UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Biotecnología

Desarrollo de un Sistema Predictivo y de Apoyo Clínico para Enfermedades Cardiovasculares mediante Machine Learning y Deep Learning: Un Enfoque en el Infarto de Miocardio

Trabajo Fin de Máster

Máster Universitario en Biotecnología Biomédica

AUTOR/A: Pascual Gisbert, Carmen

Tutor/a: Forment Millet, José Javier

Cotutor/a externo: Vallalta Rueda, Juan-Francisco

CURSO ACADÉMICO: 2024/2025

RESUMEN

Las enfermedades cardiovasculares (ECV) son un problema crítico de salud a nivel mundial y una de las principales causas de mortalidad y morbilidad. En España, el infarto de miocardio es una de las afecciones más comunes dentro de esta categoría. En este contexto, la implementación de tecnologías avanzadas como el aprendizaje automático (Machine Learning, ML) y el aprendizaje profundo (Deep Learning, DL) ofrece nuevas oportunidades para mejorar el diagnóstico y el manejo clínico de las ECV.

Este trabajo presenta el desarrollo e implementación de un sistema diseñado para apoyar la labor clínica en cardiología, combinando modelos de Machine Learning, con herramientas de procesamiento de lenguaje natural (NLP), basadas en la arquitectura Transformers. El objetivo del sistema es predecir el riesgo de enfermedad cardiovascular utilizando una base de datos de pacientes con ECV, mientras que las técnicas de NLP permiten analizar de manera efectiva documentos médicos, proporcionando información adicional útil para la toma de decisiones en la práctica clínica.

Los resultados muestran que el modelo predictivo de ML es preciso para evaluar el riesgo cardiovascular, y que el sistema de NLP facilita la extracción de información relevante de textos médicos complejos, apoyando así la toma de decisiones clínicas. Este enfoque tiene como objetivo contribuir a un diagnóstico más precoz y a una atención médica más personalizada, optimizando la atención de pacientes con factores de riesgo como hipertensión o hipercolesterolemia, contribuyendo así a una mejor planificación terapéutica y seguimiento médico.

PALABRAS CLAVE

Enfermedades cardiovasculares (ECV), Infarto de Miocardio, Machine Learning (ML), Deep Learning (DL), Inteligencia artificial en cardiología, Factores de riesgo clínicos, Asesoramiento clínico, Procesamiento Lenguaje Natural (NLP), Transformers.

ABSTRACT

Cardiovascular diseases (CVD) are a critical global health issue and one of the leading causes of mortality and morbidity. In Spain, myocardial infarction is one of the most common conditions within this category. In this context, the implementation of advanced technologies such as Machine Learning (ML) and Deep Learning (DL) offers new opportunities to improve the diagnosis and clinical management of CVDs.

This paper presents the development and implementation of a system designed to support clinical work in cardiology by combining Machine Learning models with natural language processing (NLP) tools based on the Transformers architecture. The goal of the system is to predict cardiovascular disease risk using a patient database with CVD, while NLP techniques effectively analyze medical documents, providing additional useful information for decision-making in clinical practice.

The results show that the ML predictive model is accurate in assessing cardiovascular risk, and that the NLP system facilitates the extraction of relevant information from complex medical texts, thus supporting clinical decision-making. This approach aims to contribute to earlier diagnosis and more personalized medical care, optimizing the management of patients with risk factors such as hypertension or hypercholesterolemia, thereby contributing to better therapeutic planning and medical follow-up.

KEYWORDS

Cardiovascular diseases (CVD), Myocardial Infarction, Machine Learning (ML), Deep Learning (DL), Artificial Intelligence in Cardiology, Clinical Risk Factors, Clinical Decision Support, Natural Language Processing (NLP), Transformers.

ÍNDICE

ÍNDICE DE FIGURAS

ACRÓNIMOS

AI: Artificial Intelligence (Inteligencia Artificial)

ANN: Artificial Neural Networks (Redes Neuronales Artificiales)

ARA II: Bloqueadores de los Receptores de Angiotensina II

ASCVD: Atherosclerotic Cardiovascular Disease (Enfermedad Cardiovascular Aterosclerótica)

BDS: Biodegradable Stents (Stents Biodegradables)

CNN: Convolutional Neural Network (Redes Neuronales Convolucionales)

DAMPs: Damage-Associated Molecular Patterns (Patrones Moleculares Asociados al Daño)

DL: Deep Learning (Aprendizaje Profundo)

ECG: Electrocardiogram (Electrocardiograma)

ECV: Enfermedades Cardiovasculares

HTA: Hipertensión Arterial

IAM: Infarto Agudo de Miocardio

ICP: Intervención Coronaria Percutánea

IECA: Inhibidores de la Enzima Convertidora de Angiotensina

I/R: Isquemia-Reperfusión

LDL: Low-Density Lipoprotein (Lipoproteínas de Baja Densidad)

ML: Machine Learning (Aprendizaje Automático)

NLP: Natural Language Processing (Procesamiento de Lenguaje Natural)

PCSK9: Proproteína Convertasa Subtilisina/Kexina Tipo 9

RAG: Retrieval-Augmented Generation (Generación Aumentada por Recuperación)

RME: Registros Médicos Electrónicos

ROS: Reactive Oxygen Species (Especies Reactivas del Oxígeno)

SGLT2: Cotransportador Sodio-Glucosa Tipo 2

ST: Segmento ST del electrocardiograma

TAD: Terapia Antiplaquetaria Dual

TNF-α: Tumor Necrosis Factor Alpha (Factor de Necrosis Tumoral Alfa)

ViT: Vision Transformer

1. Introducción a las Enfermedades Cardiovasculares y el Infarto de Miocardio

1.1. Prevalencia de las enfermedades cardiovasculares

En 2023, las enfermedades cardiovasculares (ECV) siguieron siendo la principal causa de muerte a nivel mundial, con más de 17,9 millones de fallecimientos, lo que representa el 30% de todas las muertes globales, según la Organización Mundial de la Salud (OMS). Entre las principales causas de estas muertes se encuentran el infarto agudo de miocardio (IAM) y los accidentes cerebrovasculares, que en conjunto representan el 85% del total. (Organización Mundial de la Salud, 2023). Además, el envejecimiento de la población, junto con el aumento de factores de riesgo como la obesidad, la diabetes y el sedentarismo, ha contribuido a un incremento en la prevalencia de las ECV (Hariharan, y otros, 2022).

Estas cifras subrayan el gran desafío que las ECV suponen para los sistemas de salud a nivel global, particularmente en los países de ingresos bajos y medianos, donde ocurre más del 80% de las muertes por ECV, lo que pone de manifiesto las profundas disparidades en el acceso a la atención médica y a la prevención de estas enfermedades (Wuriea & Cappuccio, 2012)

En España, las ECV siguieron siendo la principal causa de muerte en 2020 (Sociedad Española de Cardiología., 2020). Esta patología representa un porcentaje significativo de la mortalidad total en el país, reflejando la necesidad de continuar avanzando en su prevención y tratamiento. Aunque en las últimas décadas las tasas de mortalidad por IAM han disminuido, gracias a los avances en tratamientos, atención médica y estrategias preventivas, estas enfermedades siguen siendo una prioridad en la salud pública del país (Ministerio de Sanidad, 2023). La detección temprana de las ECV es fundamental para proporcionar un tratamiento adecuado y personalizado, lo que puede mejorar significativamente la calidad de vida de los pacientes y reducir la mortalidad.

A pesar de la reducción en las tasas de mortalidad en países con recursos como España, las ECV siguen ocupando un lugar prioritario en la agenda de salud pública debido a su carga sobre los recursos sanitarios y su impacto en la calidad de vida de las personas afectadas. Destacar que, la implementación de políticas de prevención y promoción de hábitos saludables continúa siendo crucial para enfrentar el reto que representan las ECV en el futuro.

1.2. Mecanismos Moleculares y Celulares del Infarto de Miocardio.

1.2.1. Papel de la isquemia y reperfusión en el daño cardíaco

El IAM se produce cuando una arteria coronaria se obstruye, generalmente debido a la formación de una placa aterosclerótica o un trombo, esto conduce a una reducción o interrupción completa del flujo sanguíneo al músculo cardíaco (Heusch, 2023). Esta interrupción provoca una isquemia, es decir, una disminución en el suministro de oxígeno y nutrientes al tejido miocárdico, afectando a su función y viabilidad (Ferdinandy, Hausenloy, Heusch, Baxter, & Schulz, 2022).

Durante la isquemia, las células del miocardio experimentan hipoxia y alteraciones metabólicas que pueden conducir a la muerte celular si no se restablece el flujo sanguíneo oportunamente. La restauración del flujo sanguíneo, conocida como reperfusión, es esencial para salvar el tejido cardíaco en riesgo (Sanada, Komuro, & Kitakaze, 2011). La reperfusión se logra comúnmente a través de intervenciones como la intervención coronaria percutánea (ICP) o la administración de agentes trombolíticos (Dala, 2013).

Sin embargo, aunque la reperfusión es crucial para minimizar el daño isquémico, también puede desencadenar una lesión adicional conocida como lesión por isquemia-reperfusión (I/R) (Turer & Hill, 2010). Este fenómeno implica una serie de eventos patológicos que incluyen:

- **Estrés oxidativo**: la reintroducción de oxígeno genera un exceso de especies reactivas del oxígeno (ROS), que dañan componentes celulares (Chouchani, 2013)
- **Sobrecarga de calcio intracelular**: alteraciones en la homeostasis del calcio que activan enzimas degradativas (calpaínas) y promueven la muerte celular (Ong & Hausenloy, 2010)
- **Respuesta inflamatoria:** activación del sistema inmunitario que conduce a la infiltración de células inflamatorias. Este proceso será descrito con mayor detalle en el apartado siguiente, donde se profundizará en sus mecanismos y respuesta.

Investigaciones se han centrado tradicionalmente en desarrollar intervenciones que atenúen la lesión por I/R. Estas incluyen estrategias de precondicionamiento (Sanada, Komuro, & Kitakaze, 2011), así como terapias farmacológicas dirigidas a mitigar el estrés oxidativo, modular la respuesta inflamatoria y proteger las mitocondrias.

1.2.2. Inflamación y respuesta inmunitaria en el infarto de miocardio

El infarto de miocardio (IM) desencadena un complejo proceso inflamatorio y una respuesta inmunitaria que son cruciales para eliminar el tejido necrótico y reparar el tejido cardíaco dañado (Silvis, 2020). Este proceso comienza con la liberación de Patrones Moleculares Asociados al Daño (DAMPs) tras el evento isquémico, lo que activa la respuesta inflamatoria necesaria para la recuperación del corazón.

Las citoquinas y quimioquinas (moléculas que guían a las células inmunitarias) que se liberan en respuesta a los DAMPs, promueven el reclutamiento de neutrófilos desde el torrente sanguíneo hacia el sitio del infarto. Los neutrófilos son las primeras células inmunitarias en llegar al área lesionada, generalmente dentro de las primeras horas después del evento isquémico. Como se observa en el panel c) de la Figura 1, estas células migran al área lesionada y liberan especies reactivas de oxígeno (ROS) y enzimas proteolíticas, sustancias clave para eliminar los desechos celulares y patógenos. Sin embargo, debido a su naturaleza, también pueden dañar el tejido cardíaco sano si no se regulan adecuadamente. Por lo tanto, esta respuesta es un arma de doble filo: necesaria para limpiar el tejido dañado, pero potencialmente perjudicial si se descontrola (Liu, Li, Wang, Zhang, & Zhou, 2023), (Silvis, 2020).

Posteriormente, los monocitos son reclutados al sitio del infarto (panel c)), donde se diferencian en macrófagos. Estos macrófagos desempeñan un papel clave en el proceso inflamatorio y en la reparación del tejido dañado, adoptando diferentes fenotipos según las señales del entorno. Los macrófagos proinflamatorios (M1) predominan en las primeras fases del infarto, amplificando la respuesta inflamatoria para eliminar células dañadas, mientras que los macrófagos reparadores (M2) promueven la resolución de la inflamación y favorecen la regeneración del tejido en las fases posteriores. Este equilibrio entre macrófagos M1 y M2 es crucial para una adecuada recuperación del tejido miocárdico (Nahrendorf & Swirski, 2016). En la Figura 1, el panel d) muestra la presencia de macrófagos en el espacio pericárdico, donde contribuyen al proceso de reparación.

Tanto los neutrófilos como los macrófagos liberan citoquinas proinflamatorias, como TNF-α, IL-1β e IL-6. Estas citoquinas perpetúan la inflamación y atraen más células inmunitarias al área afectada, como se aprecia en el panel c) de la Figura 1, donde se muestra la interacción entre diferentes tipos de células inmunitarias, incluyendo monocitos, neutrófilos y células T. Si la producción de estas citoquinas no se regula correctamente, puede llevar a una inflamación descontrolada, agravando el daño tisular y afectando negativamente la función cardíaca a largo plazo (Prabhu & Frangogiannis, 2016).

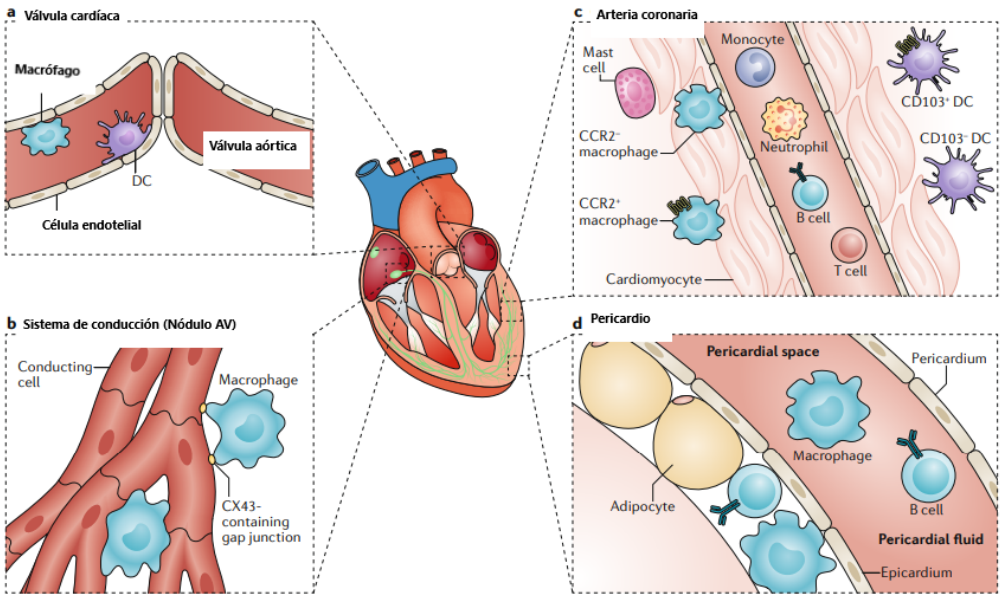


Figura 1. Proceso inflamatorio e inmunitario en el infarto de miocardio, mostrando el reclutamiento de neutrófilos y monocitos/macrófagos y su papel en la reparación del tejido dañado.

Fuente: (Sociedad Argentina de Cardiología, 2024)

Este proceso inflamatorio y de reparación es esencial para la recuperación del tejido cardíaco tras un infarto, pero requiere una regulación cuidadosa para evitar consecuencias negativas como daño tisular crónico o insuficiencia cardíaca. Las terapias actuales se enfocan en controlar este equilibrio, modulando la actividad de las citoquinas y de los macrófagos para optimizar la curación y minimizar los efectos adversos (Francisco & Del Re, 2023).

En resumen, la respuesta inmunitaria en el infarto de miocardio es un proceso complejo en el que diferentes células inmunitarias contribuyen tanto a la reparación como al potencial daño del tejido. Comprender y regular adecuadamente estos mecanismos es fundamental para mejorar los resultados clínicos y la calidad de vida de los pacientes afectados. facilitar la reparación del tejido afectado (Silvis, 2020).

1.3. Factores de riesgo y epidemiología del infarto de miocardio

El IM se relaciona estrechamente con diversos factores de riesgo, muchos de ellos prevenibles o modificables. Esto subraya la importancia crítica de las estrategias de prevención en salud pública para reducir la incidencia y el impacto de esta patología. A continuación, se describen los principales factores de riesgo asociados al IAM, basados en evidencia reciente.

1.3.1. Factores de riesgo del infarto de miocardio

Los factores de riesgo para el IAM se dividen comúnmente en modificables y no modificables. A continuación, se mencionan algunos de estos aspectos

Factores no modificables:

- **Edad**: El riesgo de IAM aumenta con la edad. Los hombres mayores de 45 años y las mujeres mayores de 55 años presentan un riesgo más elevado (Rodgers, Jones, Bolleddu, Vanthenapalli , & Panguluri , 2019).
- **Sexo**: Aunque los hombres tienen una mayor prevalencia de IAM en edades más tempranas, las mujeres presentan una mayor mortalidad tras un IAM, especialmente postmenopáusicas. (Rodgers, Jones, Bolleddu, Vanthenapalli , & Panguluri , 2019).
- **Historial familiar**: Una historia familiar de enfermedad coronaria prematura incrementa el riesgo de IAM (Bachmann, Willis, Ayers, Khera, & Berry, 2012).

Factores modificables:

- **Tabaquismo**: El hábito de fumar es uno de los principales factores de riesgo para el desarrollo del IAM. Aunque los mecanismos exactos por los cuales el tabaquismo contribuye al daño cardiovascular no se comprenden completamente, se sabe que afecta negativamente a la función endotelial y acelera el desarrollo de enfermedades cardiovasculares (Gallucci, Tartarone, Lerose, Lalinga, & Capobianco, 2020).

  El tabaco deteriora el endotelio (la capa interna de los vasos sanguíneos), lo que impide que este regule adecuadamente el flujo sanguíneo y se mantenga un estado antiinflamatorio. El tabaquismo promueve la inflamación crónica al aumentar los niveles de citoquinas proinflamatorias. Esta respuesta inflamatoria continuada da lugar a una disfunción endotelial, que contribuye al inicio y progresión de la aterosclerosis).

  Además, el tabaquismo favorece el estrés oxidativo, que incrementa la producción de ROS, agravando el daño celular y promoviendo la acumulación de placas de ateroma en las arterias. Cuando se combina con otros factores de riesgo, como la hipertensión, los efectos dañinos se potencian, acelerando el estrechamiento de los vasos sanguíneos y aumentando significativamente el riesgo de enfermedades cardiovasculares, incluido el IAM (Gallucci, Tartarone, Lerose, Lalinga, & Capobianco, 2020).

- **Dislipemia:** Es una alteración en los niveles de lípidos en la sangre, que puede incluir un aumento del colesterol total, de las *Low-Density Lipoprotein* (LDL), de los triglicéridos, o una disminución de las *High-Density Lipoprotein* (HDL). Uno de los subtipos más comunes es la hipercolesterolemia, que se caracteriza por niveles elevados de colesterol, especialmente del colesterol LDL. La hipercolesterolemia es un factor de riesgo clave para la enfermedad cardiovascular aterosclerótica (ASCVD). Los niveles elevados de colesterol LDL contribuyen directamente a la formación de placas en las arterias, lo que aumenta el riesgo de infarto de miocardio y accidente cerebrovascular (Pirillo, Casula, Olmastroni, Norata , & Catapano , 2021). Por esta razón, reducir el colesterol LDL es una estrategia fundamental en la prevención de la ASCVD. El tratamiento más común para disminuir los niveles de LDL son las estatinas, que se consideran la primera línea de defensa. Sin embargo, en algunos casos, este tratamiento se complementa con ezetimiba, un inhibidor de la proteína NPC1L1, que reduce la absorción de colesterol en el intestino (Tokgozoglu & Libby, 2022).

  Recientemente, han surgido nuevas terapias como los inhibidores de la proteína PCSK9, que han demostrado una reducción significativa del colesterol LDL y se discutirá en el punto 1.4.4. Aunque estos medicamentos son efectivos, su alto costo y la necesidad de administrarlos mediante inyecciones limitan su uso generalizado (Nishikido & Ray, 2018).

1.4. Estrategias de Prevención y Tratamiento del Infarto de Miocardio

1.4.1. Prevención Primaria: Enfocada en la reducción de los factores de riesgo antes de que se desarrolle la enfermedad.

La prevención primaria del IAM se basa en identificar y modificar los factores de riesgo cardiovasculares antes de que aparezca la enfermedad clínica. Esta estrategia es crucial para disminuir la incidencia de eventos coronarios y mejorar la salud cardiovascular de la población general (Arnett, Blumenthal, & Albert, 2019)

Cambios en el estilo de vida: constituyen la base de la prevención primaria. Una dieta saludable, como la dieta mediterránea, rica en aceite de oliva, frutas, verduras y pescados, ha demostrado ser efectiva para reducir el riesgo cardiovascular (Estruch, Ros, & Salas-Salvadó, 2018).Además, la actividad física regular, con un mínimo de 150 minutos semanales de ejercicio aeróbico de intensidad moderada, mejora la función endotelial, disminuye la presión arterial y favorece un perfil lipídico más saludable (World Health Organization, 2020).

Cesación tabáquica: es fundamental, ya que el tabaquismo sigue siendo uno de los principales factores de riesgo modificables. Estrategias como el asesoramiento conductual y las terapias farmacológicas (incluyendo sustitutos de la nicotina, bupropión o vareniclina) han demostrado aumentar las tasas de abandono del tabaco (Stead, Koilpillai, Fanshawe, & Lancaster, 2016).

Control de otros factores de riesgo: como la diabetes mellitus, que es un factor de riesgo modificable y de gran relevancia en la prevención del IAM. El manejo adecuado de la diabetes no solo es importante para el control glucémico, sino también para reducir el riesgo cardiovascular, con nuevos fármacos como los inhibidores del cotransportador sodio-glucosa tipo 2 (SGLT2) que han demostrado beneficios cardiovasculares adicionales (Mach, Baigent, & Catapano, 2020)

Manejo de la obesidad y el sedentarismo: es otro punto clave. Estrategias multidisciplinarias que involucren educación al paciente, apoyo psicológico y un seguimiento continuo han demostrado mejorar la adherencia a las recomendaciones y, como resultado, disminuir el riesgo cardiovascular global.

1.4.2. Prevención Secundaria: Estrategias dirigidas a pacientes que ya han sufrido un IM para prevenir eventos futuros.

La prevención secundaria del infarto se centra en reducir el riego de eventos cardiovasculares recurrentes en pacientes que ya han sufrido un IM. Esta estrategia es fundamental para mejorar la supervivencia al largo plazo y la calidad de vida de estos pacientes.

Los fármacos antiplaquetarios son fundamentales para prevenir nuevos problemas después de un evento cardiovascular, como un infarto. La terapia antiplaquetaria dual (TAD), que combina ácido acetilsalicílico (Aspirina) con un inhibidor del receptor P2Y12 (como el clopidogrel), es el tratamiento estándar tras un síndrome coronario agudo (SCA). Este tratamiento ayuda a evitar la formación de coágulos (trombos) en las arterias, lo que reduce el riesgo de futuros eventos trombóticos. La duración de la TAD varía según el riesgo del paciente de tener sangrados o nuevos coágulos, pero generalmente se recomienda mantenerla durante al menos 12 meses.

Las estatinas en dosis altas son esenciales para reducir los niveles de colesterol LDL y estabilizar las placas ateroscleróticas. Las guías actuales recomiendan objeticos de niveles de colesterol LDL menores de 55 mg/dl para pacientes de muy alto riesgo, como sería el caso de pacientes con IM

previo (Mach, Baigent, & Catapano, 2020). El uso de estatinas potentes como atorvastatina o rosuvastatina ha demostrado reducir significativamente la morbimortalidad cardiovascular.

Los betabloqueantes son recomendados tras un IM para reducir la carga isquémica, prevenir arritmias y disminuir la mortalidad. Estos fármacos reducen la frecuencia cardíaca y la presión arterial, disminuyendo el consumo de oxígeno del miocardio (Knuuti, Wijns, & Saraste, 2020).

Los cambios en el estilo de vida son igualmente importantes para la prevención secundaria, al igual que para la prevención primaria, el consumo de dietas saludables y la actividad física regular (siempre adaptada a las capacidades del paciente), mejoran la función endotelial. La rehabilitación cardíaca es una intervención multidisciplinar que combina ejercicio supervisado, educación y apoyo psicológico, demostrando reducciones significativas en la mortalidad y rehospitalizaciones (Dunlay, Pack, Thomas, Killian, & Roger, 2019).

La cesación tabáquica, el control estricto de la presión arterial y la glucemia son imperativos. Finalmente, se recomienda una evaluación y manejo de la adherencia al tratamiento, ya que la falta de cumplimiento terapéutico es un factor clave en la recurrencia de eventos cardiovasculares. El seguimiento regular y la educación al paciente son estrategias efectivas para mejorar la adherencia.

1.4.3. Tratamiento Agudo

El tratamiento del IAM tiene como objetico restaurar rápidamente el flujo sanguíneo en la arteria coronaria ocluida para limitar el daño miocárdico, preservar la función ventricular y mejorar la supervivencia tanto a corto como a largo plazo. Los avances en las estrategias de reperfusión y el manejo farmacológico han reducido significativamente la mortalidad asociada al IAM en las últimas décadas (Collet, Thiele, & Barbato, 2021).

La angioplastia coronaria (IPC) es el método de reperfusión preferido para pacientes con IAM con elevación del segmento ST (parte específica del electrocardiograma (ECG) que representa el intervalo entre la despolarización y la repolarización de los ventrículos cardíacos) cuando se puede realizar de manera oportuna por parte de un equipo experimentado, idealmente dentro de los 120 minutos desde el primer contacto médico (Ibanez, James, & Agewall, 2018). La ICP primaria implica la apertura mecánica de la arteria coronaria ocluida mediante angioplastia, como se muestra en la Figura 2, este procedimiento incluye el uso de un balón para dilatar el vaso sanguíneo y la colocación de un *stent* para mantener la arteria abierta. (Neumann, Sousa-Uva, & Ahlsson, 2019).

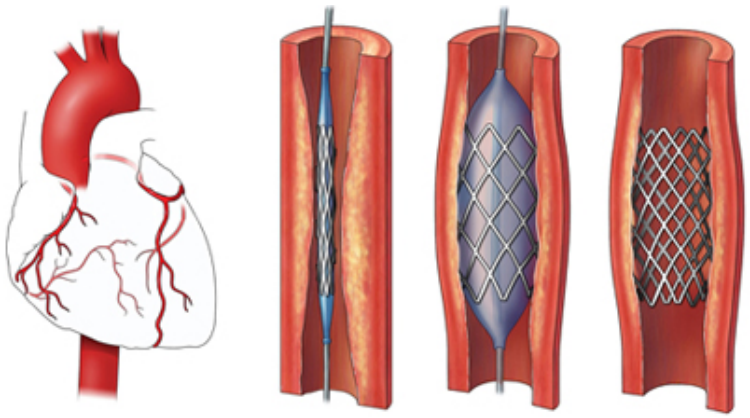


Figura 2: ICP primaria: procedimiento de apertura de la arteria coronaria mediante angioplastia con balón y colocación de stents.

Fuente: (ASSA, s.f.)

El tiempo es esencial en el tratamiento del IAM; se ha establecido el concepto de "el tiempo es músculo", ya que retrasos en la reperfusión se asocian con mayor mortalidad y complicaciones. Por tanto, se enfatiza la importancia de sistemas integrados de atención que permitan una rápida identificación, además de transporte y tratamiento de los pacientes con IAM (Ibanez, James, & Agewall, 2018).

Cuando la IPC primaria no está disponible dentro del tiempo recomendado, el tratamiento fibrinolítico se considera una alternativa efectiva su se administra dentro de las primeras 12 horas desde el inicio de los síntomas (Ibanez, James, & Agewall, 2018). Los agentes fibrinolíticos (medicamentos que se utilizan para disolver los coágulos de sangre (trombos) que obstruyen los vasos sanguíneos), presentan un mayor riesgo de complicaciones hemorrágicas en comparación con la ICP, además su eficiencia disminuye con el tiempo transcurrido desde el inicio de los síntomas.

El manejo del dolor con morfina puede ser necesario para aliviar el dolor torácico severo y reducir la ansiedad, aunque su uso debe ser cauteloso debido a posibles efectos adversos hemodinámicos y su interferencia con la absorción de los fármacos antiplaquetarios (Ibanez, James, & Agewall, 2018). La administración de oxígeno se recomienda solo en pacientes con saturación de oxígeno menor al 90%.

1.4.4. Terapias Emergentes

Las terapias emergentes en el tratamiento del IAM buscan abordar los mecanismos subyacentes de la enfermedad y promover la regeneración del tejido cardíaco dañado. Los avances en biotecnología biomédica han permitido el desarrollo de nuevas estrategias terapéuticas que prometen mejorar el pronóstico de los pacientes con IAM.

Inhibidores de PCSK9

Los inhibidores de la proteína convertasa subtilisina/kexina tipo 9 (PCSK9) han revolucionado el manejo de la hipercolesterolemia. Como se observa en la Figura 3, la PCSK9 regula la degradación de los receptores de LDL en el hígado. El proceso ilustrado muestra cómo la PCSK9 se une a los receptores de LDL-R en la superficie de las células hepáticas, promoviendo la degradación del complejo LDL-PCSK9 en los lisosomas, lo que disminuye la cantidad de receptores LDL disponibles y, por tanto, aumenta los niveles de colesterol LDL en sangre.

Los anticuerpos monoclonales como evolocumab y alirocumab inhiben esta interacción, evitando la degradación del receptor LDL. Esto aumenta la cantidad de receptores disponibles en la superficie de las células hepáticas, permitiendo que más partículas de LDL sean capturadas y eliminadas del torrente sanguíneo (Sabatine, Giugliano, & Keech, 2019). Estudios han demostrado que estos fármacos reducen el riesgo de eventos cardiovasculares mayores en pacientes con enfermedad cardiovascular aterosclerótica establecida, incluyendo aquellos con IAM previo (Schwartz, Steg, & Szarek, 2018).

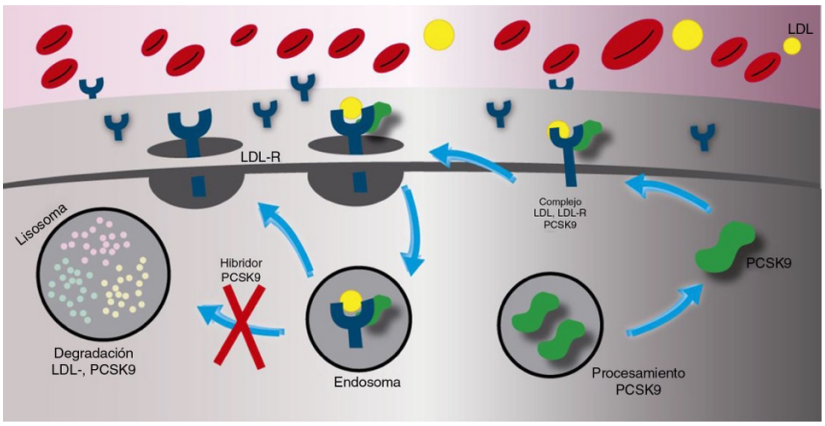


Figura 3. Mecanismo de acción de los inhibidores de PCSK9 en la reducción del colesterol LDL mediante regulación del receptor LDL

Fuente: (ScienceDirect, 2017)

Ingeniería de Tejidos y Biomateriales

La ingeniería de tejidos y el uso de biomateriales han permitido importantes avances en la reparación del tejido cardíaco tras un infarto de miocardio. Una de las innovaciones más destacadas es el desarrollo de stents degradables, también conocidos como stents biorreabsorbibles (BRS) (Huang & Ziqi, 2024). A diferencia de los stents convencionales de metal (mostrados en el panel A de la Figura 4), que permanecen de manera indefinida en las arterias tras su implantación, los stents degradables (ilustrados en el panel C) están diseñados para disolverse gradualmente y ser absorbidos por el cuerpo, lo que reduce las complicaciones a largo plazo, como la reestenosis intra-stent y la trombosis (Wu, y otros, 2021).

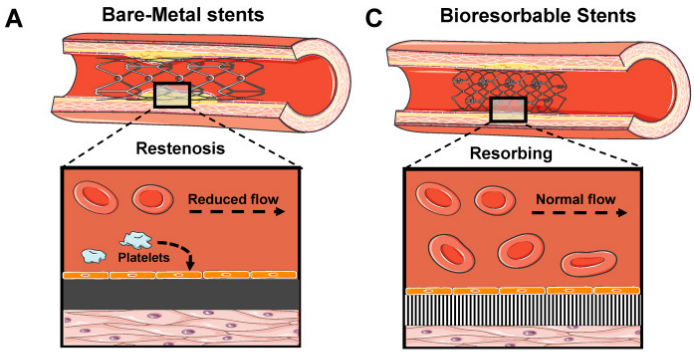


Figura 4. Comparación entre stents metálicos permanentes y stents biorreabsorbibles, destacando el proceso de degradación y la restauración del flujo sanguíneo normal.

Fuente: *(Elsevier, 2022)*

2. Introducción al Machine Learning (ML) y Deep Learning (DL) en el Ámbito Médico

2.1. Fundamentos de Machine Learning y Deep Learning

2.1.1. Conceptos Básicos y Definiciones

Definición de ML y DL

Para comprender los términos de ML y DL primeramente hay que comprender el de IA. IA es un campo de la informática que busca desarrollar sistemas capaces de realizar tareas que normalmente requieren inteligencia humana. Dentro de la IA, el ML es una rama fundamental que se centra en el desarrollo de algoritmos y modelos estadísticos que permiten a las máquinas

mejorar su desempeño en tareas específicas a través de la experiencia y el análisis de datos, sin necesidad de ser programadas explícitamente para cada acción (Ting Sim JZ, 2023).

El ML se subdivide en diferentes tipos, entre los más relevantes están el aprendizaje supervisado y el aprendizaje no supervisado (Rabbani N, 2022), siendo el aprendizaje supervisado el proceso utilizado en este trabajo. En el aprendizaje supervisado, los algoritmos se entrenan utilizando un conjunto de datos de entrada donde las salidas correspondientes son conocidas. Esto permite al modelo aprender a predecir o clasificar nuevos datos basándose en patrones previamente identificados (Ting Sim JZ, 2023). Por otro lado, el aprendizaje no supervisado implica que el modelo trabaja con datos sin etiquetas, buscando patrones o agrupaciones ocultas sin guía externa. Esto es especialmente útil para descubrir relaciones o estructuras desconocidas en los datos.

Por su parte, el DL es un subcampo del ML que usa redes neuronales artificiales (ANN) con múltiples capas para procesar y extraer características de datos complejos y no estructurados. Estas redes profundas son capaces de aprender representaciones jerárquicas de los datos, desde características de bajo nivel hasta conceptos de alto nivel, lo que las hace extremadamente eficaces en tareas como el reconocimiento de imágenes y el procesamiento del lenguaje natural (NLP) (Toma A, 2022).

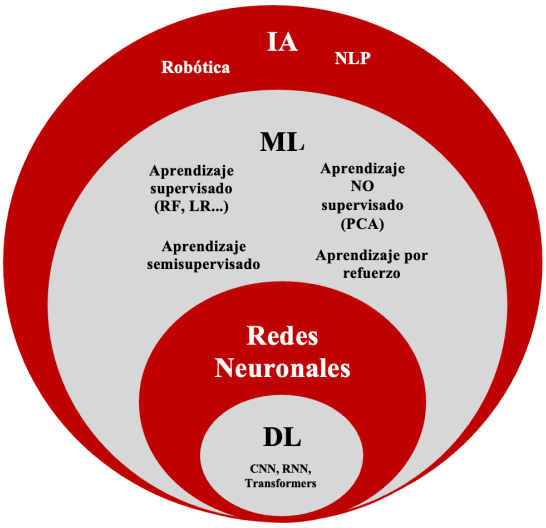


Figura 5. Diagrama de Subcampos de la Inteligencia Artificial

Fuente: (Elaboración propia, 2024)

Diferencias entre ML tradicional y DL

Pese a que el DL es una subrama del ML, existen diferencias significativas entre ambos en términos de cómo procesan los datos, su arquitectura y sus aplicaciones en el ámbito médico.

En cuanto al procesamiento de los datos, los algoritmos de ML tradicional suelen requerir una etapa previa de ingeniería de características, donde se seleccionan y extraen manualmente las variables más relevantes. Este proceso depende en gran parte del conocimiento actual y puede ser propenso a sesgos (Currie G, 2019). Por su parte, en el DL se usan redes neuronales profundas que son capaces de aprender directamente de los datos brutos (Toma A, 2022).

Mientras que el ML funciona bien con conjuntos de datos más pequeños y requiere menos poder computacional, el DL requiere grandes cantidades de datos para entrenar a modelos debido al gran número de parámetros en las redes neuronales profundas.

En términos de capacidad de modelado y rendimiento, el ML tradicional es adecuado para problemas donde las relaciones entre las variables son más simples y lineales. De esta forma, algoritmos como la regresión logística o el Random Forest son comunes y efectivos para estas situaciones. Sin embargo, su rendimiento puede verse limitado cuando se enfrentan a datos altamente complejos o no estructurados (Rabbani N, 2022).

Por el contrario, el DL es especialmente potente en el manejo de datos complejos y no estructurados, como es el caso de textos clínicos. Las redes neuronales profundas pueden modelar relaciones no lineales y capturar patrones intrincados en los datos, lo que ha llevado a avances significativos en áreas como el reconocimiento de imágenes médicas y el NLP (Toma A, 2022).

En cuanto a la interpretabilidad, los modelos de ML tradicional suelen ser más transparentes y fáciles de entender, lo cual es crucial en el ámbito médico para justificar las decisiones clínicas. Por otro lado, los modelos de DL son a menudo considerados como "cajas negras" debido a su complejidad y al gran número de parámetros internos, lo que dificulta entender cómo se llega a una predicción específica (Murdoch W. J., 2019)

2.1.2. Algoritmos y Modelos Relevantes en Medicina

Redes neuronales artificiales

Las Redes Neuronales Artificiales (ANN) son modelos computaciones inspirados en el cerebro humano (Han SH, 2018). Están compuestas por unidades interconectadas llamadas neuronas artificiales, que procesan información en paralelo y pueden aprender patrones complejos a partir de los datos. Cada neurona recibe una entrada, la procesa mediante una función de activación y transmite la salida a las neuronas de la siguiente capa (Philip J. Drew FRCS, 2000). Las ANNs son capaces de aproximar funciones no lineales y han sido ampliamente utilizadas en medicina para tareas de clasificación, predicción y reconocimiento de patrones.

El perceptrón es el modelo más sencillo de neurona artificial y fue introducido por Frank Rosenblatt en 1958. Es el componente básico de las redes neuronales y simula el comportamiento de una neurona biológica (Rosenblatt, 1958). Un perceptrón toma múltiples entradas, las pondera con pesos asociados, suma estos valores y aplica una función de activación para producir una salida. Este proceso se ilustra en la Figura 5.

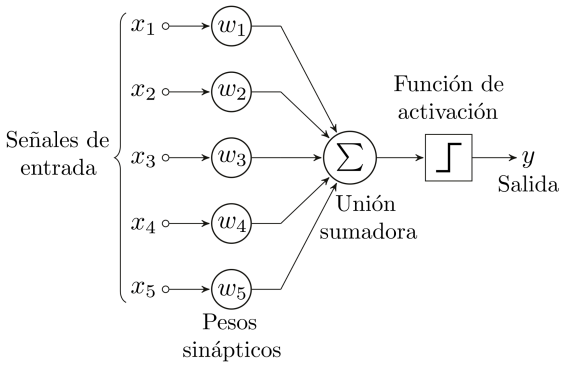


Figura 6. Esquema de un perceptrón, mostrando las señales de entrada, pesos sinápticos, unión sumadora y función de activación.

Fuente: (Wikipedia, 2017)

El perceptrón es capaz de resolver problemas de clasificación binaria que son linealmente separables. Sin embargo, presenta limitaciones al enfrentarse a problemas no linealmente separables, lo que llevó al desarrollo de redes neuronales multicapa (ANN, entre otras) que pueden manejar complejidades mayores.

Mención especial a las Redes Neuronales Convolucionales (CNN) que han revolucionado el campo del procesamiento de imágenes y han demostrado ser especialmente efectivas en el ámbito médico. Las CNN son un tipo de ANN diseñadas específicamente para trabajar con datos que tienen una estructura de cuadrícula, como las imágenes médicas (Sharma N. J., 2018). Estas redes utilizan convoluciones, aplicando un filtro o *kernel* sobre los datos en forma de matriz (generalmente en dos dimensiones) para extraer características relevantes, como bordes, texturas y patrones. Este enfoque es ideal para procesar imágenes médicas, ya que permite identificar detalles clínicos cruciales, como tumores o anomalías en los tejidos, de manera precisa y eficiente.

Redes Neuronales Transformers

Las Redes Nuruonales Transformers o simplemente Transformers, son modelos avanzados de DL inicialmente desarrollados para tareas de procesamiento del lenguaje natural (NLP), pero que han demostrado una gran versatilidad en el ámbito médico y biotecnológico (Vaswani, Shazeer, & Parmar, 2017). Su característica distintiva es el uso del mecanismo de atención (*attention mechanism*), que permite al modelo enfocarse en las partes más relevantes de los datos de entrada(Vaswani, Shazeer, & Parmar, 2017). Este enfoque está inspirado en procesos cognitivos humanos y es particularmente valioso en el contexto biomédico, donde se manejan grandes

volúmenes de datos complejos, como imágenes médicas, registros clínicos electrónicos y secuencias genéticas.

La arquitectura de un Transformer típico está compuesta por un encoder y un decoder. El encoder toma una secuencia de entrada y la transforma en una representación abstracta que captura las relaciones entre los diferentes elementos de los datos. El decoder utiliza esta representación para generar una secuencia de salida. Como se muestra en la Figura 5, cada bloque en el encoder y decoder incluye capas de atención y capas *feed-forward* totalmente conectadas, complementadas con mecanismos de normalización y conexiones residuales. Estos mecanismos mejoran la estabilidad del entrenamiento y facilitan la propagación de la información a través de múltiples capas (Vaswani, Shazeer, & Parmar, 2017).

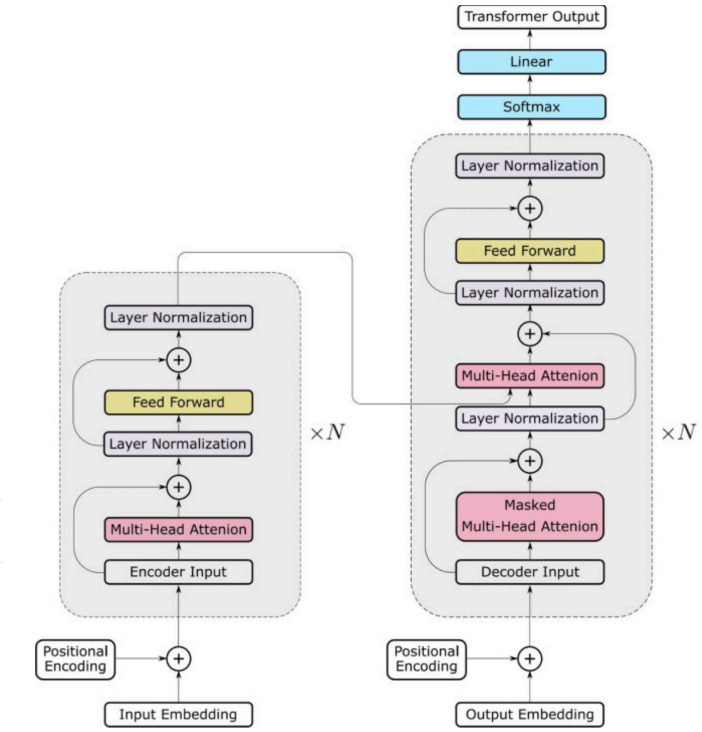


Figura 7. Arquitectura del modelo Trandformer con bloques encoder, decoder, destacado las capas de atención múltiple y conexiones residuales.

Fuente: (Aprende Machine Learning, 2024)

En el contexto médico, los Transformers han demostrado ser eficaces para extraer información relevante de registros médicos electrónicos, estudios clínicos y grandes volúmenes de texto científico, mejorando procesos de clasificación de enfermedades y predicción de resultados

clínicos. En particular, han sido adaptados para tareas de análisis de imágenes mediante modelos como el Vision Transformer (ViT) (Dosovitskiy, Beyer, & Kolesnikov, 2020).

Además, los Transformers han sido aplicados en el análisis de secuencias genómicas y proteómicas. Modelos como AlphaFold han utilizado arquitecturas basadas en Transformers para predecir la estructura tridimensional de proteínas a partir de su secuencia de aminoácidos, revolucionando el campo de la biología estructural y teniendo un impacto significativo en el desarrollo de nuevos fármacos y terapias (Jumper, Evans, & Pritzel, 2021).

2.2. Aplicaciones Clave de Machine Learning y Deep Learning en el Diagnóstico Médico

2.2.1. Deep Learning en el Análisis y Diagnóstico por Imágenes Médicas

El análisis y diagnóstico por imágenes médicas ha experimentado una transformación significativa con la introducción de técnicas de DL. Estas técnicas han permitido, entre otras cosas, avances en la detección de anomalías o patologías y en la segmentación del corazón completo, mejorando la precisión diagnóstica y facilitando la toma de decisiones clínicas en cardiología y otras áreas médicas.

Detección de anomalías o patologías

La detección automatizada de anomalías o patologías en imágenes médicas es crucial para el diagnóstico temprano y el tratamiento oportuno de diversas enfermedades. Las CNN, explicadas anteriormente, han demostrado ser especialmente efectivas en la clasificación y detección de anomalías en imágenes como radiografías, tomografías computarizadas (CT) y resonancia magnética (RM) (Litjens, Kooi, & Bejnordi, 2017).

En cardiología, las CNN han sido utilizadas para detectar enfermedades como las lesiones isquémicas. En un estudio reciente, se desarrolló una CNN profunda capaz de detectar infartos de miocardio en imágenes de RM con una precisión superior al 90% (Zhang, Chen, & Zhong, 2021). Este modelo utiliza capas convolucionales para extraer características relevantes y capas totalmente conectadas *(fully connected networks*) para clasificar las imágenes en normales o patológicas.

La detección de anomalías en imágenes de rayos X de tórax también ha sido potenciada por DL, especialmente en el contexto de la pandemia de COVID-19. Wang et al. (2020) desarrollaron un

modelo basado en DL que detecta neumonía por COVID-19 en imágenes de rayos X con alta precisión, asistiendo a los profesionales de la salud en el diagnóstico rápido de la enfermedad.

Segmentación de todo el corazón

La segmentación precisa del corazón en imágenes médicas es fundamental para evaluar la función cardíaca, planificar intervenciones y monitorear enfermedades. Las técnicas de DL han superado las limitaciones de los métodos tradicionales, proporcionando segmentaciones más exactas y eficientes.

Las CNN U-Net (arquitectura especializada para la segmentación de imágenes biomédicas) han sido ampliamente usadas para la segmentación de imágenes médicas debido a su arquitectura eficaz para capturar información tanto a nivel global como local, lo que les permite realizar segmentaciones precisas y detalladas en una variedad de aplicaciones biomédicas (Litjens, Kooi, & Bejnordi, 2017). En el contexto de la segmentación cardíaca, Oktay et al. (2018) propusieron la Anatomically Constrained Neural Network (ACNN) (arquitectura modificada a partir de una U-Net), que incorpora restricciones anatómicas (incorporación de conocimientos previos sobre la anatomía humana dentro de un modelo de DL) en el proceso de segmentación, mejorando la precisión al respetar la forma y estructura del corazón.

2.2.2. Machine Learning en el diagnóstico de Enfermedades

En el ámbito biomédico, ML ha emergido como una herramienta poderosa para el diagnóstico y pronóstico de enfermedades, permitiendo el análisis eficiente de grandes volúmenes de datos clínicos y biomédicos.

Clasificación de enfermedades a partir de datos clínicos

Una de las aplicaciones más destacadas de ML en medicina es la clasificación de enfermedades utilizando datos clínicos. Los algoritmos de ML pueden entrenarse para reconocer patrones complejos en datos de pacientes, como signos vitales, resultados de laboratorio, imágenes médicas y otras variables clínicas, para distinguir entre diferentes enfermedades o estados de salud. Los algoritmos aprenden de estos datos mediante técnicas de aprendizaje supervisado o no supervisado.

En el caso de enfermedades cardiovasculares, los algoritmos de ML han demostrado ser muy efectivos para clasificar arritmias cardíacas mediante el análisis de electrocardiogramas (ECG).

Un ejemplo notable es el estudio de Hannun et al. (2019), donde se desarrolló una red neuronal profunda capaz de clasificar más de una docena de tipos de arritmias con una precisión equivalente a la de un cardiólogo experto. Estos sistemas no solo detectan anomalías, sino que también pueden clasificar la gravedad de la afección, lo que permite una priorización más efectiva de los casos que requieren intervención médica urgente.

Estos sistemas de clasificación pueden mejorar la precisión diagnóstica y apoyar a los médicos en la toma de decisiones clínicas, reduciendo errores y optimizando los recursos sanitarios.

2.3. Ventajas y Desafíos del Uso de Inteligencia Artificial en la Medicina

La IA está transformando la cardiología, ofreciendo mejoras significativas en el diagnóstico de las distintas patologías. Entre las ventajas se destaca la precisión diagnóstica, la personalización, la capacidad de predicción y el ahorro de tiempo.

Sin embargo, su implementación presenta desafíos significativos. La privacidad y seguridad de los datos es una preocupación central, ya que el manejo de información sensible requiere cumplir con regulaciones estrictas para proteger la confidencialidad de los pacientes (Price & Cohen, 2019). Además, existen riesgos de sesgos en los algoritmos; si los datos de entrenamiento no son representativos, pueden perpetuar desigualdades en la atención médica. La calidad de los datos es crucial, pues información incompleta o errónea afecta la precisión de los modelos y puede conducir a diagnósticos incorrectos. También surgen cuestiones éticas y de responsabilidad, especialmente en situaciones críticas como el infarto de miocardio, donde decisiones basadas en IA pueden tener implicaciones vitales (Morley, y otros, 2020). Finalmente, integrar la IA en la práctica clínica requiere adaptaciones tecnológicas y formación del personal sanitario, lo cual puede ser complejo (He, Baxter, Xu, Zhou, & Zhang, 2019).

3. Hipótesis

3.1. Planteamiento de la hipótesis

Se plantea la hipótesis de que la implementación de un modelo de ML para la predicción temprana de enfermedades cardíacas, combinado con la integración de un modelo de DL como asistente biomédico, mejora significativamente la detección precoz y el manejo clínico del infarto de miocardio, y en general de todas las cardiopatías, en comparación con los métodos tradicionales. Esta aproximación integrada permitirá identificar a pacientes en riesgo con mayor precisión y ofrecer soporte avanzado en el diagnóstico y tratamiento, optimizando los resultados clínicos y reduciendo la mortalidad asociada.

4. Objetivos

4.1. Objetivo General

Desarrollar y evaluar un sistema integral basado en técnicas de ML y DL para la predicción temprana y asistencia en el diagnóstico de enfermedades cardíacas, con el fin de mejorar la detección precoz y el manejo clínico de ECV.

4.2. Objetivos Específicos

4.2.1. Procesar y analizar un conjunto de datos clínicos de pacientes para identificar las características más relevantes en la predicción del riesgo de enfermedades cardíacas, utilizando técnicas de preprocesamiento y visualización de datos.

4.2.2. Entrenar un modelo de ML, mediante Random Forest, para predecir el riesgo de enfermedad cardíaca en pacientes, mejorando la precisión en la identificación de individuos en riesgo.

4.2.3. Implementar una interfaz interactiva que permita la introducción de datos de pacientes y proporcione predicciones personalizadas sobre el riesgo de enfermedad cardíaca, facilitando su uso por parte de profesionales de la salud.

4.2.4. Desarrollar un sistema de recuperación y procesamiento de información médica, cargando y fragmentando documentos médicos especializados (como manuales de cardiología) y creando representaciones vectoriales mediante técnicas de NLP.

4.2.5. Crear un asistente biomédico basado en Deep Learning que, utilizando un modelo de lenguaje natural y el contexto recuperado, pueda responder preguntas relacionadas con la salud cardíaca y brindar asesoramiento médico experto.

4.2.6. Evaluar el rendimiento del modelo de Machine Learning y la eficacia del asistente biomédico, comparándolos con métodos diagnósticos convencionales y utilizando métricas como precisión, sensibilidad y especificidad.

5. Material y Métodos

5.1. Descripción del dataset empleado

En este estudio se utilizó el *Heart Failure Prediction Dataset* obtenido de la plataforma Kaggle (fedesoriano, 2021). Este conjunto de datos fue creado mediante la combinación de cinco *datasets* de enfermedades cardíacas previamente disponibles de forma independiente pero que no habían sido integrados en un solo conjunto de datos hasta ahora. La unificación de estos datasets proporciona una base de datos más amplia y representativa, lo que permite desarrollar modelos de ML más robustos y generalizables para la predicción de enfermedades cardíacas.

Los cinco *datasets* originales utilizados para la creación de este conjunto de datos son:

1. Cleveland Clinic Foundation Heart Disease Data: 303 observaciones.
2. Hungarian Institute of Cardiology, Budapest: 294 observaciones.
3. Switzerland University Hospital, Zurich: 123 observaciones.
4. Long Beach VA Medical Center: 200 observaciones.
5. Statlog (Heart) Data Set: 270 observaciones.

El *dataset* combinado inicial constaba de 1,190 observaciones. Sin embargo, se identificaron y eliminaron 272 observaciones duplicadas, resultando en un conjunto final de 918 observaciones correspondientes a pacientes únicos. Este proceso de depuración asegura la integridad y calidad de los datos utilizados en el estudio.

5.1.1. Características del dataset

El conjunto de datos incluye 11 variables clínicas comunes a los cinco datasets originales, las cuales son relevantes para la predicción de enfermedades cardíacas. A continuación, se describen las características incluidas:

1. Age: Edad del paciente en años (rango: 28-77 años).
2. Sex: Sexo biológico del paciente (1 = Masculino, 0 = Femenino).
3. ChestPainType: Tipo de dolor torácico experimentado:
    - TA (Typical Angina): Angina típica.
    - ATA (Atypical Angina): Angina atípica.
    - NAP (Non-Anginal Pain): Dolor no anginoso.
    - ASY (Asymptomatic): Asintomático.

4. RestingBP: Presión arterial en reposo medida en milímetros de mercurio (mm Hg).
5. Cholesterol: Nivel de colesterol sérico en miligramos por decilitro (mg/dl).
6. FastingBS: Glucemia en ayunas (1 = Nivel de glucosa en sangre ≥ 126 mg/dl, 0 = Nivel de glucosa en sangre < 126 mg/dl).
7. RestingECG: Resultados del electrocardiograma en reposo:
    - Normal: Sin anomalías.
    - ST: Presencia de anomalías en la onda ST-T (depresiones o elevaciones del segmento ST y/o inversión de la onda T).
    - LVH (Left Ventricular Hypertrophy): Hipertrofia ventricular izquierda probable o definitiva según criterios de Estes.
8. MaxHR: Frecuencia cardíaca máxima alcanzada durante una prueba de esfuerzo (latidos por minuto).
9. ExerciseAngina: Presencia de angina inducida por el ejercicio (1 = Sí, 0 = No).
10. Oldpeak: Depresión del segmento ST inducida por el ejercicio en relación con el reposo (medida en mm).
11. ST_Slope: Pendiente del segmento ST durante el ejercicio máximo:
    - Up: Ascendente.
    - Flat: Plana.
    - Down: Descendente.
12. HeartDisease: Variable objetivo que indica la presencia de enfermedad cardíaca (1 = Diagnóstico positivo de enfermedad cardíaca, 0 = Ausencia de enfermedad cardíaca).

5.1.2. Justificación de la Selección del Dataset

La elección de este dataset se basa en los siguientes criterios:

- **Tamaño y Representatividad**: Con 918 observaciones, este es uno de los datasets más grandes disponibles para el estudio de predicción de enfermedades cardíacas, lo que permite una mejor generalización de los modelos y resultados más robustos.
- **Diversidad de Datos**: Al combinar datos de diferentes fuentes y poblaciones, el dataset captura una variedad más amplia de perfiles de pacientes, lo que mejora la capacidad del modelo para aprender patrones relevantes y aplicables a diferentes grupos demográficos.
- **Variables Clínicamente Relevantes**: Las características incluidas son variables comúnmente utilizadas en la práctica clínica para la evaluación del riesgo cardiovascular, lo que facilita la interpretación y aplicabilidad de los resultados en entornos médicos reales.

5.2. Preprocesamiento de los datos

Antes de su utilización en el modelo de Machine Learning, se realizaron los siguientes pasos de preprocesamiento:

- **Conversión de Variables Categóricas**: Las variables categóricas como Sex, ChestPainType, RestingECG, ExerciseAngina y ST_Slope fueron convertidas a valores numéricos mediante asignación de códigos, facilitando su uso en algoritmos de aprendizaje automático.
- **Eliminación de Valores Nulos o Faltantes**: Se verificó la presencia de valores faltantes y, de ser necesario, se aplicaron técnicas de imputación o se eliminaron registros incompletos para mantener la integridad del análisis.
- **Normalización**: Se implementó la normalización de variables numéricas para mejorar el rendimiento y la convergencia de los algoritmos de ML.

5.3. Modelos de ML utilizados

En el presente estudio, se exploraron diversos algoritmos de ML para el desarrollo del modelo predictivo del riesgo de enfermedades cardíacas. La selección y evaluación de estos modelos se realizó con el objetivo de identificar aquel con el mejor desempeño en términos de precisión, sensibilidad y especificidad. Los algoritmos probados incluyen Regresión Logística, Árboles de Decisión, XGBoost y finalmente, Random Forest, que mostró el mejor rendimiento.

5.3.3. Random Forest

El Random Forest es un algoritmo utilizado para resolver problemas de clasificación y regresión. Funciona creando muchos árboles de decisión, donde cada uno se entrena con una parte del conjunto de datos, elegida al azar con repetición. Además, en cada punto de decisión dentro de los árboles, se selecciona aleatoriamente un grupo de características para hacer la división. Al combinar los resultados de todos estos árboles, ya sea votando en el caso de clasificación o promediando en el caso de regresión, el modelo se vuelve más preciso y robusto, reduciendo errores y mejorando su capacidad para generalizar a nuevos datos.

Ventajas de Random Forest

- **Precisión Elevada:** La capacidad de integrar múltiples árboles en el modelo reduce el riesgo de sobreajuste y mejora la precisión general.
- **Resistencia a Ruido y Outliers:** Random Forest es menos sensible a la variabilidad de los datos, lo que lo hace adecuado para conjuntos de datos con ruido o valores atípicos.
- **Importancia de las Características:** Una de las ventajas adicionales de Random Forest es que proporciona una medida de la importancia de las variables, lo que permite interpretar cuáles son las características más influyentes en el resultado.

Comparación con Otros Algoritmos:

1. **Regresión Logística**: alcanzó una exactitud de 0.8261. Aunque es un modelo interpretativo y simple, su desempeño fue inferior en comparación con Random Forest, debido a su limitación en la captura de relaciones no lineales.
2. **Árbol de Decisión**: El árbol de decisión individual obtuvo una exactitud de 0.8696. Si bien el modelo fue más preciso que la regresión logística, los árboles de decisión individuales son propensos a sobre ajustarse lo que limita su capacidad de generalización.
3. **XGBoost**: El modelo **XGBoost** alcanzó una **e**xactitud de 0.856, pero fue superado por Random Forest. XGBoost es conocido por su capacidad para manejar conjuntos de datos complejos, pero en este caso, Random Forest demostró un mejor rendimiento.

5.4. Arquitectura del algoritmo de Deep Learning

La arquitectura emplea una combinación de técnicas de procesamiento de NLP y recuperación de información avanzada, integrada con el uso de embeddings para la búsqueda eficiente de contexto en documentos médicos. Los componentes principales de la arquitectura se describen a continuación:

5.4.1. Modelo de lenguaje

El modelo de lenguaje usado en este sistema es el LLaMa 3.1 8b, un modelo de escala media alojado en el servidor de la empresa Lãberit Sistemas. LLaMA (Large Language Model Meta AI) es un modelo de generación de texto que usa la arquitectura Transformer para procesar secuencias de texto y generar respuestas coherentes a partir de las consultas del usuario. Este modelo es responsable de interpretar los datos clínicos proporcionados por el usuario y generar respuestas

basadas en los conocimientos previamente entrenados y los documentos recuperados. El modelo es de código abierto y puede encontrarse a través de la página de HugginFace (HugginFace, 2024).

5.4.2. Embeddings

Precisamente, para la recuperación de la información de los documentos se usan embeddings generados a partir del texto médico fragmentado en *chunks*. Los embeddings son representaciones numéricas de los fragmentos textuales, que permiten capturar la semántica y el contexto de las palabras en un espacio vectorial. Esto facilita la búsqueda y recuperación de los documentos más relevantes cuando se realiza una consulta (Espíndol, 2023). En este caso, se emplea un modelo especializado en la generación de embeddings semánticos en español, el modelo "jinaai/jina-embeddings-v2-base-es". Este modelo convierte los fragmentos de texto en vectores que luego se comparan para encontrar similitudes entre la consulta del usuario y los documentos médicos disponibles. Este modelo está igualmente disponible en la página de HugginFace (HugginFace, 2024).

5.4.3. FAISS (Facebook AI Similarity Search):

FAISS es la biblioteca de búsqueda vectorial empleada para indexar y buscar los embeddings de los textos médicos de manera eficiente. Utiliza búsqueda por similitud de vectores para identificar rápidamente los fragmentos de texto que más se acercan a la consulta del usuario en el espacio vectorial. Esto garantiza una recuperación rápida y precisa, incluso cuando se trata de grandes volúmenes de datos médicos.

5.4.4. Sistema de Recuperación de Información (RAG - Retrieval-Augmented Generation):

El sistema emplea un enfoque de Retrieval-Augmented Generation (RAG) para mejorar la generación de respuestas. Cuando el usuario realiza una consulta, se genera un embedding correspondiente que se utiliza para realizar una búsqueda en el índice de FAISS. Esta búsqueda por similitud de vectores (vector *similarity search*) permite recuperar los fragmentos de texto más relevantes en función de su proximidad semántica a la consulta. Los fragmentos recuperados se proporcionan como contexto adicional al modelo LLaMA 3.1B, que integra esta información para generar respuestas más precisas y basadas en evidencia.

5.5. Entrenamiento y Evaluación del Modelo de Machine Learning (*Random Forest*)

En este apartado se detalla el proceso de entrenamiento y evaluación del modelo de Machine Learning basado en Random Forest para la predicción del riesgo de enfermedades cardíacas.

5.5.1. Preparación y Preprocesamiento de los Datos

En cuanto a la lectura y conversión de los datos, se cargó el conjunto de datos desde un archivo CSV que contiene las observaciones combinadas de diferentes fuentes, como se describió en el apartado 5.1. Se realizó una conversión de variables categóricas a valores numéricos mediante mapeos específicos, lo cual es esencial para que los algoritmos de Machine Learning puedan procesar las variables de entrada. Los mapeos fueron los siguientes:

```
mapping_dict = {
    'Sex': {'M': 1, 'F': 0},
    'ChestPainType': {'ATA': 1, 'NAP': 2, 'ASY': 3, 'TA': 4},
    'RestingECG': {'Normal': 1, 'ST': 2, 'LVH': 3},
    'ExerciseAngina': {'Y': 1, 'N': 0},
    'ST_Slope': {'Up': 1, 'Flat': 2, 'Down': 3}
}
df.replace(mapping_dict, inplace=True)
```

En lo referente al análisis exploratorio de los datos, se generaron estadísticas, disponibles en el Anexo, del conjunto de datos para entender la distribución y características de las variables. Además, se verificó la presencia de valores duplicados y valores faltantes.

Para la visualización de los datos, se crearon mapas de calor para visualizar la correlación entre las variables y patrones significativos. Se generaron histogramas y gráficos de barras para las principales variables categóricas y su relación con la variable objetivo (HeartDisease), lo que ayudó a identificar distribuciones y posibles desequilibrios en las clases.

5.5.2. División del conjunto de datos

Para comenzar con el desarrollo del modelo predictivo, se procedió a separar el conjunto de datos en variables independientes (X) y la variable dependiente o variable objetivo (y),

donde y representa la presencia o ausencia de enfermedad cardíaca. Esta separación es esencial para entrenar un modelo supervisado de Machine Learning.

Se utilizó la función "`train_test_split`" de Scikit-learn para realizar la partición del conjunto de datos en conjunto de entrenamiento (80%) y conjunto de prueba (20%). Además, se estableció el parámetro "`random_state=42`" para garantizar la reproducibilidad de los resultados, de manera que en ejecuciones futuras los conjuntos de entrenamiento y prueba sean los mismos, lo que facilita la comparación y evaluación del modelo.

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

5.5.3. Escalado de las características

Antes de entrenar el modelo, se aplicó un proceso de escalado estandarizado a las características numéricas mediante el uso de la clase `StandardScaler` de Scikit-learn. Este paso es crucial, ya que garantiza que todas las características tengan una contribución equitativa en el modelo y mejora la eficiencia y la convergencia del algoritmo, evitando que las variables con valores más altos dominen el proceso de aprendizaje.

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

5.5.4. Entrenamiento del Modelo

El modelo fue entrenado utilizando el conjunto de entrenamiento escalado previamente y mediante el algoritmo de Random Forest:

```
model1 = RandomForestClassifier()
model1.fit(X_train_scaled, y_train)
```

5.5.5. Validación cruzada

Para asegurar la robustez del modelo y su capacidad de generalización en diferentes particiones del conjunto de datos, se realizó una validación cruzada de 5 pliegues (*5-fold cross-validation*) en

el conjunto de entrenamiento. La validación cruzada permite evaluar el rendimiento del modelo en diferentes subconjuntos del conjunto de entrenamiento, lo que ofrece una estimación más confiable de su capacidad predictiva.

```
scores = cross_val_score(model1, X_train_scaled, y_train, cv=5)
print(f"Cross-Validation Accuracy: {scores.mean():.4f}")
```

5.5.6. Evaluación del modelo en el conjunto de prueba

Una vez entrenado el modelo, se generaron predicciones sobre el conjunto de prueba (datos no vistos durante el entrenamiento) para evaluar su desempeño en la vida real. Se utilizó la función "`predict`" del modelo para realizar las predicciones, y posteriormente se calculó la exactitud del modelo utilizando "`accuracy_score`" de Scikit-learn.

```
y_pred = model1.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy en el conjunto de prueba: {accuracy:.4f}")
```

5.5.7. Importancia de las Características

En este caso, se calcularon las importancias de las características y se visualizaron mediante un gráfico de barras para facilitar su interpretación. Esto es particularmente útil en el ámbito médico, ya que ayuda a identificar qué variables son más influyentes en la predicción de enfermedades cardíacas.

```
importance = model1.feature_importances_
feature_importance_df = pd.DataFrame({
    'Feature': X.columns,
    'Importance': importance
}).sort_values(by='Importance', ascending=False)
```

5.5.8. Guardado y carga del modelo

Para garantizar la eficiencia en ejecuciones posteriores y evitar la necesidad de reentrenar el modelo, se guardó el modelo entrenado en un archivo utilizando `pickle.` De este modo, en

futuras ejecuciones, el modelo puede ser cargado directamente desde el archivo, ahorrando tiempo y recursos.

```
with open(model1_filename, 'wb') as f:
    pickle.dump(model1, f)
```

5.6. Librerías utilizadas

Para el presente trabajo, se han empelado diversas librerías de Python (módulos que contienen código predefinido que puede ser importado y utilizado) que han sido fundamentales para el procesamiento y análisis de datos, así como para la implementación de modelos de aprendizaje automático. A continuación, se presentan las definiciones y aplicaciones de las librerías más relevantes utilizados en el proyecto.

5.6.1. Pandas

Pandas es una librería de código abierto para Python que proporciona estructuras de datos y herramientas de análisis de datos de alto rendimiento y fáciles de usar. Es fundamental en el ecosistema de Python para la ciencia de datos y es ampliamente utilizada en análisis exploratorio, limpieza y preparación de datos (McKinney, 2011). Las características principales de pandas son:

- **Estructuras de datos: Series y DataFrame**: Permite trabajar con datos unidimensionales (Series) y tablas bidimensionales (DataFrame), donde cada columna puede ser de un tipo diferente.
- **Manejo de datos heterogéneos**: Trabaja con distintos tipos de datos como numéricos, fechas y cadenas de texto.
- **Indexación y selección**: Facilita el acceso a subconjuntos de datos usando etiquetas o posiciones.
- **Manipulación y transformación**: Herramientas para limpiar, transformar y agregar datos, y gestionar valores faltantes.

5.6.2. Langchain

LangChain es un framework de código abierto en Python diseñada para simplificar el desarrollo de aplicaciones basadas en modelos grandes de lenguaje (LLMs) (GitHub, 2024). Los LLMs, como LLaMa 3.1, se caracterizan por su capacidad de generar, analizar y comprender el texto, convirtiéndolos en herramientas valiosas para una amplia gama de aplicaciones.

LangChain se destaca como una herramienta clave para explotar el poder de los LLMs en la biotecnología biomédica, donde el acceso a la información contenida en grandes volúmenes de datos es fundamental. Ofrece una infraestructura flexible que permite el procesamiento de texto y la automatización de flujos de trabajo de manera eficiente.

5.6.3. FAISS

FAISS es otra biblioteca de código abierto desarrollada por Facebook AI Research que permite realizar búsquedas de similitud y agrupamiento de vectores densos de manera eficiente en grandes conjuntos de datos. Es especialmente útil en aplicaciones que involucran búsqueda de vecinos más cercanos (k-Nearest Neighbors, k-NN) en espacios de alta dimensión, lo cual es común en el procesamiento de lenguaje natural y en el manejo de datos embebidos (Facebook Research, 2024).

5.6.4. Transformers

Transformers, a parte de la arquitectura explicada en el punto 2.1.2, también es una biblioteca de código abierto desarrollada por Hugging Face que simplifica el uso de LLMs para una variedad de tareas de NLP. Esta biblioteca proporciona acceso a una amplia gama de modelos pre-entrenados, como el que se utiliza en este proyecto: LLaMA 3.1, proporcionado herramientas para cargar, entrenar y usar loa LLMs.

A través de Transformers, es posible cargar y utilizar el modelo preentrenado LLaMA 3.1 con solo unas pocas líneas de código.

6. Resultados

6.1. Rendimiento del modelo Random Forest en la predicción de riesgo cardiovascular

El modelo Random Forest demostró un rendimiento bueno en la predicción del riesgo cardiovascular en nuestro estudio. Específicamente, alcanzó una exactitud de 0.8678 durante la validación cruzada y mejoró a 0.8696 en el conjunto de prueba. Estos resultados superaron a los obtenidos por los demás algoritmos evaluados, destacando la eficacia del Random Forest en este contexto clínico.

6.2. Rendimiento del Modelo de Deep Learning (NLP) en la Extracción y Análisis de Información Médica

Los resultados fueron muy positivos. Al introducir una consulta, esta es correctamente vectorizada por el modelo de embeddings, que luego busca su homólogo en la librería vectorial de manera precisa, utilizando técnicas de búsqueda por similitud. Esto asegura que la información proporcionada por el sistema de RAG sea tanto correcta como relevante para la consulta médica. Además, se han incorporado metadatos que permiten un seguimiento detallado de la trazabilidad de las respuestas, lo que garantiza que la información recuperada del documento sea precisa y corresponda fielmente a los contenidos originales. Esta capacidad es especialmente útil en el ámbito médico, donde la confianza en la fuente y la precisión de los datos son fundamentales para la toma de decisiones clínicas.

El modelo presenta un tiempo de respuesta de 40 segundos, lo que, si bien es aceptable en ciertos contextos, aún puede mejorarse con optimizaciones adicionales. Una de las limitaciones actuales es el uso del modelo LLaMA 3.1 en lugar de la versión 3.2, que es más reciente y podría ofrecer un mejor rendimiento y actualización de datos, pero su implementación está limitada por la infraestructura disponible. El uso de una GPU ha permitido acelerar los tiempos de procesamiento, aunque el sistema también puede ejecutarse localmente. Sin embargo, es necesario continuar refinando el modelo para reducir los tiempos de respuesta y mejorar la eficiencia.

6.3. Visualización de los resultados: gráficos y tablas

En este apartado se presentan los gráficos y tablas generados a partir del análisis del conjunto de datos de ataques cardíacos. Estas visualizaciones son fundamentales para comprender tanto la

estructura del conjunto de datos como la relación entre las variables más importantes para la predicción de enfermedades cardíacas. Las figuras correspondientes se encuentran en el anexo del presente trabajo.

Importancia de las características

El modelo Random Forest, permite no solo realizar predicciones precisas, sino también identificar la importancia relativa de cada una de las características utilizadas en el análisis. Esta capacidad es particularmente útil para interpretar qué variables influyen más en la predicción de enfermedades cardíacas. En este caso, se ha observado que ST_Slope (pendiente del segmento ST), Oldpeak (depresión del segmento ST), y MaxHR(frecuencia cardíaca máxima) son los factores que más contribuyen al modelo predictivo. Estas características fueron fundamentales para la evaluación del riesgo, mientras que otras como el colesterol y el tipo de dolor en el pecho también aportaron valor, aunque en menor medida (ver Anexo 1).

Descripción del conjunto de datos

En Anexo 2 se presenta un mapa de calor que muestra las estadísticas descriptivas del conjunto de datos. Este gráfico incluye medidas como el conteo, media, desviación estándar, y los percentiles (25%, 50%, y 75%), lo cual nos proporciona una visión general sobre la dispersión y la variabilidad de las características. A continuación, se destacan algunos puntos relevantes observados:

- La edad de los pacientes tiene una media de 54 años, con una dispersión significativa (desviación estándar de 9.4 años), lo cual indica una población diversa en términos de edad, abarcando desde los 28 hasta los 77 años. Este rango sugiere la necesidad de ajustar el modelo para tener en cuenta el efecto de la edad en el riesgo de enfermedad cardíaca.
- El valor medio del colesterol se encuentra alrededor de 200 mg/dl, pero con una desviación estándar alta de 19, lo que indica variabilidad considerable en los niveles de colesterol de los pacientes. Además, el valor máximo de 564 mg/dl es notablemente elevado, lo cual podría ser un valor atípico y un posible riesgo significativo para los pacientes que presentan estos niveles tan altos.
- La variable Oldpeak, que indica la depresión del segmento ST, muestra valores negativos, con un mínimo de -2.6, lo cual es llamativo, ya que esta característica generalmente no tiene valores negativos. Esto sugiere posibles errores de medición o de ingreso de datos que podrían requerir una corrección durante el preprocesamiento.

- La frecuencia cardíaca máxima alcanzada (MaxHR) tiene una media de 140 ppm, con un rango de entre 60 y 202 ppm.
- Respecto a las variables categóricas, se observa que el mapeo de valores como Sex (0 para mujeres, 1 para hombres) muestra una representación mayoritaria de hombres (con una media de 0.79), indicando un sesgo de género hacia pacientes masculinos. Este sesgo es relevante, ya que puede influir en la capacidad del modelo para generalizar los resultados hacia las mujeres.
- En cuanto a la ST_Slope (pendiente del segmento ST), la mayoría de los valores tienden a agruparse alrededor de una pendiente plana (valor de 2), lo cual indica una prevalencia mayoritaria de este tipo de respuesta al ejercicio en la muestra.

Histogramas de las variables

En Anexo 3 se presentan los histogramas de las principales variables del conjunto de datos, ordenados de izquierda a derecha y de arriba a abajo:

1. **Edad**: La distribución de la edad es bastante simétrica, con la mayoría de los pacientes situados en el rango de 45 a 60 años. Esta simetría facilita el modelado, ya que los datos están bien distribuidos alrededor de la media, proporcionando una representación equilibrada de la población en términos de edad.
2. **Sexo**: La variable Sex muestra una distribución claramente desigual, con una mayor cantidad de valor 1 que valor 0.
3. **Tipo de dolor en el pecho**: Los tipos de dolor en el pecho se distribuyen de forma desigual, predominando el tipo asintomático. Esta desigualdad puede tener un impacto en el modelo, ya que las categorías de dolor menos comunes pueden ser subrepresentadas.
4. **Presión arterial en reposo**: La presión arterial en reposo tiene una distribución sesgada hacia la derecha, con la mayoría de los valores concentrados entre 120 y 140 mmHg, pero con algunos valores mucho más altos que se consideran extremos. Este sesgo sugiere la necesidad de aplicar transformaciones para evitar que los valores altos influyan desproporcionadamente en el modelo.
5. **Colesterol**: Esta variable también presenta un sesgo positivo, con un pequeño número de pacientes con niveles extremadamente altos. Además, algunos registros tienen valores de colesterol igual a 0, lo cual no es fisiológicamente posible y sugiere errores de entrada o datos faltantes. Estos valores podrían afectar negativamente al modelo si no se tratan adecuadamente.
6. **Azúcar en sangre en ayunas**: La mayoría de los pacientes tienen niveles normales de azúcar en sangre en ayunas (valor 0), mientras que un número menor tiene niveles

elevados (valor 1). Esto refleja una baja prevalencia de hiperglucemia, lo cual puede ser relevante para el análisis del riesgo cardiovascular.

7. **Electrocardiograma en reposo**: Esta variable tiene una distribución discreta, con tres categorías principales: Normal, Anormalidad en ST-T, e Hipertrofia ventricular izquierda. La mayoría de los registros se encuentran en la categoría "Normal", pero hay una proporción significativa de pacientes con anormalidades, lo cual podría ser un factor de riesgo importante.
8. **Frecuencia cardíaca máxima alcanzada**: La frecuencia cardíaca máxima presenta una distribución simétrica, con valores en su mayoría entre 130 y 170 ppm. Esta simetría facilita su uso en el modelado, ya que proporciona un rango equilibrado de valores sin sesgo significativo.
9. **Angina inducida por ejercicio**: La mayoría de los pacientes no presentan angina inducida por ejercicio (valor 0), mientras que un número menor de pacientes la presenta (valor 1).
10. **Depresión ST**: La variable Oldpeak, que indica la depresión del segmento ST, presenta valores tanto positivos como negativos, con una distribución sesgada hacia valores bajos. Algunos valores negativos podrían indicar posibles errores en la entrada de datos, lo cual debe ser considerado durante el preprocesamiento.
11. **Pendiente del segmento ST**: La mayoría de los pacientes presentan una pendiente plana o ascendentedel segmento ST, con menos pacientes teniendo una pendiente descendente. La pendiente descendente se considera un indicador de mayor riesgo cardiovascular, por lo que esta variable es crucial para el análisis.
12. **Enfermedad cardíaca**: Finalmente, la variable HeartDisease indica la presencia (valor 1) o ausencia (valor 0) de enfermedad cardíaca. La distribución muestra una proporción significativa de pacientes con la enfermedad.

Relación entre las principales variables y las enfermedades cardíacas

A continuación, se describen las principales observaciones de cada gráfico del Anexo 4:

1. **Sexo**: En el primer gráfico, se observa que el valor 1.0 (hombres) está representado de manera más frecuente que el valor 0.0 (mujeres), indicando un sesgo de género en el *dataset* hacia una mayor representación de hombres. Además, los hombres presentan una mayor prevalencia de enfermedades cardíacas en comparación con las mujeres. Esta mayor incidencia sugiere que los hombres tienen un mayor riesgo de padecer enfermedades cardíacas en esta muestra.

2. **Tipo de dolor en el pecho**: Los tipos de dolor están representados por valores 1.0, 2.0, y 3.0. El tipo 3.0 (dolor asintomático) tiene la mayor prevalencia de enfermedades cardíacas, seguido por los otros tipos de dolor. Esto indica que los pacientes con dolor asintomático están en mayor riesgo de enfermedad cardíaca, posiblemente porque la falta de síntomas claros puede retrasar el diagnóstico y el tratamiento.
3. **Electrocardiograma en reposo**: Se observa que el valor 1.0(electrocardiograma normal) tiene una prevalencia significativa tanto en pacientes con enfermedades cardíacas como sin ellas, siendo el grupo mayoritario en ambos casos. Sin embargo, los valores 2.0 y 3.0 (anomalías en ST-T e hipertrofia ventricular izquierda, respectivamente) muestran un desequilibrio ligero, donde la prevalencia de enfermedad cardíaca es algo mayor en comparación con los pacientes sin la enfermedad. Esto indica que los pacientes con anomalías en el ECG tienen una mayor probabilidad de presentar la enfermedad.
4. **Angina inducida por el ejercicio**: El gráfico de ExerciseAngina muestra que los pacientes sin angina inducida por el ejercicio (valor 0.0) son mucho más numerosos, y predominan aquellos sin enfermedad cardíaca. Por otro lado, los pacientes que presentan angina inducida por el ejercicio (valor 1.0) tienen una prevalencia mucho mayor de enfermedad cardíaca, lo cual sugiere que la angina durante el ejercicio es un indicador importante de problemas cardíacos.
5. **Pendiente del segmento ST**: En el gráfico de ST_Slope, los valores de la pendiente del segmento ST se representan por 1.0 (ascendente), 2.0 (plana), y 3.0 (descendente). Se observa claramente que los pacientes con una pendiente ascendente tienen una prevalencia considerablemente mayor de enfermedad cardíaca, lo cual está relacionado con problemas graves de perfusión cardíaca. En contraste, los pacientes con pendiente descendente muestran un menor riesgo de enfermedad, lo cual resalta la importancia de ST_Slope como un factor de predicción clave en el riesgo de enfermedad cardíaca.
6. **Colesterol**: Se observa la relación entre los niveles de colesterol y la presencia de enfermedad cardíaca. Aunque no se aprecia una separación clara, se nota una ligera tendencia a que los pacientes con niveles más altos de colesterol tienden a tener una mayor prevalencia de enfermedad cardíaca (barra naranja). Además, se observan valores extremadamente bajos, incluso 0, que probablemente reflejan errores en la entrada de datos o datos faltantes.
7. **Frecuencia cardíaca máxima alcanzada**: En el gráfico de MaxHR, se muestra que aquellos pacientes con una frecuencia cardíaca máxima más baja tienden a tener una mayor prevalencia de enfermedades cardíacas. Esto puede indicar una menor capacidad cardiovascular, la cual suele estar asociada con problemas subyacentes en el sistema cardiovascular y, por tanto, con un mayor riesgo de enfermedad.

6.4. Discusión de los resultados obtenidos en base a la literatura existente

Los resultados de este estudio permiten entender mejor las variables que tienen mayor peso a la hora de predecir ECV, así como las características demográficas y clínicas de la población analizada. Estos hallazgos son coherentes con estudios anteriores y ofrecen perspectivas interesantes sobre las variables clave en los modelos de predicción basados en aprendizaje automático.

En el modelo Random Forest, las variables más importantes para predecir enfermedades del corazón fueron la pendiente del segmento ST, la depresión del segmento ST y la frecuencia cardiaca máxima. Estos resultados concuerdan con investigaciones previas que también identifican estas variables como factores relevantes para predecir problemas cardiovasculares.

En la población estudiada, la edad promedio es de 54 años, con edades que oscilan entre los 28 y los 77 años. Este rango es representativo de la población en riesgo de enfermedades del corazón, aunque la amplia variabilidad sugiere la necesidad de ajustar los modelos para diferentes grupos de edad. Además, se observó un claro sesgo hacia pacientes masculinos, ya que el 79% de la muestra corresponde a hombres. Este desequilibrio está alineado con la mayor prevalencia de enfermedades del corazón en hombres, como se documenta en la literatura.

Se identificaron también posibles errores en los datos, como valores negativos en la variable `Oldpeak` y niveles de colesterol de cero. Los valores negativos en `Oldpeak` no tienen sentido fisiológico, lo que sugiere errores de medición o registro. Los valores de colesterol iguales a cero tampoco son fisiológicamente posibles, por lo que probablemente se trate de datos faltantes o errores de registro. Además, se observaron niveles de colesterol muy altos (hasta 564 mg/dl), lo cual indica la necesidad de manejar estos valores extremos con cuidado para evitar distorsiones en los resultados.

El análisis de la relación entre las variables clínicas y la presencia de enfermedades del corazón reveló que los hombres presentan una mayor prevalencia de la enfermedad en comparación con las mujeres, lo cual concuerda con los datos epidemiológicos actuales. Además, los pacientes con dolor torácico asintomático presentaron una mayor incidencia de enfermedad del corazón, lo que resalta la importancia de una evaluación clínica integral, incluso en ausencia de síntomas típicos.

En general, los resultados de este estudio confirman lo que se sabe de investigaciones previas y muestran la importancia de abordar los sesgos y las anomalías en los datos para contar con modelos predictivos precisos y útiles en la clínica. La integración de conocimientos médicos con

técnicas avanzadas de análisis de datos representa un camino adecuado para mejorar la predicción y prevención de enfermedades del corazón en el futuro.

7. Conclusión

7.1. Impacto de los modelos de Machine Learning y Deep Learning en la mejora de la predicción de enfermedades cardíacas

La aplicación de modelos de ML y DL está revolucionado el campo de la cardiología, entre otras áreas médicas, al ofrecer herramientas avanzadas para la predicción y diagnóstico de las enfermedades. Estos modelos tienen la capacidad de analizar grandes volúmenes de datos y detectar patrones complejos que podrían pasar desapercibidos mediante métodos tradicionales.

Es fundamental destacar que, aunque estas tecnologías están adquiriendo la capacidad de realizar tareas cada vez más sofisticadas, su papel debe ser visto como el de un facilitador y complemento para los profesionales de la salud, y no como un sustituto. La inteligencia artificial puede procesar y analizar datos a una velocidad y escala inigualables, pero carece del juicio clínico y la comprensión contextual que poseen los médicos. La combinación de la inteligencia humana y artificial permite una atención al paciente más efectiva y personalizada, donde los médicos pueden tomar decisiones informadas respaldadas por análisis de datos avanzados. La correcta colaboración entre personal médico y herramientas basadas en ML y DL puede dar lugar a diagnósticos más precisos, tratamientos personalizados, predicción de riesgos…

7.2. Contribución del estudio al diagnóstico precoz y manejo clínico personalizado

Este estudio ha explorado el uso de modelos de ML para la predicción de patologías y el uso de modelos de DL para el diagnóstico y asesoramiento clínico. A pesar de las limitaciones debidas a los datos e infraestructura, se ha evidenciado el potencial de estas tecnologías.

El modelo de ML implementado mostró capacidades buenas, pero limitadas debido a la cantidad insuficiente de variables y pacientes en el conjunto de datos tabulares. Se destaca la importancia de incorporar más variables clínicas y demográficas, así como un mayor número de pacientes, para mejorar el rendimiento de los modelos predictivos. La literatura existente apoya esta noción, señalando que la inclusión de datos multimodales y de gran volumen es crucial para el éxito de los modelos de ML en medicina (Esteva, y otros, 2019).

Además, se identificó la necesidad de un enfoque más integral que incluya datos de electrocardiogramas (ECG) procesados mediante redes neuronales convolucionales (CNN) pre-entrenadas y ajustadas a través de técnicas de *fine-tuning*. Esta integración permitiría capturar

características más complejas y sutiles asociadas con las enfermedades cardíacas, mejorando potencialmente la precisión diagnóstica.

En cuanto al uso de DL como asistente médico, los resultados fueron satisfactorios pero dependientes del modelo subyacente y su entrenamiento. Aunque no se utilizó DL para la predicción directa, su aplicación como herramienta de apoyo demostró ser valiosa. Sin embargo, la falta de integración en el flujo de trabajo médico actual limita su aplicabilidad inmediata, además, el modelo con "bajas capacidades" impide su implementación clínica directa. La colaboración entre desarrolladores y profesionales de la salud es esencial para diseñar sistemas que se ajusten a las necesidades clínicas y se integren de manera fluida en la práctica diaria.

7.3. Limitaciones del estudio y posibles mejoras futuras

El presente estudio enfrenta varias limitaciones que afectan la generalización y aplicabilidad de los resultados:

- **Conjunto de datos limitado**: La escasez de variables y pacientes limita la capacidad del modelo para aprender patrones complejos y generalizar a poblaciones más amplias.
- **Falta de datos multimodales**: La ausencia de datos de ECG y otros registros médicos electrónicos impide una comprensión más profunda de las características clínicas relevantes. La literatura sugiere que la incorporación de datos multimodales mejora significativamente la precisión de los modelos predictivos en cardiología.
- **Sesgos en los datos**: Se identificó un sesgo de género hacia pacientes masculinos, lo que puede afectar la capacidad del modelo para generalizar a pacientes femeninos. Además, la falta de equilibrio en otras variables demográficas puede introducir sesgos adicionales.
- **Aspectos técnicos**: Los modelos utilizados pueden no haber sido suficientemente complejos para capturar las relaciones no lineales en los datos. Además, no se exploraron exhaustivamente técnicas de preprocesamiento y selección de características que podrían mejorar el rendimiento.
- **Integración clínica limitada**: La falta de integración con los flujos de trabajo médicos actuales dificulta la adopción de estas herramientas en la práctica clínica. La ausencia de consideraciones sobre usabilidad y adaptación al entorno clínico limita su aplicabilidad.

7.4. Implicaciones para la investigación futura en el uso de inteligencia artificial en la medicina

La investigación y el desarrollo de modelos de IA aplicados al campo de la medicina, en particular en la cardiología, abren un amplio abanico de posibilidades para mejorar la predicción, diagnóstico y tratamiento de diversas patologías. Los resultados de este estudio resaltan varias implicaciones importantes para la investigación futura.

Primero, la necesidad de **integrar datos multimodales** se destaca como un factor crítico. La incorporación de datos provenientes de electrocardiogramas (ECG), imágenes médicas, registros electrónicos de salud y otras pruebas clínicas permitiría crear modelos de aprendizaje profundo (DL) más robustos. Estos datos enriquecidos pueden mejorar la capacidad de los algoritmos para capturar relaciones más complejas y sutiles en la fisiopatología de los pacientes. En estudios futuros, la inclusión de diferentes fuentes de información permitirá desarrollar modelos de IA que no solo sean predictivos, sino también explicativos, lo que facilitará una mejor comprensión de las causas subyacentes de enfermedades cardíacas.

Además, el **desarrollo de modelos de IA más complejos**, como CNNs y redes neuronales recurrentes (RNNs) aplicadas a datos temporales, tiene el potencial de capturar mejor las dinámicas clínicas a lo largo del tiempo. Estos enfoques podrían ser utilizados para monitorear a los pacientes en tiempo real, ofreciendo la posibilidad de predicciones más dinámicas y precisas que alerten sobre la evolución de condiciones críticas antes de que se manifiesten clínicamente.

Otro punto relevante es la importancia de la **mitigación de sesgos** en los modelos de IA. Este estudio ha identificado sesgos de género y posibles desequilibrios en otras variables demográficas, lo que indica que los futuros desarrollos deben abordar estos problemas mediante la inclusión de datos más representativos y la utilización de estrategias técnicas para equilibrar las predicciones. Los sesgos en la IA no solo pueden comprometer la precisión de los modelos, sino también introducir riesgos éticos significativos, lo que requiere una vigilancia constante y el desarrollo de estándares éticos sólidos para la IA en medicina.

Finalmente, la **usabilidad y aceptación** clínica son aspectos clave que deben abordarse en investigaciones futuras. La IA puede mejorar significativamente la práctica médica, pero su implementación depende de cómo se integre en el flujo de trabajo clínico. La colaboración entre desarrolladores y profesionales de la salud es esencial para diseñar interfaces y herramientas que sean intuitivas y eficientes. El uso de modelos de IA en la práctica médica requerirá esfuerzos adicionales para garantizar que los sistemas no solo sean precisos, sino también prácticos y

accesibles para el personal clínico. Esto incluye también el desarrollo de estrategias de capacitación para que los profesionales de la salud puedan utilizar estas herramientas de manera efectiva y segura.

7.5 Disponibilidad de datos

La disponibilidad de los datos es un aspecto clave en el desarrollo y evaluación de modelos de IA para aplicaciones médicas. El *Heart Failure Prediction Dataset* utilizado en este estudio está disponible públicamente en la plataforma Kaggle (fedesoriano, 2021), lo que permite la replicación de este estudio por otros investigadores y facilita la colaboración entre la comunidad científica. Este acceso abierto a los datos es fundamental para mejorar la transparencia, reproducibilidad y validez de los estudios basados en inteligencia artificial, lo que refuerza la importancia de las iniciativas de datos abiertos en el campo de la medicina.

No obstante, la investigación futura debe abordar las limitaciones asociadas con la calidad y representatividad de los datos. Si bien el conjunto de datos utilizado es una fuente valiosa, es necesario recopilar más datos clínicos de fuentes diversas para mejorar la robustez de los modelos predictivos. Además, debe considerarse la necesidad de disponer de datos multimodales, como registros electrónicos de salud, datos de imágenes médicas, señales de ECG y otra información clínica, lo que enriquecerá los modelos y permitirá desarrollar herramientas de inteligencia artificial más precisas y aplicables en una mayor variedad de escenarios clínicos.

Finalmente, es importante subrayar que el acceso a datos de pacientes debe estar en concordancia con normativas de protección de datos y privacidad como el Reglamento General de Protección de Datos (GDPR) en Europa o la HIPAA en Estados Unidos. La anonimización y manejo seguro de los datos es una prioridad en el diseño de cualquier investigación basada en datos médicos, lo que garantizará que las innovaciones tecnológicas no comprometan los derechos de los pacientes. En el futuro, será clave crear colaboraciones entre instituciones médicas y tecnológicas que permitan acceder a datos clínicos bajo acuerdos que prioricen la ética y la privacidad.

ANEXO I - Código Fuente

```
from flask import Flask, render_template, request, jsonify
import dill as pickle
import torch
from transformers import AutoTokenizer, AutoModel
from langchain.embeddings.base import Embeddings
from langchain_community.vectorstores import FAISS
from langchain.docstore.document import Document
from langchain.text_splitter import RecursiveCharacterTextSplitter
import PyPDF2
import requests
import os
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import StandardScaler
from sentence_transformers import SentenceTransformer
from metapub import PubMedFetcher
import os
from metapub import PubMedFetcher
import json
import traceback

app = Flask(__name__)
# Cargar el scaler y el modelo entrenado
scaler_filename = 'scaler.pkl'
model1_filename = 'modelo_entrenado.pkl'

# Verificar que scaler y modelo están cargados correctamente
try:
    with open(scaler_filename, 'rb') as f:
        scaler = pickle.load(f)
    print(f"{scaler_filename} cargado correctamente.")
    print(f"Tipo de scaler: {type(scaler)}")

    with open(model1_filename, 'rb') as f:
        model1 = pickle.load(f)
    print(f"{model1_filename} cargado correctamente.")
    print(f"Tipo de modelo: {type(model1)}")
except Exception as e:
    print("Error al cargar scaler o modelo:")
    traceback.print_exc()

# Configuración del tokenizer y modelo de embeddings
model_name = "jinaai/jina-embeddings-v2-base-es"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model_embeddings = AutoModel.from_pretrained(model_name, trust_remote_code=True)
device = torch.device("cpu")
model_embeddings.to(device)

from metapub import PubMedFetcher

os.environ["NCBI_API_KEY"] = "386c272f5229817d334a9ffc3919db050408"

# Cargar el modelo de embeddings para similitud semántica
semantic_model = SentenceTransformer('all-MiniLM-L6-v2')  # Modelo ligero de similitud semántica

# Configuración de la API de PubMed
os.environ["NCBI_API_KEY"] = "386c272f5229817d334a9ffc3919db050408"
fetch = PubMedFetcher()

# def extract_most_similar_phrase(query, num_articles=5):
#     """
#     Extrae la frase en la consulta que es más similar a un título de PubMed.

#     Args:
#         query (str): La consulta del usuario.
#         num_articles (int): Número de artículos a recuperar de PubMed.

#     Returns:
#         str: La frase de la consulta más similar al título en PubMed.
#     """
#     # Dividir la consulta en frases
#     query_phrases = query.split('. ')

#     # Realizar búsqueda en PubMed
#     keyword = query  # O extraer una palabra clave específica
#     pmids = fetch.pmids_for_query(keyword, retmax=num_articles)
#     if not pmids:
#         print(f"No se encontraron artículos para '{keyword}' en PubMed.")
#         return None

#     # Extraer títulos de los artículos recuperados
#     pubmed_titles = []
#     for pmid in pmids:
#         article = fetch.article_by_pmid(pmid)
#         if article and article.title:
#             pubmed_titles.append(article.title)
```

```
#     # Calcular la similitud entre cada frase de la consulta y cada título de PubMed
#     max_similarity = 0
#     most_similar_phrase = None

#     for phrase in query_phrases:
#         phrase_embedding = semantic_model.encode(phrase, convert_to_tensor=True)

#         for title in pubmed_titles:
#             title_embedding = semantic_model.encode(title, convert_to_tensor=True)
#             similarity_score = util.pytorch_cos_sim(phrase_embedding, title_embedding).item()

#             if similarity_score > max_similarity:
#                 max_similarity = similarity_score
#                 most_similar_phrase = phrase

#     print(f"Frase más similar: '{most_similar_phrase}' (Similitud: {max_similarity:.2f})")
#     return most_similar_phrase


def generar_referencias_pubmed(articles_data):
    """
    Genera una lista de referencias bibliográficas a partir de los datos de artículos obtenidos de PubMed.

    Args:
        articles_data (list): Lista de artículos con datos obtenidos de PubMed.

    Returns:
        list: Lista de referencias formateadas.
    """
    referencias = []
    for article in articles_data:
        if isinstance(article, dict) and article.get("Title"):
            autores = ', '.join(article.get("Author", ["Autores no especificados"]))
            year = article.get("Year", "Año no especificado")
            title = article.get("Title", "Título no disponible")
            journal = article.get("Journal", "Revista no especificada")
            volume = article.get("Volume", "Volumen no especificado")
            issue = article.get("Issue", "Número no especificado")
            link = article.get("Link", "#")

            referencia = (
                f'- {autores} ({year}). "{title}" en {journal} '
                f'(Vol. {volume}, N.º {issue}). '
                f'Disponible en: [PubMed]({link}).'
            )
            referencias.append(referencia)
        else:
            referencias.append("- Información del artículo no disponible en PubMed")

    return referencias



import time

def find_pubmed(keywords, num_of_articles=5, retries=3):
    """
    Realiza una búsqueda en PubMed utilizando palabras clave y recupera los artículos más relevantes.

    Args:
        keywords (list): Lista de palabras clave para la búsqueda en PubMed.
        num_of_articles (int): Número de artículos a recuperar por palabra clave.
        retries (int): Número de reintentos en caso de error.

    Returns:
        list: Lista de artículos con datos bibliográficos obtenidos de PubMed.
    """
    api_key = os.getenv("NCBI_API_KEY")
    if not api_key:
        print("⚠️ La API key no está configurada en el entorno.")
        return []

    articles_data = []
    for keyword in keywords:
        attempts = 0
        while attempts < retries:
            try:
                pmids = fetch.pmids_for_query(keyword, retmax=num_of_articles)
                if not pmids:
                    print(f"❌ No se encontraron artículos para '{keyword}' en PubMed.")
                    break

                for pmid in pmids:
                    article = fetch.article_by_pmid(pmid)
                    if article:
                        articles_data.append({
                            "pmid": pmid,
                            "Title": article.title,
                            "Abstract": article.abstract,
                            "Author": article.authors,
                            "Year": article.year,
```

```
                            "Volume": article.volume,
                            "Issue": article.issue,
                            "Journal": article.journal,
                            "Citation": article.citation,
                            "Link": f"https://pubmed.ncbi.nlm.nih.gov/{pmid}/"
                        })
                    break  # Exit retry loop if successful
                except Exception as e:
                    print(f"⚠️ Error al recuperar artículos para '{keyword}': {e}. Reintentando ({attempts +
1}/{retries})...")
                    time.sleep(2)  # Espera 2 segundos antes de intentar nuevamente
                    attempts += 1

    return articles_data

def generar_referencias_completas(referencias_generadas_por_modelo, referencias_pubmed):
    """
    Combina las referencias generadas por el modelo y las referencias obtenidas de PubMed en un solo texto.

    Args:
        referencias_generadas_por_modelo (list): Lista de referencias generadas originalmente.
        referencias_pubmed (list): Lista de referencias relacionadas obtenidas de PubMed.

    Returns:
        str: Texto con ambas secciones de referencias combinadas.
    """
    referencias_texto = "Referencias:\n"
    referencias_texto += "\n".join(referencias_generadas_por_modelo)

    referencias_texto += "\n\nBibliografía recomendada por PubMed:\n"
    referencias_texto += "\n".join(referencias_pubmed)

    return referencias_texto

def consulta_pubmed(question, num_of_articles=5):
    # Extraer palabras clave de la pregunta utilizando el modelo
    keywords = extract_keywords_with_model(question)

    # Buscar artículos en PubMed usando las palabras clave
    articles_data = find_pubmed(keywords, num_of_articles=num_of_articles)

    # Generar referencias bibliográficas
    referencias_pubmed = generar_referencias_pubmed(articles_data)

    if referencias_pubmed:
        print("Referencias generadas de PubMed:")
        print(referencias_pubmed)
    else:
        print("No se encontraron referencias para la búsqueda.")

    return referencias_pubmed







def extract_keywords_with_model(question):
    # Definir el prompt para pedirle al modelo que extraiga palabras clave
    prompt = f"Por favor, extrae las palabras clave médicas más relevantes de la siguiente pregunta. Lista las
palabras clave separadas por comas.\n\nPregunta: {question}\n\nPalabras clave:"

    # Preparar los mensajes para el asistente
    messages = [
        {"role": "system", "content": "Eres un asistente que extrae palabras clave médicas de las preguntas de los
usuarios."},
        {"role": "user", "content": prompt}
    ]

    # Utilizar la función send_chat del asistente para obtener las palabras clave
    keywords_text = assistant.send_chat(messages)

    # Procesar la respuesta para obtener una lista de palabras clave
    keywords = [word.strip() for word in keywords_text.split(',') if word.strip()]

    return keywords




# Definir la clase LangChain_Embeddings
class LangChain_Embeddings(Embeddings):
    def __init__(self, embedder: AutoModel):
        self.embedder = embedder
        super().__init__()

    def embed_documents(self, texts: list[str]) -> list[list[float]]:
```

```
        inputs = tokenizer(
            texts, return_tensors="pt", padding=True, truncation=True, max_length=512
        )
        with torch.no_grad():
            embeddings = self.embedder(**inputs).last_hidden_state.mean(dim=1)
        return embeddings.cpu().numpy().tolist()

    def embed_query(self, text: str) -> list[float]:
        inputs = tokenizer(
            [text], return_tensors="pt", padding=True, truncation=True, max_length=200
        )
        with torch.no_grad():
            embedding = self.embedder(**inputs).last_hidden_state.mean(dim=1)
        return embedding.cpu().numpy().tolist()[0]

def tryload(file_name):
    try:
        with open(file_name, 'rb') as f:
            return pickle.load(f)
    except (FileNotFoundError, AttributeError) as e:
        print(f"Error al cargar {file_name}: {e}")
        return None

def save(file_name, data):
    with open(file_name, 'wb') as f:
        pickle.dump(data, f)
    print(f"Archivo guardado correctamente en {file_name}")

def load_document_pypdf2(pdf_path):
    documents = []
    try:
        with open(pdf_path, 'rb') as pdf_file:
            reader = PyPDF2.PdfReader(pdf_file)
            for page_number in range(len(reader.pages)):
                page = reader.pages[page_number]
                content = page.extract_text()
                if content:
                    documents.append({
                        "page_content": content.strip(),
                        "metadata": {"page": page_number + 1}
                    })
    except Exception as e:
        print(f"No se pudo cargar el archivo PDF: {e}")
    return documents

# Función para dividir documentos en fragmentos más pequeños
def split_documents(documents):
    chunks = []
    splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=100)
    for doc in documents:
        doc_obj = Document(
            page_content=doc["page_content"], metadata=doc["metadata"]
        )
        chunks.extend([
            Document(page_content=chunk, metadata=doc["metadata"])
            for chunk in splitter.split_text(doc_obj.page_content)
        ])
    return chunks

# Cargar o crear el índice FAISS
faiss_index_filename = 'faiss_indexML2.pkl'
if os.path.exists(faiss_index_filename):
    with open(faiss_index_filename, 'rb') as f:
        faiss_index = pickle.load(f)
else:
    pdf_path = "Manual_AMIR_12da_ed_Cardiologia_y_Cirugi.pdf"
    document = load_document_pypdf2(pdf_path)
    chunks = split_documents(document)
    faiss_index = FAISS.from_documents(chunks, LangChain_Embeddings(model_embeddings))
    save(faiss_index_filename, faiss_index)

# Configurar el buscador con FAISS
faiss_retriever = faiss_index.as_retriever(search_kwargs={'k': 10})

# Función para obtener documentos relevantes
def get_documents(query, retriever):
    unique_docs = retriever.get_relevant_documents(query)
    documents_with_pages = []
    context = ""
    for doc in unique_docs:
        page_info = doc.metadata.get('page', 'Página desconocida')
        content = f"[Página {page_info}]\n{doc.page_content}\n\n"
        context += content
        documents_with_pages.append({
            "page": page_info,
            "content": doc.page_content,
        })
    return context, documents_with_pages
```

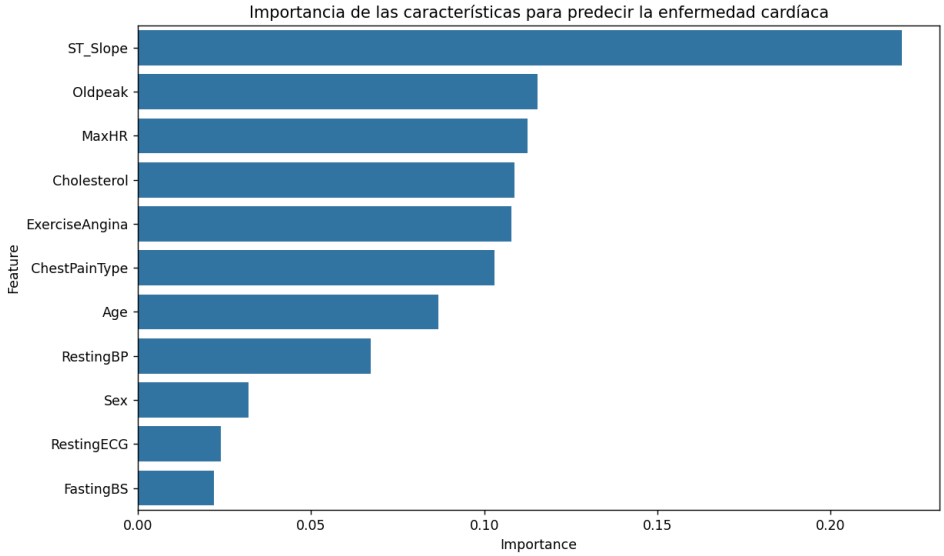
```
# Clase para manejar el asistente biomédico
class BiomedicalAssistant:
    def __init__(self, base_url, retriever, api_key=None):
        self.base_url = base_url
        self.retriever = retriever
        self.api_key = api_key
        self.sys_message = """
Eres un asistente especializado en enfermedades cardíacas. Tu tarea es proporcionar respuestas claras, objetivas y
basadas exclusivamente en los datos médicos y el contenido del documento proporcionado. Debes ofrecer interpretaciones
precisas para el diagnóstico y manejo de condiciones cardiovasculares, como infarto de miocardio, hipertensión,
arritmias, entre otras.

Instrucciones de Respuesta:
Contenido Relevante: Responde únicamente con la información relevante del documento o de fuentes científicas cuando no
haya datos en el documento.
Formato: Organiza las respuestas en párrafos bien estructurados. Usa negritas para conceptos clave y listas numeradas
cuando sea necesario.
Referencias: Al final de la respuesta, incluye una sección titulada "Referencias:" en un párrafo separado. Enumera las
referencias en este formato: "- [1] Documento proporcionado (Página X)." Solo usa referencias directas del documento
y, si no hay, utiliza el nombre de la fuente científica, indicando el artículo o fuente específica EN FORMATO APA Y EN
INGLÉS.

Respuestas Específicas:
Si la consulta es "¿qué enfermedad tiene el paciente?", utiliza exclusivamente los datos proporcionados en "Datos del
paciente".
Si la pregunta es "¿cuáles son los últimos avances...?", busca únicamente en artículos recientes de PubMed.
"""

    def send_request(self, prompt):
        headers = {'Content-Type': 'application/json'}
        if self.api_key:
            headers['Authorization'] = f'Bearer {self.api_key}'

        payload = {'prompt': prompt, 'system_message': self.sys_message}
        response = requests.post(f'{self.base_url}/chat', json=payload, headers=headers)

        if response.status_code == 200:
            return response.json()['response']
        else:
            raise Exception(f'Error en la solicitud: {response.status_code} - {response.text}')

    def send_chat(self, messages):
        headers = {'Content-Type': 'application/json'}
        if self.api_key:
            headers['Authorization'] = f'Bearer {self.api_key}'

        payload = {'messages': messages, "model": "modelo_vacio"}
        response = requests.post(f'{self.base_url}/v1/chat/completions', json=payload, headers=headers)

        if response.status_code == 200:
            return response.json()["choices"][0]["message"]["content"]
        else:
            raise Exception(f'Error en la solicitud: {response.status_code} - {response.text}')

    def generate_references(self, user_query):
        # Esta función debe ser definida para generar referencias desde el contenido del documento proporcionado.
        return []  # Placeholder hasta implementar la generación de referencias.

    def extract_keywords_with_model(self, user_query):
        # Prompt ajustado para pedir únicamente palabras clave
        prompt = (
            f"Extrae solo las palabras clave médicas más relevantes de la siguiente pregunta. "
            f"Lista las palabras clave en formato simple, separadas por comas, y no incluyas detalles
adicionales:\n\n"
            f"Pregunta: {user_query}\n\n"
            f"Palabras clave:"
        )

        # Utilizar el modelo para obtener la lista de palabras clave
        keywords_text = self.send_chat([{"role": "user", "content": prompt}])

        # Procesar la respuesta para extraer las palabras clave en una lista
        keywords = [word.strip() for word in keywords_text.split(',') if word.strip()]
        return keywords

    def find_pubmed(self, keywords, num_of_articles=5):
        # Buscar en PubMed con las palabras clave
        articles_data = []
        for keyword in keywords:
            try:
                pmids = fetch.pmids_for_query(keyword, retmax=num_of_articles)
                for pmid in pmids:
                    article = fetch.article_by_pmid(pmid)
                    if article:
                        articles_data.append({
                            "pmid": pmid,
                            "Title": article.title,
                            "Abstract": article.abstract,
                            "Author": article.authors,
                            "Year": article.year,
                            "Volume": article.volume,
```

```
                            "Issue": article.issue,
                            "Journal": article.journal,
                            "Citation": article.citation,
                            "Link": f"https://pubmed.ncbi.nlm.nih.gov/{pmid}/"
                        })
            except Exception as e:
                print(f"Error al recuperar artículos para '{keyword}': {e}")
        return articles_data

    def generar_referencias_pubmed(self, articles_data):
        referencias = []
        for article in articles_data:
            autores = ', '.join(article.get("Author", ["Autores no especificados"]))
            year = article.get("Year", "Año no especificado")
            title = article.get("Title", "Título no disponible")
            journal = article.get("Journal", "Revista no especificada")
            link = article.get("Link", "#")
            referencia = f'- {autores} ({year}). "{title}" en {journal}. Disponible en: [PubMed]({link}).'
            referencias.append(referencia)
        return referencias

    def generar_referencias_completas(self, referencias_documento, referencias_pubmed):
        texto_referencias = "Referencias:\n" + "\n".join(referencias_documento)
        texto_referencias += "\n\nBibliografía recomendada por PubMed:\n" + "\n".join(referencias_pubmed)
        return texto_referencias

    def generate_response(self, context: dict, user_query: str) -> str:
        try:
            retrieved_context, retrieved_docs = get_documents(user_query, self.retriever)
            if not retrieved_context:
                return "No se encontró contenido relevante en el documento para esta consulta."

            prompt_user = (
                f"Contexto del documento:\n{retrieved_context}\n\n"
                f"Datos del paciente: {context.get('data', '')}\n\n"
                f"Pregunta: {user_query}\n\n"
                "Instrucciones de Formato:\n"
                "Responde en párrafos separados. Al final, coloca 'Referencias' con las referencias extraídas, "
                "y luego 'Bibliografía recomendada por PubMed' con los artículos relacionados.\n\n"
            )
            messages = [{"role": "system", "content": self.sys_message}, {"role": "user", "content": prompt_user}]
            generated_text = self.send_chat(messages)
            return self.process_plain_text(generated_text, retrieved_docs)
        except Exception as e:
            return f"Error al generar la respuesta: {str(e)}"

    def process_plain_text(self, generated_text, retrieved_docs):
        return generated_text.strip()

# Ejecución de la lógica con la consulta recibida
def chat():
    raw_data = request.get_data(as_text=True)

    try:
        data = json.loads(raw_data)
        if 'message' not in data:
            return jsonify({'error': 'No se proporcionó una consulta.'}), 400

        user_query = data['message']
        print("Mensaje recibido:", user_query)

        # Generar referencias del contenido relevante
        referencias_documento = assistant.generate_references(user_query)

        # Extraer palabras clave
        keywords = assistant.extract_keywords_with_model(user_query)
        print(f"🔍 Palabras clave extraídas para PubMed: {keywords}")

        # Buscar artículos en PubMed
        articles_data = assistant.find_pubmed(keywords, num_of_articles=5)

        # Generar la bibliografía recomendada por PubMed
        referencias_pubmed = assistant.generar_referencias_pubmed(articles_data)

        # Combinar las referencias
        referencias_completas = assistant.generar_referencias_completas(referencias_documento, referencias_pubmed)

        # Crear contexto
        context = {"data": user_query, "referencias": referencias_completas}

        # Generar la respuesta final
        response_text = assistant.generate_response(context=context, user_query=user_query)

        return jsonify({'response': response_text})

    except json.JSONDecodeError:
        return jsonify({'error': 'La solicitud no contiene un JSON válido.'}), 400


model1 = tryload(model1_filename)
assistant = BiomedicalAssistant(base_url="http://172.20.8.120:2345", retriever= faiss_retriever)
```

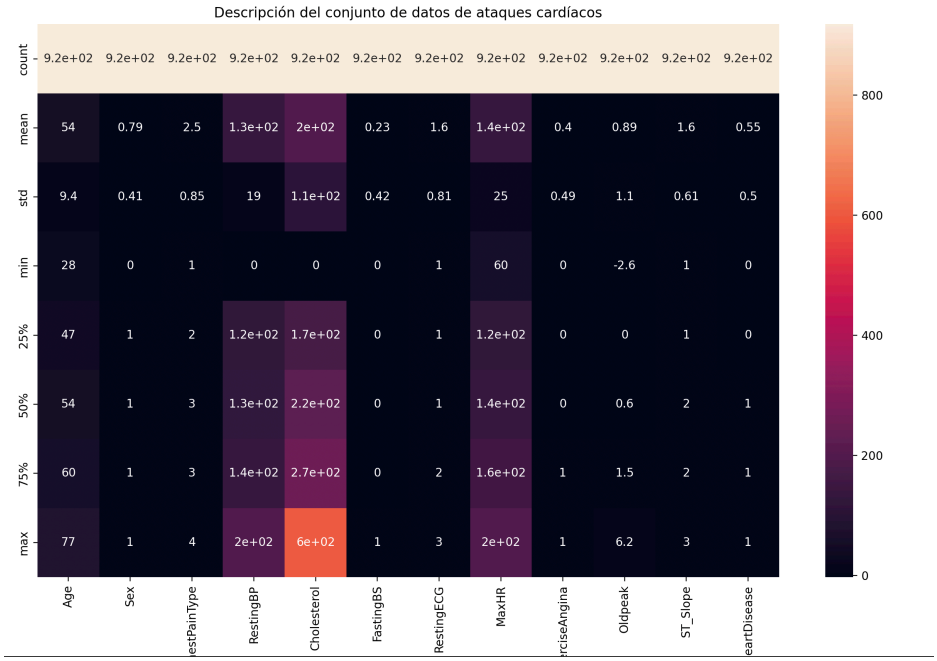
```
scaler = StandardScaler
import numpy as np
def predict_heart_disease(model1, patient_data):
    try:
        # Realizar la predicción
        prediction = model1.predict(patient_data)
        print("Resultado de la predicción (crudo):", prediction)  # Imprime la salida original

        # Verifica si es un array y si tiene el valor esperado (0 o 1)
        if isinstance(prediction, (list, np.ndarray)):
            prediction_value = prediction[0]
            print("Valor de predicción:", prediction_value)  # Imprime el valor de predicción específico
            return prediction_value
        else:
            print("Formato de predicción inesperado:", prediction)
            return None  # Devuelve None si el formato es inesperado
    except Exception as e:
        print("Error en la predicción del modelo:", str(e))
        return None


# Rutas de Flask
@app.route('/')
def index():
    return render_template('home.html')

@app.route('/buscar_pubmed', methods=['GET'])
def buscar_pubmed_route():
    keyword = request.args.get('keyword', 'cardiovascular disease')
    num_of_articles = int(request.args.get('num_of_articles', 10))
    print(f"🔍 Realizando búsqueda en PubMed con '{keyword}' y {num_of_articles} artículos")  # Debug
    articles = find_pubmed(keyword, num_of_articles)

    # Comprobar si se han encontrado artículos
    if articles:
        print(f"✅ {len(articles)} artículos encontrados")  # Debug
    else:
        print("⚠️ No se encontraron artículos o la API falló")  # Debug

    return jsonify(articles)

@app.route('/consulta_pubmed', methods=['GET'])
def consulta_pubmed():
    keyword = request.args.get('keyword', 'cardiovascular disease')
    num_of_articles = int(request.args.get('num_of_articles', 10))

    # Realizar la búsqueda en PubMed
    articles_data = find_pubmed(keyword, num_of_articles)

    # Generar referencias con datos reales
    referencias = generar_referencias_pubmed(articles_data)
    return jsonify({
        "referencias": referencias
    })

@app.route('/predict', methods=['POST'])
def predict():
    try:
        print("Obteniendo datos del formulario...")

        # Obtener datos del formulario
        patient_data = {
            'Age': request.form.get('age'),
            'Sex': request.form.get('sex'),
            'ChestPainType': request.form.get('cp'),
            'RestingBP': request.form.get('resting_bp'),
            'Cholesterol': request.form.get('cholesterol'),
            'FastingBS': request.form.get('fasting_bs'),
            'RestingECG': request.form.get('resting_ecg'),
            'MaxHR': request.form.get('max_hr'),
            'ExerciseAngina': request.form.get('exercise_angina'),
            'Oldpeak': request.form.get('oldpeak'),
            'ST_Slope': request.form.get('st_slope')
        }
        print(f"Datos del formulario recibidos: {patient_data}")

        # Convertir a DataFrame y asegurar que todos los valores son numéricos
        patient_df = pd.DataFrame([patient_data])
        patient_df = patient_df.apply(pd.to_numeric, errors='coerce')
        patient_df = patient_df.astype('float64')


        # Cargar el scaler en esta ruta específica
        with open('scaler.pkl', 'rb') as f:
            scaler = pickle.load(f)


        patient_scaled = scaler.transform(patient_df)
```

```
        # Realizar la predicción
        prediccion = predict_heart_disease(model1, patient_scaled)
        if prediccion == 1:
            prediction_text = 'Alto riesgo de enfermedad cardíaca'
        elif prediccion == 0:
            prediction_text = 'Bajo riesgo de enfermedad cardíaca'
        else:
            prediction_text = 'Error: resultado inesperado de la predicción'

        return jsonify({'prediction': prediction_text})

    except Exception as e:
        print("Error en la predicción:")
        traceback.print_exc()  # Imprime la traza completa del error
        return jsonify({'error': 'Ocurrió un error al predecir el riesgo.'}), 500
from flask import json
import json

@app.route('/chat', methods=['POST'])
def chat():
    raw_data = request.get_data(as_text=True)

    try:
        data = json.loads(raw_data)
        if 'message' not in data:
            return jsonify({'error': 'No se proporcionó una consulta.'}), 400

        user_query = data['message']
        print("Mensaje recibido:", user_query)

        # Generar referencias del contenido relevante
        referencias_documento = assistant.generate_references(user_query)

        # Extraer palabras clave
        keywords = assistant.extract_keywords_with_model(user_query)
        print(f"🔍 Palabras clave extraídas para PubMed: {keywords}")

        # Buscar artículos en PubMed
        articles_data = assistant.find_pubmed(keywords, num_of_articles=5)

        # Generar la bibliografía recomendada por PubMed
        referencias_pubmed = assistant.generar_referencias_pubmed(articles_data)

        # Combinar las referencias
        referencias_completas = assistant.generar_referencias_completas(referencias_documento, referencias_pubmed)

        # Crear contexto
        context = {"data": user_query, "referencias": referencias_completas}

        # Generar la respuesta final
        response_text = assistant.generate_response(context=context, user_query=user_query)

        return jsonify({'response': response_text})

    except json.JSONDecodeError:
        return jsonify({'error': 'La solicitud no contiene un JSON válido.'}), 400

if __name__ == '__main__':
    app.run(debug=False)
```

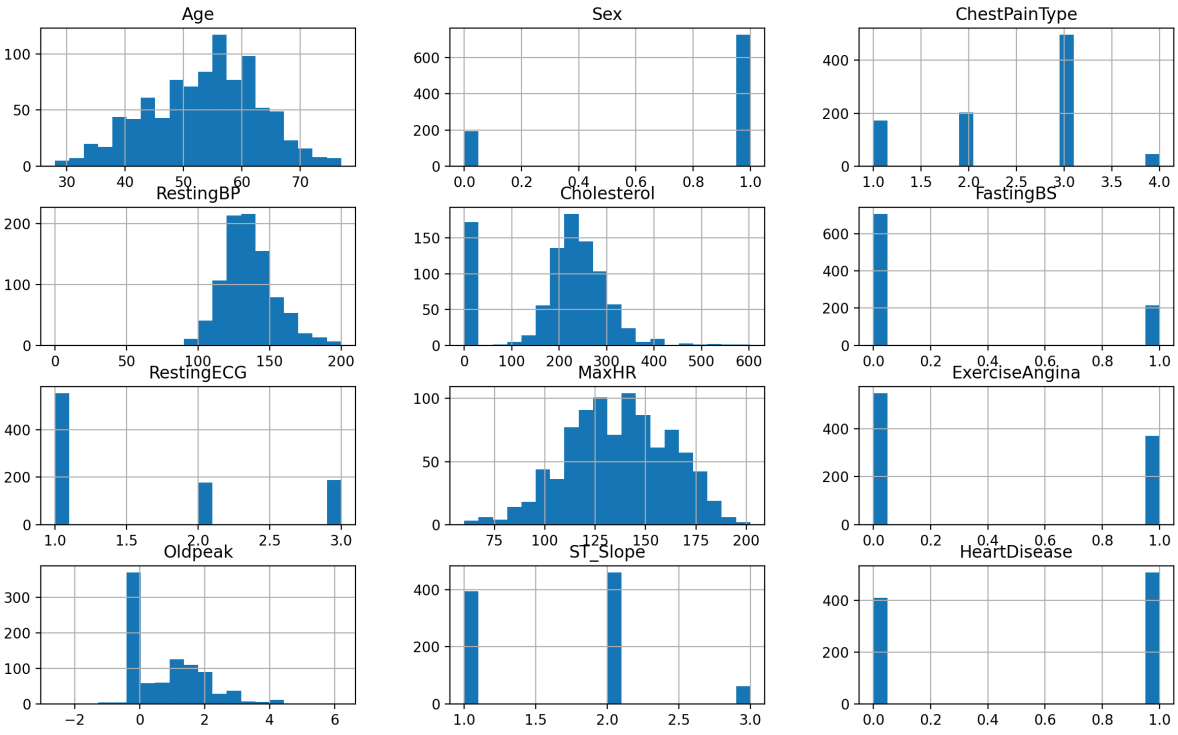
ANEXO II – Gráficas generadas



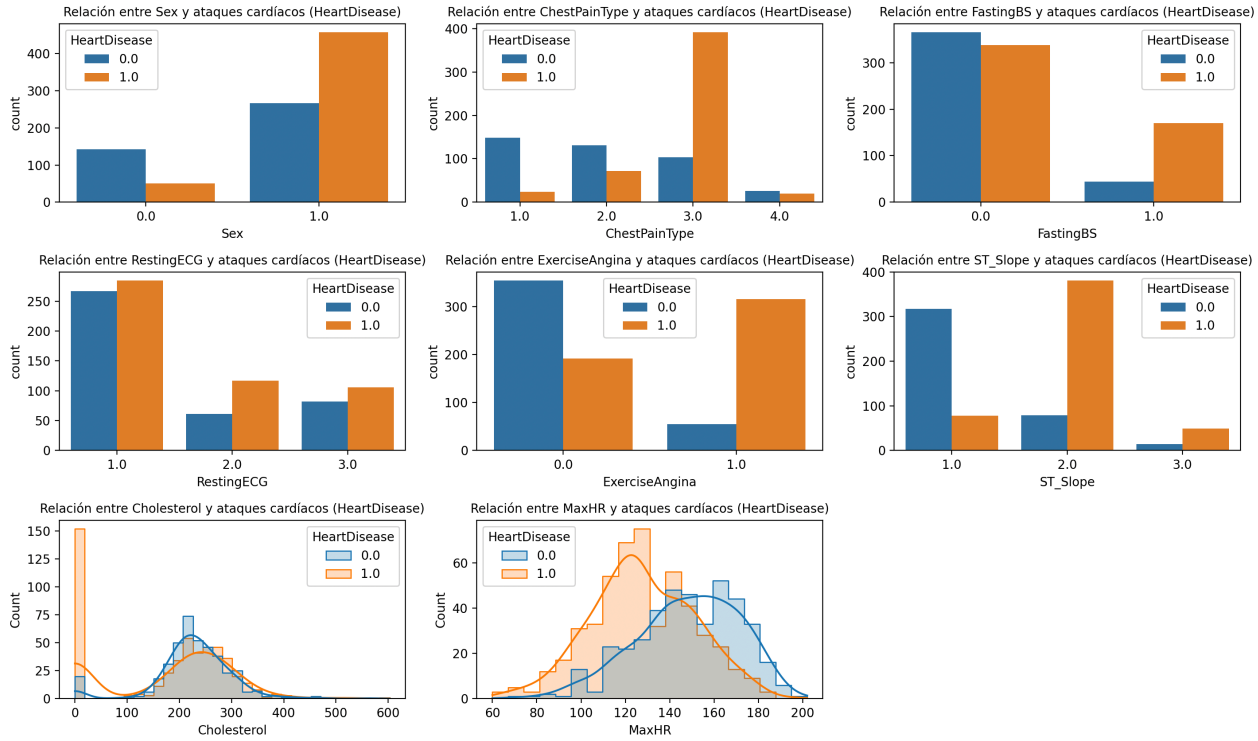
Anexo 1. Importancia de las características para predecir la enfermedad cardíaca según el modelo Random Forest



Anexo 2. Descripción estadística del conjunto de datos de ataques cardíacos



Anexo 3. Distribución de las variables del conjunto de datos de ataques cardíacos



Anexo 4. Relación entre variables clínicas y ataques cardíacos (HeartDisease)

Contribución a los Objetivos de Desarrollo Sostenible (Agenda 2030)

El presente trabajo tiene como objetivo principal mejorar el diagnóstico y manejo de las enfermedades cardiovasculares (ECV) mediante el uso de tecnologías avanzadas como el Machine Learning (ML) y el Deep Learning (DL). Estas patologías, que son una de las principales causas de muerte a nivel mundial, están íntimamente relacionadas con el objetivo número 3 de los Objetivos de Desarrollo Sostenible (ODS): "Garantizar una vida sana y promover el bienestar para todos en todas las edades".

El trabajo responde a este objetivo al desarrollar un sistema que predice el riesgo cardiovascular, ayudando a identificar de manera temprana a los pacientes que están en riesgo. Esto contribuye a la meta de reducir la mortalidad prematura por enfermedades no transmisibles a través de la prevención, el tratamiento y la promoción de la salud mental y el bienestar.

Además, este estudio se alinea con el ODS 3 al proporcionar una herramienta innovadora para mejorar la atención sanitaria mediante el uso de modelos predictivos basados en inteligencia artificial. Este enfoque no solo facilita una atención médica más personalizada y precisa, sino que también optimiza los recursos sanitarios y apoya a los profesionales de la salud en la toma de decisiones, promoviendo sistemas de salud más sostenibles y eficientes.

Por otro lado, al mejorar la capacidad de diagnóstico temprano y reducir la carga de las ECV, el trabajo también contribuye indirectamente a otros ODS como el ODS 10, que busca reducir las desigualdades dentro de los países y entre ellos. El uso de inteligencia artificial en la medicina tiene el potencial de democratizar el acceso a herramientas diagnósticas avanzadas, haciéndolas más accesibles y equitativas para diversas poblaciones.

En resumen, este trabajo contribuye de manera significativa a los ODS al avanzar en la comprensión y el manejo de las ECV, ofreciendo soluciones que pueden transformar los sistemas de salud y mejorar la calidad de vida de millones de personas en todo el mundo.

8. Bibliografía

Arnett, D., Blumenthal, R., & Albert, M. (2019). 2019 ACC/AHA Guideline on the Primary Prevention of Cardiovascular Disease. *Circulation*, 140(11).

Bachmann, J., Willis, B., Ayers, C., Khera, A., & Berry, J. (2012). Association between family history and coronary heart disease death across long-term follow-up in men: the Cooper Center Longitudinal Study. *Circulation*, 125(25):3092–8.

Cho, Y., Kwon, J., Kim, K.-H., Medina-Inojosa, J., Jeon, K.-H., Cho, S., & Lee, S. (2020). Artificial intelligence algorithm for detecting myocardial infarction using six-lead electrocardiography. *Scientific Reports*, 10, 20495.

Chouchani, E. (2013). Cardioprotection by S-nitrosation of a cysteine switch on mitochondrial complex I. *Nat Med*, 19(6):753-759.

Collet, J., Thiele, H., & Barbato, E. (2021). 2020 ESC Guidelines for the management of acute coronary syndromes in patients presenting without persistent ST-segment elevation. *European Heart Journa*, 42(14), 1289–1367.

Currie G, H. K. (2019). Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging. *J Med Imaging Radiat Sci*, 477-487.

Dala, J. (2013). Role of thrombolysis in reperfusion therapy for management of AMI: Indian scenario . *Indian Heart J*, 65(5): 566–585.

Dosovitskiy, A., Beyer, L., & Kolesnikov, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint*, arXiv:2010.11929.

Du , K.-L., Leung, C.-S., Mow, w., & Swamy, M. (2012). eptron: Learning, Generalization, Model Selection, Fault Tolerance, and Role in the Deep Learning Era. *Mathematicals*, 10(24), 4730.

Dunlay, S., Pack, Q., Thomas, R., Killian, J., & Roger, V. (2019). Participation in cardiac rehabilitation, readmissions, and death after acute myocardial infarction. *The American Journal of Medicine*, 127(6), 538-546.

Elgendy, I., Mahtta, D., & Pepine, C. (2019). Medical Therapy for Heart Failure Caused by Ischemic Heart Disease. *Circulation Research (Circ. Res.)*, 24:1520–1535.

Espíndol, G. (2023). ¿Qué son los embeddings y cómo se utilizan en la inteligencia artificial con python? *Medium*, 1.

Esteva, A., Krupel, B., A Nova, R., Ko , J., M Swetter, S., M Blau , H., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542:115–8.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., & Chou, K. (2019). A guide to deep learning in healthcare. *Nature medicine*, 25(1), 24-29.

Estruch, R., Ros, E., & Salas-Salvadó, J. (2018). Primary prevention of cardiovascular disease with a Mediterranean diet supplemented with extra-virgin olive oil or nuts. *New England Journal of Medicine*, 378(25).

Facebook Research. (2024). *GitHub*. Retrieved from FAISS: A library for efficient similarity search and clustering of dense vectors: https://github.com/facebookresearch/faiss

fedesoriano. (2021, September). *Kaggle*. Retrieved from Heart Failure Prediction: https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

Ferdinandy, P., Hausenloy, D., Heusch, G., Baxter, G., & Schulz, R. (2022). Interaction of Cardiovascular Risk Factors, Comorbidities, and Comedications with Ischemia/Reperfusion Injury and Cardioprotection. *Pharmacol Rev*, 74(2):639-684.

Fernández-Avilés, F., Sanz-Ruiz, R., & Climent, A. (2018). Global position paper on cardiovascular regenerative medicine. *European Heart Journal*, 38(33), 2532-2546.

Francisco, J., & Del Re, D. (2023). Inflammation in Myocardial Ischemia/Reperfusion Injury: Underlying Mechanisms and Therapeutic Potential. *Antioxidants (Basel)*, 12(11): 1944.

Gallucci, G., Tartarone, A., Lerose, R., Lalinga, A., & Capobianco, A. (2020). Cardiovascular risk of smoking and benefits of smoking cessation. *Journal of Thoracic Disease*, 12(7), 3866–3876.

Gao, Y., Zhou, M., Metaxas, D., & Chen, B. (2021). UTNet: A hybrid transformer architecture for medical image segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 52–61.

GitHub. (2024). *langchain-ai*. Retrieved from langchain: https://github.com/langchain-ai/langchain

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. In I. Goodfellow, *Deep Learning* (pp. 167-171). Cambridge, Massachusetts, USA: MIT Press.

Grundy, S., Stone, N., & Bailey, A. (2019). 2018 AHA/ACC/AACVPR/AAPA/ABC/ACPM/ADA/AGS/APhA/ ASPC/NLA/PCNA Guideline on the Management of Blood Cholesterol. *Journal of the American College of Cardiology*, 73(24).

Gulshan, V., Peng, L., Coram, M., C Stumpe, M., Wu, D., Narayanaswamy, A., . . . Webster, R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316:2402–10.

Han SH, K. K. (2018). Artificial Neural Network: Understanding the Basic Concepts without Mathematics. *Dement Neurocogn Disord*, 83-89.

Hannun, A., Rajpurkar, P., & Haghpanahi, M. (2019). Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nat Med25*, 65–69.

Hariharan, R., Odjidja, E., Scott, D., Shivappa, N., Hébert, J., Hodge, A., & de Courten, B. (2022, Enero(Jan)). The dietary inflammatory index, obesity, type 2 diabetes, and cardiovascular risk

factors and diseases. *Obesity Reviews*. Retrieved from Obesity Reviews.: https://doi.org/10.1111/obr.13349

Haykin, S. (2008). *Neural Networks and Learning Machines.* Upper Saddle River, Nueva Jersey: Pearson.

He , K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, 770-778.

He, J., Baxter, S., Xu, J., Zhou, X., & Zhang, K. (2019). he practical implementation of artificial intelligence technologies in medicine. *Nature Medicine*, 25(1), 30–36.

He, K. G. (2023). Transformers in medical image analysis. *Intelligent Medicine*, 59-78.

Heusch, G. (2023). Reperfusion Injury: Pathophysiology, Phenotyping, and Clinical Implications. *Am J Physiol Heart Circ Physiol*, 324(1).

Huang , Y., & Ziqi, D. (2024). Biomaterials for cardiovascular diseases. *Biomedical Technology*, 1-14.

HugginFace. (2024). *jinaai*. Retrieved from jina-embeddings-v2-base-es: https://huggingface.co/jinaai/jina-embeddings-v2-base-es

HugginFace. (2024). *Meta Llama*. Retrieved from meta-llama/Llama-3.1-8B: https://huggingface.co/meta-llama/Llama-3.1-8B

Ibáñez, B. H. (2015). Evolving therapies for myocardial ischemia/reperfusion injury. *Journal of the American College of Cardiology (J Am Coll Cardiol)*, 65(14), 1454-1471.

Ibanez, B., James, S., & Agewall, S. (2018). 2017 ESC Guidelines for the management of acute myocardial infarction in patients presenting with ST-segment elevation. *European Heart Journal*, 9(2), 119–177.

Isensee, F., Jaeger, P., & Kohl, S. (2021). nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nat Methods*, 18, 203–211 (2021).

Jumper, J., Evans, R., & Pritzel, A. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873), 583–589.

Kim, M., Yung , J., Cho, Y., Shin, K., Jang, R., Bae, H., & Kim, N. (2019). Deep Learning in Medical Imaging. *Neurospine*, 16(4):657-668.

Kjeldsen, S. (2018). Hypertension and cardiovascular risk: General aspects. *Pharmacol Res*, 129:95-99.

Knuuti, J., Wijns, W., & Saraste, A. (2020). 2019 ESC Guidelines for the diagnosis and management of chronic coronary syndromes. *European Heart Journal*, 41(3), 407-477.

Krijnen PA, N. R. (2002). Apoptosis in myocardial ischaemia and infarction. *J Clin Pathol.*

Lakhani P, S. B. (2017). deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*, 284:574–82.

Li Z, C. Y. (2022). ision transformer-based weakly supervised histopathological image analysis of primary brain tumors. *iScience*, 26.

Litjens, G., Kooi, T., & Bejnordi, B. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60–88.

Liu, Y., Li, L., Wang, Z., Zhang, J., & Zhou, Z. (2023). Myocardial ischemia-reperfusion injury: Molecular mechanisms and prevention. *Microvascular Research*, Volume 149, 104565.

Mach, F., Baigent, C., & Catapano, A. (2020). 2019 ESC/EAS Guidelines for the management of dyslipidaemias: lipid modification to reduce cardiovascular risk. *European Heart Journal*, 41(1), 111-188.

Marso, S., Daniels, G., & Brown-Frandsen, K. (2016). Liraglutide and cardiovascular outcomes in type 2 diabetes. *New England Journal of Medicine*, 375(4).

McKinney, W. (2011). Pandas: A foundational Python library for data analysis and statistics. *In Python for high performance and scientific computing* , 1-9.

Mills, E. L. (2016). Succinate Dehydrogenase Supports Metabolic Repurposing of Mitochondria to Drive Inflammatory Macrophages. *Cell*, 167, 457–470.e13.

Ministerio de Sanidad. (2023). *Informe anual del Sistema Nacional de Salud 2023*. Retrieved from INFORMES, ESTUDIOS E INVESTIGACIÓN: inisterio de Sanidad. (2024). Informe anual del Sistema Nacional de Salud 2023. INFORMES, ESTUDIOS E INVESTIGACIÓN.

Morley, J., Machado, C., Burr, C., Cowls, J., Taddeo, M., & Floridi, L. (2020). The ethics of AI in health care: A mapping review. *ocial Science & Medicine*, 260, 113172.

Murdoch W. J., S. C.-A. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44).

Murphy, K. (2012). Machine learning: a probabilistic perspective. *Cambridge (MA): MIT Press*, 1-5.

Nahrendorf, M., & Swirski , F. (2016). Innate immune cells in ischaemic heart disease: does myocardial infarction beget myocardial infarction? *Eur Heart J*, 868-872.

Nahrendorf, M., & Swirski, F. K. (2016). Abandoning M1/M2 for a Network Model of Macrophage Function. *Circulation Research*, 414–417.

National Heart, Lung and Blood Institute. (2022, Marzo 24). *National Heart, Lung and Blood Institute*. Retrieved from NIH: https://www.nhlbi.nih.gov/es/salud/neumonia

Neumann, F., Sousa-Uva, M., & Ahlsson, A. (2019). 2018 ESC/EACTS Guidelines on myocardial revascularization. *European Heart Journal*, 40(2), 87–165.

Nishikido, T., & Ray, K. (2018). Non-antibody Approaches to proprotein convertase subtilisin kexin 9 inhibition: siRNA, antisense oligonucleotides, adnectins, vaccination, and new attempts at small-molecule inhibitors based on new discoveries. *Front. Cardiovasc*, 5:199.

Ong, S., & Hausenloy, D. (2010). Mitochondrial morphology and cardiovascular disease. *Cardiovasc Res*, 88(1):16-29.

Organización Mundial de la Salud. (2023). *Enfermedades cardiovasculares.* Retrieved from Organización Mundial de la Salud.: https://www.who.int/es/health-topics/cardiovascular-diseases#tab=tab_1.

Organization, W. H. (2020). WHO Guidelines on Physical Activity and Sedentary Behaviou. Ginebra: OMS.

Oyama, J. B. (2004). Reduced myocardial ischemia-reperfusion injury in toll-like receptor 4-deficient mice. *Circulation*, 109, 784–789.

Philip J. Drew FRCS, J. R. (2000). Artificial neural networks. *Surgery*, 127.

Pirillo, A., Casula, M., Olmastroni, E., Norata , G., & Catapano , A. (2021). Global epidemiology of dyslipidaemias. *Nat. Rev. Cardiol*, 18:689–700.

Prabhu, S., & Frangogiannis, N. (2016). The Biological Basis for Cardiac Repair After Myocardial Infarction: From Inflammation to Fibrosis. *Circulation Research*, 119(1), 91–112.

Price, W., & Cohen, I. (2019). Privacy in the age of medical big data. *Nature Medicine*, 25(1), 37–43.

Puhl, S. L. (2019). Neutrophils in Post-myocardial Infarction Inflammation: Damage vs. Resolution? *Frontiers in Cardiovascular Medicine (Front. Cardiovasc. Med.)*, 6, 25.

Rabbani N, K. G. (2022). Applications of machine learning in routine laboratory medicine: Current state and future directions. *Clin Biochem*, 103:1-7.

Ray, K., Wright, R., & Kallend, D. (2020). Two Phase 3 Trials of Inclisiran in Patients with Elevated LDL Cholesterol. *New England Journal of Medicine*, 382(16), 1507-1519.

Rodgers, J., Jones, J., Bolleddu, S., Vanthenapalli , S., & Panguluri , S. (2019). Cardiovascular Risks Associated with Gender and Aging. *J Cardiovasc Dev Dis*, 6(2): 19.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6).

Sabatine, M., Giugliano, R., & Keech, A. (2019). Evolocumab and Clinical Outcomes in Patients with Cardiovascular Disease. *New England Journal of Medicine*, 376(18), 1713-1722.

Sanada, S., Komuro, I., & Kitakaze, M. (2011). Pathophysiology of myocardial reperfusion injury: preconditioning, postconditioning, and translational aspects of protective measures. *Am J Physiol Heart Circ Physiol*, 301(5).

Saraste A, P. K. (2000). Apoptosis in myocardial ischaemia and infarction. *Cardiovasc Res.*, 528-37.

Sarvamangala, D., & Kulkarni, R. (2022). Convolutional neural networks in medical image understanding: a survey. *Evolutionary Intelligence*, 15(4):1-22.

Schwartz, G., Steg, P., & Szarek, M. (2018). Alirocumab and Cardiovascular Outcomes after Acute Coronary Syndrome. *New England Journal of Medicine*, 379(22), 2097-2107.

Sharma, N. J. (2018). An analysis of convolutional neural networks for image classification. *Procedia Computer Science*, 377-384.

Sharma, V., Rai, S., & Dev, A. (2012). A Comprehensive Study of Artificial Neural Networks. *International Journal of Advanced Research in Computer Science and Software Engineering*, 1-7.

Silvis, M. J. (2020). Damage-Associated Molecular Patterns in Myocardial Infarction and Heart Transplantation: The Road to Translational Success. *Frontiers in Immunology*, 11, 599511.

Sociedad Española de Cardiología. (2020, Noviembre). *Mortalidad cardiovascular en España en 2020.* Retrieved from Sociedad Española de Cardiología.: https://secardiologia.es/publicaciones/infografias/13105-mortalidad-cardiovascular-en-espana-en-2020.

Sreejit, G. N.-S. (2022). Retention of the NLRP3 Inflammasome-Primed Neutrophils in the Bone Marrow Is Essential for Myocardial Infarction-Induced Granulopoiesis. *Circulation*, 145, 31–44.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research 15*, 1929-1958.

Stead, L., Koilpillai, P., Fanshawe, T., & Lancaster, T. (2016). Combined pharmacotherapy and behavioural interventions for smoking cessation. *Cochrane Database of Systematic Reviews*, 3.

Tighe, P., Shickel, B., & Bihorac, T. (2018). Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis. *IEEE Journal of Biomedical and Health Informatics*, 22(5), 1589–1604.

Ting Sim JZ, F. Q. (2023). Machine learning in medicine: what clinicians should know. *Singapore Med J*, 91-97.

Tokgozoglu, L., & Libby, P. (2022). The dawn of a new era of targeted lipid-lowering therapies. *Eur. Heart J*, 43:3198–3208.

Toma A, D. G. (2022). Deep Learning in Medicine. *JACC*, 1(1).

Turer, A., & Hill, J. (2010). Pathogenesis of myocardial ischemia-reperfusion injury and rationale for therapy. *Am J Cardiol*, 106(3):360-368.

Valgimigli, M., Bueno, H., & Byrne, R. (2018). 2017 ESC focused update on dual antiplatelet therapy in coronary artery disease developed in collaboration with EACTS. *European Heart Journal*, 39(3), 213-260.

Vaswani, A., Shazeer, N., & Parmar, N. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.

Wang, H., Zhou, Z., & Li, Y. (2017). Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18F-FDG PET/CT images. *EJNMMI Res*, 7:11.

Wu, X., Wu, S., Kawashima, H., Hara, H., Ono, M., Gao, C., . . . Onuma, Y. (2021). Current perspectives on bioresorbable scaffolds in coronary intervention and other fields. *Expert Review of Medical Devices*, 18(4), 351–365.

Wuriea, H., & Cappuccio, F. (2012, Jan (Enero) 1). Cardiovascular disease in low- and middle-income countries: an urgent priority. *Ethn Health* , 543-50. Retrieved from Ethn Health: https://doi.org/10.1080/13557858.2012.778642

Yusuf, S., Joseph, P., & Dans, A. (2018). Polypill with or without aspirin in persons without cardiovascular disease. *New England Journal of Medicine*, 384(3), 216-228.

Yusuf, S., Joseph, P., & Rangarajan, S. (2020). Modifiable risk factors, cardiovascular disease, and mortality in 155,722 individuals from 21 high-income, middle-income, and low-income countries (PURE): a prospective cohort study. *The Lancet*, 795-808.

Zhang, J., Chen, J., & Zhong, Y. (2021). A Deep Learning Framework for the Detection of Infarction in Cardiac MRI. *EEE Transactions on Medical Imaging*, 40(1), 254–265.

Zhang, R. Y. (2023). Impact of Reperfusion on Temporal Immune Cell Dynamics after Myocardial Infarction. *Journal of the American Heart Association (J. Am. Heart Assoc.)*, 12, e027600.

Zinman, B., Wanner, C., & Lachin, J. (2015). Empagliflozin, cardiovascular outcomes, and mortality in type 2 diabetes. *New England Journal of Medicine*, 373(22), 2117–2128.