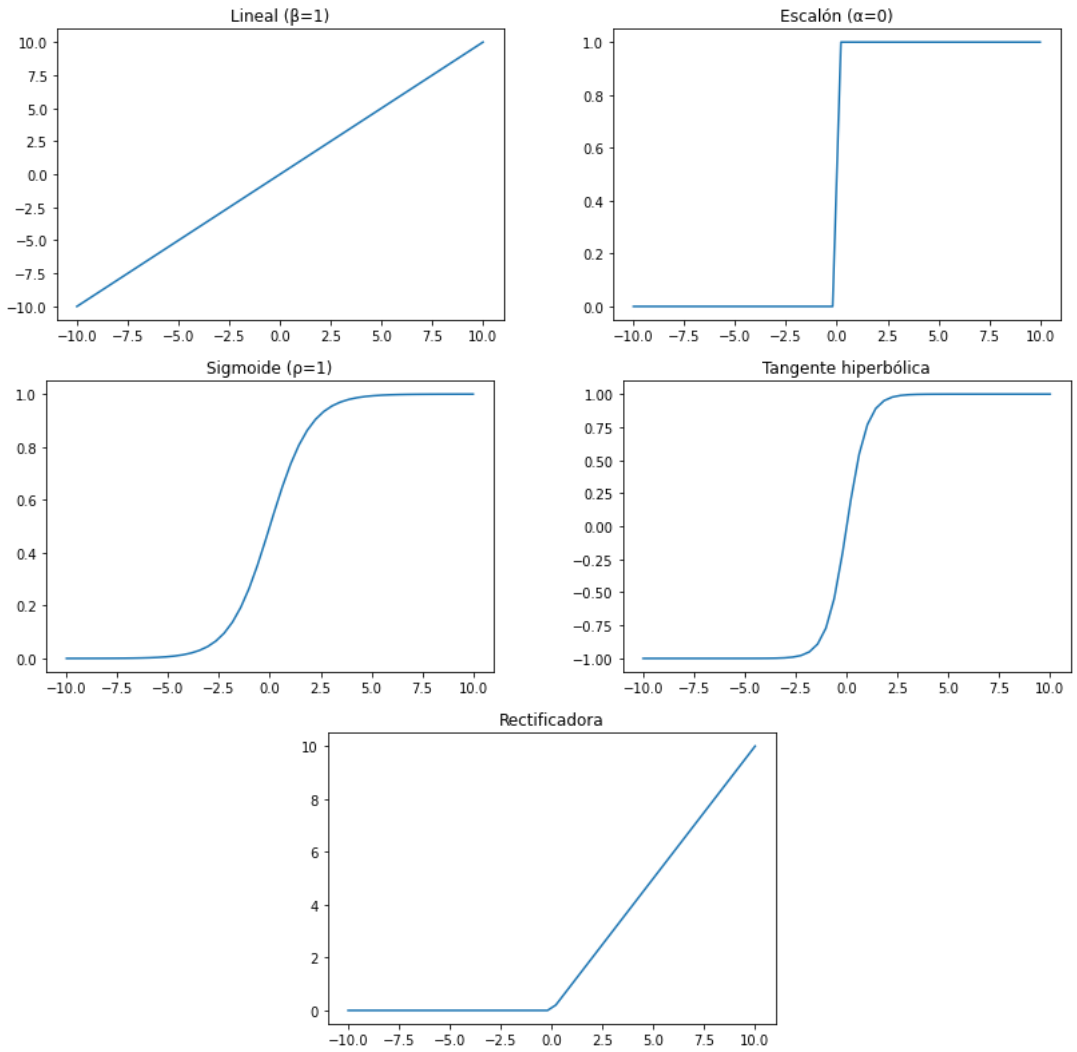


Figura 4: Funciones de activación más frecuentes.
Fuente: elaboración propia.

2.2.1. Arquitectura de las redes neuronales

En una red neuronal, la arquitectura, también denominada topología, se refiere a la organización de sus neuronas en distintas capas. La arquitectura dependerá del tipo de problema a resolver y de la calidad de los resultados que se deseen obtener, teniendo en cuenta que tendrán un distinto coste computacional en función a la complejidad de la estructura resultante.

Los ejemplos típicos de arquitectura de redes neuronales son los siguientes [12]:

- Neurona única: es la topología más simple, con una neurona conectada a una serie de entradas que proporciona una única salida. Este tipo de red podría efectuar tareas sencillas, similares a las funciones de regresión no lineal.
- Monocapa: en esta arquitectura, la red presenta una capa de entrada, una capa oculta de procesamiento con un conjunto variable de neuronas, y una capa de salida con una o múltiples neuronas. Este tipo de redes pueden realizar predicciones sobre patrones de entrada más complejos y de dimensionalidad más alta comparadas con las redes de neurona única. La capa oculta produce una mayor capacidad de procesamiento, pero se aumenta el riesgo de sobreajuste y que el modelo no sea capaz de generalizar correctamente. La capa

de salida depende del problema a resolver, por ejemplo, en una clasificación binaria, una única neurona de salida es suficiente.

- Multicapa: tomando la arquitectura monocapa, es posible aumentar el número de capas ocultas y aumentar la complejidad de la arquitectura de la red. Se utilizan para realizar predicciones sobre patrones aún más complejos, pero con el consiguiente riesgo de sobreajuste. Cuando el número de capas en esta arquitectura es muy elevado se suele denominar red neuronal profunda.

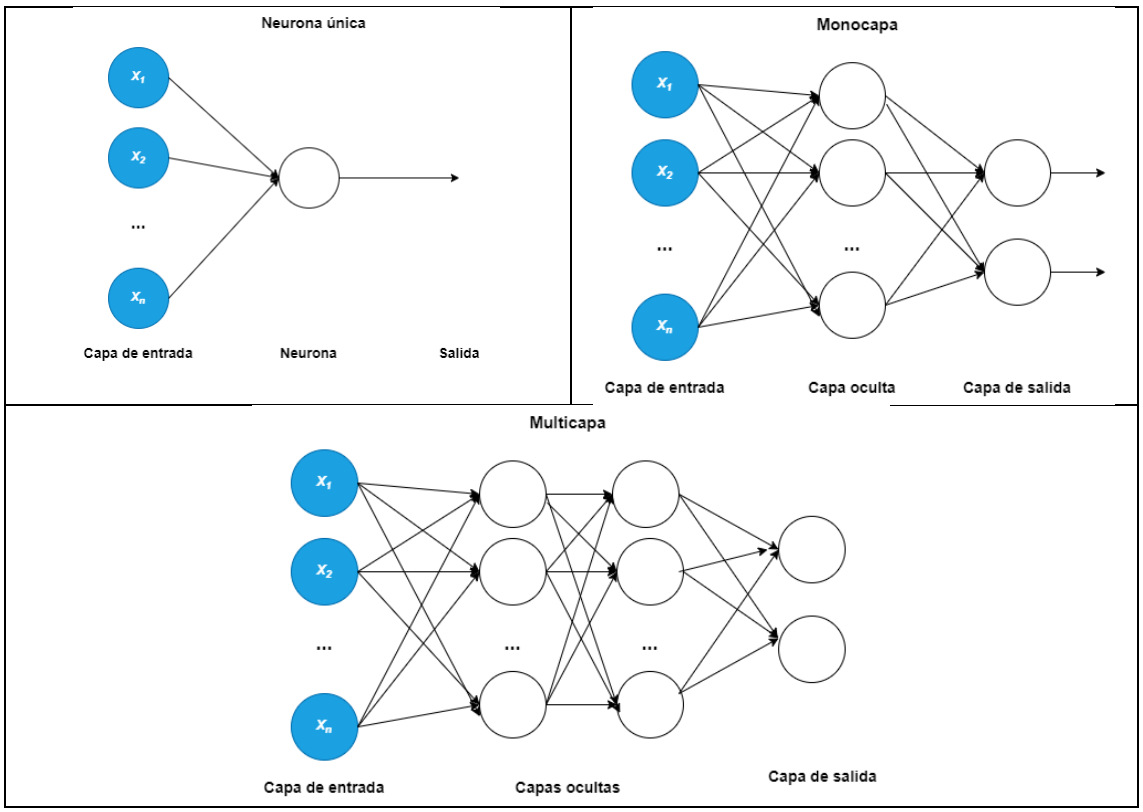


Figura 5: Arquitecturas típicas de las redes neuronales.
Fuente: elaboración propia.

2.3. Redes neuronales convolucionales (CNN)

Uno de los retos más importantes a los que se enfrenta un sistema de visión por computador es el hecho de que el número en entradas al sistema es muy elevado. Por ejemplo "para una imagen de 1.000 píxeles de alto por 1.000 píxeles de ancho con tres canales de color RGB, se necesita un vector de 3 millones de dimensiones para almacenar esta imagen" [12]. Si se considerase una red neuronal con una capa oculta de 1.000 neuronas, la dimensión de la primera matriz sería de $1000 \times 3 \times 10^6 = 3 \times 10^9$ posiciones, es decir 3 mil millones de posiciones, que equivaldría al número de parámetros necesarios para configurar una red neuronal completamente conectada.

Esta enorme cantidad de parámetros producen importantes problemas computaciones y es prácticamente imposible tener suficientes datos de entrenamiento para prevenir el sobreajuste. Así pues, es imprescindible implementar algún tipo de mecanismo que permita reducir la dimensionalidad de una imagen, conservando el máximo posible la información útil que contenga. Una solución utilizada en este campo es la operación de convolución, que es el fundamento básico de las redes neuronales convolucionales (CNN), donde esta operación es utilizada en lugar de la multiplicación de matrices en sus capas.

La convolución [11] "es una operación matemática lineal sobre dos funciones *(f, g)* que produce una tercera función *s"*. Aplicada al ámbito de las CNN, la función *f* se suele corresponder a la entrada (*input*) y la función *g* se suele denominar el filtro (*kernel*). La salida de la operación de convolución *s* se suele denominar mapa de características (*feature map*). De manera matemática, la operación de convolución, que se suele denotar con un asterisco $*$, se puede expresar de la siguiente manera para datos discretos [12]:

$$s(t) = (f * g)(t) = \sum_{\tau=-\infty}^{\infty} f(\tau) g(t-\tau)$$

En el caso de aplicaciones de aprendizaje automático, la entrada *f* es un vector de datos de múltiples dimensiones y el filtro *g* es un vector de parámetros también multidimensional. Cuando se trabaja con imágenes en dos dimensiones, las convoluciones se suelen aplicar sobre las dos dimensiones de la imagen. Particularizando la operación de convolución para una imagen *I* de dos dimensiones con tamaño *i x j con* un filtro *K* de dos dimensiones de tamaño *m x n,* la fórmula para la operación de convolución es la siguiente [12]*:*

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n) K(i-m, j-n)$$

Como la operación de convolución es conmutativa, se puede expresar de manera equivalente [12]:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n) K(m,n)$$

Por lo general, esta segunda fórmula es más sencilla de implementar por una función de aprendizaje automático, porque hay menos variación en el rango de valores válidos de *m* y *n*.

La propiedad conmutativa de la convolución se cumple porque se ha invertido el *kernel* en relación con la entrada, en el sentido de que a medida que aumenta *m*, el índice en la entrada aumenta, pero el índice en el *kernel* disminuye. La razón para invertir el *kernel* es cumplir la propiedad conmutativa, pero esta propiedad no es necesaria en la implementación de las CNN, y la mayoría de las bibliotecas que trabajan con estas redes implementan la función de *cross-correlation* para mejorar su eficiencia, aunque se suele seguir llamando operación de convolución, aplicando la siguiente fórmula [12]:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n)$$

La siguiente figura muestra un ejemplo de operación de convolución (estrictamente hablando sería una operación de *cross-correlation*) sobre una imagen de 2 dimensiones de tamaño 4x3 píxeles, aplicando un *kernel* de 2 dimensiones de tamaño 2x2. Para la salida, se dibujan cuadros con una flecha para indicar como se forma el elemento superior izquierdo de la matriz de 2 dimensiones de salida, aplicando el *kernel* a la región superior izquierda correspondiente a la matriz de 2 dimensiones de entrada. El resto de las posiciones de la matriz de salida se calculan de manera equivalente, trasladado los cuadrados a la derecha y hacia abajo, teniendo en cuenta que se emplean solo las posiciones donde el *kernel* cuadra con la imagen, es decir, que no se sale del borde de la imagen.

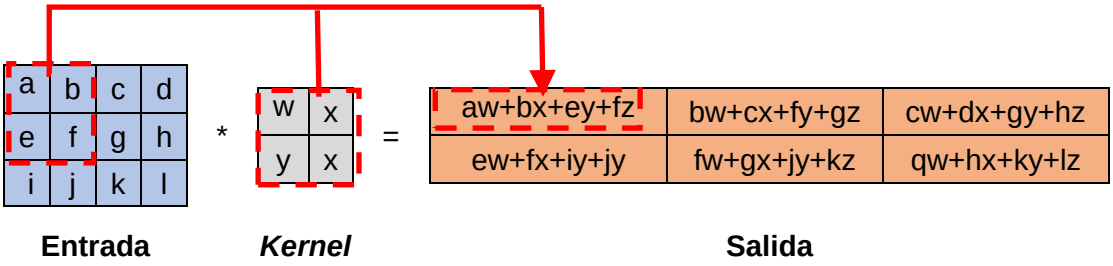


Figura 6: Ejemplo de operación de convolución.
Fuente: [12].

La operación de convolución descrita permite a las CNN una mejora sustancial en dos aspectos importantes como son "el tratamiento de interacciones o conexiones dispersas (cuando las neuronas de una capa solo se conectan con un subconjunto de las neuronas de la capa siguiente) y la compartición de parámetros (cuando los pesos de las neuronas de una capa son los mismos)" [11].

Además de las capas convolucionales, las CNN suelen incluyen también otros tipos de capas que se describen a continuación [12]:

Capa de agrupamiento (*pooling*): la más utilizada es la de *max-pooling*, que selecciona el valor máximo de entre un conjunto de valores de entrada. Por ejemplo, en una entrada con dimensiones 4x4, si se aplica un *max-pooling* de 2 se dividiría el conjunto de entrada en 4 regiones y se tomaría el valor máximo de cada región, resultando una salida de dimensiones 2x2, extrayendo aquellas características más frecuentes y relevantes de cada zona. De esta manera, se consigue que las características más importantes tengan más peso en el red, lo que produce CNN más robustas y eficientes.

Capa rectificadora (ReLU – *rectified linear unit*): es habitual aplicar esta capa después de una capa de convolución, para aplicar no linealidad al sistema, siendo la función ReLU (función rampa) una de las más utilizadas.

Capa totalmente conectada (*fully connected*): en una CNN con múltiples capas ocultas, de manera general no están todas las neuronas de una capa conectadas con todas las neuronas de la capa anterior y posterior. Sin embargo, hay casos donde puede ser necesario hacerlo, por ejemplo a la salida de la CNN donde se realiza una clasificación de varias categorías y el resultado es un vector con la predicción que consiste en la probabilidad de cada una de las categorías.

Capa de abandono (*dropout*): son capas muy empleadas para evitar el sobreajuste en una CNN. Su mecanismo consiste en desactivar un número aleatorio de las entradas de la capa de abandono, lo que impide que la red memorice los datos de entrada durante el entrenamiento, lo que ayuda a prevenir el sobreajuste. Esta capa se utiliza solo durante la fase de entrenamiento de la CNN.

2.4. CNN EfficientNet

EfficientNet fue propuesto en el año 2019 por Google AI en el artículo *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks* para la *International Conference on Machine Learning* [13].

Una manera típica de mejorar el rendimiento de una CNN aplicada a un problema de visión por computadora consiste en aumentar los recursos computacionales asignados a la red y dotarla de mayor profundidad (más capas), incrementar su anchura (más neuronas por cada capa) y aumentar la resolución de las imágenes de entrada para aumentar la resolución.

Según el artículo de Google AI, "estudiando las dimensiones de profundidad, anchura y resolución, se llega a la situación de que el incremento de manera independiente de estas dimensiones provoca una rápida saturación de la red". El estudio concluye con: "si se aumentan los recursos de computación se consigue un mayor rendimiento escalando estas tres dimensiones al mismo tiempo con lo que se denomina escalado compuesto".

Basándose en ese principio, se creó la arquitectura EfficientNet, cuya arquitectura de base es el modelo EfficientNet-B0, a partir de la cual se puede escalar hasta el modelo EfficientNet-B7:

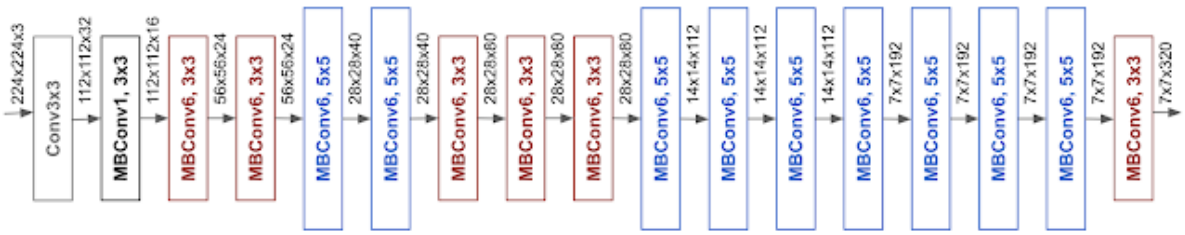


Figura 7: EfficientNet.
Fuente: [13]

Con esta arquitectura base, Google ha desarrollado una familia de modelos, llamados EfficientNets, que pueden llegar, según propias estimaciones de Google, a ofrecer un rendimiento de "hasta 10 veces superior comparado con otros sistemas basados en CNN" [13].

2.5. Transferencia de aprendizaje

La técnica de transferencia de aprendizaje consiste en "entrenar una red de origen con un conjunto de datos de origen para una tarea de origen, transferir el conocimiento aprendido (por ejemplo, características o pesos) a una nueva red para una tarea de aprendizaje diferente y/o en un dominio objetivo diferente" [14].

Esta técnica de optimización permite entrenar de forma más rápida y mejorar el rendimiento de un modelo, por lo cual es muy popular en el área del aprendizaje profundo, debido a la gran cantidad de datos y recursos computacionales que se requieren para entrenar un modelo de estas características.

Como se ha explicado con anterioridad, el entrenamiento de una red neuronal consiste en ajustar los pesos de la red utilizando un conjunto de datos. Estos pesos, si la red neuronal es lo suficientemente generalista, en principio se pueden transferir a otra red neuronal, en lugar de entrenar la red neuronal desde cero.

Sin embargo, hay que destacar que "esta técnica solo funciona si las características que ha aprendido el modelo durante el entrenamiento de la primera tarea son lo suficientemente generales para ser útiles y aplicables en otros entornos similares. Esta forma de transferir el aprendizaje que se usa en *deep learning* se llama transferencia inductiva (*inductive transfer*)" [12] .

Para generar una red neuronal generalista para la transferencia de aprendizaje se emplea a menudo [15] el conjunto de datos de imágenes ImageNet [16], el cual está organizado según la jerarquía de WordNet (base de datos léxica de dominio público). Cada concepto significativo en WordNet, descrito por varias palabras o frases de palabras, se denomina "conjunto de sinónimos" o "*synset*". Hay más de 100.000 *synsets* en WordNet, siendo la mayoría de ellos sustantivos (80.000+).

ImageNet [16] es "un esfuerzo de investigación con el objetivo de proporcionar a los investigadores de todo el mundo datos de imágenes para entrenar modelos de reconocimiento de objetos a gran escala". El objetivo final es proporcionar un promedio de 1.000 imágenes para ilustrar cada *synset*. Las imágenes de cada concepto tienen control de calidad y anotaciones humanas. Al completarse, se espera que ImageNet ofrezca decenas de millones de imágenes ordenadas y etiquetadas de manera correcta para la mayoría de los conceptos en la jerarquía de WordNet.

2.6. Medidas de evaluación del rendimiento del modelo de clasificación

Las métricas de evaluación del rendimiento juegan un papel fundamental para lograr el modelo óptimo de un modelo de clasificación. Para poder obtener estas métricas, se suele dividir el conjunto de datos de entrada al modelo en tres subconjuntos de manera aleatoria, pero de manera que contengan de manera aproximada la misma proporción de categorías a clasificar que el conjunto original.

El primer subconjunto se emplea para entrenar el modelo, el segundo se utiliza para validar los resultados obtenidos por el modelo y para ajustar sus parámetros. Por último, el tercer subconjunto de datos, llamado subconjunto de prueba o *test*, se emplea para evaluar el modelo mediante las métricas correspondientes.

Es fundamental que el subconjunto de prueba este aislado de la fase de entrenamiento y sea nuevo para el modelo en el momento de la evaluación, de manera que no haya ningún tipo de filtrado de información entre este subconjunto y los otros dos subconjuntos que han servido para entrenar el modelo.

Para definir las métricas de evaluación en el problema de clasificación de melanomas se pueden definir las siguientes categorías de predicciones del modelo:

- Verdadero positivo (*true positive*, TP): el modelo predice como melanoma una imagen etiquetada como melanoma.
- Falso positivo (*false positive*, FP): el modelo predice como melanoma una imagen que no está etiquetada como melanoma.
- Verdadero negativo (*true negative*, TN): el modelo predice como no melanoma una imagen que no está etiquetada como melanoma.
- Falso negativo (*false negative*, FN): el modelo predice como no melanoma una imagen que está etiquetada como melanoma.

El etiquetado de las imágenes se refiere al diagnóstico de expertos dermatólogos, que se considera que es la variable del mundo real que se modela con el sistema de aprendizaje automático. Una vez que se obtiene el modelo, se introduce el subconjunto de prueba, sin las etiquetas de diagnóstico, y los resultados se suelen exponer en la llamada matriz de confusión, que relaciona las predicciones realizadas por el modelo con la realidad.

		Predicción	
		Melanoma	No Melanoma
Realidad	Melanoma	TP	FN
	No Melanoma	FP	TN

Figura 8: Matriz de confusión.
Fuente: elaboración propia.

A continuación, se explican distintas métricas asociadas a la matriz de confusión para evaluar el rendimiento del modelo. Todas ellas varían entre 0 y 1, siendo 1 el mejor valor posible.

Exactitud (*accuracy*): indica el número de casos clasificados de forma correcta en comparación con el número total de casos.

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

Precisión (*precision*): esta métrica representa el número de verdaderos positivos que son realmente positivos en comparación con el número total de valores positivos predichos.

$$P = \frac{TP}{TP + FP}$$

Sensibilidad, exhaustividad o ratio de verdaderos positivos (*recall o sensitivity*): es la cantidad de verdaderos positivos que el modelo ha clasificado en función del número total de valores positivos. Representa la probabilidad de una imagen con melanoma sea identificada de manera correcta.

$$R = \frac{TP}{TP + FN}$$

Especificidad o ratio de verdaderos negativos (*specifity*): es la cantidad de verdaderos negativos que el modelo ha clasificado en función del número total de valores positivos. Representa la probabilidad de una imagen sin melanoma sea identificada de manera correcta.

$$S = \frac{TN}{TN + FP}$$

Puntuación F1: Esta métrica es la combinación de las métricas de precisión y sensibilidad y sirve como compromiso entre ambas.

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

El objetivo ideal del sistema de diagnóstico es pronosticar de manera correcta todos los casos positivos, es decir, todas las imágenes con melanoma. Es preferible que el sistema haga una predicción de falso positivo a una de falso negativo. En este escenario donde se penalizan los falsos negativos, las métricas de precisión y sensibilidad serían las más importantes. Como es lógico, la predicción de falsos positivos tendría también un coste humano asociado, en este caso podría ser la realización de una biopsia que determinase un caso benigno de lesión cutánea considerado como melanoma por el modelo.

Curva ROC (*Receiver Operating Characteristic* - Característica Operativa del Receptor): en los sistemas de clasificación binaria se establece un umbral de probabilidad a partir de cual se considera un caso como positivo (por defecto es 0.5). Para el caso de detección de melanoma, donde interesa maximizar los verdaderos positivos, se puede emplear la curva ROC, que es una representación gráfica que muestra el rendimiento de un modelo de clasificación según el umbral que se determine. La curva ROC muestra la representación del ratio de verdaderos positivos (TPR) respecto al ratio de falsos positivos (FPR) con diferentes umbrales de clasificación.

$$TPR = \frac{TP}{TP + FN} \quad FPR = \frac{FP}{TP + TN}$$

Reducir el umbral de clasificación predice más casos como positivos, pero aumentando tanto los falsos positivos como los verdaderos positivos.

AUC (*Area Under the ROC Curve* – Área bajo la curva ROC): es la medida del área que queda bajo la curva ROC, lo cual da una medida de rendimiento a través de todos los posibles umbrales de clasificación. Un modelo con todas sus predicciones erróneas obtendría un valor de AUC=0, mientras el que tuviese todas sus predicciones correctas obtendría un AUC=1. Un clasificador aleatorio obtendría un valor AUC=0.5

Curva PR (*precision-recall* - precisión-sensibilidad): esta curva muestra la relación entre la precisión y la sensibilidad de un clasificador binario. En general, si se entrena un clasificador para aumentar la precisión, disminuirá su sensibilidad, y viceversa. Por ello, puede ser interesante representar ambas métricas de una manera gráfica.

PR AUC (*Area Under the PR Curve* – Área bajo la curva PR): es la medida del área bajo la curva PR. Cuanto más se acerque a 1 (alta precisión y alta sensibilidad), mejor será el rendimiento del modelo de clasificación binaria.

2.7. Medidas de evaluación de la equidad del modelo de clasificación

Como marco de trabajo para evaluar la equidad en los resultados del modelo de diagnóstico se emplea para este proyecto el marco para la equidad algorítmica de la iniciativa Aequitas [7]. Este marco sugiere una serie de métricas para evaluar los algoritmos de inteligencia artificial en función al uso que se hacen de los mismos. En función del tipo de problema que se intenta resolver, se puede aplicar el llamado "árbol de decisión de la equidad" para determinar que métricas son las mejores para realizar la evaluación [7]:

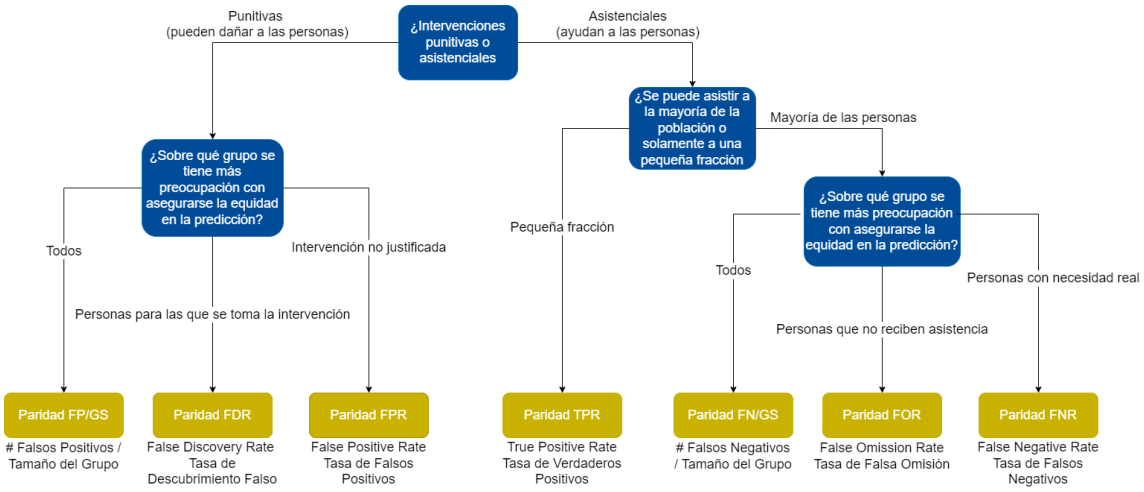


Figura 9: Árbol de decisión de la equidad.
Fuente: [7].

Las fórmulas completas para el cálculo de las métricas de equidad se pueden consultar en el *Anexo I: Métricas de equidad de Aequitas*

Teniendo en cuenta que la finalidad del sistema de diagnóstico es asistencial, es decir, ayudar a las personas, las métricas que resultan más interesantes para la evaluación de la equidad son las que se encuentran en las ramas derechas del árbol, es decir [17]:

Tasa de verdaderos positivos (también denominada sensibilidad, *recall o sensitivity*). Es la probabilidad de que la imagen se clasifique como melanoma si es realmente un melanoma.

$$R = \frac{TP}{TP + FN}$$

Tasa de falsos negativos respecto al tamaño del grupo.

Tasa de falsa omisión. Representa la probabilidad de que una imagen sea un melanoma si ha sido detectada como no melanoma.

$$FOR = \frac{FN}{TN + FN}$$

Tasa de falsos negativos. Es la probabilidad de que la imagen sea clasificada como no melanoma pero realmente es melanoma.

$$FNR = \frac{FN}{TP + FN}$$

Con estas métricas puede ser interesante comparar los valores de los distintos grupos de personas (agrupadas por sexo, tono de piel, edad, o cualquier otra característica). Si se toma un grupo de referencia, la disparidad de sesgo *j* para un grupo a_i respecto a ese grupo de referencia se calcula de la siguiente manera [7]:

$$disparidad_{j,a_i} = \frac{métrica_{j,a_i}}{métrica_{j,a_{grupo\ de\ referencia}}}$$

2.8. Aprendizaje automático aplicado al diagnóstico de melanoma

En el campo de la visión por computadora hay diferentes usos del aprendizaje automático que se pueden emplear para analizar una imagen que se pueden aplicar a este proyecto [12]:

- Clasificación de imágenes: dada una imagen, se pretende determinar a qué clase pertenece. En el caso de imágenes dermatoscópicas, se puede clasificar una imagen en un tipo determinado de lesión cutánea (melanoma, angiofibroma, dermatofibroma, lentigo, etc.).
- Detección de objetos: se identifica la presencia de objetos con un cuadro delimitador y predice a que clase pertenecen los objetos ubicados en una imagen. Correspondería a detectar distintas lesiones cutáneas en una imagen y delimitarlas.
- Segmentación semántica: se clasifican todos los píxeles de una imagen en clases de objetos. Estas clases son interpretables semánticamente y se asocian a categorías. Por ejemplo, se podría aislar todos los píxeles asociados a un melanoma y colorearlos con una tonalidad distinta al resto de la imagen.
- Segmentación de instancias: se señalan los píxeles de varios objetos u anomalías concretas, identificando cada instancia de forma individual.

Este proyecto se orienta hacia la tarea de clasificación binaria de imágenes de riesgo de melanoma. Asociada a la clasificación binaria se proporciona la probabilidad calculada por el modelo.

También es importante destacar que se puede obtener una mejora en los sistemas de detección de melanoma [18] mediante el procesamiento de imágenes para contrarrestar la presencia de artefactos (por ejemplo, pelo o regla de medir) y la variabilidad en las características de las imágenes (por ejemplo, contraste, intensidad o ángulo).

Un estudio de referencia en el ámbito de aplicación el aprendizaje automático aplicados al diagnóstico de melanoma es el *Dermatologist-level classification of skin cancer with deep neural networks* [19] realizado en el año 2017 por un equipo de la Universidad de Stanford liderado por Andre Esteva. Con un modelo entrenado con más de 125.000 imágenes y basado en una adaptación de CNN InceptionV3 con transferencia de aprendizaje del *ImageNet Large Scale Visual Recognition Challenge* [20], el estudio mostró "un ratio de éxito en sus predicciones de melanoma en torno al 94%, similar al ratio de éxito en el diagnóstico de 21 dermatólogos con los que se comparó el estudio" [19] .

A partir del éxito del estudio de la Universidad de Stanford, en los últimos años "se ha adoptado en gran medida la utilización de CNN para los sistemas de clasificación de imágenes dermatoscópicas" [21]. En el *Anexo II: Arquitecturas típicas de las CNN aplicadas a la detección de riesgos de melanoma* de este documento se describen y comparan las arquitecturas más habituales.

En cuanto a las fuentes de datos de imágenes, una iniciativa internacional para crear una base de datos de imágenes dermatoscópicas a las que se le puedan aplicar sistemas automáticos de diagnósticos es la asociación International Skin Imaging Collaboration (ISIC). Esta asociación está formada por la colaboración entre más de 30 centros académicos y empresas de todo el mundo [22].

El Memorial Sloan Kettering Cancer Center de Nueva York actúa como centro coordinador del proyecto, a la vez que cuenta con grupos de trabajo de temáticas específicas (tecnología, metadatos, inteligencia artificial, educación, privacidad, etc.) compuesto por médicos, investigadores clínicos, ingenieros e informáticos de la industria y la academia de todo el mundo. En España cuenta con investigadores del Hospital Clinic de la Universitat de Barcelona, el Hospital de la Santa Creu i Sant Pau de la Universitat Autònoma de Barcelona, la Universitat de Girona y la Fundació Clínic per a la Recerca Biomèdica de Barcelona.

El principal objetivo clínico de la ISIC es "apoyar los esfuerzos para reducir las muertes relacionadas con el melanoma y las biopsias innecesarias al mejorar la precisión y la eficiencia de la detección temprana del melanoma mediante la aplicación de imágenes digitales de la piel" [22]. Para promocionar el uso de imágenes digitales para la educación y el diagnóstico asistido por Inteligencia Artificial, la ISIC está creando recursos para las comunidades de Dermatología y Ciencia de Datos, mediante la implementación de un gran archivo de datos abierto con imágenes de la piel, metadatos y diagnósticos asociados.

ISIC ha patrocinado desde 2016 desafíos anuales para la comunidad de científicos de datos en asociación con las principales conferencias sobre visión artificial. Los desafíos han crecido en escala, complejidad y participación, utilizando datos de entrenamiento de alta calidad validados por expertos humanos y conjuntos de datos de miles de imágenes y metadatos. Algunos de estos desafíos se han centrado en la precisión diagnóstica para distinguir el melanoma de otras lesiones cutáneas.

El último desafío realizado hasta la fecha es el del año 2020, dedicado al análisis de lesiones cutáneas para la detección de melanomas. A este desafío se presentaron un total de 3.314 equipos [23]. Estudiando los métodos de clasificación empleados por los tres mejores equipos [24]–[26], así como los del mejor equipo del desafío de 2019 [27] se ha descubierto que son del tipo CNN EfficientNet. Teniendo en cuenta estos antecedentes, para el modelo de diagnóstico del sistema de diagnóstico automático desarrollado en este proyecto ha sido también CNN del tipo EfficientNet.

2.9. Detección aproximada del fototipo de piel

En las bases de datos de imágenes que se han comentado en el apartado anterior, pueden existir asociados a cada imagen una serie de metadatos. como el sexo los pacientes, la edad, la zona donde se encuentra la lesión u otra información sobre sus historiales clínicos. Sin embargo, el fototipo de piel, es decir, el tono de color, no se encuentra recogido en estos metadatos. Esta información puede ser importante a la hora de evaluar la equidad de un modelo de diagnóstico.

Una manera de detectar de manera aproximada el fototipo de piel en una imagen consiste en el cálculo del ángulo de tipología individual (ITA - *individual typology angle*) [28] de la piel sana en cada imagen. Para calcular el tono de piel aproximado se emplea la siguiente fórmula:

$$ITA = arctan\left(\frac{L - 50}{b}\right) \cdot \frac{180}{\pi}$$

Donde *L* y *b* se obtienen al convertir los valores RGB de cada píxel al espacio de color CIELAB [29].

Se puede establecer una relación aproximada entre el valor del ITA y la clasificación de Fitzpatrick [28], [30] de fototipos de piel humana:

Fototipo	Descripción	ITA
I	Tienden a quemarse con facilidad con la exposición a sol, personas rubias o pelirrojas de piel muy clara	ITA > 55
II	Se broncean algunas veces, pero se queman con facilidad con la exposición al sol. Personas con pelo rubio y pelirrojas con pecas.	55 ≥ ITA > 41
III	Se brocean con facilidad, pudiéndose quemar de manera moderada con la exposición al sol. Personas con pelo castaño claro y ojos verdes o marrones.	41 ≥ ITA > 28
IV	Nunca se queman con la exposición solar y se broncean. Personas con pelo castaño oscuro y ojos de color marrón.	28 ≥ ITA > 19
V	Nunca se queman, bronceado fácil. Personas con piel moderadamente pigmentada, ojos oscuros y pelo negro.	19 ≥ ITA > 10
VI	Nunca se queman. Personas de raza negra.	10 ≥ ITA

Tabla 4: Relación aproximada entre clasificación Fitzpatrick y valor ITA.
Fuente: [28], [30].

Para todos los tonos de piel, las lesiones dermatológicas suelen tener un color más oscuro que la piel circundante, por lo que un método para el cálculo del ITA consiste en "tomar parches de piel no enfermos, como por ejemplo 8 muestras de 20x20 píxeles de alrededor de los bordes de cada imagen y use la muestra con el valor de ITA más alto (tono de piel más claro) como el tono de piel estimado" [28]. Antes de tomar los parches, es conveniente realizar un filtro de la imagen de manera que se elimine el pelo.

Un método alternativo consiste en realizar la segmentación de la imagen y descartar la zona de la lesión, sin embargo, el método de los parches descrito reduce el impacto de las condiciones de iluminación variable, seleccionando la muestra más clara en lugar de la totalidad de la piel sana. Asimismo, el método de los parches es más sencillo de implementar [28].

3. Diseño del sistema

3.1. Modelo de diagnóstico

Para el modelo de diagnóstico se ha empleado tecnología de Inteligencia Artificial con aprendizaje automático supervisado. Esta técnica consiste en crear una función de predicción a partir de un conjunto de datos de ejemplo, llamados de entrenamiento, ya etiquetados con la predicción correcta, donde el etiquetaje de los datos de entrenamiento en general es realizado por expertos humanos en la materia. Una vez creada la función de predicción, el sistema debe ser capaz de predecir el valor correspondiente a datos no vistos antes, por lo que el sistema debe tener la capacidad de generalizar a partir de los ejemplos presentados durante el entrenamiento.

3.1.1. Plataforma Google Cloud

La plataforma de Google Cloud se ha empleado en este proyecto para la realización de experimentos con el objetivo de obtener un modelo de diagnóstico óptimo.

Google Colab o Colaboratory es "un producto de Google Research disponible gratuitamente o con coste según la potencia computacional necesaria para que cualquier usuario pueda desarrollar y ejecutar código en Python directamente desde un navegador web" [31]. Su funcionamiento se basa en cuadernos de Jupyter desde donde se ofrece accesos a recursos informáticos en Google Cloud.

La gran ventaja que aporta utilizar Google Colab consiste en la disponibilidad de *Cloud Tensor Processing Units* (TPU) [32]. Estas unidades de procesamiento de tensores son circuitos integrados del tipo ASIC (*Application Specific Integrated Circuit*) desarrollados a medida por Google que se utilizan para acelerar de manera muy significativa las cargas de trabajo en el entrenamiento de modelos de aprendizaje automático.

La API de TensorFlow está diseñada para obtener el máximo rendimiento de las TPU. Como contrapartida, el flujo de datos hacía las TPU solo se puede realizar desde una fuente de datos almacenada en Google Cloud, no estando permitida la ingestión de datos desde un repositorio local.

3.1.2. Plataforma AWS Sagemaker

Para la implementación del modelo de diagnóstico se ha empleado la plataforma en la nube AWS SageMaker, que "permite la implementación integral (*end-to-end*) de proyectos de Ciencia de Datos desde su fase de desarrollo a la implementación en producción" [33]. Esta plataforma permite automatizar y estandarizar prácticas de integración continua e implementación rápida y repetible de modelos de inteligencia artificial en una organización de manera que se pueden crear, entrenar, implementar y administrar estos modelos a cualquier escala.

Las fases típicas de del proyecto de ciencia de datos con SageMaker son [33]:

- Acceso a los datos: permite la conexión a una gran variedad de fuentes de datos almacenados en S3, Apache Spark, Amazon Redshift, etc.
- Preparación de los datos: se preparan datos estructurados y no estructurados a gran escala para lograr la máxima calidad del modelo. Permite transformar datos explorando fuentes de datos, metadatos, esquemas y realizar consultas sobre los datos para extraer los que se deseen.

- Construcción de modelos de aprendizaje automático: se proporcionan herramientas y bibliotecas para crear modelos de aprendizaje automático, facilita el proceso de probar de forma iterativa diferentes algoritmos y evaluar su precisión para encontrar el mejor modelo.
- Entrenamiento y ajuste de los modelos de aprendizaje automático: reduce el tiempo y el costo de entrenar y ajustar modelos a escala sin necesidad de administrar la infraestructura subyacente.
- Despliegue de modelos para realizar predicciones: facilita la implementación de modelos de aprendizaje automático para realizar predicciones.
- Monitorización: permite mantener la calidad al detectar las desviaciones del modelo en tiempo real de los modelos desplegados.

Con la plataforma de SageMaker es posible seguir prácticas de integración y desarrollo continuo (CI/CD) aplicadas al aprendizaje automático y a las prácticas de MLOps, como son mantener la equivalencia entre los entornos de desarrollo y producción, el control de versiones y orígenes, las pruebas A/B y la automatización integral.

3.2. Aplicación *web* de diagnóstico

La aplicación *web* sirve para realizar diagnósticos a partir de nuevas imágenes. Se trata de una aplicación con *front-end* realizado en JavaScript nativo y *back-end* en NodeJS.

La aplicación consta de dos páginas:

Página de diagnóstico: se muestra el diagnóstico realizado por el modelo sobre una imagen introducida por el usuario. Se muestra un valor cuantitativo entre 0 y 100, un valor cualitativo y un código de color.

Figura 10: Aplicación *web* – página de diagnóstico.
Fuente: elaboración propia.

La correspondencia de estos valores respecto a la probabilidad (valores entre 0 y 1) proporcionada por el modelo es la siguiente:

Probabilidad	Valor cuantitativo	Valor cualitativo	Color
[0 – 0.25)	[0 – 25)	BAJO	
[0.25 – 0.50)	[25 – 50)	MEDIO	
[0.50 – 0.75)	[50 – 75)	ALTO	
[0.75 – 1]	[75 – 100]	MUY ALTO	

Tabla 5: Correspondencia entre probabilidad y valores de la aplicación *web*.
Fuente: elaboración propia.

La página de información contiene una breve descripción del proyecto, a la vez que se encuentra un enlace a un fichero con una serie de imágenes seleccionadas aleatoriamente de entre el conjunto de pruebas, es decir, no vistas antes por el modelo de diagnóstico durante el entrenamiento.

Figura 11: Aplicación *web* – página de información.
Fuente: elaboración propia.

Nota: La aplicación *web* esta activada en la siguiente URL:

https://d39fz5pvr27vh5.cloudfront.net

3.3. Integración de la aplicación *web* y el modelo de diagnóstico

AWS SageMaker proporciona un punto de enlace *(endpoint)* de inferencia mediante una API de tipo REST que permite su integración con otras aplicaciones, de forma general a través de un API *gateway*. Un modelo típico de integración [34] sugerido por AWS presenta la siguiente arquitectura:

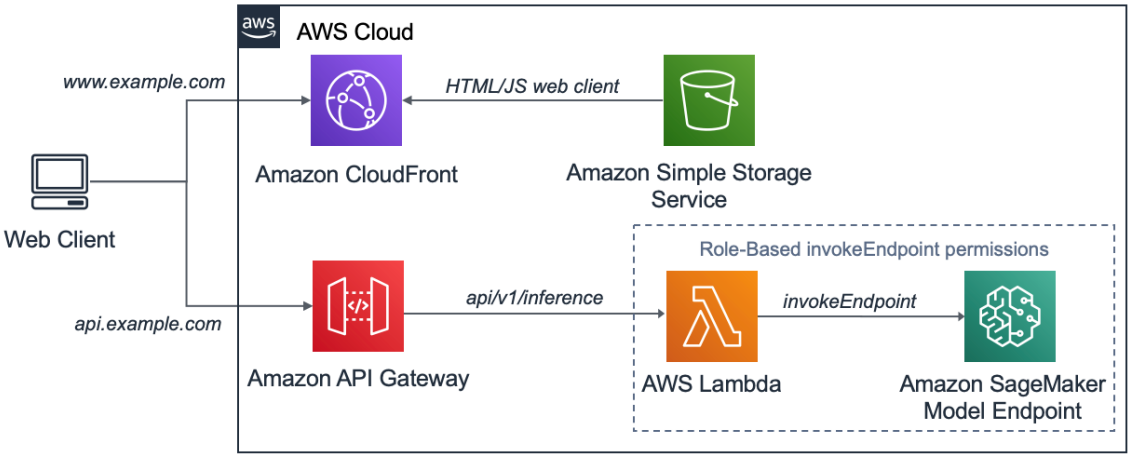


Figura 12: Diseño de la integración.
Fuente: [34].

Los componentes de la integración son los siguientes:

- *Cloud Front*: servicio de entrega de contenido (CDN).
- *Simple Storage Service* (S3): sistema de almacenamiento en la nube donde se aloja la aplicación web.
- API *Gateway*: gestor de tráfico entre la aplicación web y el *endpoint* de inferencia de SageMaker.
- Función Lambda: efectúa las llamadas hacia/desde el punto de enlace. Las funciones lambda son ejecutadas por la infraestructura de AWS sin necesidad de asignar una máquina virtual por parte del usuario.
- SageMaker Model Endpoint: punto de enlace para introducir datos y obtener respuestas del modelo de diagnóstico puesto en producción.

4. Desarrollo del sistema

4.1. Preparación del entorno de desarrollo

Para este proyecto se han combinado las capacidades de los entornos de desarrollo de Google Colab y de AWS SageMaker. Originalmente, el planteamiento era realizar todas las tareas de desarrollo en la plataforma AWS SageMaker, no obstante, debido a carencias de potencia computacional en el *hardware* puesto a disposición por AWS en su *free tier* gratuito o de bajo coste accesible a usuarios particulares, se ha optado por desarrollar las tareas de experimentos y entrenamiento en la plataforma de Google Colab.

4.1.1. Preparación del entorno de desarrollo de Google Colab

Para preparar el entorno de entrenamiento, se creó una cuenta de Google Colab Pro que permite un mayor uso de recursos de computación. Una vez creada la cuenta, se accede a https://colab.research.google.com/ donde se presenta una interfaz *web* para crear o subir cuadernos de Python al entorno de desarrollo de Jupyter. Por otra parte, el entorno está integrado con Google Drive de manera que se puede acceder al almacenamiento disponible en la nube a través de este servicio.

4.1.2. Preparación del entorno de desarrollo de AWS SageMaker

Para este proyecto se ha utilizado el entorno de desarrollo SageMaker Studio. Se trata de entorno de desarrollo integrado (*integrated development environment* - IDE) en entorno *web* para el aprendizaje automático que permite crear, entrenar, depurar, implementar y monitorear modelos.

En una sola interfaz visual unificada, se pueden realizar las siguientes tareas:

- Desarrollar y ejecutar código en cuadernos Jupyter.
- Preparar datos para modelos de aprendizaje automático.
- Crear y entrenar modelos de aprendizaje automático.
- Implementar los modelos y supervisar el rendimiento de sus predicciones.

Para preparar el sistema de desarrollo en la plataforma SageMaker se siguieron las instrucciones de *Amazon SageMaker Developer Guide* [35]. Los pasos seguidos se detallan en *10.4. Anexo IV: Preparación del entorno de desarrollo en Amazon Sagemaker.*

Al ser un entorno colaborativo de trabajo se pueden dar de alta tantos usuarios como sean necesarios para desarrollar el proyecto. En el caso concreto de este proyecto solo se registró un usuario al ser un proyecto individual.

4.2. Repositorio de código fuente

Se ha creado un repositorio para el código fuente empleado en:
https://github.com/zardemostoles/zardemostoles-uoc.edu

El orden de ejecución y el propósito de cada elemento son los siguientes:

Orden	Elemento	Descripción
1	TFG_Comun.py	Variables comunes del proyecto.
2	TFG_Calculo_Tipo_Tono_Piel_ITA.ipynb	Cálculo aproximado del fototipo.
3	TFG_EDA.ipynb	Análisis exploratorio de datos.
4	TFG_Modelo_Diagnostico.ipynb	Experimento y creación del modelo de diagnóstico óptimo.
5	TFG_Evaluacion.ipynb	Evaluación del rendimiento y equidad del modelo.
6	TFG_Crear_Modelo_AWS.ipynb	Preparación del modelo en formato apropiado para AWS.
7	TFG_Despliegue_Modelo_AWS.ipynb	Despliegue del modelo en AWS con su correspondiente punto de enlace.
8	inference.py	Programa para convertir imágenes a tensores utilizado por el modelo desplegado.
9	requirements.txt	Módulos que necesita instalar el modelo desplegado en AWS.
10	TFG_Aplicacion_Web.tar.gz	Aplicación *web* de diagnóstico.

Tabla 6: Elementos del repositorio de código fuente.
Fuente: elaboración propia.

4.3. Modelo de diagnóstico

En este apartado se describen las distintas fases que fueron necesarias para la realización del modelo de diagnóstico de riesgo de melanoma basado en técnicas de aprendizaje automático.

4.3.1. Análisis de los conjuntos de datos de la base de datos de ISIC

Como se ha descrito con anterioridad, los conjuntos de datos de ISIC se han convertido en un repositorio de referencia para los investigadores en aprendizaje automático para el análisis de imágenes médicas aplicadas al campo de la detección del cáncer de piel y la evaluación de malignidad.

ISIC ha patrocinado desde 2016 desafíos anuales para la comunidad de científicos de datos en asociación con las principales conferencias sobre visión artificial. Los desafíos han crecido en escala, complejidad y participación, utilizando datos de entrenamiento de alta calidad validados por humanos y conjuntos de datos con miles de imágenes y metadatos.

La evolución de las imágenes y datos suministrados en los desafíos es la siguiente [36]:

Año	Descripción	Entreno	Prueba
2016 [37]	Etiquetados tanto para los datos de entrenamiento como de prueba, indicando si la lesión es benigna o maligna.	900	379
2017 [38]	Etiquetados tanto para los datos de entrenamiento como de prueba, con 4 categorías de diagnóstico. Se incluyen metadatos con la edad aproximada y sexo.	2.000	600
2018[39]	Etiquetados los datos de entrenamiento, con 7 categorías de diagnóstico. Se incluyen metadatos con la edad aproximada y sexo.	10.015	1.512
2019 [40]–[42]	Etiquetados los datos de entrenamiento, con 8 categorías de diagnóstico. Contiene múltiples imágenes de la misma lesión con distintos detalles de acercamiento. Se incluyen metadatos con la edad aproximada, sexo y lugar anatómico de la lesión.	25.331	238
2020 [43]	Etiquetados los datos de entrenamiento, con 8 categorías de diagnóstico. Contiene múltiples imágenes del mismo paciente con distintos detalles de acercamiento. Se incluyen metadatos con el identificador de paciente, edad aproximada, sexo y lugar de la lesión.	33.126	10.982

Tabla 7: Datos e imágenes de los desafíos ISIC.
Fuente: [36].

.

La distribución de las etiquetas de diagnósticos de los 4 conjuntos de datos que presentan categorías de lesión es la siguiente:

Diagnóstico	2017	2018	2019	2020
Melanoma	374 (18.7%)	1.113 (11.1%)	4.522 (17.8%)	584 (1.8%)
Proliferación melanocítica atípica				1
Mácula café con leche				1
Lentigo				44
Queratosis liquenoide				37
Nevus				5.193
Queratosis seborreica	254			135
Lentigo solar				7
Nevus melanocíticos		6.705	12.875	
Carcinoma de células basales		514	3.323	
Queratosis actínica		327	867	
Queratosis benigna		1.099	2.624	
Dermatofibroma		115	239	
Lesión vascular		142	253	
Carcinoma espinocelular			628	
Otros / desconocido	1.372			27.124
Total	2.000	10.015	25.331	33.126

Tabla 8: Distribución de diagnósticos en datos de los desafíos ISIC.
Fuente: [36].

Se muestran los porcentajes de diagnóstico de melanoma sobre el número total de imágenes de cada año. En el año 2020, los casos con diagnóstico desconocido se consideran benignos.

Teniendo en cuenta el número de casos diagnosticados como melanoma y el volumen total de imágenes, se pueden realizar las siguientes consideraciones:

- Año 2017: el número total de imágenes es muy reducido.
- Año 2018: el número total de imágenes en mayor que el año anterior y el porcentaje de casos de melanoma es del 11%.
- Año 2019: el número total de imágenes en mayor en años anteriores y el porcentaje de casos de melanoma es de casi el 18%. Se debe tener en cuenta que contiene múltiples imágenes de la misma lesión con distintos detalles de acercamiento de la misma lesión.
- Año 2020: es el año que presenta un número total de imágenes, sin embargo, la mayoría de ellas no tiene diagnóstico y el porcentaje de casos de melanoma es muy bajo. Se debe considerar que contiene múltiples imágenes del mismo paciente (2.056 identificadores de paciente únicos) con distintos detalles de acercamiento.

El conjunto de datos del año 2019 es, en principio, el que aparece como más apropiado como datos de entrada para realizar el aprendizaje y evaluación del modelo de diagnóstico. Hay que tener en cuenta que presenta imágenes de la misma lesión que se deben tratar de manera adecuada.

4.3.2. Análisis del conjunto de datos del desafío ISIC del año 2019

En este apartado se analiza el fichero de metadatos correspondiente al conjunto de datos del desafío de ISIC del año 2019. Se trata de una tabla en la que cada registro se corresponde con una imagen del conjunto de datos. Se compone de los siguientes campos:

- *image_name*: nombre del fichero que contiene la imagen
- *patient_id:* identificador del paciente. Todos los registros presentan valor -1.
- *sex*: sexo del/la paciente.
- *age_aprox*: edad aproximada del/la paciente.
- *anatom_site_general_challenge*: zona anatómica donde se encuentra la lesión.
- *diagnosis:* diagnóstico de la enfermedad.
- *target:* 1 si el diagnóstico es de melanoma, 0 si en caso contrario.

En este fichero de metadatos no se encuentra información sobre el fototipo de piel de los pacientes, por lo que se realiza un cálculo aproximado de su ITA, como se explicó en *2.9. Detección aproximada del fototipo de piel*, y se agrupa según la clasificación de seis tipos de Fitzpatrick.

4.3.2.1. Análisis exploratorio de los metadatos

Como se ha comentado con anterioridad, existen imágenes que se tratan de la misma lesión con distinto grado de acercamiento (prefijo *_downsampled*):

Número total de registros	Número de registros *_downsampled*	Número total de registros de distintas lesiones
25.331	2.074 (8.2%)	23.257

Tabla 9: Distribución de imágenes ISIC 2019.
Fuente: elaboración propia.

Para el entrenamiento y validación del modelo, en principio no interesa que existan fotos de la misma lesión debido a que puede presentarse una fuga de información entre los datos de entrenamiento y de validación. Así pues, se eliminan las imágenes que presenten esta característica (*_downsampled*). Se obtiene la siguiente distribución de casos de melanoma:

Número total de registros	No melanoma	Melanoma
23.257	19.109 (82.2%)	4.148 (17.8%)

Tabla 10: Distribución de diagnósticos ISIC 2019.
Fuente: elaboración propia.

Se aprecia que existe un fuerte desbalanceo entre las dos clases (no melanoma / melanoma), lo que se tuvo en cuenta en el diseño del modelo de diagnóstico.

La variable edad aproximada se ha discretizado en grupos de 15 años cada uno.

Con la variable edad aproximada discretizada y sin considerar la zona anatómica, la distribución del número total de las imágenes respecto a las distintas variables categóricas y su diagnóstico es la siguiente:

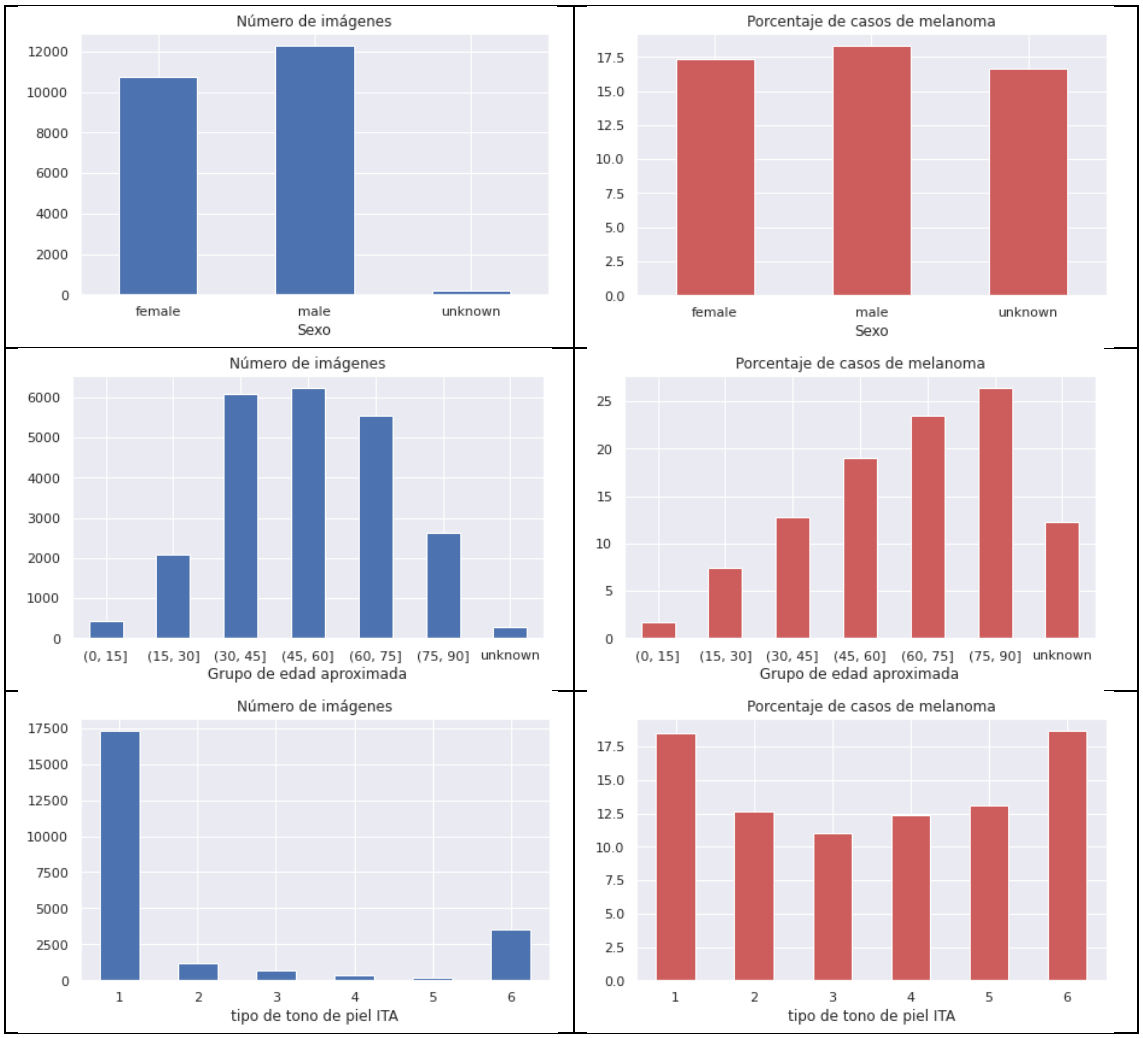


Figura 13: Distribución del número total de imágenes.
Fuente: elaboración propia.

Se pueden realizar las siguientes observaciones:

- Variable sexo: el porcentaje de imágenes pertenecientes a pacientes del sexo masculino y femenino son similares. Los porcentajes de casos de melanoma también son similares.
- Variable edad aproximada: la mayoría de pacientes se encuentra entre los 30 y los 75 años, creciendo la incidencia de melanoma según aumenta la edad. La franja de edad de 0 a 15 años está casi sin representación y su porcentaje de casos de melanoma es mínimo.
- El fototipo de piel calculado es una aproximación poco precisa, ya que casi el 75% se encuentra en el fototipo de piel muy clara. Sin embargo, el tipo VI está sobrerepresentado, debido probablemente a la existencia de muchas imágenes con bordes de color oscuro. Estos datos están en consonancia con el estudio que se ha tomado como referencia [28].

4.3.3. Preparación de los datos de entrenamiento, validación y prueba

Para poder realizar de manera más eficiente la entrada de imágenes de entrenamiento, validación y prueba para el modelo de diagnóstico se utilizó el formato TFRecord. Este formato presenta importantes beneficios en el rendimiento del mecanismo que se utilizó para el entrenamiento del modelo [44]:

- Almacenamiento más eficiente: los datos almacenados como TFRecord pueden ocupar menos espacio que los datos originales y se pueden agrupar por archivos. Esto supone un importante beneficio ya que se trabajó con unas decenas de archivos en lugar de más de 25.000 ficheros de imágenes.
- E/S más rápida: este formato se puede leer con operaciones de E/S paralelas, lo cual es útil para TPU o múltiples hosts que se utilizaron en el entrenamiento del modelo de diagnóstico.
- Archivos autónomos: los datos almacenados en este formato se pudieron leer desde una sola fuente y no fue necesario crear otros ficheros o subdirectorios.

Una preparación de datos ampliamente utilizada con formato TFRecord es la realizada por Chris Deotte [45] con los conjuntos de datos de ISIC del desafío de 2019, que se encuentra disponible en la plataforma Kaggle. Se dispone de ficheros con el conjunto total de las imágenes con conversión uniforme a varias resoluciones. Además de las ventajas mencionadas sobre los ficheros en este formato, este conjunto de datos presenta otras ventajas adicionales:

- Los tamaños de las imágenes son homogéneos para todo el conjunto de datos, de forma cuadrada tomando la parte central si es necesario recortar. Este tamaño de imágenes homogénea favorece el rendimiento en el entrenamiento en las CNN.
- Los ficheros TFRecord se encuentran ya disponibles en la nube de Google. Este es un requisito indispensable si se quiere utilizar *hardware* TPU en la plataforma Google Colab.

Para el entrenamiento y evaluación del modelo de diagnóstico se utilizaron los siguientes conjuntos de ficheros:

Conjunto de ficheros	Resolución	Canales de color RGB	Número de ficheros	Tamaño total
isic2019-256x256	256x256	3	30	441 MB
isic2019-384x384	384x384	3	30	891 MB
isic2019-512x512	512x512	3	30	1 GB
isic2019-768x768	768x768	3	30	3 GB

Tabla 11: Ficheros con imágenes ISIC 2019.
Fuente: elaboración propia.

4.3.4. Carga de datos para el entrenamiento y prueba del modelo

Una vez que se consiguieron los datos, el siguiente paso consistió en la carga de estos en el entorno de desarrollo. Por motivos prácticos y económicos fue necesario realizar experimentos de configuración del modelo en un entorno de desarrollo de Google Colab que permite el uso de hardware TPU que acelera de manera muy significativa el proceso de entrenamiento de múltiples configuraciones de modelos de diagnóstico. Una vez seleccionado el modelo óptimo, era posible realizar más experimentos en el sistema de desarrollo de AWS SageMaker pero se decidió transferir el modelo creado a AWS y realizar su implantación en producción.

En primer lugar, se crearon 4 *datasets* de TensorFlow, descartando los registros con imágenes que contuviesen "_downsampled" en su nombre de fichero.

Para comprobar que realizó la carga de forma correcta, se mostraron 20 ejemplos de cada clase de diagnóstico (melanoma, no melanoma)

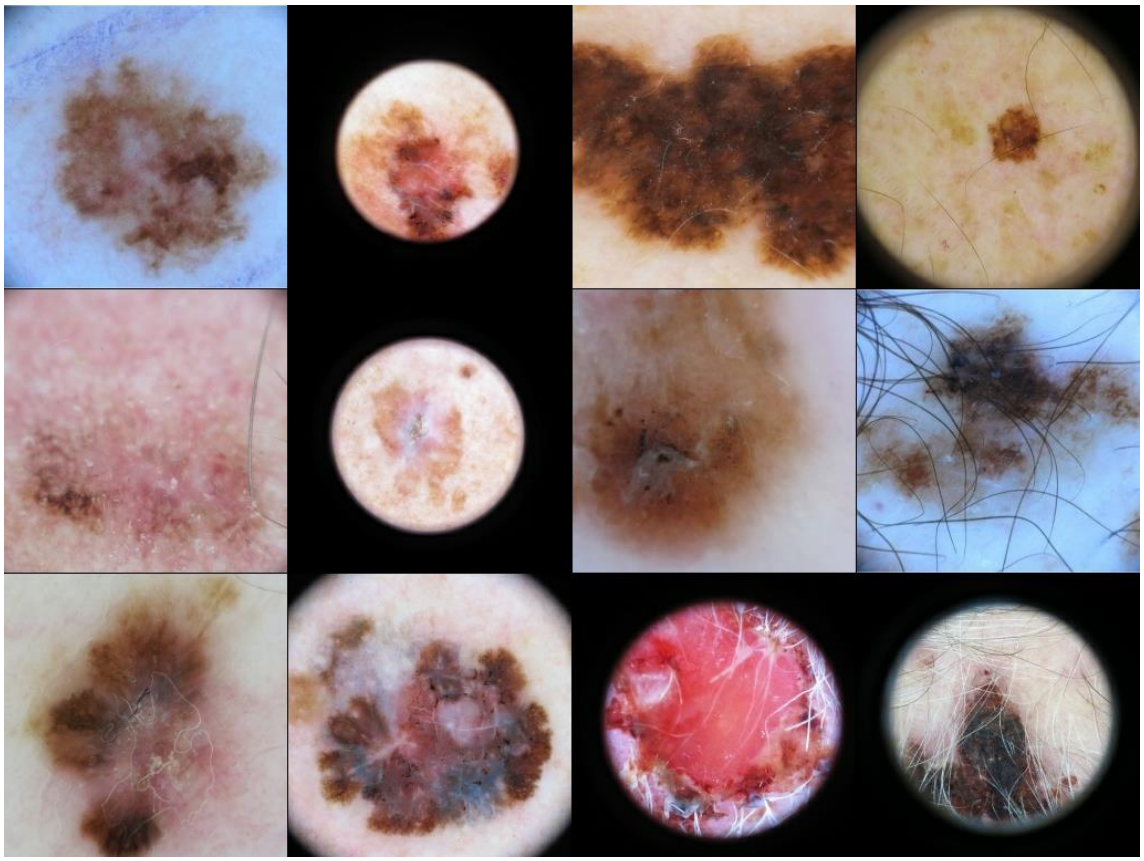


Figura 14: Ejemplos de imágenes de ISIC con diagnóstico de melanoma.
Fuente: elaboración propia.

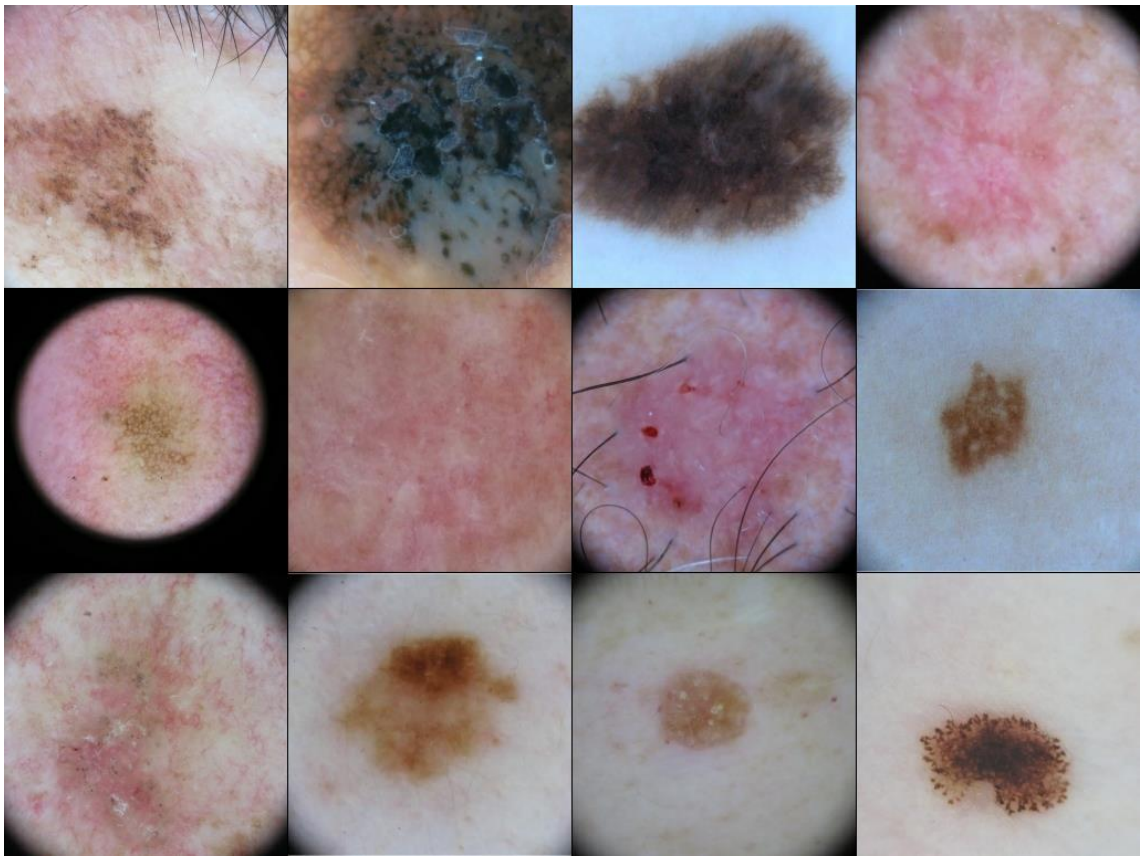


Figura 15: Ejemplos de imágenes de ISIC con diagnóstico distinto a melanoma.
Fuente: elaboración propia

Se comprobó a continuación el número total de imágenes preparadas en los *datasets* y la distribución por diagnóstico.

Número total de registros	No melanoma	Melanoma
23.218	19.076 (82.2%)	4.142 (17.8%)

Tabla 12: Distribución de diagnósticos.
Fuente: elaboración propia.

Se apreció que en la conversión de las imágenes a ficheros TFRecord se había producido la pérdida de 39 imágenes (33 con diagnóstico no melanoma y 6 con diagnóstico melanoma). Al ser una proporción insignificante de perdida, sin variación en la distribución, se continuó con los registros ya preparados y cargados en el sistema de desarrollo.

El siguiente paso consistió en extraer un conjunto de datos de prueba que sirvieron para evaluar el modelo final. Es importante destacar que este conjunto de datos no fue utilizado en ningún momento en el entrenamiento del modelo, ya que de esta manera se pudieron utilizar para evaluar predicciones realizadas por el modelo sobre datos no vistos antes.

El porcentaje de datos que se dedicaron para la prueba fue el 15%, que es un porcentaje en general adecuado para este volumen de datos. De esta manera, para cada uno de los conjuntos de datos de las 4 resoluciones se realizó una partición aleatoria estratificada que conservó la distribución de los diagnósticos. Al conjunto restante se le denominó *experimentos*.

Conjunto de datos	Número total de registros	No melanoma	Melanoma
experimentos	19.735	16.214 (82.2%)	3.521 (17.8%)
pruebas	3.483	2.862 (82.2%)	621 (17.8%)

Tabla 13: Distribución de diagnósticos en datos de experimentos y pruebas.
Fuente: elaboración propia.

4.3.5. Experimentos para configurar el modelo de diagnóstico óptimo

Para poder realizar una validación durante el periodo de entrenamiento, es habitual dividir el conjunto de datos disponibles en conjunto de datos de entrenamiento y conjunto de datos de validación. Los datos de validación sirven para comprobar el rendimiento del modelo y ajustar sus parámetros durante el entrenamiento. Cuando hay escasez de datos disponibles, se recomienda realizar un proceso de validación cruzada donde se combinan distintas divisiones de datos de entrenamiento y validación y se recomiendo también utilizar medidas estadísticas que aseguren que los resultados obtenidos son independientes a la división realizada.

Por razones de volumen, para realizar los experimentos para configurar el modelo de diagnóstico óptimo no se consideró utilizar validación cruzada, sino que se seleccionó una partición aleatoria estratificada que conservó la distribución de los diagnósticos, de manera similar a como se seleccionó el conjunto de datos de prueba. Se tomó el 15% del conjunto de datos *experimentos* para realizar la validación de los distintos modelos que se experimentaron. Las dos particiones resultantes se denominaron *entrenamiento* y *validación*.

Conjunto de datos	Núm. total de registros	No melanoma	Melanoma
entrenamiento	16.774	13.781 (82.2%)	2.993 (17.8%)
validación	2.961	2.433 (82.2%)	528 (17.8%)

Tabla 14: Distribución de diagnósticos en datos de entrenamiento y de validación.
Fuente: elaboración propia.

Se realizaron una serie de experimentos para evaluar distintas configuraciones del entrenamiento de CNN de tipo EfficientNet. Se comenzó con una configuración básica de EfficientNet-B0 con transferencia de conocimiento de ImageNet, donde en la última capa se situó un clasificador binario que se entrenó con los datos del conjunto de datos *experimentos*. Todos los experimentos se realizaron durante 15 épocas.

En primer lugar, se realizaron experimentos con distintos valores de *learning rate*. Los resultados de las métricas de rendimiento obtenidas para el conjunto de datos *validación* fueron los siguientes:

Learning rate	AUC ROC	AUC PR	Precisión	Sensibilidad	F1
0.00500	0.830	0.582	0.413	0.716	0.524
0.00100	0.892	0.751	0.728	0.708	0.718
0.00050	0.912	0.794	0.707	0.718	0.712
0.00010	0.921	0.809	0.767	0.693	0.728
0.00005	0.916	0.802	0.778	0.652	0.709

Tabla 15: Métricas de rendimiento variando el *learning rate*.
Fuente: elaboración propia.

Se aprecia que disminuyendo el valor del *learning rate* mejoraron en general las métricas. Los dos valores más bajos no presentaron grandes diferencias.

Como las categorías se encuentran desbalanceadas se aplicaron técnicas que mejorasen el rendimiento del modelo de clasificación. Se empleó, tomando los tres valores menores de *learning rate*, la técnica de utilizar el sesgo inicial en la CNN, lo que puede ayudar a la convergencia [46].

Learning rate	Sesgo inicial	AUC ROC	AUC PR	Precisión	Sensibilidad	F1
0.00050	Sí	0.908	0.796	0.795	0.718	0.755
0.00010	Sí	0.921	0.797	0.742	0.742	0.742
0.00005	Sí	0.930	0.809	0.710	0.731	0.720

Tabla 16: Métricas de rendimiento aplicando sesgo inicial.
Fuente: elaboración propia.

Efectivamente, se apreció una ligera mejora en las métricas respecto a los experimentos donde no se había aplicado el sesgo inicial.

Un método para prevenir sobreajuste en CNN es la incorporación de una capa de *dropout*. Esta capa tiene un parámetro para indicar el ratio de entradas que se desactivan. Dejando fijo *el learning rate* al menor valor y con sesgo inicial, se realizaron experimentos con distintos valores de *dropout rate.*

Dropout rate	*Learning rate*	Sesgo inicial	AUC ROC	AUC PR	Precisión	Sensibilidad	F!
0.2	0.00005	Sí	0.924	0.805	0.820	0.636	0.716
0.3	0.00005	Sí	0.924	0.796	0.767	0.659	0.709
0.4	0.00005	Sí	0.934	0.813	0.682	0.754	0.716
0.5	0.00005	Sí	0.930	0.821	0.837	0.631	0.720

Tabla 17: Métricas de rendimiento variando el *dropout rate*.
Fuente: elaboración propia.

De nuevo se apreció una ligera mejora en las métricas respecto a los experimentos anteriores, tomando como *dropout rate* 0.4 o 0.5. Para los siguientes experimento se tomó el valor de *dropout rate* = 0.4.

Hasta ese momento se había utilizado el modelo EfficientNetB0 y la resolución de imágenes de 256x256. A continuación, dejando fijos los parámetros de *drop connect rate, learning rate* y sesgo inicial, se realizaron pruebas con distintas resoluciones de imágenes y el resto de los modelos de EfficientNet. Para ello, se consideraron las resoluciones óptimas para la que están configuradas los modelos de EfficientNet [47]:

Modelo	Resolución
EfficientNetB0	224x224
EfficientNetB1	240x240
EfficientNetB2	260x260
EfficientNetB3	300x300
EfficientNetB4	380x380
EfficientNetB5	456x456
EfficientNetB6	528x528
EfficientNetB7	600x600

Tabla 18: Resoluciones óptimas para los modelos EfficientNet.
Fuente: [47].

Se realizaron experimentos con distintas resoluciones, dejando fijos los parámetros *dropout rate=0.4; learning rate*=0.00005; sesgo inicial = Sí

Modelo base	Res. imágenes	AUC ROC	AUC PR	Precisión	Sensibilidad	F1
EfficientNetB1	384x384	0.939	0.815	0.760	0.744	0.752
EfficientNetB2	384x384	0.926	0.814	0.753	0.735	0.744
EfficientNetB3	384x384	0.941	0.838	0.721	0.824	0.769
EfficientNetB4	384x384	0.945	0.853	0.797	0.788	0.792
EfficientNetB5	512x512	0.950	0.868	0.822	0.820	0.821
EfficientNetB6	512x512	0.954	0.879	0.773	0.833	0.802
EfficientNetB7	512x512	n/d	n/d	n/d	n/d	n/d

Tabla 19: Métricas de rendimiento variando el modelo de EfficientNet.
Fuente: elaboración propia.

Nota: el entrenamiento con EfficientNetB7 no se pudo realizar por fallos en el entorno de desarrollo de falta de memoria.

Se aprecia que los modelos con las versiones superiores B5 y B6 mostraron las mejores métricas. Se consideró para el modelo óptimo EfficientNet-B5, al necesitar menos recursos computacionales.

4.3.6. Entrenamiento del modelo de diagnóstico

Con la configuración óptima conseguida en el apartado anterior:

- Modelo: EfficientNetB5
- Resolución de imágenes: 512x512
- *drop connect rate:* 0.4
- learning rate: 0.00005
- sesgo inicial: Sí

se realizó el entrenamiento del modelo con dicha configuración, almacenando el modelo que generó el valor superior de la métrica de AUC de la curva de precisión-sensibilidad.

Las métricas de rendimiento obtenidas sobre el conjunto de datos de validación se resumen en la siguiente tabla:

Época	AUC ROC	AUC PR	Precisión	Sensibilidad	F1
15	0.947	0.880	0.829	0.805	0.817

Tabla 20: Métricas de rendimiento del modelo óptimo.
Fuente: elaboración propia.

Las gráficas de las métricas de rendimiento sobre el conjunto de datos de entrenamiento y el conjunto de datos de validación respecto a las épocas fueron:

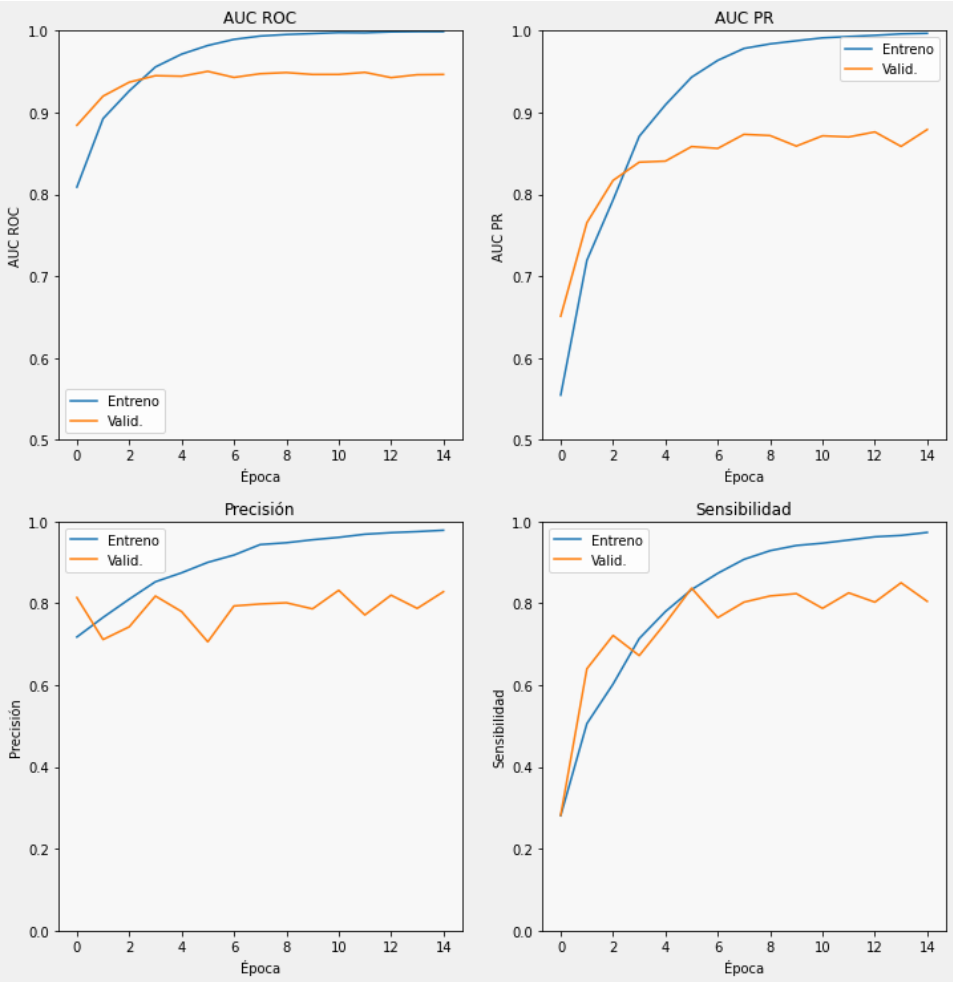


Figura 16: Gráficas de métricas durante el entrenamiento.
Fuente: elaboración propia.

Se aprecia que las métricas en ambos conjuntos de datos fueron estables, y se puede considerar que a partir de la décima época la mejora en el conjunto de validación fue inapreciable.

4.3.7. Evaluación del rendimiento del modelo de diagnóstico

Una vez que se creó y entrenó el modelo, el siguiente paso fue evaluar su rendimiento en datos que no había visto con anterioridad, es decir, los datos de prueba que se reservaron desde un primer momento con esta finalidad.

Considerando los casos de diagnóstico de melanoma como positivo, se obtuvieron los siguientes resultados:

Verdadero negativo	Falso Positivo	Falso negativo	Verdadero positivo
2.770	92	121	500

Tabla 21: Resultados del modelo optimo en los datos de prueba.
Fuente: elaboración propia.

Estos resultados corresponden a la siguiente matriz de confusión:

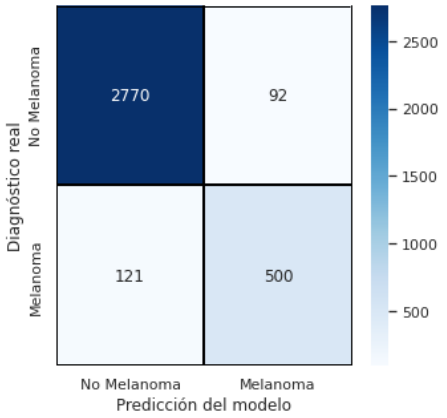


Figura 17: Matriz de confusión sobre el conjunto de datos de prueba.
Fuente: elaboración propia.

Las métricas de rendimiento que se obtuvieron fueron las siguientes:

AUC ROC	AUC PR	Precisión	Sensibilidad	F1
0.961	0.886	0.845	0.805	0.824

Figura 18: Métricas de rendimiento del modelo sobre los datos de prueba.
Fuente: elaboración propia.

Sobre los datos de prueba, el modelo proporcionó unos valores de las métricas similares a los obtenidos durante la fase de entrenamiento. Estos datos sugieren que el modelo es capaz de generalizar la resolución del problema planteado de manera correcta.

4.3.8. Evaluación de la equidad del modelo de diagnóstico

A continuación, se analizan los valores para las métricas de precisión, sensibilidad, tasa de falsa omisión y tasas de falsos negativos obtenidas sobre los datos de prueba para las distintas categorías de las variables categóricas.

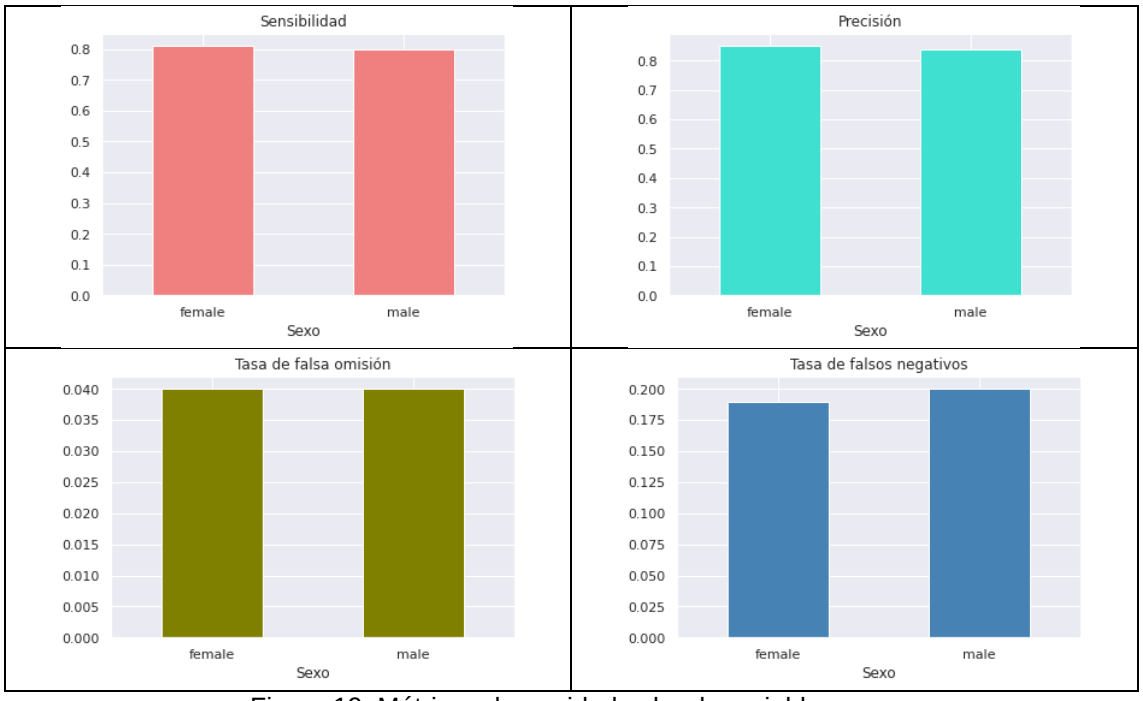


Figura 19: Métricas de equidad sobre la variable *sexo*.
Fuente: elaboración propia.

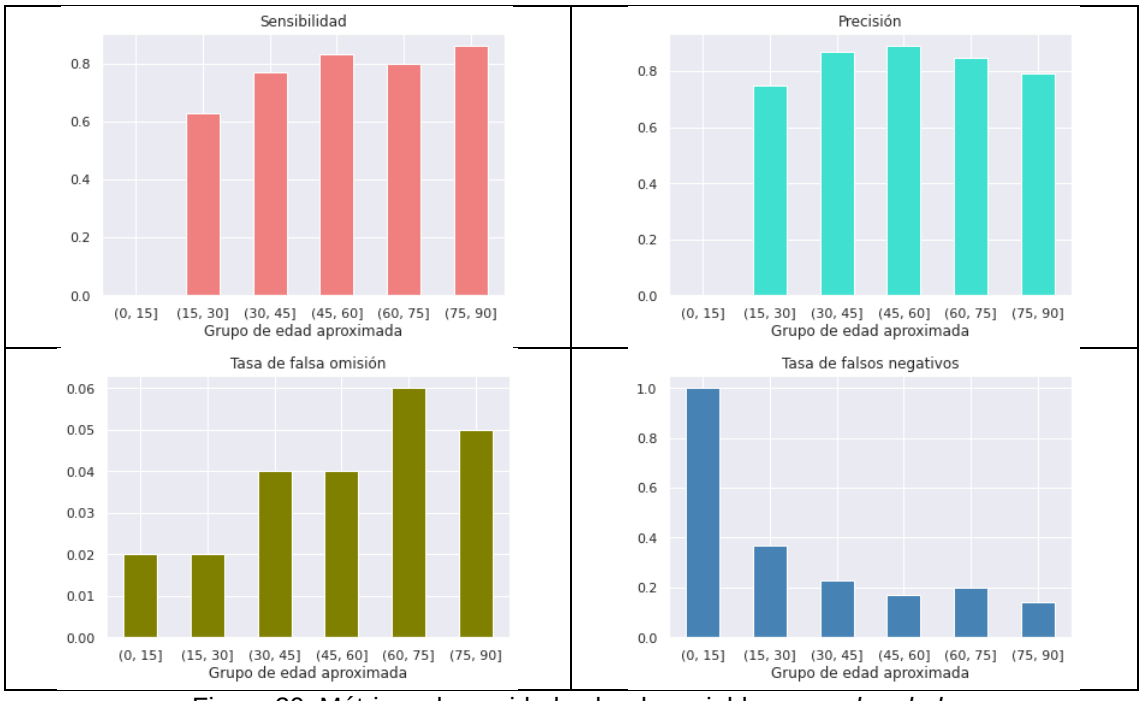


Figura 20: Métricas de equidad sobre la variable *grupo de edad*.
Fuente: elaboración propia.

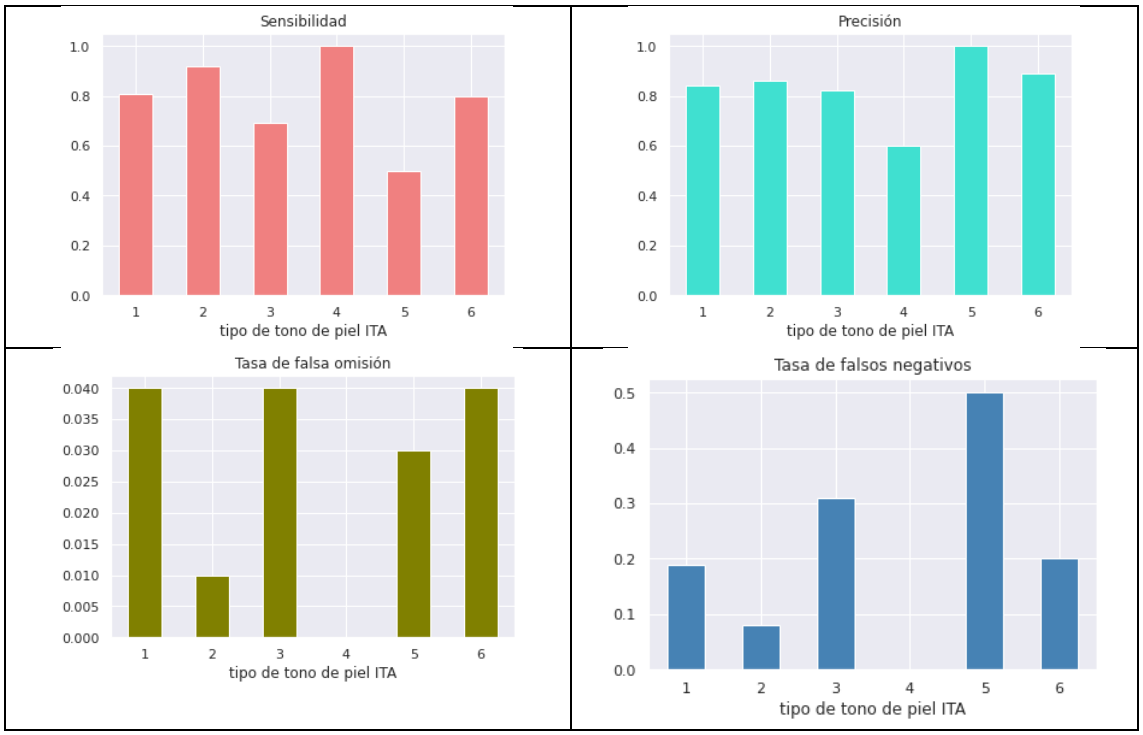


Figura 21: Métricas de equidad sobre la variable *tono de piel ITA*.
Fuente: elaboración propia.

Se pueden realizar las siguientes conclusiones sobre la equidad del modelo:

- Variable *sexo*: las métricas son similares, con valores ligeramente superiores de la tasa de falsos negativos para el caso de sexo masculino. El modelo presenta equidad con respecto a ambos grupos.
- Variable *grupo de edad*: se observan las peores métricas, a excepción de la tasa de falsa omisión, en el grupo de menor edad, y a continuación las del grupo de personas entre 15 y 30 años. Estos datos sugieren que el modelo no se debe emplear en niños y jóvenes, que por otra parte también son los grupos de edad donde el melanoma es menos frecuente. La tasa de falsa omisión es baja para todos los grupos.
- Variable *tipo de tono de piel ITA*: al ser el valor aproximado no se pueden realizar afirmaciones concluyentes. Sin embargo, sin considerar el grupo 6 que está muy influenciado por las imágenes con borde oscuro, el grupo 5, que corresponde al segundo tono más oscuro, presenta un valor de sensibilidad muy bajo, que se corresponde con una tasa de falsos negativos muy alta. Estas métricas sugieren que el modelo no sería válido para tonos de piel más oscura y que habría que realizar las oportunas correcciones, mejorando la calidad de los datos obtenido metadatos más precisos que incluyesen el tono de piel correspondiente a la imagen.

Así pues, teniendo en cuenta estas conclusiones, este modelo sería más apropiado para personas adultas, tanto mujeres como hombres, mayores de 30 años, con tonos de piel morena o clara, es decir, fototipos de la escala de Fitzpatrick I, II, III y IV.

4.4. Aplicación *web* de diagnóstico

Para la realización de la aplicación *web* de diagnóstico se tomó como código base el proporcionado por AWS en su aplicación *Amazon Sagemaker Inference Client Application* [34]. Esta aplicación consta de un cliente *web* JavaScript y un *back-end* con NodeJS, integrados con una función lambda y una API Gateway para comunicarse con el punto de enlace del modelo de diagnóstico. Como herramienta de desarrollo se empleó AWS Amplify, herramienta que permite crear y alojar aplicaciones en AWS.

4.5. Integración de la aplicación *web* y el modelo de diagnóstico

Una vez entrenado, un modelo de diagnóstico tiene que hospedarse y exponerse de manera que sea accesible para las aplicaciones cliente. Esta conexión se realizó través de un punto de enlace de Amazon Sagemaker.

5. Implantación del sistema

Una vez que se creó el modelo de diagnóstico óptimo y se desarrolló la aplicación *web* de diagnóstico, el siguiente paso fue implementar todo el sistema en la nube de AWS. Esta implantación sería equivalente a una puesta en producción del sistema, que en este proyecto se realizó de manera manual. La implementación se podría también realizar de una manera automatizada en el marco de trabajo de una práctica MLOps, como se explica en *7. Conclusiones y trabajos futuros.*

5.1. Implantación del modelo de diagnóstico

El modelo óptimo se entrenó en la plataforma de desarrollo Google Colab, por lo que se requirió realizar una serie de operaciones para implantar el modelo en la plataforma Amazon SageMaker de AWS.

Para realizar esta implantación se siguió la guía de AWS: *Deploy trained Keras or TensorFlow models using Amazon SageMaker* [48].

Los pasos a seguir fueron los siguientes:

- Exportar el modelo al formato ProtoBuf de TensorFlow.
- Convertir el modelo en formato ProtoBuf a la estructura de ficheros apropiada para SageMaker.
- Crear un fichero comprimido con la estructura de ficheros realizada con anterioridad.
- Desplegar el modelo en AWS, creando un punto de enlace para ser empleado por la aplicación *web* de diagnóstico.

Para desplegar el modelo se tuvieron en cuenta tres puntos importantes:

- Se debió crear un fichero de requerimientos donde se indican los módulos que no se encuentren por defecto en la implementación de AWS y que necesite el modelo.
- Se necesitó realizar un punto de entrada al modelo con la funcionalidad necesaria para tratar la entrada y salida de este. En este caso, fue necesario realizar una conversión de la imagen suministrada por la aplicación *web* a un tensor de dimensiones [1,512,512,3] y a su vez transformarlo a formato JSON.
- Existen dos modalidades de servicio para un modelo desplegado en AWS:
    - Servicio dedicado: una o varias máquinas virtuales se encargan de dar servicio al modelo. Las máquinas están siempre activas y suponen un coste durante todo el tiempo de servicio.
    - Servicio de inferencia sin servidor: AWS lanza recursos de computación de manera automática para el modelo según sea necesario. El coste se reduce al tiempo durante el cual se hacen peticiones al servicio, lo que resulta más económico para este proyecto. Por el contrario, puede presentarse un pequeño retardo en la respuesta si el servicio no se ha usado durante un tiempo para que se puedan crear automáticamente los recursos de computación necesarios. Por razones económicas, se optó por implementar esta modalidad de servicio.

5.2. Implantación de la aplicación *web* de diagnóstico

Para la implantación de la aplicación *web* se desplegaron mediante AWS Amplify los siguientes elementos:

- Aplicación AWS Amplify.
- API Gateway.
- Función lambda.
- *Bucket* S3.

La aplicación *web* está activada en la siguiente URL:

https://d39fz5pvr27vh5.cloudfront.net

6. Resultados

Los resultados obtenidos se consideran satisfactorios según los objetivos planteados, ya que se ha conseguido implementar con éxito un ejemplo de aplicación de Ciencia de Datos mediante la realización de un sistema informático capaz de realizar el diagnóstico automático de casos de riesgo de melanoma basado en imágenes dermatoscópicas.

Para llegar a realizar la implantación del sistema, ha sido necesario obtener resultados satisfactorios en todas las fases que el modelo CRISP-DM propone para la fase de ejecución de un proyecto de Ciencia de Datos [9]:

- Entendimiento del negocio: se ha contextualizado y justificado el Trabajo y se han entendido las implicaciones en sostenibilidad, ético-social y de diversidad, explicando cómo la Ciencia de Datos se puede aplicar en el ámbito de la salud para apoyar a profesionales médicos.
- Entendimiento de los datos: se han analizado en profundidad cinco conjuntos de datos abiertos provenientes de los desafíos de ISIC. Se ha seleccionado el conjunto del desafío 2019 por su mayor proporción de casos de melanoma comparado con los otros conjuntos. Los resultados completos del análisis realizado se encuentran en *4.3.1. Análisis de los conjuntos de datos de la base de datos de ISIC* y *4.3.2. Análisis del conjunto de datos del desafío ISIC del año 2019.*
- Preparación de los datos: se ha seleccionado una preparación de datos optimizada para utilizar TensorFlow con TPU, se han eliminado las imágenes que aparecían con distintos grados de acercamiento y se han utilizado 23.218 imágenes en las resoluciones 256x256, 384x384 y 512x512 con 3 canales de color RGB. De estas imágenes, el 17.8% están etiquetadas por parte de expertos con diagnóstico de melanoma.
  Sobre este conjunto se ha realizado una selección aleatoria estratificada para mantener la relación de diagnósticos de melanoma y se ha reservado el 15% de imágenes para realizar la posterior evaluación del modelo. A su vez, de las imágenes dedicadas al entrenamiento, se ha apartado el 15 % de las mismas, mediante otra selección aleatoria estratificada, para realizar la validación del modelo durante el entrenamiento. Los resultados obtenidos están incluidos en *4.3.3 Preparación de los datos de entrenamiento, validación y prueba.*
- Modelado: se han realizado un total de 18 experimentos con distintos valores de hiperparámetros y modelos de la CNN EfficientNet, con transferencia de conocimiento de ImageNet en el entorno de desarrollo Google Colab, para obtener el resultado de un modelo de diagnóstico óptimo sobre EfficientNet-B5. Los resultados detallados de los experimentos se encuentran en *4.3.5 Experimentos para configurar el modelo de diagnóstico óptimo*
- Evaluación: con el modelo óptimo seleccionado, se ha procedido a realizar la evaluación de este, tanto en lo que respecta a las métricas de rendimiento como respecto a las métricas de equidad. Los resultados obtenidos se pueden considerar satisfactorios, con la recomendación de usar el sistema en pacientes mayores de 30 años y con tono de piel clara o morena. Los resultados de la evaluación se detallan en *4.3.7 Evaluación del rendimiento del modelo de diagnóstico y 4.3.8 Evaluación de la equidad del modelo de diagnóstico.*
- Implementación: se ha implantado en la plataforma en la nube de AWS el modelo y la aplicación *web* para introducir nuevas imágenes para diagnosticar. La estructura del sistema de diagnóstico se encuentra detallada en *10.3 Anexo III: Elementos desplegados en AWS.*

7. Conclusiones y trabajos futuros

El ámbito de la salud es probablemente uno de los que más uso de la Ciencia de Datos está haciendo hoy en día, con proyección de una utilización todavía más intensiva en un futuro próximo, teniendo en cuenta las posibilidades ofrecidas de tratamiento y análisis de cantidades masivas de datos. Este Trabajo ha presentado un caso de uso de Ciencia de Datos aplicada en el ámbito de la salud, que demuestra cómo se puede plantear, desarrollar e implantar un sistema de diagnóstico de riesgo de melanoma que sirva de apoyo a la labor de los profesionales médicos en esta materia.

Los resultados, teniendo en cuenta los medios limitados en tiempo y recursos humanos y económicos, se pueden considerar satisfactorios, y se podría considerar el modelo implantado como una “prueba de concepto” que muestra la viabilidad de un sistema de apoyo al diagnóstico médico de este tipo. Se debe considerar que, además de mostrar la manera de seleccionar datos, prepararlos, elegir un modelo mediante experimentos e implantarlo en un sistema en producción, el Trabajo también muestra mecanismos para evaluar el modelo desde un punto de vista de rendimiento y de equidad.

De vital importancia a la hora de demostrar la viabilidad de un sistema es también asegurar el paso de un sistema de desarrollo a un sistema en producción, donde se garantice la seguridad, escalabilidad y disponibilidad del servicio. Para realizar esta demostración, se ha realizado una implantación en la plataforma SageMaker de AWS en la nube, de tal manera que el sistema es accesible 24x7 desde cualquier parte del mundo y se certifica que el sistema puede dar respuesta a prácticamente cualquier tipo de demanda de una manera segura, eficaz y viable desde el punto de vista económico.

Respecto a la planificación inicial del Trabajo, se tuvo que realizar un cambio sustancial en el proyecto debido a la imposibilidad de entrenar modelos complejos de una manera adecuada en AWS. Además del costo asociado, para la cuenta de una persona particular es necesario realizar una solicitud cada vez que se requieren recursos de computación adicionales, tardando varios días en ser concedida cada solicitud.

Finalmente, se decidió crear un entorno de desarrollo en Google Colab, donde, por un coste muy reducido se cuenta con *hardware* de tipo TPU que acelera en gran manera el entrenamiento de este tipo de modelos. Este cambio supuso una semana de retraso respecto a la planificación prevista, que tuvo que recuperarse añadiendo carga de trabajo en el último mes del proyecto.

Respecto a la dificultad para realizar el entrenamiento de AWS, cabía también la posibilidad de investigar sobre otros tipos de *hardware* en AWS o realizar el proyecto en su totalidad en la plataforma de Google Cloud. La decisión final, teniendo en cuenta los conocimientos del estudiante y el tiempo disponible, fue adoptar una solución hibrida con experimentos y entrenamiento de modelos en Google Colab e implantación en AWS. Como beneficio adicional, se demuestra cómo se pueden integrar diversos proveedores dentro del producto, que es una situación muy habitual en cualquier proyecto de cierta envergadura.

Teniendo en cuenta el alcance de este proyecto, las métricas de rendimiento y equidad se pueden considerar adecuadas, con la recomendación de su uso en pacientes adultos, tanto mujeres como hombres, mayores de 30 años, con tonos de piel morena o clara. Como es lógico, este sistema desarrollado no sería apto para un sistema de apoyo al diagnóstico a la comunidad médica, sino que como se ha comentado con anterioridad podría servir como prueba de concepto.

Existen algunas recomendaciones y técnicas específicas al problema de las imágenes médicas que pueden mejorar un modelo basado en Ciencia de Datos y que no se han considerado en el presente Trabajo por el alcance reducido del mismo. Estas recomendaciones y técnicas constituyen líneas de trabajo a explorar para sistemas de diagnóstico más avanzados.

En primer lugar, como en todo proyecto de Ciencia de Datos, la principal área de mejora consiste en obtener más datos y en incrementar la calidad de los mismos. En la base de datos de ISIC existen más de 70.000 imágenes que se pueden emplear para entrenar un modelo, además de otras bases de datos de imágenes dermatoscópicas que no son de acceso libre y que se pueden incorporar en el proceso de entrenamiento mediante la correspondiente licencia de uso. Relacionado con la calidad de los datos, sería importante poder contar con bases de datos de imágenes con sus correspondientes metadatos que incluyesen el tipo de piel, ya que el modelo en este proyecto da unos resultados mediocres en este aspecto.

Otros aspectos que se pueden aplicar como líneas de trabajo futuro para mejorar la calidad del modelo de diagnóstico son:

- Incluir los metadatos (sexo, edad, zona anatómica, tono piel, etc.) de los pacientes en el modelo. Se puede realizar mediante un algoritmo de clasificación que se ensamble con el modelo de diagnóstico de las imágenes [49].
- Preparar varios modelos con distintas configuraciones y realizar la predicción con un ensamblado de estos.
- Procesar las imágenes para contrarrestar la presencia de artefactos (por ejemplo, pelo o regla de medir) y la variabilidad en las características de las imágenes (por ejemplo, contraste, intensidad, ángulo) [18].
- Utilizar técnicas de segmentación, ya que se puede mejorar la tarea de clasificación de una imagen dermatoscópica [21] si antes se separan las zonas enfermas de las zonas sanas y se clasifica la zona enferma.
- Eliminar el borde negro que aparece en numerosas imágenes [50].
- Aplicar técnicas de aumentado de datos (*data augmentation*) que consiste en realizar transformaciones en las imágenes como pueden ser rotaciones, cambios de brillo, segmentaciones de zonas enfermas, etc. para producir datos artificiales y evitar el sobreajuste [51].
- Aplicar aumento de datos en la inferencia (*test time augmentation*) que consiste en realizar las técnicas de aumentado de datos sobre la imagen nueva sobre la que se desea realizar el diagnóstico [52].

Respecto a la aplicación *web* desarrollada para la introducción de nuevas imágenes para diagnosticar, puede extenderse de manera que contenga una gestión de usuarios autorizados, que solicite los metadatos de los pacientes o que almacene en una base de datos las imágenes y los diagnósticos realizados. Estos diagnósticos pueden ser posteriormente analizados por expertos en la materia y servir de retroalimentación al sistema, de modo que se realice un proceso de mejora continua, realizando ajustes del modelo en producción.

Asociado al proceso de mejora continua, que implica cambios en los sistemas y aplicaciones informáticas, se han desarrollado prácticas en los últimos años que agrupan el desarrollo de aplicaciones con las operaciones en sistemas informáticos, que se ha venido a denominar DevOps o en el caso de sistemas de aprendizaje automático MLOps, de manera que los cambios se hagan de la manera más eficaz posible, con un alto grado de automatización en las tareas de despliegues de nuevas versiones.

Una opción muy interesante a tener en cuenta para implementar MLOps es el servicio de Amazon SageMaker Pipelines que permite construir, automatizar y gestionar flujos de trabajo de sistemas de aprendizaje automático [53]. Una posible configuración de una práctica de MLOps que abarcaría los flujos de trabajo desde el desarrollo a la puesta en producción es la siguiente:

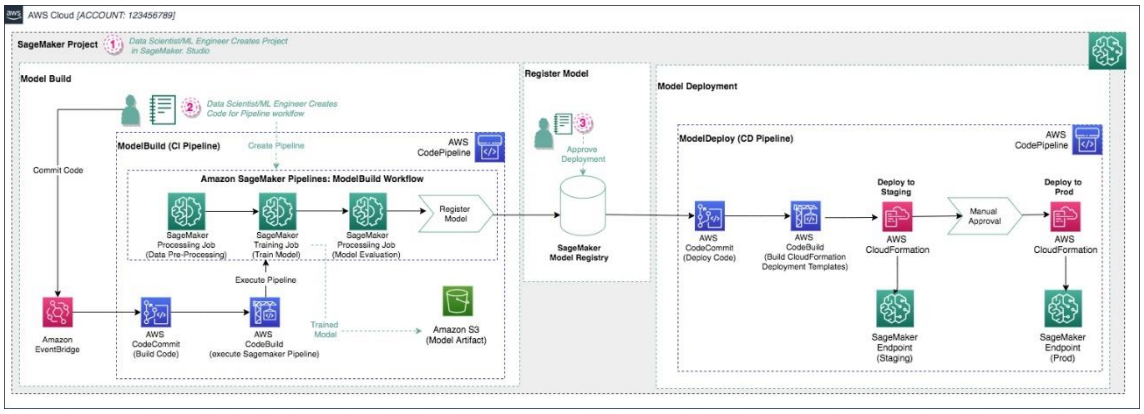


Figura 22: Ejemplo de práctica MLOps basado en Amazon SageMaker Pipelines. Fuente: [53].

Por último, es importante destacar el problema al que se enfrentan en general todos los sistemas que aplican redes neuronales de aprendizaje profundo para crear un modelo y en particular los dedicados al cuidado de pacientes con regulaciones muy estrictas, que es cómo hacer que el proceso de diagnóstico sea transparente, es decir, saber explicar cómo funciona el modelo y cómo realiza el diagnóstico. Este problema de interpretabilidad implica que el modelo puede llegar a ser considerado como una "caja negra" que emite un resultado sin explicar las razones, lo que a efectos de la comunidad de potenciales usuarios e incluso de la legislación aplicable puede hacer que el sistema sea inviable.

Para solventar este problema, "se puede incorporar interpretabilidad durante el proceso de diseño de la red neuronal. Los métodos de interpretabilidad post-hoc proporcionan explicaciones para las predicciones después de que el modelo ha sido entrenado" [54].

Existen varios métodos de interpretabilidad para sistemas de imágenes médicas, resumidos en la siguiente tabla [54]:

Método	Descripción
Modelo de aprendizaje conceptual	Se predicen en primer lugar conceptos clínicos de alto nivel y la clasificación se realiza utilizando esos conceptos.
Modelo basado en casos	Se crean prototipos discriminativos y la clasificación se realiza comparando características extraídas de las imágenes con los prototipos.
Explicación contrafáctica	Las imágenes de entrada se perturban de una manera realista para generar la predicción contraria.
Atribución de concepto	Se generan explicación para cuantificar la influencia a alto nivel de características de la imagen diagnosticada.
Descripción textual	Se proporcionan explicaciones textuales junto a la predicción.
Espacio latente	Se utiliza para descubrir los factores de variación más importantes respecto al conocimiento clínico, de manera que se identifican similitudes y valores atípicos.
Mapa de atributos	Se proporciona las explicaciones destacando las regiones de la imagen de entrada que el modelo considera importantes para el diagnóstico.

Método	Descripción
Representación interna de la red	Se visualizan y explican las diferentes características aprendidas por los diferentes filtros de la red.

Tabla 22: Métodos de interpretabilidad para sistemas de imágenes médicas.
Fuentes: [54]

Una vez obtenido el diagnóstico y su explicación correspondiente, se evalúa el modelo en un entorno controlado por profesionales en la materia y de esta manera se puede conseguir una solución clínicamente aceptable.

En conclusión, este sistema de diagnóstico automático de casos de riesgo de melanoma basado en imágenes dermatoscópicas podría ser utilizado por la comunidad médica si se realizasen las mejoras en la calidad del diagnóstico indicadas, se consiguiese su interpretabilidad en un contexto clínico y se proporcionase una práctica de MLOps para gestionar nuevas versiones en un proceso de mejora continua del sistema.

8. Glosario

API: *Application Programming Interface*. Interfaz de programación de aplicaciones.

AWS: *Amazon Web Services*. Proveedor de servicios computacionales en la nube.

CNN: *Convolutional Neural Network*. Red neuronal convolucional.

ISIC: *International Skin Imaging Collaboration*. Colaboración internacional de imágenes de la piel.

HTTP: Hypertext Transfer Protocol. Protocolo de transferencia de hipertexto. Se utiliza para transmitir información en Internet.

JSON: *JavaScript Object Notation.* Notación de objeto de JavaScript. Formato de texto utilizado para el intercambio de datos.

NodeJS: entorno de programación de código abierto para servidores basado en JavaScript.

REST: *Representational State Transfer*: Transferencia de estado representacional. Tipo de arquitectura software que se utiliza en sistemas que obtienen datos o ejecutan operaciones para datos utilizando HTTP.

TensorFlow: biblioteca de código abierto especializada en tareas de aprendizaje automático. Desarrollada y mantenida por Google Brain Team.

TPU: *Tensor Processing Unit*. Unidad de proceso de tensores. Es un circuito integrado específico para aplicaciones de inteligencia artificial desarrollado por Google para aplicaciones desarrolladas con TensorFlow.

9. Bibliografía

[1] **Organización Mundial de la Salud**, “Cáncer. Datos y cifras.”, 2022. https://www.who.int/es/news-room/fact-sheets/detail/cancer (consultado jun. 03, 2022).

[2] **American Cancer Society**, “Estadísticas importantes sobre el cáncer de piel tipo melanoma”, 2202. https://www.cancer.org/es/cancer/cancer-de-piel-tipo-melanoma/acerca/estadisticas-clave.html (consultado jun. 03, 2022).

[3] **P. Zaballos, C. Carrera, S. Puig, y J. Malvehy**, “Criterios dermatoscópicos para el diagnóstico del melanoma”, *Med Cutan Iber Lat Am*, vol. 32, núm. 1, pp. 3–17, 2004.

[4] **A. R. Lopez, X. Giro-i-Nieto, J. Burdick, y O. Marques**, “Skin lesion classification from dermoscopic images using deep learning techniques”, en *2017 13th IASTED international conference on biomedical engineering (BioMed)*, 2017, pp. 49–54.

[5] **A. Lashbrook**, “AI-Driven Dermatology Could Leave Dark-Skinned Patients Behind - The Atlantic”, 2018. https://www.theatlantic.com/health/archive/2018/08/machine-learning-dermatology-skin-color/567619/ (consultado oct. 13, 2022).

[6] **J. Curto Diaz y D. Cabanillas Becerril**, *Aplicaciones para la toma de decisiones*. Barcelona: Universitat Oberta de Catalunya, 2021.

[7] **P. Saleiro *et al.***, *Aequitas: A Bias and Fairness Audit Toolkit*. Center for Data Science and Public Policy. University of Chicago., 2019.

[8] **Project Management Institute**, “PMBOK Guide”, 2022. https://www.pmi.org/pmbok-guide-standards/foundational/pmbok?sc_camp=D750AAC10C2F4378CE6D51F8D987F49D (consultado oct. 20, 2022).

[9] **Data Science Process Alliance**, “CRISP-DM - What is CRISP DM?” https://www.datascience-pm.com/crisp-dm-2/ (consultado jun. 05, 2022).

[10] **T. M. Mitchell**, *Machine Learning*. Nueva York: McGraw-Hill, 1997.

[11] **Ian Goodfellow, Yoshua Bengio, y Aaron Courville**, *Deep Learning*. MIT Press, 2016.

[12] **A. Bosh Rué, J. Casas Roma, y T. Lozano Bagén**, *Deep Learning - Principios y Fundamentos*. Barcelona: Universitat Oberta de Catalunya, 2019.

[13] **M. Tan y Q. v. Le**, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”, *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10691–10700, may 2019, doi: 10.48550/arxiv.1905.11946.

[14] **J. Yosinski, J. Clune, Y. Bengio, y H. Lipson**, “How transferable are features in deep neural networks?”, *Adv Neural Inf Process Syst*, vol. 27, núm. Dec. 2014, pp. 3320–3328, 2014.

[15] **M. Huh, P. Agrawal, y A. A. Efros**, “What makes ImageNet good for transfer learning?”.

[16] **ImageNet**, “About ImageNet”. https://www.image-net.org/about.php (consultado nov. 06, 2022).

[17] **MedicalStat**, “Calculate the sensitivity and specificity and similar concepts”, 2022. https://www.medicalstat.org/CalculateSensitivitySpecificity.html (consultado ene. 12, 2023).

[18] **A. N. Hoshyar, A. Al-Jumaily, y A. N. Hoshyar**, “The Beneficial Techniques in Preprocessing Step of Skin Cancer Detection System Comparing”, *Procedia Comput Sci*, vol. 42, núm. C, pp. 25–31, ene. 2014, doi: 10.1016/J.PROCS.2014.11.029.

[19] **A. Esteva *et al.***, “Dermatologist-level classification of skin cancer with deep neural networks”, *Nature 2017 542:7639*, vol. 542, núm. 7639, pp. 115–118, ene. 2017, doi: 10.1038/nature21056.

[20] **O. Russakovsky *et al.***, “ImageNet Large Scale Visual Recognition Challenge”, *Int J Comput Vis*, vol. 115, núm. 3, pp. 211–252, dic. 2015, doi: 10.1007/S11263-015-0816-Y/FIGURES/16.
[21] **A. Adegun y S. Viriri**, “Deep learning techniques for skin lesion analysis and melanoma cancer detection: a survey of state-of-the-art”, *Artif Intell Rev*, vol. 54, núm. 2, pp. 811–841, feb. 2021, doi: 10.1007/s10462-020-09865-y.
[22] **International Skin Imaging Collaboration**, “Goals of ISIC”. https://www.isic-archive.com/#!/topWithHeader/tightContentTop/about/aboutIsicOverview (consultado may 13, 2022).
[23] **Society for Imaging Informatics in Medicine**, “The Melanoma Classification Challenge”, 2021. https://siim.org/page/melanoma_classification_challenge (consultado may 12, 2022).
[24] **I. Pan**, “[2nd place] Solution Overview”, 2020. https://www.kaggle.com/c/siim-isic-melanoma-classification/discussion/175324 (consultado may 13, 2022).
[25] **C. Rota**, “3rd place solution overview”, 2020. https://www.kaggle.com/c/siim-isic-melanoma-classification/discussion/175633 (consultado may 13, 2022).
[26] **Q. Ha, B. Liu, y F. Liu**, “Identifying Melanoma Images using EfficientNet Ensemble: Winning Solution to the SIIM-ISIC Melanoma Classification Challenge”, 2020. [En línea]. Available: https://github.com/haqishen/SIIM-ISIC-Melanoma-Classification-1st-Place-Solution
[27] **N. Gessert, M. Nielsen, M. Shaikh, R. Werner, y A. Schlaefer**, “Skin lesion classification using ensembles of multi-resolution EfficientNets with meta data”, *MethodsX*, vol. 7, p. 100864, ene. 2020, doi: 10.1016/J.MEX.2020.100864.
[28] **P. J. Bevan y A. Atapour-Abarghouei**, “Detecting Melanoma Fairly: Skin Tone Detection and Debiasing for Skin Lesion Classification”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13542 LNCS, pp. 1–11, feb. 2022, doi: 10.48550/arxiv.2202.02832.
[29] **M. R. Luo**, “CIELAB”, *Encyclopedia of Color Science and Technology*, pp. 1–7, 2015, doi: 10.1007/978-3-642-27851-8_11-1.
[30] **N. A. Gil Coca, E. H. Hernández Rincón, y J. Contreras Ruíz**, “El impacto de la prevención primaria y secundaria en la disminución del cáncer de piel”, *CES Salud Pública*, 2016, doi: 10.21615/CESSP.7.2.4.
[31] **Google Cloud**, “Google Colab”. https://research.google.com/colaboratory/intl/es/faq.html (consultado ene. 08, 2023).
[32] **Google Cloud**, “Cloud Tensor Processing Units (TPUs)”. https://cloud.google.com/tpu/docs/tpus (consultado ene. 08, 2023).
[33] **Amazon Web Services**, “Amazon SageMaker for Data Scientist”, 2022. https://aws.amazon.com/es/sagemaker/data-scientist/?nc1=h_ls (consultado oct. 10, 2022).
[34] **Amazon Web Services**, “Amazon Sagemaker Object Detection Inference Endpoint visualization client web application hosted and managed by AWS Amplify.” https://github.com/awslabs/amazon-sagemaker-inference-client (consultado dic. 23, 2022).
[35] **Amazon Web Services**, *Amazon SageMaker Developer Guide*. 2022.
[36] **B. Cassidy, C. Kendrick, A. Brodzicki, J. Jaworek-Korjakowska, y M. H. Yap**, “Analysis of the ISIC image datasets: Usage, benchmarks and recommendations”, *Med Image Anal*, vol. 75, p. 102305, ene. 2022, doi: 10.1016/J.MEDIA.2021.102305.
[37] **D. Gutman *et al.***, “Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on Biomedical Imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC)”, may 2016, doi: 10.48550/arxiv.1605.01397.

[38] **N. C. F. Codella *et al.***, “Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC)”, *Proceedings - International Symposium on Biomedical Imaging*, vol. 2018-April, pp. 168–172, oct. 2017, doi: 10.48550/arxiv.1710.05006.

[39] **N. Codella *et al.***, “Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC)”, feb. 2019, doi: 10.48550/arxiv.1902.03368.

[40] **M. Combalia *et al.***, “BCN20000: Dermoscopic Lesions in the Wild”, ago. 2019, Consultado: dic. 21, 2022. [En línea]. Available: http://arxiv.org/abs/1908.02288

[41] **N. C. F. Codella *et al.***, “Skin lesion analysis toward melanoma detection: A challenge at the 2017 International symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC)”, *Proceedings - International Symposium on Biomedical Imaging*, vol. 2018-April, pp. 168–172, may 2018, doi: 10.1109/ISBI.2018.8363547.

[42] **P. Tschandl, C. Rosendahl, y H. Kittler**, “Data descriptor: The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions”, *Sci Data*, vol. 5, ago. 2018, doi: 10.1038/SDATA.2018.161.

[43] **V. Rotemberg *et al.***, “A Patient-Centric Dataset of Images and Metadata for Identifying Melanomas Using Clinical Context”, *Sci Data*, vol. 8, núm. 1, ago. 2020, doi: 10.48550/arxiv.2008.07360.

[44] **Keras**, “Creating TFRecords”. https://keras.io/examples/keras_recipes/creating_tfrecords/ (consultado dic. 21, 2022).

[45] **C. Deotte**, “ISIC 2019 TFRecords 512x512”. https://www.kaggle.com/datasets/cdeotte/isic2019-512x512 (consultado dic. 21, 2022).

[46] **TensorFlow**, “Classification on imbalanced data”. https://www.tensorflow.org/tutorials/structured_data/imbalanced_data#optional_set_the_correct_initial_bias (consultado dic. 22, 2022).

[47] **Keras**, “Image classification via fine-tuning with EfficientNet”. https://keras.io/examples/vision/image_classification_efficientnet_fine_tuning/ (consultado dic. 22, 2022).

[48] **P. Ponnapalli, J. G. Piccinini, y S. Senthivel**, “Deploy trained Keras or TensorFlow models using Amazon SageMaker”, *AWS Machine Learning Blog*, 2022. https://aws.amazon.com/blogs/machine-learning/deploy-trained-keras-or-tensorflow-models-using-amazon-sagemaker/ (consultado ene. 10, 2023).

[49] **D. N. Anggraini Ningrum *et al.***, “Deep Learning Classifier with Patient’s Metadata of Dermoscopic Images in Malignant Melanoma Detection”, *J Multidiscip Healthc*, vol. 14, p. 877, 2021, doi: 10.2147/JMDH.S306284.

[50] **S. W. Pewton y M. H. Yap**, “Dark Corner on Skin Lesion Image Dataset: Does it matter?”, Consultado: ene. 14, 2023. [En línea]. Available: https://github.com/mmu-dermatology-

[51] **F. Perez, C. Vasconcelos, S. Avila, y E. Valle**, “Data Augmentation for Skin Lesion Analysis”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11041 LNCS, pp. 303–311, sep. 2018, doi: 10.1007/978-3-030-01201-4_33.

[52] **H. Ashraf, A. Waris, M. F. Ghafoor, S. O. Gilani, y I. K. Niazi**, “Melanoma segmentation using deep learning with test-time augmentations and conditional random fields”, *Scientific Reports 2022 12:1*, vol. 12, núm. 1, pp. 1–16, mar. 2022, doi: 10.1038/s41598-022-07885-y.

[53] **S. Morgan, H. Weishahn, y S. Eigenbrode**, “Building, automating, managing, and scaling ML workflows using Amazon SageMaker Pipelines”, *AWS Machine Learning Blog*, 2021. https://aws.amazon.com/es/blogs/machine-learning/building-automating-managing-and-scaling-ml-workflows-using-amazon-sagemaker-pipelines/ (consultado ene. 13, 2023).

[54] **Z. Salahuddin, H. C. Woodruff, A. Chatterjee, y P. Lambin**, “Transparency of deep neural networks for medical image analysis: A review of interpretability methods”, *Comput Biol Med*, vol. 140, p. 105111, ene. 2022, doi: 10.1016/J.COMPBIOMED.2021.105111.

10. Anexos

10.1. Anexo I: Métricas de equidad de Aequitas

Las métricas que propone el marco de equidad algorítmica de Aequitas [7] son las siguientes:

Métrica	Descripción	Fórmula
Predicted Positive	Número de entidades dentro de un grupo donde la predicción es positiva ($\hat{Y} = 1$).	PP_g
Total Predictive Positive	Número total de entidades predichas como positiva a lo largo de grupos definidos por *A*.	$K = \sum_{A=a_1}^{A=a_n} PP_{g(a_i)}$
Predicted Negative	Número de entidades dentro de un grupo donde la predicción es negativa ($\hat{Y} = 0$).	PN_g
Predicted Prevalence	Fracción de entidades dentro de un grupo predichas como positivas.	$PPrev_g = \frac{PP_g}{\lvert g \rvert} = \Pr(\hat{Y} = 1 \mid A = a_i)$
Predicted Positive Rate	Fracción de entidades predichas como positiva que pertenecen a un determinado grupo	$PPR_g = \frac{PP_g}{K} = \Pr(A = a_i \mid \hat{Y} = 1)$
False Positive	Número de entidades de un grupo predichas erróneamente como positivas.	FP_g
False Negative	Número de entidades de un grupo predichas erróneamente como *negativa*.	FN_g
True Positive	Número de entidades de un grupo predichas correctamente como positiva.	TP_g
True Negative	Número de entidades de un grupo predichas correctamente como negativa.	TN_g
False Discovery Rate	Fracción de *False Positive* de un grupo respecto a *Predicted Positive* de ese grupo.	$FDR_g = \frac{FP_g}{PP_g} = \Pr(Y = 0 \mid \hat{Y} = 1, A = a_i)$
False Omission Rate	Fracción de *False Negative* de un grupo respecto a *Predicted Negative* de ese grupo.	$FOR_g = \frac{FP_g}{PP_g} = \Pr(Y = 0 \mid \hat{Y} = 1, A = a_i)$
False Positive Rate	Fracción de *False Positive* de un grupo respecto a las entidades etiquetadas como *negative* en ese grupo.	$FPR_g = \frac{FP_g}{LN_g} = \Pr(\hat{Y} = 1 \mid Y = 0, A = a_i)$

Métrica	Descripción	Fórmula
False Negative Rate	Fracción de *False Negative* de un grupo respecto a las entidades etiquetadas como *positive* en ese grupo.	$FNR_g = \frac{FN_g}{LP_g} = \Pr(\hat{Y} = 0 \mid Y = 1, A = a_i)$

Tabla 23: Métricas de equidad algorítmica
Fuente: [7]

La disparidad de sesgo *j* para un grupo a_i se calcula de la siguiente manera [7]:

$$disparidad_{j,a_i} = \frac{métrica_{j,a_i}}{métrica_{j,a_{grupo\ de\ referencia}}}$$

10.2. Anexo II: Arquitecturas típicas de las CNN aplicadas a la detección de riesgos de melanoma

El diseño general de una CNN es el siguiente:

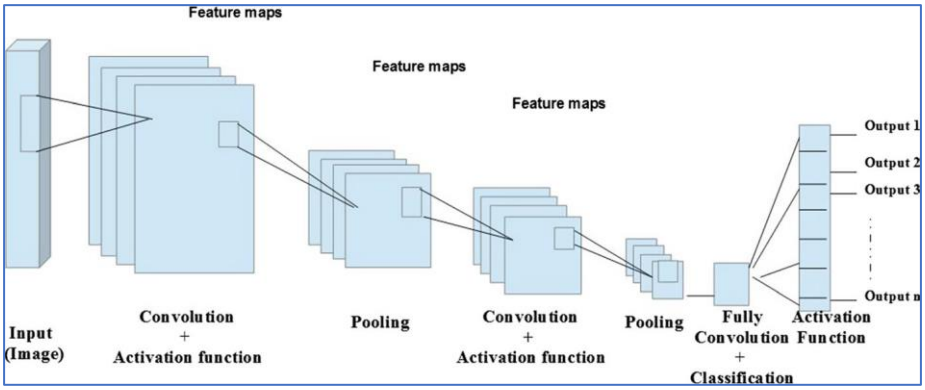


Figura 23: Diseño general de una CNN.
Fuente: [21].

En los últimos años se ha adoptado en gran medida la utilización de CNN para los sistemas de clasificación de imágenes dermatoscópicas. A continuación, se comparan las arquitecturas más habituales [21]:

Técnica	Descripción	Ventajas	Inconvenientes
AlexNet	Utiliza filtros de tamaño grande y pequeño en capa inicial (5x5 y 11x11) y última (3x3) para la extracción de características.	Introduce regularización en la CNN y emplea el uso paralelo de GPU como acelerador para tratar arquitecturas complejas	Tiene tendencia al sobreajuste a partir de los mapas de características aprendidos con filtros de tamaño grande.
VGG (Visual Graphics Group)	Se caracteriza por una forma piramidal y se compone de una serie de capas convolucionales seguidas de capas de agrupación con las capas de agrupación contribuyendo a la forma más estrecha de la arquitectura.	Utiliza un campo receptivo efectivo con una topología simple y homogénea.	Utiliza capas completamente conectadas computacionalmente costosas.
GoogleNet	Utiliza filtros multiescala dentro de las capas y emplea un sistema de división, transformación y fusión de procesos.	Se reduce el número de parámetros mediante el uso de una capa de cuello de botella, con una agrupación promedio global en la última capa.	Utiliza una topología heterogénea que dificulta la selección de parámetros. Información útil también se puede perder debido a la capa de cuello de botella.

Técnica	Descripción	Ventajas	Inconvenientes
Inception-V3, Inception-V4	Emplean jerarquías de características profundas y representación de características multinivel.	Los filtros asimétricos y la capa de cuello de botella son empleados para reducir el coste computacional.	Emplean un diseño de arquitectura complejo. Estas arquitecturas carecen de homogeneidad.
ResNet	Emplea conexiones de salto basadas en identidad y arquitectura de aprendizaje residual.	Utiliza el aprendizaje residual y reduce el efecto del problema del gradiente de fuga.	La arquitectura es compleja y se degrada la información del mapa de características durante el proceso de reenvió de alimentación.
Inception-ResNet	Combina las arquitecturas ResNet e Inception.	Combina el poder del aprendizaje residual y de inception.	La fase de aprendizaje es generalmente lenta.
DenseNet	Emplea dimensiones a través de profundidad y capas cruzadas para maximizar el flujo de datos entre las capas en la red.	Elimina los mapas de características redundantes.	Hay un gran aumento en la cantidad de parámetros debido al aumento de mapas de características en cada capa.
Xception	Emplea convolución separable en profundidad.	Utiliza cardinalidad para aprender abstracciones de características.	Su coste computacional es alto.
ResNeXt	Utiliza transformación residual agregada y procesos de cardinalidad en cada capa.	Utiliza la personalización de parámetros y convolución agrupada.	Su coste computacional es alto.
EfficientNet	Aplica un método de escalado que escala uniformemente todas las dimensiones de profundidad/ancho/resolución utilizando un coeficiente compuesto.	Se consiguen resultados similares a otras arquitecturas utilizando un numero mucho menor de parámetros, con coste computacional más reducido.	Generalmente la fase de entrenamiento es compleja.

Tabla 24: Arquitecturas para la clasificación de imágenes dermatoscópicas.
Fuente: [21]

10.3. Anexo III: Elementos desplegados en AWS

La estructura en AWS de los elementos del sistema de diagnóstico es la siguiente:

Elemento	Nombre	Localización en AWS
Mejor modelo en el formato apropiado para AWS	TFG_mejor_modelo	s3://uoc-tfg/TFG_mejor_modelo.tar.gz
Modelo de inferencia	TFG-Modelo	arn:aws:sagemaker:us-east-1:361247709919:model/tfg-modelo
Configuración del punto de enlace	TFG-endpoint	arn:aws:sagemaker:us-east-1:361247709919:endpoint-config/tfg-endpoint
Punto de enlace	TFG-endpoint	arn:aws:sagemaker:us-east-1:361247709919:endpoint/tfg-endpoint
Aplicación Amplify	awsamplifysagemaker	Espacio AWS Amplify del usuario TFG
API Gateway	smInferenceClient	Espacio API Gateway del usuario TFG (id: 4kuwbvjc6k)
Función Lambda	awsamplifysagemaker-dev	amplify-awsamplifysagemaker-dev-193202-functionawsamplifysagemaker-1E7ILXUSWAZSX
Aplicación *web: front-end*	amazonsagemakerinfer-20221217211614-hostingbucket-dev	arn:aws:s3:::amazonsagemakerinfer-20221217211614-hostingbucket-dev
Aplicación *web: back-end*	amplify-amazonsagemakerinfer-dev-211434-deployment	arn:aws:s3:::amplify-amazonsagemakerinfer-dev-211434-deployment
Aplicación *web*: URL	Diagnóstico de riesgo de melanoma	https://d39fz5pvr27vh5.cloudfront.net

Tabla 25: Estructura de los elementos desplegados en AWS.
Fuente: elaboración propia.

10.4. Anexo IV: Preparación del entorno de desarrollo en Amazon Sagemaker

Se siguieron las instrucciones del *Amazon SageMaker Developer Guide* [35]

En primer lugar, se necesitó crear un usuario administrador de AWS y un dominio de SageMaker. Un dominio de SageMaker consta de un volumen asociado de Elastic File System (EFS); una lista de usuarios autorizados; y una variedad de configuraciones de seguridad, aplicaciones, políticas y Virtual Private Cloud (VPC). Los usuarios dentro de un dominio de SageMaker pueden compartir archivos de notebook y otros artefactos entre sí.
Los pasos a seguir para crear un usuario AWS administrador de SageMaker y configurar un dominio de SageMaker fueron los siguientes:

1. Crear una cuenta AWS. Al crear una nueva cuenta en Amazon Web Services (AWS), esta cuenta de AWS se registra automáticamente para todos los servicios de AWS, incluido SageMaker.
2. Crear un grupo de administrador de IAM (*Identity and Access Management*) con las políticas de permisos adecuadas a un administrador de SageMaker.
3. Crear un usuario que pertenezca al grupo de administradores de SageMaker.
4. Crear un alias para que el usuario administrador del SageMaker (u otro cualquier usuario no administrador que se quiera añadir al sistema) tenga una URL directa a la consola de AWS. Se crea el alias `tfg-uoc` que proporciona la URL https://tfg-uoc.signin.aws.amazon.com/console
5. A continuación, se necesitó realizar la incorporación al dominio Amazon SageMaker. Con el usuario administrador creado, se accedió a la consola de SageMaker y se realiza la configuración estándar de dominio de SageMaker. Se crea el rol de ejecución predeterminado y se utiliza la VPN predeterminada.
6. A continuación, se realizó la configuración de SageMaker Studio, con la versión Jupyter Lab 3.0 con la ubicación s3://sagemaker-studio-uoc-tfg/sharing como ubicación de S3 para recursos de bloc de notas de uso compartido.