

Yasmina Karimi Hammadi

**Desarrollo de modelos predictivos basados en
inteligencia artificial y radiómica para asistir en las
decisiones clínicas para pacientes con cáncer de recto.**

TRABAJO FINAL DE GRADO

dirigido por el Dr. Hernández y la Dra. Arenas

Grado de Ingeniería Biomédica



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2024

Tabla de contenido

Propósito	3
1 Introducción	4
1.1 Contexto: Cáncer de recto	4
1.2 Radiómica	8
2 Metodología	10
2.1 Adquisición de datos	12
2.2 Extracción de características	13
2.3 Preprocesamiento de datos	19
2.4 Optimización de características	19
2.5 Desarrollo de modelos predictivos	24
2.5.1 Bosque Aleatorio (RF).....	26
2.5.2 Support Vector Machine (SVM)	26
2.5.3 Regresión Logística (RL).....	27
2.5.4 Red Neuronal Artificial	27
2.6 Validación de modelos predictivos	29
3 Resultados	32
3.1 Selección de características.....	32
3.2 Evaluación y confirmación de modelos.....	35
4 Discusión	47
5 Conclusiones.....	50
6 Referencias consultadas.....	51
7 Anexo: Código fuente.....	54
7.1 Selección de características.....	54
Random forest.....	55
Gradient Boosting	56
Recursive Feature Elimination (RFE)	57
Bosques Extra	58
7.2 Construcción de modelos predictivos	59

Propósito

Este proyecto se centra en pacientes con cáncer de recto sometidos a tratamiento neoadyuvante, que implica la administración de quimioterapia y radioterapia antes de la cirugía. Cuando se hace referencia al tratamiento neoadyuvante, se está hablando de la aplicación de estas terapias antes de la intervención quirúrgica. La perspectiva de este estudio es que la combinación de la radiómica junto con el aprendizaje automático permitirá predecir las respuestas post quimio-radioterapia (QT/RT). Esta estrategia tiene el potencial de mejorar significativamente la toma de decisiones terapéuticas en la práctica clínica al proporcionar una evaluación más precisa de las variables que afectan el pronóstico y la identificación de la respuesta patológica.

Estos modelos desempeñan un papel crucial al identificar tempranamente a los pacientes que no responderán positivamente al tratamiento, lo que facilita la adaptación de estrategias terapéuticas para minimizar la toxicidad asociada. Además, contribuyen a la predicción de respuestas completas, lo que podría evitar intervenciones quirúrgicas posteriores y, por ende, reducir complicaciones y secuelas, como la necesidad de colostomía.

El objetivo principal de este estudio consiste en desarrollar modelos de predicción para anticipar la respuesta al tratamiento en las dos aplicaciones específicas previamente mencionadas. Por un lado, se busca prever la respuesta negativa al tratamiento con el fin de evitar la toxicidad asociada y, por otro lado, se pretende anticipar la respuesta completa para evitar la cirugía en caso de no ser necesaria.

En este contexto, el enfoque se centra en la construcción de modelos de inteligencia artificial basados en biomarcadores radiómicos extraídos de imágenes de TAC de pacientes con cáncer de recto que han sido sometidos a neoadyuvancia y cirugía. La intención es aprovechar estas características radiómicas para desarrollar modelos predictivos que proporcionen una visión precisa de la respuesta al tratamiento en este grupo específico de pacientes.

Con el propósito de lograr estos objetivos, se plantea la construcción de una base de datos integral que incluya las características radiómicas extraídas de las imágenes TAC correspondientes a los pacientes tratados en el Hospital Sant Joan de Reus.

Finalmente, utilizando las características radiómicas extraídas como variables predictivas, se desarrollarán modelos de inteligencia artificial con el objetivo de predecir la respuesta patológica y evaluar su capacidad predictiva. Este enfoque busca contribuir significativamente a la personalización de los tratamientos, evitando así complicaciones innecesarias y optimizando la toma de decisiones clínicas en el manejo de pacientes con cáncer de recto.

1 Introducción

1.1 Contexto: Cáncer de recto

El cáncer colorrectal es caracterizado principalmente por su gran impacto en el colon y recto, siendo uno de los tipos de cáncer más comunes a nivel mundial. Este tipo de cáncer es considerado la segunda causa principal de muertes atribuidos al cáncer en escala mundial. Según la Organización Mundial de la Salud (OMS), durante el año 2020 se estima que hubo más 1,9 millones de nuevos casos y se registraron más de 930.000 muertes debido a este cáncer en todo el mundo [1]. De éstos, el 70% se localiza en el colon mientras que el otro 30% se encuentra en el recto. El diagnóstico del cáncer de recto se suele producir en etapas más avanzadas en comparación con el cáncer de colon. En el momento del diagnóstico, alrededor del 25% de los cánceres de colon ya han invadido la capa muscular de la pared del intestino, mientras que, en el cáncer de recto, esta invasión se observa aproximadamente en el 50% de los pacientes diagnosticados [2].

Según el Centro Internacional de Investigaciones sobre el Cáncer (IARC), el cáncer colorrectal se clasificó como la segunda causa más común de cáncer en escala mundial. En 2020, se estima que hubo un número significativo de casos prevalentes, abarcando tanto diagnósticos recientes como aquellos que persisten bajo tratamiento durante un periodo de 5 años (Figura1) [3].

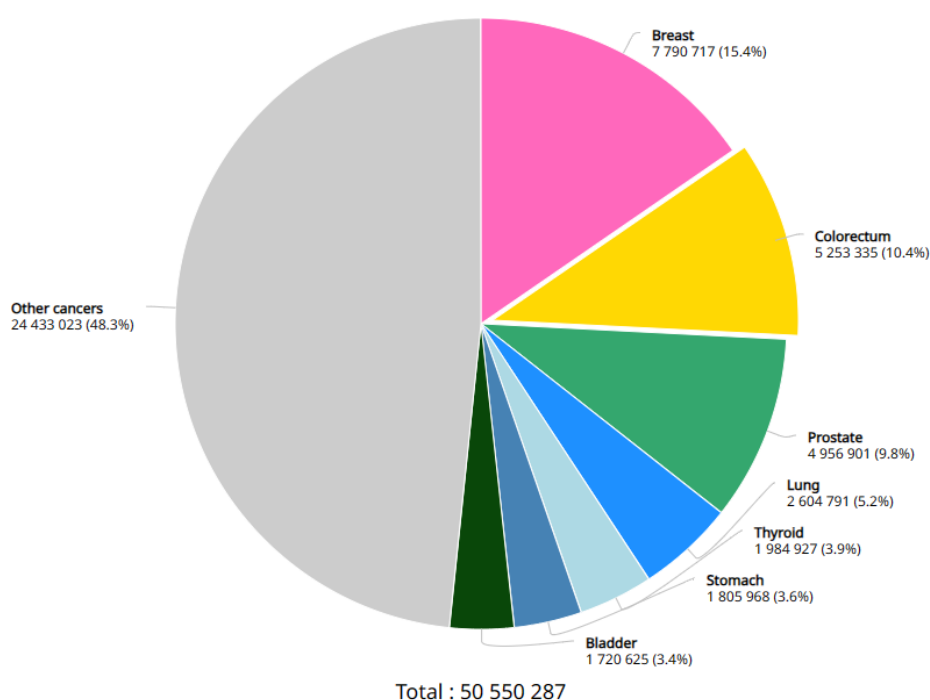


Figura 1: Estimación global de casos de cáncer colorrectal en 2020 (5 años). [3]

El cáncer de recto localmente avanzado (LARC) representa un desafío clínico en la oncología moderna. El abordaje terapéutico de esta enfermedad ha evolucionado hacia la integración de modalidades terapéuticas con el principal objetivo de mejorar los

resultados en términos de control local y supervivencia, sin aumentar la toxicidad [4]. En este contexto, el tratamiento neoadyuvante, específicamente la quimio-irradiación preoperatoria, ha cobrado especial relevancia.

Dentro del espectro de opciones terapéuticas, se destacan tres enfoques clave que desempeñan un papel crucial en el tratamiento del cáncer de recto: adyuvante, neoadyuvante y postadyuvante.

El tratamiento adyuvante se administra después de la cirugía para eliminar cualquier célula cancerosa residual y prevenir la recurrencia. Esta estrategia busca consolidar los resultados obtenidos con la intervención quirúrgica.

Por otro lado, el tratamiento neoadyuvante implica la aplicación de terapias como quimioterapia y radioterapia antes de la cirugía. Este enfoque tiene como objetivo reducir el tamaño del tumor, facilitar la cirugía o mejorar los resultados a largo plazo.

Finalmente, el tratamiento postadyuvante se administra después de la cirugía y el tratamiento neoadyuvante. Su objetivo es consolidar los efectos de las terapias anteriores y minimizar el riesgo de recurrencia.

La radioterapia preoperatoria, tratamiento neoadyuvante, ofrece beneficios significativos en el manejo del cáncer de recto localmente avanzado. Estudios han demostrado que esta aproximación puede aumentar el control local y la supervivencia en comparación con la cirugía exclusiva [4]. Consecuentemente, la quimio-irradiación preoperatoria se ha establecido como una parte crucial del tratamiento neoadyuvante [4].

La administración concomitante de quimioterapia durante la irradiación potencia los mecanismos de lesión sobre las células tumorales propios de cada uno de estos tratamientos citotóxicos [5]. En este sentido, la quimioterapia concomitante ha sido estudiada ampliamente y se ha asociado con un descenso de estadificación (downstaging) y con un incremento en las tasas de cirugías conservadoras del esfínter [4]. El intervalo entre la radioterapia y la cirugía, la dosis y el fraccionamiento de la radioterapia son variables que pueden modificar la práctica clínica actual.

En la radioterapia, la unidad utilizada para expresar la dosis de radiación absorbida por los tejidos es el gray (Gy). Representa la absorción de un julio de energía radiante por un kilogramo de masa. Esta medida es crucial para cuantificar y administrar con precisión la cantidad de radiación entregada a los tejidos durante el tratamiento oncológico

En el contexto de la neoadyuvancia, los pacientes se someten a una dosis total de 50.4 Gy, dividida en 28 fracciones de radioterapia. Esto significa que cada sesión entrega aproximadamente 1.8 Gy. La distribución en fracciones permite maximizar la efectividad terapéutica, focalizando la radiación en las células cancerosas mientras se minimiza la exposición de los tejidos circundantes. Esta estrategia no solo busca controlar eficientemente el tumor sino también reducir los posibles efectos secundarios al permitir la recuperación de los tejidos normales entre sesiones. [6]

Los avances en el tratamiento adyuvante también han contribuido a mejorar los resultados. El desarrollo de nuevas estrategias, como la optimización de la terapéutica adyuvante y los progresos en la técnica quirúrgica, han tenido un papel fundamental en el control local tumoral.

La capacidad predictiva de la respuesta a la quimio-irradiación neoadyuvante es un área de interés creciente. Un entendimiento más preciso de la importancia del descenso de estadificación puede influir significativamente en los objetivos clínicos del tratamiento del cáncer de recto, como la promoción del control pélvico y la supervivencia. Las tasas de control local pélvico y la factibilidad técnica de la cirugía pueden verse afectadas por la respuesta al tratamiento neoadyuvante [4].

El sistema de estadificación más comúnmente utilizado para evaluar el cáncer colorrectal es el sistema TNM (Tumor-Nodo-Metástasis) del *American Joint Committee on Cancer* (AJCC) [7]. En este contexto, la letra T hace referencia al tamaño y extensión del tumor primario, la letra N indica la presencia y extensión de los ganglios linfáticos afectados, y la letra M se refiere a la existencia de metástasis, es decir, la propagación del cáncer a otras partes del cuerpo.

Los números y letras que siguen las letras T, N y M proporcionan detalles adicionales sobre cada uno de estos factores. En general, los valores más altos indican mayor grado de avance del cáncer. Una vez que se han determinado estas categorías, esta información se amalgama en un proceso denominado agrupación por etapas. Este procedimiento tiene como objetivo asignar una etapa general que refleje la gravedad y extensión del cáncer en el paciente.

La tecnología de la tomografía computarizada (TAC) juega un papel importante en la determinación del estadio clínico del cáncer de recto. En varios estudios se ha visto que la TAC, complementada con la ecografía endorrectal y la resonancia magnética nuclear (RMN), puede aportar información valiosa sobre la extensión local del tumor y el compromiso ganglionar [8].

En este contexto, el análisis radiómico de imágenes TAC emerge como una herramienta prometedora. A través de la extracción de características cuantitativas de las imágenes, se pueden identificar patrones que no son perceptibles a simple vista. Estos patrones tienen el potencial de correlacionarse con la respuesta histopatológica y otros resultados clínicos [9]. De hecho, el *American Joint Cancer Committee* ha reconocido la importancia de la clasificación TNM para el cáncer colorrectal, proporcionando una base para la implementación del análisis radiómico en la práctica clínica [10].

La capacidad de predecir la respuesta al tratamiento neoadyuvante es esencial, ya que puede influir en las decisiones posteriores al tratamiento, como la necesidad de terapia adyuvante [11]. Además, una predicción precisa de la respuesta podría facilitar la individualización de los regímenes de tratamiento, mejorando potencialmente los resultados clínicos y la calidad de vida de los pacientes [12].

El proceso de diagnóstico y tratamiento del cáncer de recto (Figura 2) implica una secuencia de pasos muy importantes para determinar la extensión de la enfermedad y brindar la atención adecuada. El estándar de tratamiento para la enfermedad en etapa localmente avanzada (LARC) es la terapia multimodal seguida de cirugía con escisión total del mesorrecto (TME), con la quimiorradioterapia neoadyuvante (CRT) jugando un papel crucial en la mejora del control locorregional y posiblemente en la tasa de respuestas patológicas completas (pCR) [13]. En este contexto, se destaca la relevancia de la evaluación clínica inicial, que incluye colonoscopias, tomografías computarizadas y resonancias magnéticas. En particular, el uso de tomografías computarizadas (TAC) juega un papel fundamental, proporcionando imágenes que permiten la extracción del TNM clínico, para establecer el pronóstico y guiar la estrategia terapéutica.

Una vez obtenido el TNM clínico, los pacientes son sometidos a un tratamiento neoadyuvante, combinando quimioterapia y radioterapia concomitante. Con la intención de reducir el tamaño del tumor antes de la cirugía, facilitando así la extirpación quirúrgica. En este punto, se obtiene el ypTNM, una clasificación patológica que se basa en imágenes y que proporciona información sobre la respuesta del paciente al tratamiento neoadyuvante.

Es esencial señalar que hay alrededor de un 30% de posibilidades de lograr una respuesta completa sin recurrir a la cirugía en pacientes que han recibido tratamiento neoadyuvante. Este hallazgo tiene implicaciones significativas, ya que sugiere la posibilidad de omitir la cirugía en estos casos, evitando así las consecuencias secundarias asociadas a procedimientos quirúrgicos y mejorando la calidad de vida de los pacientes.

El siguiente paso en el proceso es la cirugía, donde se lleva a cabo la extracción del recto afectado, permitiendo obtener el pTNM (TNM patológico) a partir de la pieza quirúrgica extraída y evaluar la extensión final del cáncer. Posteriormente, algunos pacientes pueden requerir tratamiento adyuvante para asegurar una recuperación completa y prevenir posibles recurrencias.

Finalmente, se establece un seguimiento y cuidados posteriores, crucial para monitorear la salud a largo plazo del paciente. Este proceso completo, representado de manera esquemática, destaca la importancia de cada etapa y subraya la posibilidad de personalizar el tratamiento según la respuesta individual de cada paciente. La integración de información radiológica, clasificaciones TNM, tratamiento neoadyuvante, cirugía, y cuidados postoperatorios se convierte en un enfoque integral para abordar el cáncer de recto, permitiendo una atención más precisa y personalizada.

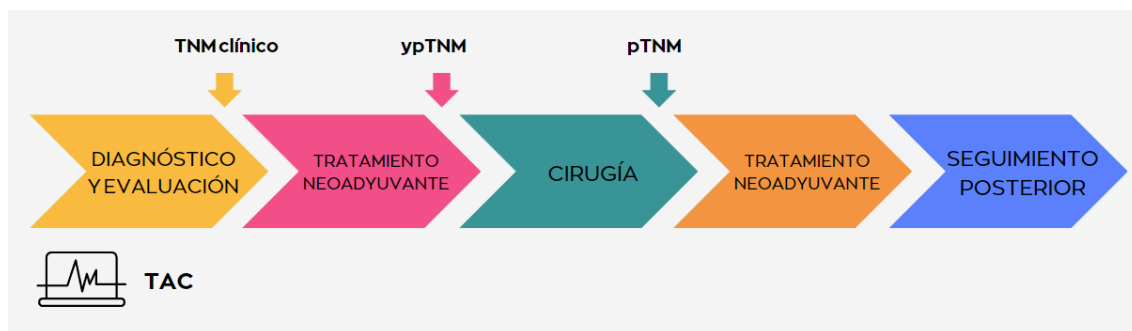


Figura 2. Figura representativa del proceso de diagnóstico y tratamiento de pacientes con cáncer de recto.

1.2 Radiómica

En los últimos años, la radiómica ha surgido como un área prometedora en la oncología, proporcionando así un análisis cuantitativo de las imágenes médicas como el TAC que puede revelar patrones no percibidos por el ojo humano. [14][15]. El diagnóstico radiómico se nutre de la extracción de características de las imágenes médicas, tales como la intensidad, el tamaño, la forma, el volumen y la textura, que ofrecen información adicional sobre la naturaleza del tumor y pueden correlacionarse con la respuesta al tratamiento. [16] La radiómica se ha mostrado prometedora para identificar biomarcadores no invasivos que correlacionan con resultados clínicos y patológicos, mejorando así la personalización del tratamiento. [15]

Junto con técnicas de aprendizaje automático, la radiómica se ha utilizado para mejorar la predicción de la respuesta al tratamiento neoadyuvante en pacientes con cáncer de recto. Al analizar las imágenes TAC, es posible identificar patrones radiómicos que se relacionan con la respuesta completa a la quimio y radioterapia. Además, estas características mencionadas se pueden integrar en modelos predictivos para proporcionar estimaciones más precisas sobre la eficacia del tratamiento [15].

La aplicación de la radiómica implica una serie de pasos cruciales (ver Figura 3). En primer lugar, comienza con la obtención de datos provenientes de imágenes médicas, siendo la calidad de estas imágenes crucial para asegurar la precisión y consistencia del modelo radiómico [17]. Posteriormente, se lleva a cabo la segmentación de las imágenes con el propósito de identificar la región de interés (ROI), un paso de gran relevancia que sienta las bases para la extracción subsiguiente de características. Debido a posibles variaciones en las imágenes debido a distintos modelos y parámetros de escaneo, resulta esencial estandarizar las imágenes antes de definir la ROI. La segmentación del ROI puede ser abordada de tres maneras: manual, semiautomática y automática, siendo la segmentación manual considerada el estándar de oro, aunque también es la más laboriosa y susceptible a la variabilidad del operador [17].

En la tercera etapa, se eligen y extraen las características radiómicas, las cuales se pueden agrupar en cuatro categorías principales: tamaño y forma, estadísticas de primer orden, estadísticas de segundo orden y características asociadas con cambios, como intensidad, forma, textura y propiedades de ondas. La textura destaca como una característica

comúnmente empleada. Este procedimiento de elección y extracción de características implica la reducción de la dimensionalidad de los datos, y puede realizarse de manera supervisada mediante métodos como el operador de selección y contracción mínima absoluta (LASSO) y el análisis discriminante lineal, o de forma no supervisada, a través de técnicas como el análisis de componentes principales y el análisis de conglomerados [17].

La cuarta etapa involucra el análisis y la creación de modelos. El objetivo principal en esta fase es establecer un modelo práctico que permita predecir resultados clínicos. Los modelos comúnmente empleados incluyen la regresión logística, LASSO, el algoritmo de bosque aleatorio o la máquina de vectores de soporte. Es importante destacar que, en la práctica, es frecuente la necesidad de utilizar diversos métodos de modelado en conjunto para mitigar las limitaciones inherentes a cada uno y lograr un modelo radiómico más preciso [18].

Finalmente, en el quinto paso, se aplica el modelo resultante, lo que puede abarcar el diagnóstico, estadificación, pronóstico y predicción de la respuesta al tratamiento de tumores u otras aplicaciones clínicas relevantes.

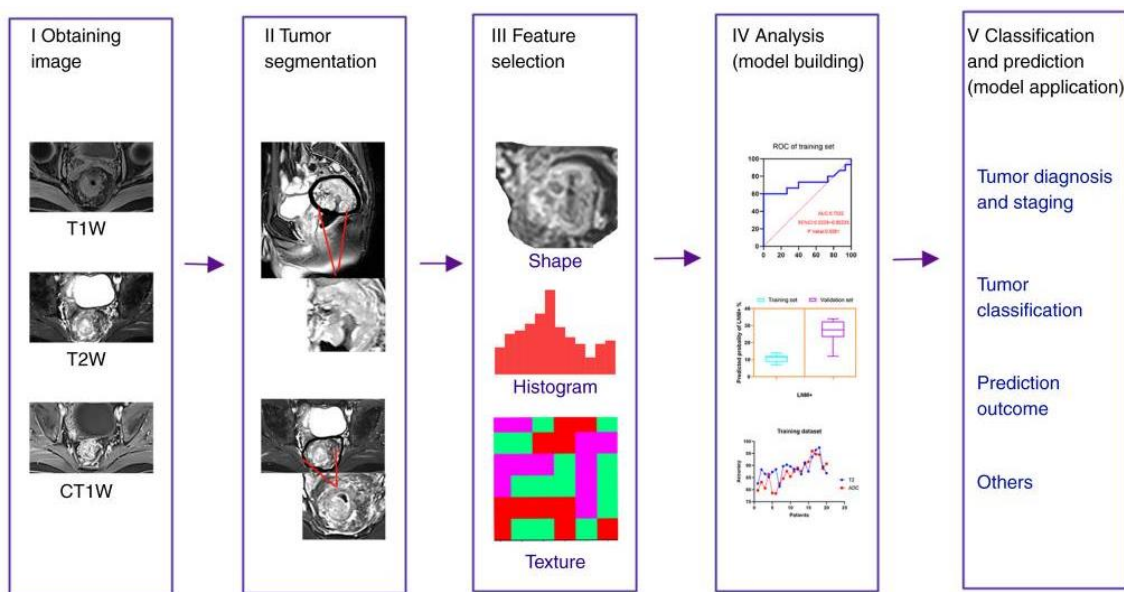


Figura 3: Obtenida de “*Role of MRI-based radiomics in locally advanced rectal cancer*” [17]. Representa el flujo de trabajo.

La capacidad de predecir qué pacientes responderán favorablemente a la terapia neoadyuvante puede transformar la práctica clínica al permitir tratamientos más personalizados. En un estudio reciente, se demostró que las características radiómicas de textura obtenidas por PET/CT pueden ser útiles para predecir la respuesta al tratamiento en pacientes con cáncer de mama [19], lo cual sugiere un potencial similar en el contexto del cáncer de recto.

Estudios recientes han revelado que la categoría N preoperatoria tiene un impacto significativo en la predicción de la pCR; en un análisis de 141 pacientes tratados con CRT

neoadyuvante seguido de cirugía radical, se observó que la pCR estaba significativamente asociada con la categoría nodal preoperatoria [20]. Además, se ha observado que la radiómica puede ayudar a discernir las características clínico-patológicas y pronósticas en pacientes tratados con quimiorradioterapia neoadyuvante [20]. La radiómica ofrece una oportunidad para identificar marcadores predictivos no invasivos que puedan influir en el manejo personalizado de los pacientes.

La aplicación de la radiómica en la evaluación de imágenes TAC puede revelar patrones subvisuales que correlacionan con cambios histopatológicos, lo que podría anticipar la respuesta al tratamiento antes de la manifestación clínica de estos cambios. Por ende, la radiómica podría desempeñar un papel transformador en la predicción de la pCR y la recurrencia, lo que a su vez podría mejorar la selección de pacientes para protocolos de tratamiento específicos y la gestión de la toxicidad relacionada con la CRT.

Según el estudio del paper “*Breast Cancer Tissue Markers, Genomic Profiling, and Other Prognostic Factors: A Primer for Radiologists*” [21], la aplicación de quimioterapia neoadyuvante favorece la respuesta del 30% de los pacientes con cáncer de mama y reduce la tasa de recurrencia hasta en un 50%. Sin embargo, la NAC no produce una respuesta patológica completa (pCR) en todos los pacientes debido a la heterogeneidad de los tumores mamarios.

Por ello, el objetivo de la radiómica es ayudar a predecir la respuesta a la NAC mediante el análisis de imágenes, principalmente MRI en este ámbito, para evitar tratamientos innecesarios y sus consecuencias tóxicas, a la vez que potenciar otros que sí vayan a tener una respuesta positiva en el paciente. [21]

2 Metodología

La metodología del trabajo se estructura en varias etapas para analizar y prever la respuesta post-neoadyuvante en pacientes con cáncer de recto. Empezando con la selección de pacientes y culminando en la evaluación del modelo predictivo, cada fase contribuye de manera crucial para comprender los factores que impactan la respuesta post-neoadyuvante y desarrollar un modelo preciso.

El análisis de los modelos predictivos se estructuró en dos partes fundamentales, alineadas con los objetivos principales del estudio. Cada parte aborda un aspecto clave en la toma de decisiones clínicas, destacando los resultados obtenidos y su relevancia para la predicción de respuestas a tratamientos y la toma de decisiones quirúrgicas.

Una aplicación esencial de este enfoque se encuentra en la capacidad de anticipar y evitar el tratamiento de Radioterapia (RT) y Quimioterapia (QT) en pacientes que probablemente no responderán positivamente. Este abordaje tiene el potencial de prevenir la exposición innecesaria a tratamientos agresivos, minimizando la toxicidad asociada y mejorando significativamente la calidad de vida de los pacientes. La predicción se enfoca en determinar si la respuesta al tratamiento será Completa/Parcial (Clase I) o si la enfermedad permanecerá Estable/Progresará (Clase II). Al identificar con precisión los

pacientes no respondedores, se optimiza la toma de decisiones terapéuticas, permitiendo una atención más personalizada y efectiva.

Otra aplicación crucial radica en la capacidad de prever la respuesta completa en pacientes, lo que puede llevar a evitar procedimientos quirúrgicos innecesarios. En particular, se busca identificar aquellos pacientes que han logrado una respuesta completa al tratamiento, con el objetivo de evitar complicaciones y secuelas, como la necesidad de colostomía. Este enfoque no solo mejora la calidad de vida de los pacientes al reducir la probabilidad de complicaciones quirúrgicas, sino que también contribuye a una gestión más eficiente de los recursos médicos al evitar intervenciones no fundamentadas. La predicción se orienta hacia la clasificación de respuestas como Completa (Clase I) o Parcial/Estable/Progresión (Clase II), permitiendo una toma de decisiones clínicas más informada y adaptada a las necesidades individuales de los pacientes.

Durante la extracción de características radiómicas a partir de las regiones GTV de las imágenes TAC, se emplea el software Slicer 3D para explorar y analizar los volúmenes de tumor en las tomografías computarizadas. Este software facilita la segmentación de las regiones de interés, sentando las bases para obtener hasta 130 características radiómicas distribuidas en 8 grupos. Estas características proporcionan datos cuantitativos derivados de las imágenes médicas, ofreciendo una visión detallada de las propiedades de las lesiones y garantizando la captura integral de información para el análisis y predicción posteriores.

En la etapa de selección de características, se aplican técnicas avanzadas para identificar las características radiómicas más relevantes en relación con la respuesta post-neoadyuvante. Esto se logra mediante un análisis de los conjuntos de datos, priorizando las características con fuerte correlación con la variable objetivo (respuesta patológica). Así, se construye una base de datos que incorpora las características más significativas de cada paciente.

La construcción del modelo predictivo constituye una fase crucial, donde técnicas de inteligencia artificial y modelos de aprendizaje automático son empleados. Entrenando el modelo con las características radiómicas seleccionadas, se logra capturar patrones complejos y relaciones no lineales, asegurando una capacidad predictiva robusta para la respuesta post-neoadyuvante en pacientes con cáncer de recto.

La metodología aborda dos aplicaciones específicas, en el primer caso, se busca evitar el tratamiento de radioterapia (RT) y quimioterapia (QT) en aquellos pacientes que no mostrarán una respuesta positiva, con el objetivo de prevenir la toxicidad asociada. Aquí, se pretende anticipar el tipo de respuesta, ya sea Completa/Parcial (clase I) o Estable/Progresión (clase II).

Por otro lado, en la segunda aplicación, se procura eludir la intervención quirúrgica en pacientes que han manifestado una respuesta completa. La razón es prevenir posibles complicaciones y secuelas derivadas de la cirugía, como, por ejemplo, la necesidad de

una colostomía. En esta situación, el objetivo es predecir la respuesta completa, considerando las clases I (Completa) y II (Parcial, Estable, Progresión).

Finalmente, la evaluación del rendimiento del modelo es esencial. Métricas fundamentales como precisión, sensibilidad y especificidad se utilizan para medir la eficacia y precisión de las predicciones en el contexto clínico. Estas métricas proporcionan una comprensión detallada de la capacidad predictiva del modelo, contribuyendo a su validez y utilidad en la predicción de la respuesta post-neoadyuvante en pacientes con cáncer de recto.

2.1 Adquisición de datos

Los datos que se usaron en este estudio fueron obtenidos a través del Departamento de Oncología Radioterápica del Hospital Universitario Sant Joan de Reus. Durante una estada de prácticas se pudieron adquirir imágenes de tomografía computarizada de simulación (TAC) de 50 pacientes anteriores a cualquier tratamiento o cirugía, de los cuales se procedió a la extracción de las características radiómicas.

En el proceso de obtención de tomografías, se hizo uso del entorno Aria de Varian, este sistema de información centraliza el historial de visitas y sesiones de tratamiento en el servicio oncológico. De esta manera, se consigue visualizar los TACs realizados durante el tratamiento de cada paciente mediante su ID.

En el mismo contexto, a partir de las tomografías computarizadas (TACs) de planificación, se realiza la descarga de las imágenes en formato DICOM (Comunicación e Imagen Digital en Medicina), un estándar internacional utilizado en la imagenología digital y comunicaciones médicas que facilita el intercambio de datos e imágenes [22]. Cumpliendo con las normativas de protección de datos del hospital y respetando el acuerdo de confidencialidad establecido, el proceso de descarga se ejecuta de forma anónima.

En el análisis de la base de datos que abarca a los 50 pacientes sujetos a estudio, se han identificado los diversos patrones de respuesta al tratamiento. De manera significativa, 34 pacientes exhibieron una respuesta parcial, evidenciando una notable mejoría en la condición clínica sin alcanzar la remisión completa. Además, se observó una progresión en un caso.

Es relevante destacar que, conforme a lo discutido en el marco teórico, los 12 pacientes que lograron una respuesta completa al tratamiento representan aproximadamente el 30% de la muestra total. Este resultado es especialmente alentador, ya que indica una erradicación exitosa de la enfermedad en un porcentaje significativo de la cohorte estudiada. Por otro lado, 3 pacientes mostraron una respuesta estable, indicando una situación donde la enfermedad no progresó ni regresó.

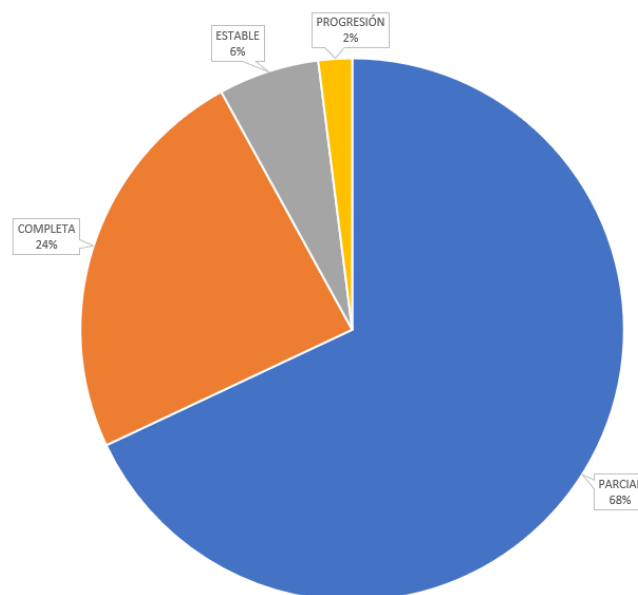


Figura 4. Diagrama circular de la representación y clasificación de la respuesta patológica de los pacientes.

2.2 Extracción de características

La extracción de características radiómicas se llevó a cabo utilizando el software "3D Slicer", una herramienta altamente especializada y destacada en el ámbito de la visualización y procesamiento de imágenes médicas. Este programa se ha consolidado como una opción de referencia, especialmente diseñada para aplicaciones clínicas y biomédicas.

El proceso de extracción de imágenes comienza en el software *Advantage Image*, disponible en los ordenadores del hospital, donde se almacenan las imágenes de todos los pacientes en formato DICOM. El formato DICOM es fundamental en medicina, estandarizando el intercambio de imágenes médicas con metadatos clínicos.

Inicialmente, se realiza un filtrado por el número de historial de los pacientes incluidos en el estudio. Luego, se descargan las imágenes de tomografía computarizada (TAC) en un dispositivo USB para su posterior manipulación y análisis.

Posteriormente, las imágenes descargadas se procesan utilizando el software Slicer 3D, tal y como hemos comentado previamente. En esta etapa, se podría llevar a cabo la delimitación del tumor, una herramienta que nos facilitaría la manipulación de estas imágenes sin usar otros programas diferentes. En este caso, la región de interés (GTV) estaba correctamente delimitada. Estas características radiómicas representan medidas cuantitativas de diversas propiedades de la imagen y se extraen de la región previamente identificada como la ubicación del tumor.

En este contexto, se obtienen un total de 130 características radiómicas utilizando el software Slicer 3D que coinciden con el máximo de características que puede ofrecernos este software. Estas características se dividen en ocho grupos distintos, abarcando una

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

amplia gama de propiedades radiológicas y proporcionando una descripción detallada y cuantitativa de las características de la región tumoral.

La división en ocho grupos puede referirse a categorías específicas de características, como estadísticas de intensidad, textura, forma, entre otras. Cada grupo puede contener características que ofrecen información única sobre la composición y la estructura de la lesión detectada en las imágenes TAC. Este enfoque permite una importante evaluación de la heterogeneidad tumoral y proporciona información valiosa para el análisis y pronóstico del cáncer de recto.

En las siguientes imágenes, podemos observar los tres cortes de las imágenes de tomografía computarizada (TAC): axial, coronal y sagital, con el volumen del tumor (GTV) resaltado en rojo/verde, junto con una representación tridimensional del tumor.

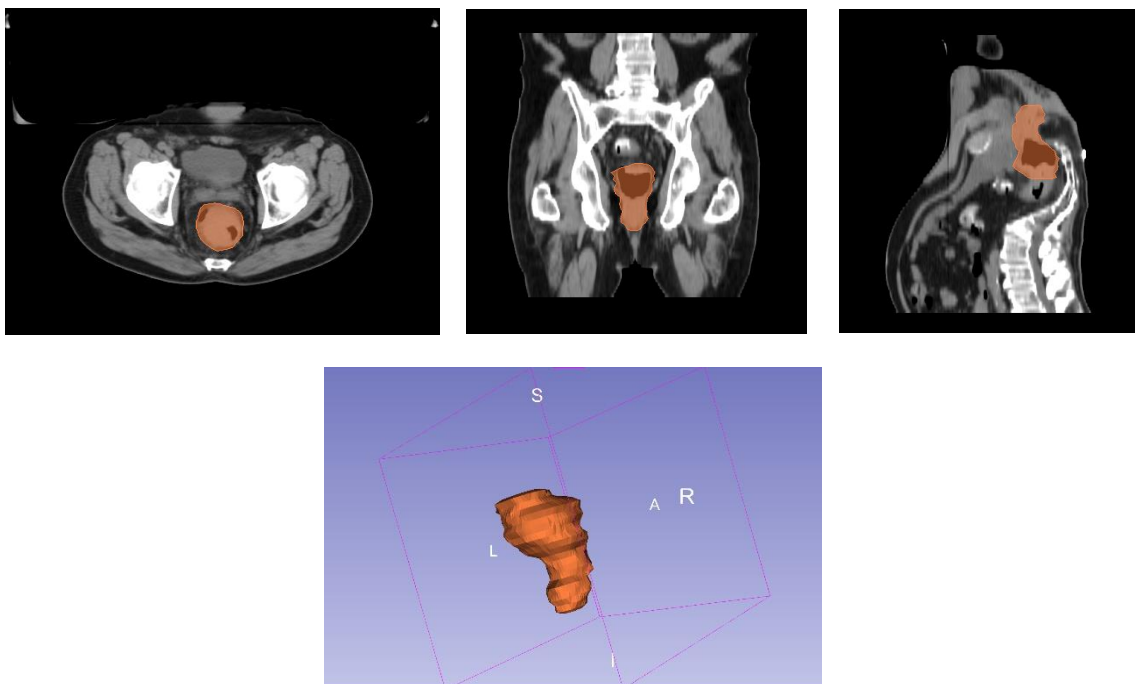


Figura 5: imágenes de tomografía computarizada (TAC) del paciente 1 en los planos axial, coronal y sagital. Estas imágenes, extraídas con el software *Slicer 3D*, resaltan el *Gross Tumor Volume* (GTV).

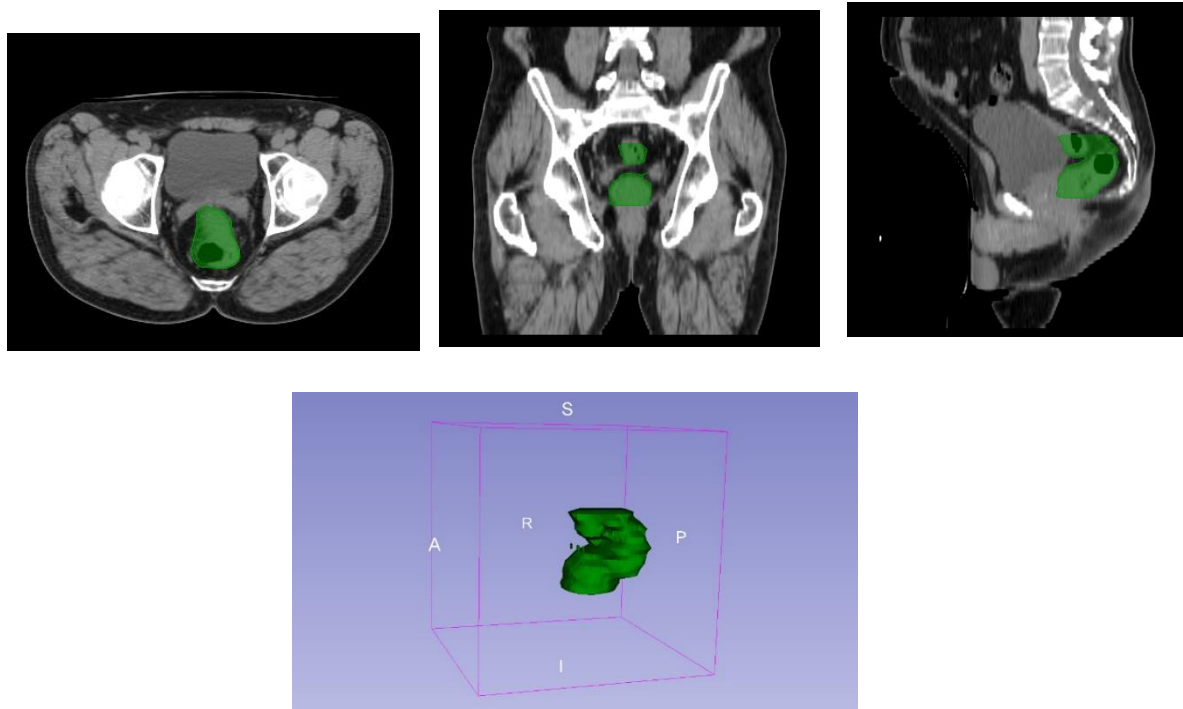


Figura 6: imágenes de tomografía computarizada (TAC) del paciente 2 en los planos axial, coronal y sagital. Estas imágenes, extraídas con el software *Slicer 3D*, resaltan el *Gross Tumor Volume (GTV)*.

Las características que se obtuvieron fueron las siguientes:

Las características radiómicas se agrupan en distintas categorías, cada una proporcionando información detallada sobre la distribución de intensidades y textura de las imágenes médicas de los tumores. Veamos estas categorías y las características específicas que abarcan [23]:

a) First Order Features:

Estas características capturan la disposición de las intensidades en cada píxel volumétrico dentro de la región tumoral. Entre las variables se destacan:

- Energy
- Interquartile Range
- Total Energy
- Range
- Entropy
- Mean Absolute Deviation (MAD)
- Minimum
- Robust Mean Absolute Deviation (rMAD)
- 10th percentile
- Root Mean Squared (RMS)

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

- 90th percentile
- Uniformity
- Maximum
- Skewness
- Mean
- Kurtosis
- Median
- Variance

b) Gray Level Cooccurrence Matrix Features (GLCM) Features:

Estas características se refieren a las propiedades extraídas a partir de la Matriz de Coocurrencia de Niveles de Gris (GLCM). Entre las variables se encuentran:

- Autocorrelation
- Informational Measure of Correlation (IMC1)
- Cluster Prominence
- Informational Measure of Correlation (IMC2)
- Joint Average
- Inverse Difference Moment (IDM)
- Cluster Shade
- Inverse Difference Moment Normalized (IDMN)
- Cluster Tendency
- Maximal Correlation Coefficient (MCC)
- Contrast
- Inverse Difference (ID)
- Correlation
- Inverse Difference Normalized (IDN)
- Difference Average
- Inverse Variance
- Difference Entropy
- Maximum Probability
- Difference Variance
- Sum Average
- Joint Energy
- Sum Entropy
- Joint Entropy
- Sum of Squares

c) Gray Level Dependence Matrix (GLDM) Features:

Estas características corresponden a propiedades derivadas de la Matriz de Dependencia de Niveles de Gris (GLDM). Algunas de las variables incluyen:

- Dependence Entropy (DE)
- Low Gray Level Emphasis (LGLE)
- Dependence Variance (DV)
- High Gray Level Emphasis (HGLE)

- Gray Level Variance (GLV)
- Gray Level Non-Uniformity (GLN)
- Dependence Non-Uniformity (DN)
- Dependence Non-Uniformity Normalized (DNN)
- Small Dependence Emphasis (SDE)
- Large Dependence Emphasis (LDE)
- Small Dependence High Gray Level Emphasis (SDHGLE)
- Small Dependence Low Gray Level Emphasis (SDLGLE)
- Large Dependence Low Gray Level Emphasis (LDLGLE)
- Large Dependence High Gray Level Emphasis (LDHGLE)

d) Gray Level Run Length Matrix (GLRLM) Features:

Estas características se relacionan con propiedades derivadas de la Matriz de Longitud de Carrera de Niveles de Gris (GLRLM). Entre las variables se encuentran:

- Short Run Emphasis (SRE)
- Run Length Non-Uniformity (RLN)
- Long Run Emphasis (LRE)
- Run Length Non-Uniformity Normalized (RLNN)
- Run Percentage (RP)
- Gray Level Non-Uniformity (GLN)
- Run Variance (RV)
- Gray Level Non-Uniformity Normalized (GLNN)
- Run Entropy (RE)
- High Gray Level Run Emphasis (HGLRE)
- Low Gray Level Run Emphasis (LGLRE)
- Short Run Low Gray Level Emphasis (SRLGLE)
- Short Run High Gray Level Emphasis (SRHGLE)
- Long Run Low Gray Level Emphasis (LRLGLE)
- Long Run High Gray Level Emphasis (LRHGLE)

e) Gray Level Size Zone Matrix (GLSZM) Features:

Estas características se vinculan con propiedades derivadas de la Matriz de Tamaño de Zona de Niveles de Gris (GLSZM). Entre las variables se encuentran:

- Small Area Emphasis (SAE)
- Zone Percentage (ZP)
- Large Area Emphasis (LAE)
- Zone Variance (ZV)
- Gray Level Variance (GLV)
- Zone Entropy (ZE)
- Gray Level Non-Uniformity (GLN)
- Gray Level Non-Uniformity Normalized (GLNN)
- Size-Zone Non-Uniformity (SZN)
- Size-Zone Non-Uniformity Normalized (SZNN)
- Low Gray Level Zone Emphasis (LGLZE)

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

- High Gray Level Zone Emphasis (HGLZE)
- Small Area Low Gray Level Emphasis (SALGLE)
- Small Area High Gray Level Emphasis (SAHGLE)
- Large Area Low Gray Level Emphasis (LALGLE)
- Large Area High Gray Level Emphasis (LAHGLE)

f) Neighbouring Gray Tone Difference Matrix (NGTDM) Features:

Estas características están relacionadas con propiedades derivadas de la Matriz de Diferencia de Tonos Grises Vecinos (NGTDM). Entre las variables se encuentran:

- Coarseness
- Complexity
- Contrast
- Strength
- Busyness

g) Shape 2D Features:

Estas características 2D de forma describen aspectos geométricos y estructurales de la región de interés en la imagen. Entre las variables se encuentran:

- Mesh Surface
- Spherical Disproportion
- Pixel Surface
- Maximum 2D diameter
- Perimeter
- Major Axis Length
- Perimeter to Surface ratio
- Minor Axis Length
- Sphericity
- Elongation

h) Shape 3D Features:

Estas características 3D de forma describen aspectos geométricos y estructurales de la región de interés en el espacio tridimensional. Entre las variables se encuentran:

- Mesh Volume
- Maximum 3D diameter
- Voxel Volume
- Maximum 2D diameter (Slice)
- Surface Area
- Maximum 2D diameter (Column)
- Surface Area to Volume ratio
- Maximum 2D diameter (Row)
- Sphericity
- Major Axis Length
- Elongation

- Flatness

2.3 Preprocesamiento de datos

En la fase inicial, se llevó a cabo un proceso de preprocesamiento de datos, enfocándonos específicamente en la estandarización de las características radiómicas. Este paso es esencial para garantizar que todas las variables estén en la misma escala y unidad antes de realizar la selección de características.

La estandarización es crucial cuando trabajamos con datos biomédicos, especialmente en el contexto de características radiómicas, que a menudo pueden variar ampliamente en rango y magnitud. Para asegurar que todas las variables contribuyan equitativamente a los análisis subsiguientes, es necesario alinearlas en una escala común.

Para llevar a cabo la estandarización de manera efectiva, se empleó el *StandardScaler* de la librería *sklearn.preprocessing* en *Python*. Esta herramienta realiza una transformación z-score, asegurando que las características tengan una media de cero y una desviación estándar de uno. Esta uniformidad facilita la comparación y el análisis de las características radiómicas, eliminando sesgos debidos a diferencias en las escalas originales.

El proceso con *StandardScaler* se aplicó a cada característica radiómica de manera individual. Primero, se calcularon la media y la desviación estándar de cada característica en el conjunto de datos completo. Luego, estos valores se utilizaron para estandarizar cada punto de datos, asegurando que todas las características compartieran la misma escala.

Este enfoque de estandarización sienta las bases para análisis más coherentes y resultados confiables en la selección de características. La uniformidad en las escalas y unidades optimiza la interpretación y la evaluación precisa de las relaciones y patrones en los datos radiómicos.

2.4 Optimización de características

La selección de características desempeña un papel fundamental en el análisis de datos y el desarrollo de modelos de aprendizaje automático, permitiendo la identificación de variables clave para predecir un resultado. En este contexto, diversos métodos de selección ofrecen enfoques variados, cada uno con sus propias ventajas y aplicaciones.

En conjuntos de datos con muchas características, utilizar todas ellas para construir modelos de Machine Learning puede generar problemas como bajo rendimiento debido a ruido adicional, tiempos de entrenamiento más largos y asignación innecesaria de recursos a características poco útiles. Se hace necesario abordar la selección de características de manera estratégica para evitar estos inconvenientes.



Figura 8: Obtenida de “Método no supervisado para la selección de rasgos en problemas de regionalización” [25]. Algoritmos de agrupamiento.

A continuación, se explican los diferentes tipos de métodos de selección: [25][26]

- **Métodos de filtro (*Filter Method*):**

Los métodos de filtro eligen las características importantes basándose directamente en los datos, es decir, se evalúan los atributos según sus propiedades inherentes sin depender de algoritmos de agrupación que orienten la búsqueda de atributos relevantes. La principal ventaja de los métodos de filtro se basa en su velocidad y capacidad de adaptarse a conjuntos de datos de diferentes tamaños de manera eficiente.

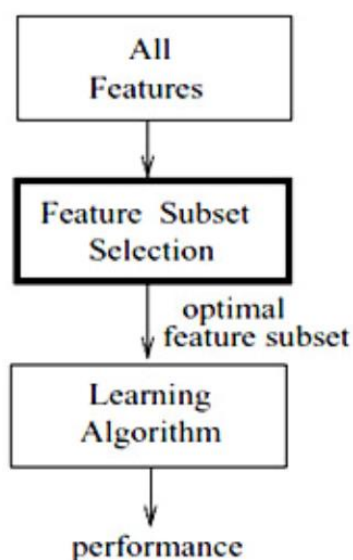


Figura 9: Obtenida de “A Genetic Based Wrapper Feature Selection Approach Using Nearest Neighbour Distance Matrix” [26]. Representación del método de filtro.

- **Métodos de envoltura (*Wrapper Method*):**

Los métodos de envoltura, como la búsqueda hacia adelante (**Forward Selection**), hacia atrás (**Backward Selection**) y bidireccional (**Recursive Feature Elimination**), consideran la interacción entre características y rendimiento del modelo. Aunque pueden ser computacionalmente costosos, proporcionan una selección más precisa. SVM y Gradient Boosting, al ser algoritmos que ajustan modelos de manera iterativa, pueden aprovecharse de técnicas de envoltura para evaluar la relevancia de las características de manera más dinámica durante el proceso de entrenamiento.

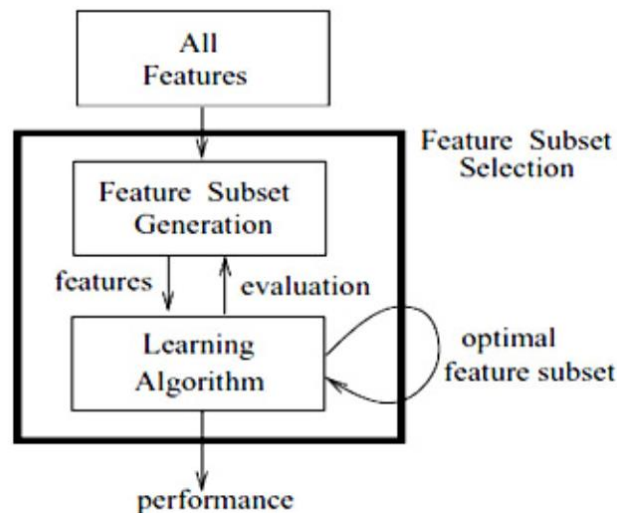


Figura 10: Obtenida de “A Genetic Based Wrapper Feature Selection Approach Using Nearest Neighbour Distance Matrix” [26]. Representación del método de envoltura.

- **Métodos empotrados (*Embedded Method*):**

En métodos empotrados, la selección de características se realiza internamente durante el entrenamiento del algoritmo, sin ser un paso previo al modelo. Cada algoritmo específico lleva a cabo esta selección de características durante su proceso de entrenamiento, buscando las más relevantes entre subconjuntos de características e hipótesis. Esta estrategia permite identificar dependencias con menor costo computacional en comparación con métodos envolventes. En el caso de Random Forest, la construcción de árboles de decisión también incluye una evaluación interna de la importancia de las características, convirtiéndolo en un método empotrado.

En el presente proyecto, se implementaron diversos métodos de selección de características para optimizar el rendimiento del modelo. Entre los enfoques utilizados se encuentran Random Forest como método de filtrado, LASSO como método empotrado, Recursive Feature Elimination (RFE) utilizando un clasificador SVM como método de envoltura, y Gradient Boosting como método empotrado y de envoltura.

2.4.1.1 Bosque Aleatorio (RF)

La selección de características utilizando Random Forest implica la aplicación del algoritmo *RandomForestClassifier*, que se encuentra en la biblioteca *scikit-learn*. En este procedimiento, se comienza importando la clase *RandomForestClassifier* y luego se establece un modelo de clasificación basado en la técnica de Bosques Aleatorios. Es importante destacar que, aunque Random Forest no está diseñado específicamente como un método de selección de características, en este caso se utiliza con ese propósito. Este enfoque aprovecha la potencia de múltiples árboles de decisión para mejorar la precisión y robustez del modelo global [27].

Para realizar la selección de características, se utiliza un conjunto de datos que incluye características radiómicas y una variable objetivo, en este caso, la "Respuesta". Durante la fase de entrenamiento, el modelo utiliza estas características radiómicas para aprender patrones y relaciones en los datos con el fin de realizar predicciones sobre la variable objetivo.

Una vez completado el entrenamiento, se accede a la propiedad "*feature_importances_*" del modelo. Esta propiedad ofrece una visión detallada de la importancia relativa de cada característica en el proceso de toma de decisiones del modelo. La importancia se cuantifica según la contribución de cada característica a la reducción de la impureza en los nodos de los árboles de decisión que componen el Bosque Aleatorio.

Después de obtener la cuantificación de la contribución de cada característica mediante la propiedad "*feature_importances_*" del modelo de Random Forest se establece un umbral, y las características cuya importancia relativa supere este umbral son seleccionadas para formar un conjunto más reducido y significativo, compuesto por aproximadamente 10 a 15 características. Este enfoque de umbral permite destacar las variables que más influyen en las predicciones del modelo, simplificando así el conjunto de características y facilitando una interpretación más clara y eficiente del modelo de aprendizaje automático.

En resumen, el método de selección de características mediante Random Forest no solo implica la construcción de un modelo de clasificación robusto (usado más adelante), sino que también brinda información valiosa sobre qué características radiómicas tienen un impacto más significativo en la capacidad predictiva del modelo, facilitando así la identificación de aspectos clave en los datos.

2.4.1.2 LASSO

LASSO (Least Absolute Shrinkage and Selection Operator) es una técnica de regularización utilizada en modelos de regresión que impone una penalización L1 a la función de costo durante el proceso de entrenamiento. Esta penalización tiene el efecto de reducir algunos coeficientes del modelo a cero, lo que conduce a la selección automática de características y, por ende, a la simplificación del modelo. [28]

Este método se implementa mediante la regresión logística. Primero, el conjunto de datos se divide en conjuntos de entrenamiento y prueba. Luego, se inicializa un modelo de

regresión logística configurado para aplicar la penalización L1 mediante el parámetro “*penalty=’l1’*”. Este modelo se entrena con el conjunto de entrenamiento, ajustando los coeficientes a los datos mientras aplica la regularización L1.

Después del entrenamiento, se evalúa el rendimiento del modelo en el conjunto de prueba utilizando la métrica de precisión. Sin embargo, la característica distintiva de LASSO es la identificación de las características seleccionadas. Esto se logra examinando los coeficientes del modelo, y las características asociadas a coeficientes no nulos se consideran seleccionadas por LASSO. El código revela que se han seleccionado 13 características, ya que algunas de las características originales han sido reducidas a cero debido a la penalización L1.

En resumen, LASSO, aplicado a través de la regresión logística en este contexto, ofrece una metodología efectiva para la selección automática de características al forzar algunos coeficientes a cero. El código implementa este proceso al entrenar el modelo, evaluar su rendimiento y determinar las características seleccionadas por LASSO, proporcionando así un modelo más eficiente e interpretable.

2.4.1.3 Eliminación Recursiva de Características (RFE)

Recursive Feature Elimination (RFE) es un método de selección de características que utiliza un modelo dado, en este caso, una máquina de soporte vectorial (SVM), para iterativamente eliminar las características menos importantes. Funciona ajustando el modelo al conjunto de datos completo, evaluando la importancia de cada característica y eliminando la menos importante. Este proceso se repite hasta alcanzar el número deseado de características seleccionadas.

En términos generales, el proceso comienza con la creación de un clasificador SVM con un kernel lineal y un parámetro de regularización establecido en 1. Este clasificador será utilizado por RFE para evaluar la importancia de las características. A continuación, se emplea la técnica de validación cruzada Leave-One-Out (LOOCV), donde en cada iteración, una observación se selecciona como conjunto de prueba y las demás como conjunto de entrenamiento.

Durante el bucle de LOOCV, RFE ajusta sucesivamente el modelo al conjunto de entrenamiento, seleccionando características de manera iterativa y evaluando su relevancia. Después de cada ajuste, se entrena el clasificador SVM con las características seleccionadas y se realizan predicciones en el conjunto de prueba.

El resultado final es la identificación de las características más influyentes, y estas características seleccionadas se almacenan en un *DataFrame*. En este caso, se ha establecido que se seleccionen 10 características. Este enfoque permite construir un modelo más eficiente y fácilmente interpretable al centrarse en un subconjunto de características esenciales para la predicción del modelo SVM.

2.4.1.4 Gradient Boosting

El método de selección de características mediante XGBoost se basa en la utilización de Gradient Boosting, un algoritmo de aprendizaje automático que combina la predicción de múltiples modelos más simples (típicamente árboles de decisión débiles) para mejorar la precisión global. XGBoost, en particular, se destaca por su eficiencia y capacidad para manejar conjuntos de datos grandes, así como por su capacidad para realizar selección de características de manera integrada.

En el contexto de XGBoost, las importancias de las características se determinan mediante la evaluación de cómo cada característica contribuye al rendimiento del modelo durante el proceso de construcción de árboles. El algoritmo asigna un valor de importancia a cada característica, reflejando su influencia en la capacidad predictiva del modelo. Estas importancias se utilizan posteriormente para identificar y seleccionar las características más informativas.

Se comienza inicializando un clasificador XGBoost mediante la creación de una instancia de `xgb.XGBClassifier()`. Este clasificador se entrena con el conjunto de datos de características y la variable objetivo "Respuesta". Posteriormente, se accede a la propiedad `feature_importances_` del modelo para obtener las importancias de cada característica.

Luego, se establece un umbral de importancia y se seleccionan solo aquellas características cuya importancia supera dicho umbral. Esto ayuda a reducir el conjunto de características a las más relevantes según los criterios definidos.

Finalmente, se seleccionaron 15 características, luego de aplicar el umbral de importancia. Este enfoque permite una visión más detallada de las características más influyentes según el modelo XGBoost, facilitando la interpretación y toma de decisiones.

2.5 Desarrollo de modelos predictivos

En el proceso de desarrollo de modelos predictivos, la elección adecuada de algoritmos de predicción fue esencial. La diversidad de enfoques disponibles en inteligencia artificial y radiómica requería una cuidadosa consideración antes de tomar decisiones fundamentales.

Una vez que se definieron los modelos a utilizar, el siguiente paso crucial fue la división del conjunto de datos en tres partes: entrenamiento, prueba y validación. En este caso, se optó por una división del 80% para entrenamiento, utilizando la técnica de validación cruzada con k-fold ($k=5$) para dividir este conjunto en subconjuntos de entrenamiento y prueba en cada iteración. Esto proporcionó una evaluación más robusta del rendimiento del modelo al realizar predicciones iniciales en diferentes conjuntos de prueba.

Simultáneamente, se reservó el 20% restante para el conjunto de validación. Este enfoque permitió entrenar los modelos con la mayoría de los datos disponibles y, al mismo tiempo, disponer de un conjunto independiente para evaluar y validar la capacidad predictiva mediante diversas métricas.

En la fase de validación, una vez que los modelos fueron entrenados con la técnica de k-fold, se realizaron nuevas predicciones utilizando el conjunto de datos completamente nuevo y no visto durante el proceso de entrenamiento. Esto permitió evaluar la capacidad de generalización de los modelos a datos no utilizados previamente, asegurando así su utilidad y fiabilidad en situaciones del mundo real.

Este enfoque estructurado en el desarrollo de modelos garantizó una evaluación rigurosa y la optimización continua de su desempeño, proporcionando resultados más precisos y confiables para asistir en las decisiones clínicas para pacientes con cáncer de recto.

En el ámbito del aprendizaje automático, se distinguen dos enfoques esenciales: el aprendizaje supervisado y el aprendizaje no supervisado. Mientras que en el aprendizaje supervisado el modelo se nutre de datos de entrada y sus salidas correspondientes conocidas para prever resultados futuros, el aprendizaje no supervisado se enfoca en descubrir patrones intrínsecos o estructuras específicas en los datos de entrada, prescindiendo de información de salida conocida para guiar el proceso de aprendizaje. Ahora, procedamos a explorar la aplicación de estos enfoques en el desarrollo de modelos predictivos.

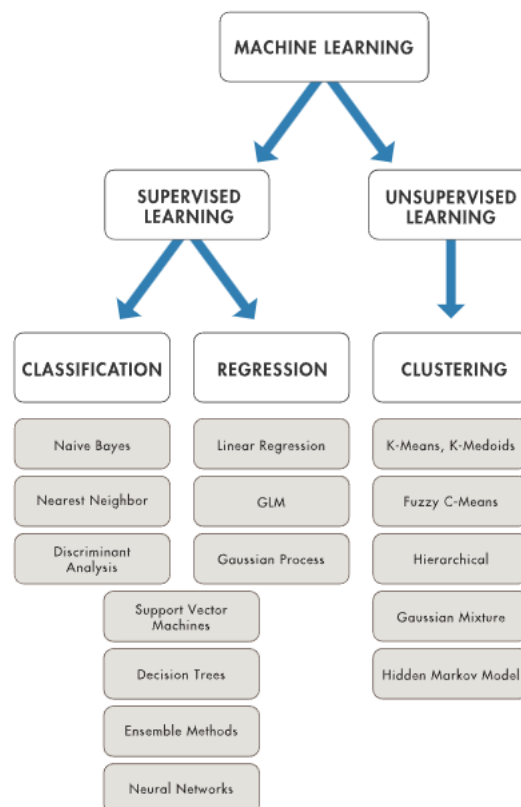


Figura 11. Obtenida de “Machine Learning - MathWorks” [29]. Técnicas de Machine Learning.

En este estudio, se construyeron cuatro modelos predictivos basados en modelos supervisados. Se utilizaron Random Forest, Regresión Logística, Máquinas de Soporte Vectorial (SVM) y Redes Neuronales.

2.5.1 Bosque Aleatorio (RF)

Random Forest es un algoritmo de aprendizaje supervisado que utiliza múltiples árboles. Cada árbol se construye con un conjunto de datos aleatorio, reduciendo el sobreajuste y mejorando la robustez del modelo frente a datos nuevos [27].

En la configuración del modelo Random Forest Classifier se establece que el bosque estará compuesto por 100 árboles, proporcionando así diversidad al modelo. La profundidad máxima de los árboles se define como ilimitada (*None*), permitiendo que los árboles crezcan hasta su máxima capacidad. Además, se especifica que se requieren al menos 2 muestras para formar un nodo hoja y 2 muestras para realizar una división interna en cada árbol. Estos parámetros se eligen para equilibrar la complejidad del modelo y su capacidad para adaptarse a los datos.

Para evaluar la robustez del modelo, se implementa la técnica de validación cruzada K-Fold con 5 divisiones. Este enfoque implica dividir el conjunto de datos en cinco partes, utilizando cuatro partes para entrenar el modelo y la restante para evaluar su rendimiento. Este proceso se repite en cinco iteraciones, asegurando que cada parte del conjunto de datos sirva como conjunto de prueba al menos una vez. La validación cruzada es esencial para obtener estimaciones más confiables del rendimiento del modelo, especialmente cuando los datos son limitados.

El proceso de entrenamiento y evaluación del modelo se lleva a cabo en un bucle que itera sobre las divisiones de la validación cruzada. En cada iteración, se separan los datos en conjuntos de entrenamiento y prueba, y el modelo se entrena utilizando el conjunto de entrenamiento. Luego, se realizan predicciones en el conjunto de prueba, y se registran las respuestas reales y predichas. Este procedimiento se repite para cada división de la validación cruzada.

Al final del proceso, se obtiene un conjunto de predicciones consolidadas, que incluye tanto las respuestas reales como las predichas para cada conjunto de prueba en todas las iteraciones de la validación cruzada. Este conjunto de predicciones será utilizado para evaluar el rendimiento global del modelo en términos de su capacidad para predecir con precisión las respuestas en nuevos datos.

2.5.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) es un algoritmo versátil de aprendizaje supervisado para clasificación y regresión. Busca un hiperplano efectivo para separar clases (clasificación) o ajustarse óptimamente a los datos (regresión) en un espacio multidimensional. Utiliza funciones de kernel para tratar datos no linealmente separables y es robusto ante atípicos, centrándose en vectores de soporte clave. Su versatilidad se destaca al gestionar múltiples clases mediante clasificadores binarios. [30]

La SVM elige puntos extremos o vectores de soporte para construir el hiperplano. Estos casos críticos son los "vectores de soporte", de ahí el nombre de Support Vector Machine. La imagen siguiente ilustra la clasificación de dos categorías mediante un límite de decisión o hiperplano:

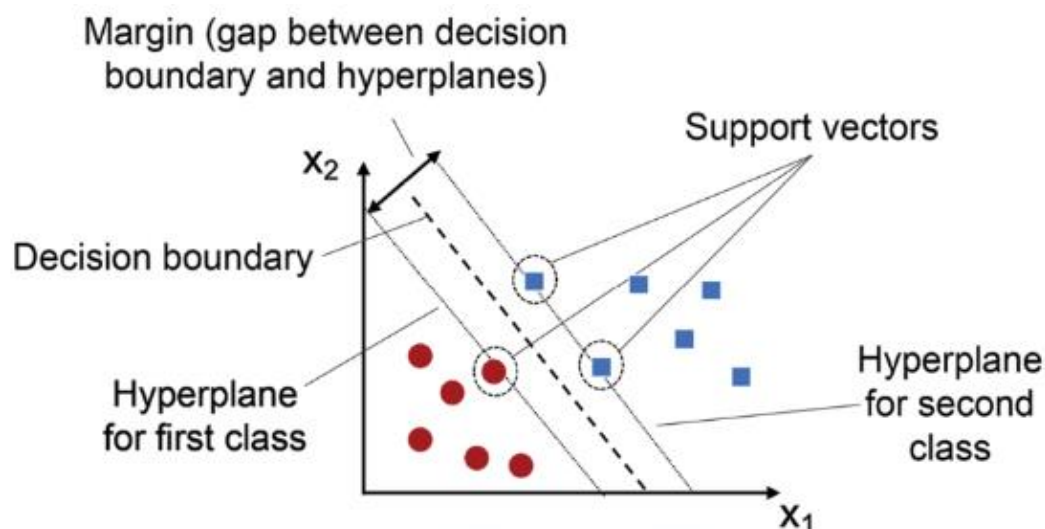


Figura 13. Obtenida de “Classification Model: SVM Classifier in Python with Example” [31]. SVM con hiperplano de margen máximo y vectores de soporte.

Este modelo utiliza la clase *SVC* de la biblioteca *scikit-learn*, con la especificación de un kernel radial “*rbf*”. El modelo se inicializa con un estado aleatorio constante “*random_state=42*” para garantizar reproducibilidad.

Posteriormente, se emplea la técnica de validación cruzada K-Fold con 5 divisiones para evaluar el rendimiento del modelo de manera robusta. En cada iteración de la validación cruzada, se divide el conjunto de datos en conjuntos de entrenamiento y prueba. El modelo SVM se entrena con el conjunto de entrenamiento y se utilizan las características del conjunto de prueba para realizar predicciones.

2.5.3 Regresión Logística (RL)

El modelo de regresión logística es una herramienta eficaz en el aprendizaje automático para abordar problemas de clasificación. Su enfoque se basa en estimar la probabilidad de pertenencia a una clase particular, facilitando decisiones de clasificación fundamentadas en dichas probabilidades.

Este modelo sigue un enfoque similar al anterior, estableciendo un esquema de validación cruzada con 5 pliegues. En este proceso, el conjunto de datos se divide en cinco partes, utilizando cuatro de ellas para entrenar el modelo y la restante para evaluar su rendimiento. Esta estrategia garantiza una evaluación robusta al permitir que cada parte del conjunto de datos sirva alternativamente como conjunto de entrenamiento y prueba.

A continuación, se instancia un modelo de regresión logística, configurado con parámetros clave, como el número máximo de iteraciones y la estrategia multiclase para abordar problemas con más de dos clases. La etapa subsiguiente involucra la realización de predicciones utilizando el modelo entrenado en el conjunto de prueba. Estas predicciones se almacenan para cada iteración del proceso de validación cruzada.

2.5.4 Red Neuronal Artificial

Las redes neuronales, también llamadas RNA o RNS, son esenciales en el aprendizaje automático y constituyen la base de los algoritmos de aprendizaje profundo. Estas redes imitan la comunicación entre neuronas biológicas y se componen de capas de nodos, incluyendo entrada, una o más capas ocultas y salida. Cada neurona, interconectada con otras, tiene un peso y umbral. Cuando la salida supera el umbral, se activa y transmite datos a la siguiente capa; de lo contrario, la información no se propaga. El entrenamiento con datos ajusta pesos y umbrales para mejorar la capacidad predictiva. Una vez ajustadas con precisión, estas redes se vuelven potentes herramientas de inteligencia artificial, capaces de procesar datos eficientemente y realizar tareas complejas a alta velocidad. [32]

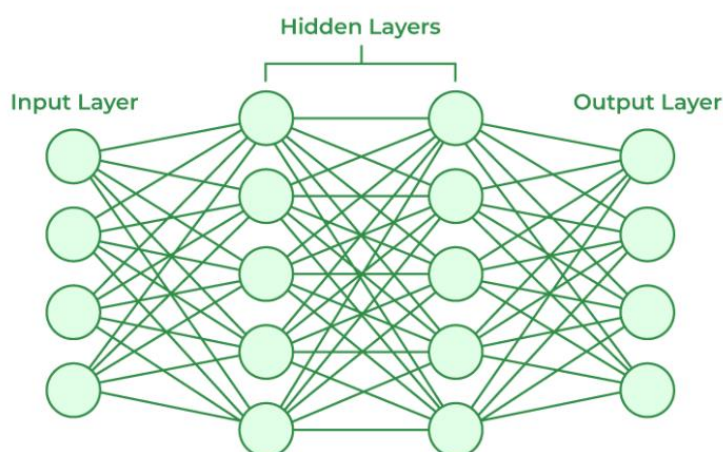


Figura 14. Obtenida de “*Artificial Neural Networks and its Applications*” [33].
Arquitectura de una red neuronal artificial.

En el contexto de este proyecto, se está implementando una Red Neuronal Artificial (RNA) utilizando la biblioteca Keras, y se ha adoptado un enfoque específico denominado Validación Cruzada Leave-One-Out (LOOCV). Este enfoque implica que, durante la validación cruzada, se entrena el modelo con todos los datos excepto uno, y se repite este proceso hasta que cada punto de datos ha sido excluido y utilizado como conjunto de prueba una vez. Esta metodología ayuda a evaluar la capacidad de generalización del modelo de manera exhaustiva.

La estructura del modelo se define como secuencial en *Keras*, lo que significa que las capas se organizan de manera lineal, una tras otra. Además, se emplean capas densas, donde cada neurona de una capa está conectada a todas las neuronas de la capa siguiente. Este enfoque secuencial y denso es común en la construcción de modelos neuronales.

La primera capa de la red, que actúa como la capa de entrada, consta de 28 neuronas y utiliza la función de activación *ReLU* (Rectified Linear Unit). La elección de *ReLU* es habitual en las capas intermedias de una red neuronal debido a su capacidad para introducir no linealidades en el modelo, permitiendo la representación de relaciones complejas en los datos.

La segunda capa es una capa oculta que contiene 16 neuronas. Además de la activación *ReLU*, se aplica una técnica llamada regularización L2. Esta regularización tiene como objetivo controlar el sobreajuste, penalizando los pesos grandes en la red. Al hacerlo, se busca mejorar la generalización del modelo, es decir, su capacidad para hacer predicciones precisas en datos no vistos.

Después de la capa oculta, se introduce una capa de *dropout* con una tasa del 0.3. El *dropout* es una estrategia de regularización que implica aleatorizar el apagado o ignorar algunas neuronas durante el entrenamiento. Esta aleatorización ayuda a prevenir el sobreajuste al introducir variabilidad en el modelo, evitando que las neuronas dependan demasiado unas de otras.

Finalmente, la capa de salida consta de una sola neurona con activación *sigmoid*. La elección de *sigmoid* es adecuada para problemas de clasificación binaria, ya que produce una salida en el rango de 0 a 1, interpretado como probabilidades. Una salida cercana a 0 indica una clase, mientras que una salida cercana a 1 indica la otra.

La configuración específica de la red, como el número de neuronas en cada capa, la regularización L2, el *dropout* y la activación *sigmoid* en la capa de salida, es crucial para entender cómo se ha ajustado la RNA para abordar la tarea específica de clasificación binaria. Estas elecciones se derivan de consideraciones teóricas y empíricas para lograr un equilibrio entre el ajuste preciso a los datos de entrenamiento y la generalización a datos nuevos.

En términos de sobreajuste (*overfitting*), es un fenómeno que puede ocurrir cuando un modelo se adapta demasiado a los datos de entrenamiento y no generaliza bien a datos no vistos. Las estrategias de regularización, como la regularización L2 y el *dropout*, desempeñan un papel crucial en abordar este problema. La regularización L2 controla la complejidad del modelo, mientras que el *dropout* introduce variabilidad, ambos contribuyendo a una mejor capacidad de generalización del modelo. Estas medidas buscan asegurar que la red sea capaz de hacer predicciones precisas en una variedad de situaciones y no se vuelva demasiado específica para el conjunto de entrenamiento. [34]

Adicionalmente, otras estrategias podrían fortalecer aún más la resistencia contra el sobreajuste. La detención temprana, por ejemplo, podría ser implementada para monitorear el rendimiento del modelo en un conjunto de validación durante el entrenamiento y detenerlo cuando el rendimiento deja de mejorar, evitando así la sobre optimización. Asimismo, el aumento de datos podría ser empleado para introducir variabilidad mediante transformaciones en los datos de entrada, asegurando que el modelo no memorice características específicas y pueda generalizar de manera más efectiva a nuevos escenarios. Estas estrategias adicionales contribuirían a consolidar un modelo robusto y generalizable en diversas condiciones. [34]

2.6 Validación de modelos predictivos

Tras la construcción de los modelos de predicción y la configuración de sus parámetros se procedió a su evaluación y validación. La selección del modelo más adecuado se basa

en su capacidad para clasificar de manera precisa los datos, cumpliendo con los objetivos iniciales del estudio. Se seleccionaron las métricas más comúnmente empleadas para evaluar el rendimiento de estos modelos.

2.6.1.1 Exactitud

La **exactitud** es una métrica que evalúa la proporción de predicciones correctas realizadas por un modelo en relación con el total de predicciones realizadas. Se calcula dividiendo el número de predicciones correctas (verdaderos positivos más verdaderos negativos) entre el número total de predicciones realizadas (verdaderos positivos más verdaderos negativos más falsos positivos más falsos negativos).

La fórmula para la exactitud es:

$$\text{Exactitud} = \frac{\text{Verdaderos Positivos} + \text{Verdaderos Negativos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos} + \text{Falsos Negativos} + \text{Verdaderos Negativos}}$$

Esta métrica proporciona una visión general de la efectividad del modelo al clasificar las instancias correctamente. Sin embargo, es importante tener en cuenta que la exactitud puede ser engañosa en situaciones donde las clases están desbalanceadas, ya que un modelo podría obtener una alta exactitud simplemente prediciendo siempre la clase mayoritaria.

2.6.1.2 Precisión

La **precisión** es una métrica que evalúa la proporción de predicciones positivas realizadas correctamente en relación con el total de predicciones positivas realizadas por el modelo. En otras palabras, la precisión mide la capacidad del modelo para evitar realizar predicciones positivas incorrectas. Un valor alto de precisión indica que el modelo tiene menos probabilidad de clasificar incorrectamente instancias negativas como positivas.

La fórmula para la precisión es:

$$\text{Precisión} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}}$$

La precisión es especialmente relevante en situaciones donde los falsos positivos tienen un impacto significativo o cuando se busca minimizar los errores de clasificación en la clase positiva.

2.6.1.3 Sensibilidad

La **sensibilidad**, también conocida como **recall**, es una métrica que evalúa la capacidad de un modelo para identificar correctamente todas las instancias positivas en un conjunto de datos. Se centra en minimizar los falsos negativos y es especialmente útil cuando es crucial identificar todas las instancias positivas, incluso a costa de aumentar los falsos positivos.

La fórmula para la sensibilidad es:

$$\text{Sensibilidad} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$$

La sensibilidad se interpreta como la proporción de instancias positivas que el modelo logra identificar correctamente con respecto a todas las instancias positivas reales en el conjunto de datos. Un valor alto de sensibilidad indica que el modelo es efectivo para capturar la mayoría de las instancias positivas, aunque pueda haber un aumento en los falsos positivos.

2.6.1.4 F1 score

El **F1 score** es una métrica que combina las métricas de precisión y sensibilidad en un solo valor, proporcionando una medida más equilibrada del rendimiento de un modelo de clasificación. Esta métrica es especialmente útil cuando existe un desequilibrio entre las clases o cuando se quiere encontrar un equilibrio entre la precisión y la capacidad de identificar positivos.

La fórmula del F1 score es:

$$F1 = 2 \times \frac{\text{Precisión} \times \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}}$$

El F1 score varía entre 0 y 1, donde un valor más alto indica un mejor rendimiento. Se busca un equilibrio entre la precisión y la sensibilidad, ya que ambos componentes contribuyen al puntaje final. Si tanto la precisión como la sensibilidad son altas, el F1 score será cercano a 1, indicando un rendimiento sólido en la capacidad del modelo para clasificar correctamente ambas clases.

2.6.1.5 Matriz de confusión

La matriz de confusión es usada en la evaluación de modelos de clasificación. Proporciona una visión detallada del rendimiento del modelo al comparar las predicciones con los valores reales de las clases. La matriz de confusión organiza los resultados en cuatro categorías:

- **Verdaderos Positivos (TP):** Son los casos en los que el modelo predijo correctamente la clase positiva.
- **Verdaderos Negativos (TN):** Representa los casos en los que el modelo predijo correctamente la clase negativa.
- **Falsos Positivos (FP):** Indica los casos en los que el modelo predijo incorrectamente la clase positiva.
- **Falsos Negativos (FN):** Representa los casos en los que el modelo predijo incorrectamente la clase negativa.

La estructura de la matriz de confusión es la siguiente:

Real = YES	TP	FN
	FP	TN
Real = NO		
	Predicted = YES	Predicted = NO

Figura 19. Obtenida de “Calculando la precisión en un modelo de clasificación multiclase” [35]. Matriz de confusión para clasificador binario.

La interpretación de la matriz de confusión es esencial para entender las fortalezas y debilidades del modelo. Además, se utiliza para calcular varias métricas de evaluación, como precisión, sensibilidad (recall), especificidad y otras, que proporcionan una visión más detallada del rendimiento en diferentes aspectos.

2.6.1.6 Curva ROC

Las **curvas ROC (Receiver Operating Characteristic)** son herramientas gráficas utilizadas para evaluar y comparar el rendimiento de modelos de clasificación en distintos umbrales de discriminación. Estas curvas representan la tasa de verdaderos positivos (sensibilidad) frente a la tasa de falsos positivos (1 - especificidad) para diferentes valores de umbral. En la representación gráfica de una curva ROC, el eje X muestra la tasa de falsos positivos, mientras que el eje Y muestra la tasa de verdaderos positivos.

El área bajo la curva ROC (AUC-ROC) proporciona una medida numérica del rendimiento del modelo. Un AUC-ROC cercano a 1 indica un buen rendimiento del modelo, mientras que un valor cercano a 0.5 sugiere un rendimiento similar al azar. Las curvas ROC y el AUC-ROC son herramientas útiles para la comparación de varios modelos y la selección del más adecuado para la tarea de clasificación específica.

3 Resultados

3.1 Selección de características

En este estudio, se ha establecido como objetivo la selección de entre 10 y 15 características mediante cada uno de estos métodos, asegurando así la obtención de un conjunto de variables significativas y pertinentes para el modelo final.

En representación gráfica inicial (ver figura 15), mediante el método de Random Forest, se observa la importancia de todas las características radiómicas consideradas, destacando la contribución de cada una de ellas. Posteriormente, se aplicó un umbral para la selección de las características más relevantes.

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

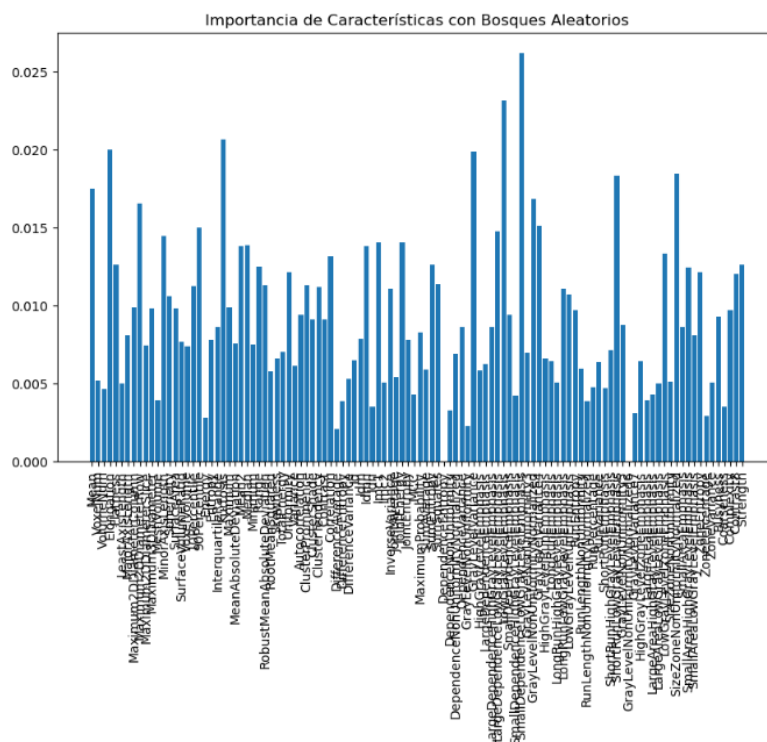
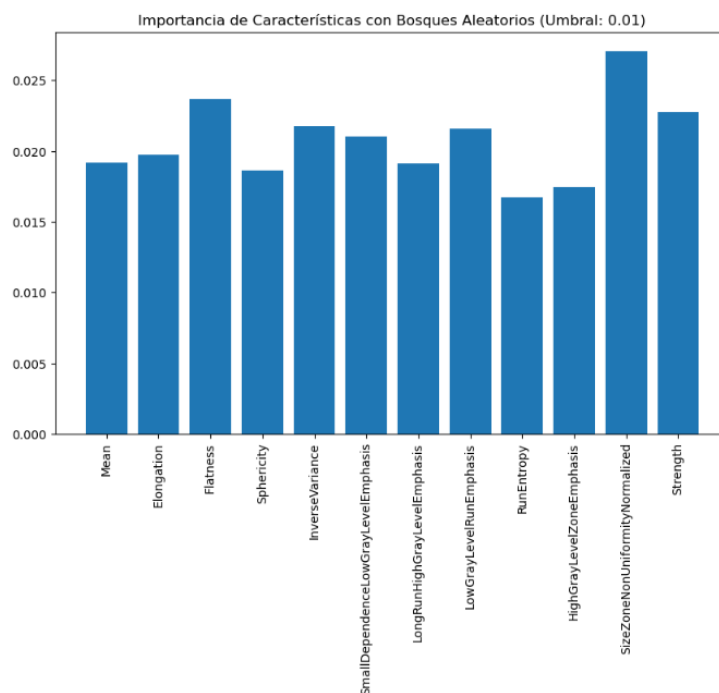


Figura 15. Importancia de las características radiómicas con el método Random Forest.

Una vez aplicado el umbral (ver figura 16), nos permite construir un conjunto final de 12 características compuesto por 1 First Order Feature, 1 Shape 2D Feature, 5 GLCM Features, 1 GLDM Feature, 3 GLRLM Features, 3 GLSZM Features y 1 NGTDM Feature.



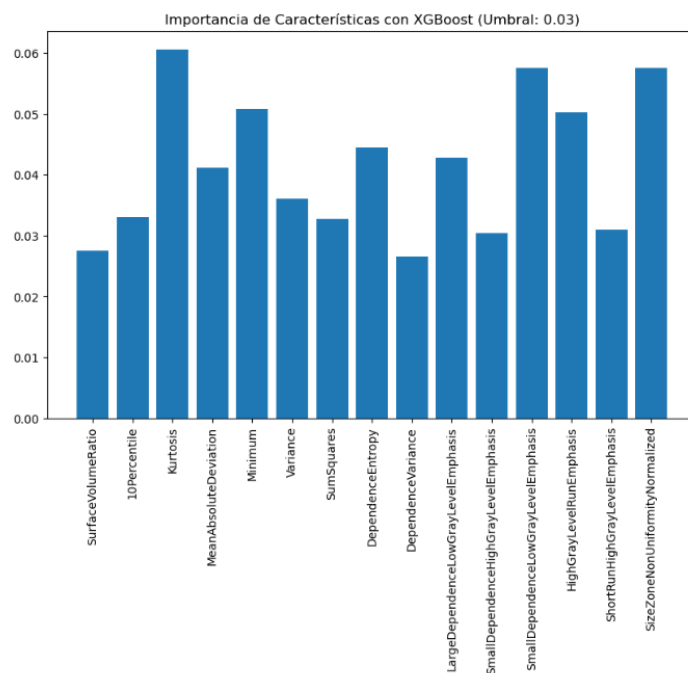


Figura 18. Importancia de las 15 características radiómicas reducidas con el método Gradient Boosting.

3.2 Evaluación y confirmación de modelos

3.2.1.1 Primera aplicación

En esta primera sección, se presentan las curvas Receiver Operating Characteristic (ROC) relacionadas con los conjuntos de características seleccionadas mediante los 4 métodos de selección, en la aplicación de los modelos predictivos. Se han visualizado las curvas ROC-AUC para cada clase individual y sus respectivas medias.

- **Método de selección: Random Forest.**

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

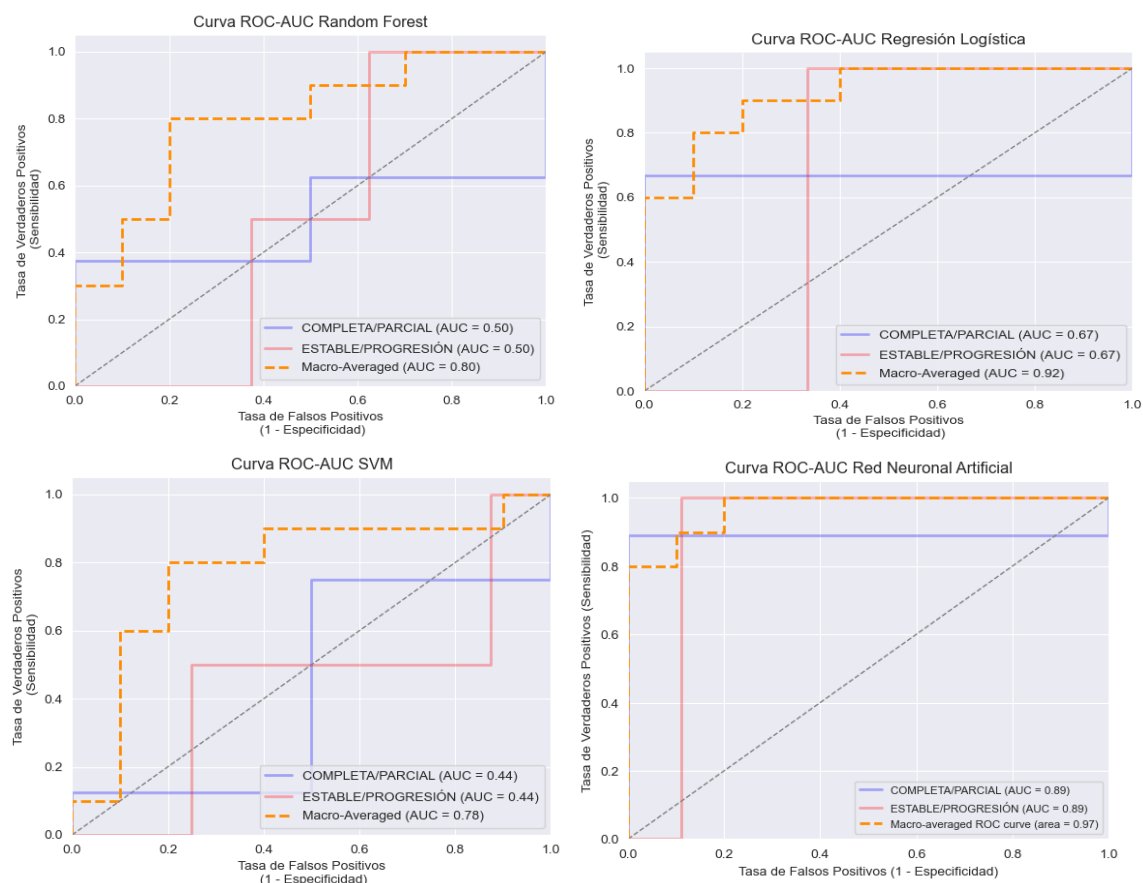


Figura 20: Curvas ROC-AUC de cada clase y media entrenadas por los modelos Random Forest, Regresión Logística, SVM y Red Neuronal Artificial con el conjunto de características seleccionadas mediante el método Random Forest.

En términos generales, se observan resultados diversos en las AUC medias, que cuantifican la capacidad discriminativa de los modelos: 0.80 para Random Forest, 0.92 para Regresión Logística, 0.78 para SVM y 0.97 para Red Neuronal Artificial.

Al examinar detalladamente el rendimiento de cada modelo en la clasificación de respuestas, se identifican patrones distintivos:

- **Random Forest:** Se evidencia una capacidad limitada para distinguir entre respuesta completa/parcial y respuesta estable/progresión, con AUC de 0.50 para ambas clases, indicando una clasificación aleatoria.
- **Regresión Logística:** Muestra una mejora en comparación con Random Forest, con AUC de 0.67 para ambas clases. Sin embargo, persisten limitaciones en la discriminación precisa.
- **SVM:** Exhibe un desempeño menos robusto, con AUC de 0.44 para ambas clases, sugiriendo dificultades en la separación efectiva de las respuestas.
- **Red Neuronal Artificial:** Destaca con AUC de 0.89 para ambas clases, revelando una buena capacidad para discriminar entre respuestas completa/parcial y estable/progresión.

- **Método de selección: Recursive Feature Elimination (RFE).**

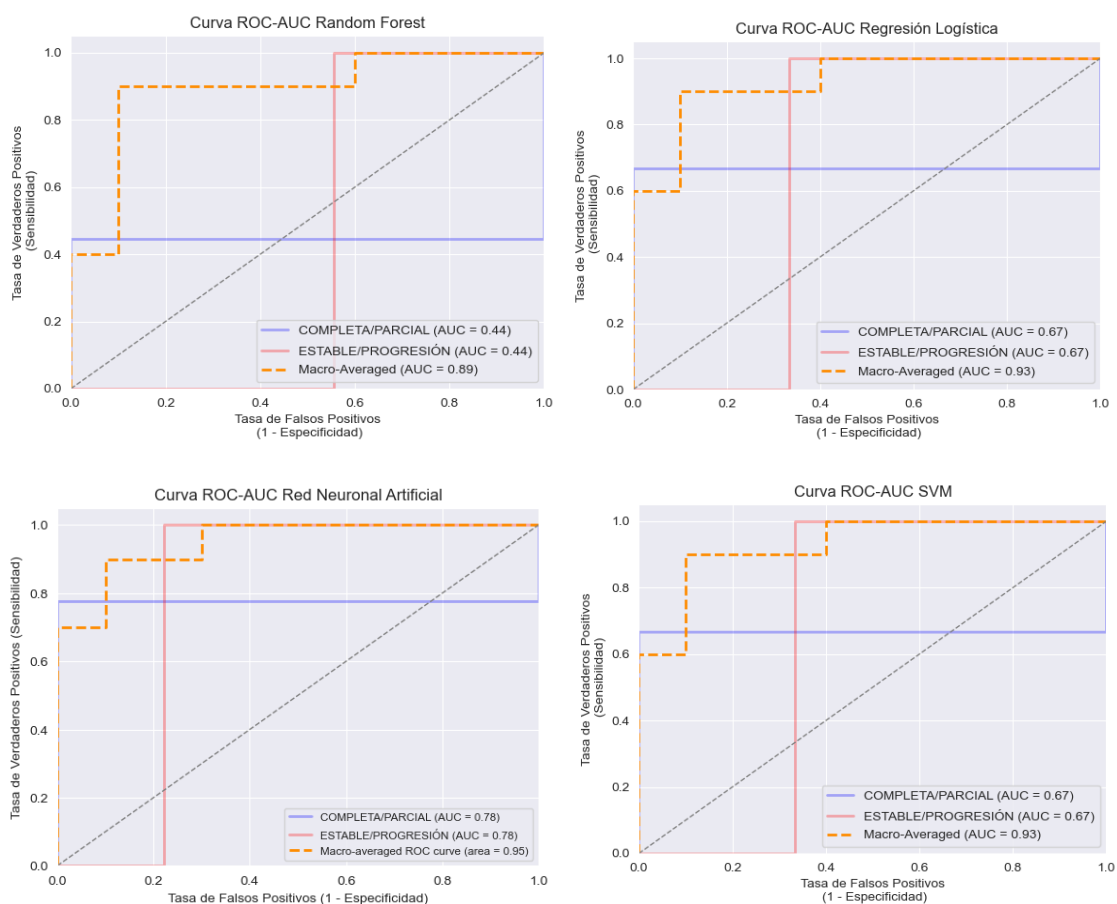


Figura 21: Curvas ROC-AUC de cada clase y media entrenadas por los modelos Random Forest, Regresión Logística, SVM y Red Neuronal Artificial con el conjunto de características seleccionadas mediante el método Recursive Feature Elimination (RFE).

Se implementó el método Recursive Feature Elimination (RFE) para la selección de características, evaluando su impacto en el rendimiento de diversos algoritmos de predicción. A continuación, se presentan los resultados obtenidos para cada modelo:

- **Random Forest:** Bajo la aplicación de RFE, el modelo de Random Forest demostró una AUC media de 0.89. No obstante, al clasificar entre respuesta completa/parcial y respuesta estable/progresión, se registraron AUC de 0.44 para ambas clases, indicando una capacidad limitada para distinguir entre estas respuestas.
- **Regresión Logística:** La Regresión Logística, beneficiándose de la selección de características mediante RFE, experimentó una mejora en la AUC media, alcanzando un valor de 0.93. En términos de clasificación, se observó una AUC de 0.67 para ambas clases, denotando una discriminación moderada entre respuesta completa/parcial y respuesta estable/progresión.
- **Red Neuronal Artificial:** La Red Neuronal Artificial, al ser sometida a RFE, mostró una destacada AUC media de 0.95. Al evaluar la clasificación entre

respuesta completa/parcial y respuesta estable/progresión, se obtuvieron AUC de 0.78 para ambas clases, indicando una capacidad sólida de discriminación.

- **SVM:** Con la aplicación de RFE, el modelo de SVM presentó una AUC media de 0.93. Sin embargo, al clasificar entre las respuestas deseadas, se registró una AUC de 0.67 para ambas clases, señalando limitaciones en la separación efectiva de las respuestas.

- **Método de selección: Gradient Boosting.**

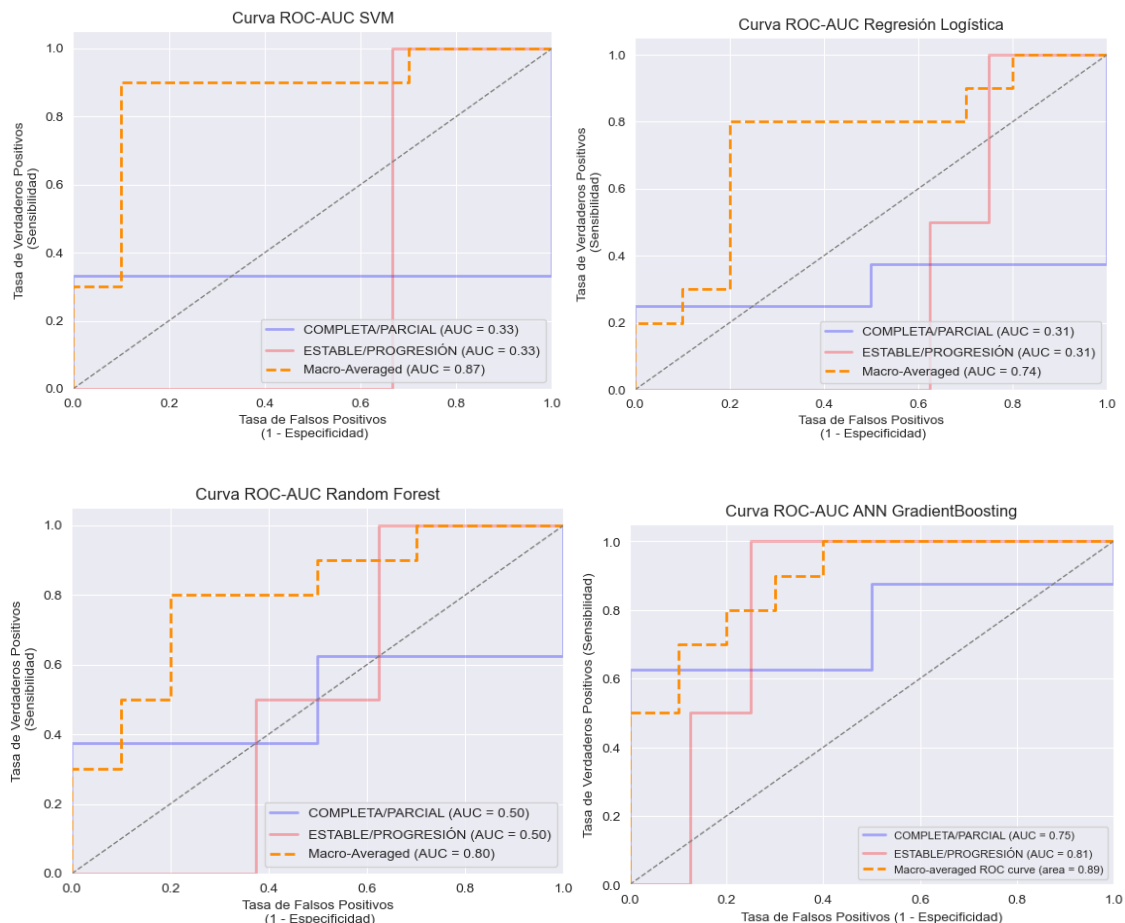


Figura 22: Curvas ROC-AUC de cada clase y media entrenadas por los modelos Random Forest, Regresión Logística, SVM y Red Neuronal Artificial con el conjunto de características seleccionadas mediante el método Gradient Boosting.

- **SVM:** AUC media de 0.87. Clasificación entre respuesta completa/parcial y respuesta estable/progresión con AUC de 0.33 para ambas clases, indicando limitaciones en la discriminación.
- **Regresión Logística:** AUC media de 0.74. AUC de 0.31 para ambas clases al clasificar entre respuesta completa/parcial y respuesta estable/progresión, revelando desafíos en la discriminación precisa.
- **Random Forest:** AUC media de 0.80. Clasificación aleatoria entre respuesta completa/parcial y respuesta estable/progresión con AUC de 0.50 para ambas clases.

- **Red Neuronal Artificial:** AUC media de 0.89. Buena capacidad de discriminación al clasificar entre respuesta completa/parcial y respuesta estable/progresión, con AUC de 0.75 y 0.81 respectivamente.

- **Método de selección: LASSO.**

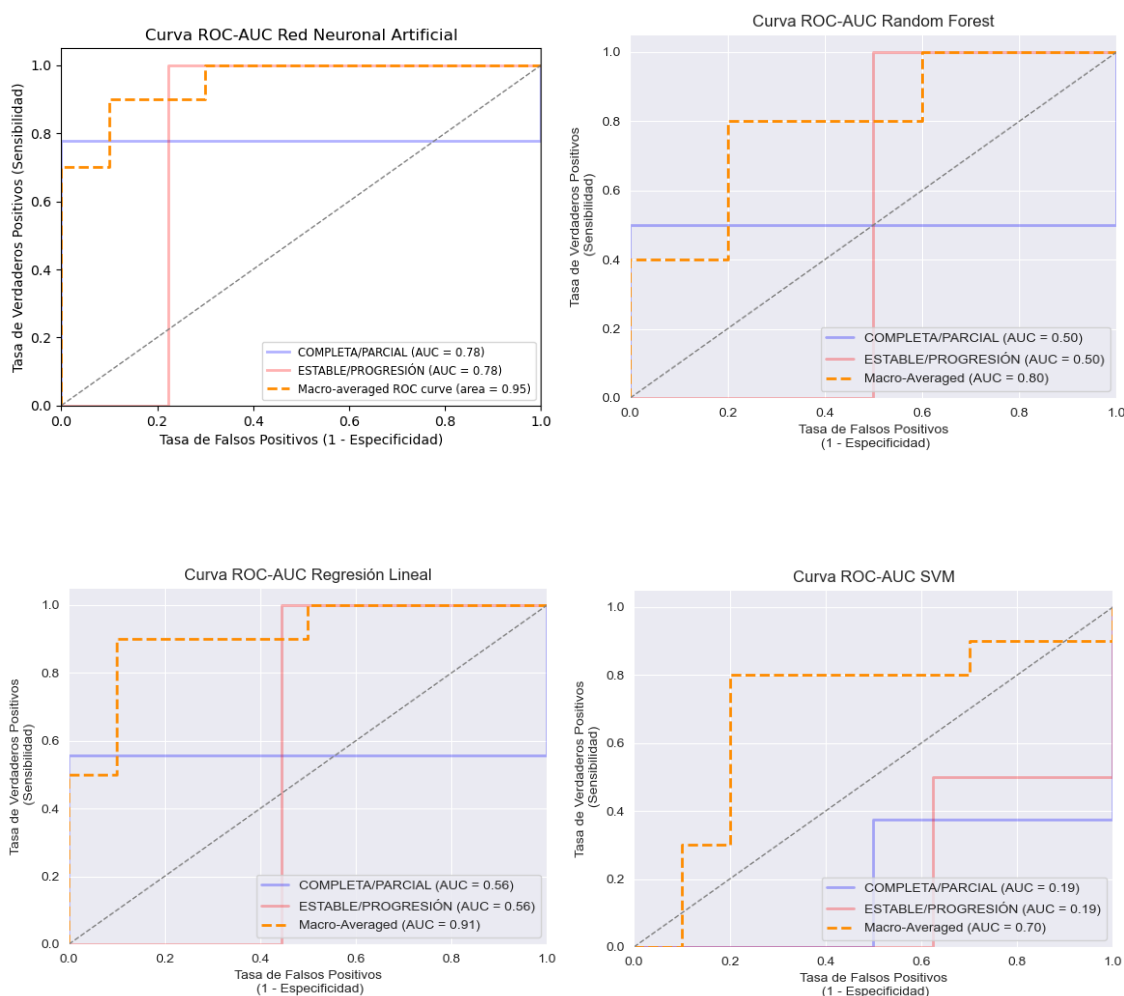


Figura 23: Curvas ROC-AUC de cada clase y media entrenadas por los modelos Random Forest, Regresión Logística, SVM y Red Neuronal Artificial con el conjunto de características seleccionadas mediante el método LASSO.

- **Red Neuronal Artificial:** Mantuvo su robustez con AUC media de 0.95. AUC de 0.78 para ambas clases indicó fuerte capacidad de discriminación.
- **Random Forest:** AUC media de 0.80. Sin embargo, clasificación aleatoria con AUC de 0.50 para ambas clases al evaluar respuesta completa/parcial y respuesta estable/progresión.
- **Regresión Lineal:** AUC media de 0.91. AUC de 0.56 para ambas clases al clasificar respuesta completa/parcial y respuesta estable/progresión.
- **SVM:** AUC media de 0.70. Limitaciones en capacidad de discriminación con AUC de 0.19 para ambas clases al clasificar respuesta completa/parcial y respuesta estable/progresión.

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

En la siguiente tabla se resumen los valores del Área bajo la Curva (AUC) obtenidos de las curvas Receiver Operating Characteristic (ROC) para cada modelo predictivo, considerando los diferentes métodos de selección de características radiómicas utilizados en ambas aplicaciones.

Tabla 1. Valores AUC de los 4 modelos de predicción en los 4 conjuntos de datos.

ROC - AUC	Red Neuronal Artificial	SVM	Random Forest	Regresión Logística
Random Forest	0,97	0,78	0,80	0,92
RFE	0,95	0,93	0,89	0,93
Gradient Boosting	0,89	0,87	0,80	0,74
LASSO	0,95	0,70	0,80	0,91

A continuación, se presenta la matriz de confusión correspondiente al modelo predictivo de Red Neuronal Artificial, utilizando el conjunto de características seleccionadas mediante el método Random Forest.

Matriz de confusión Red Neuronal Artificial

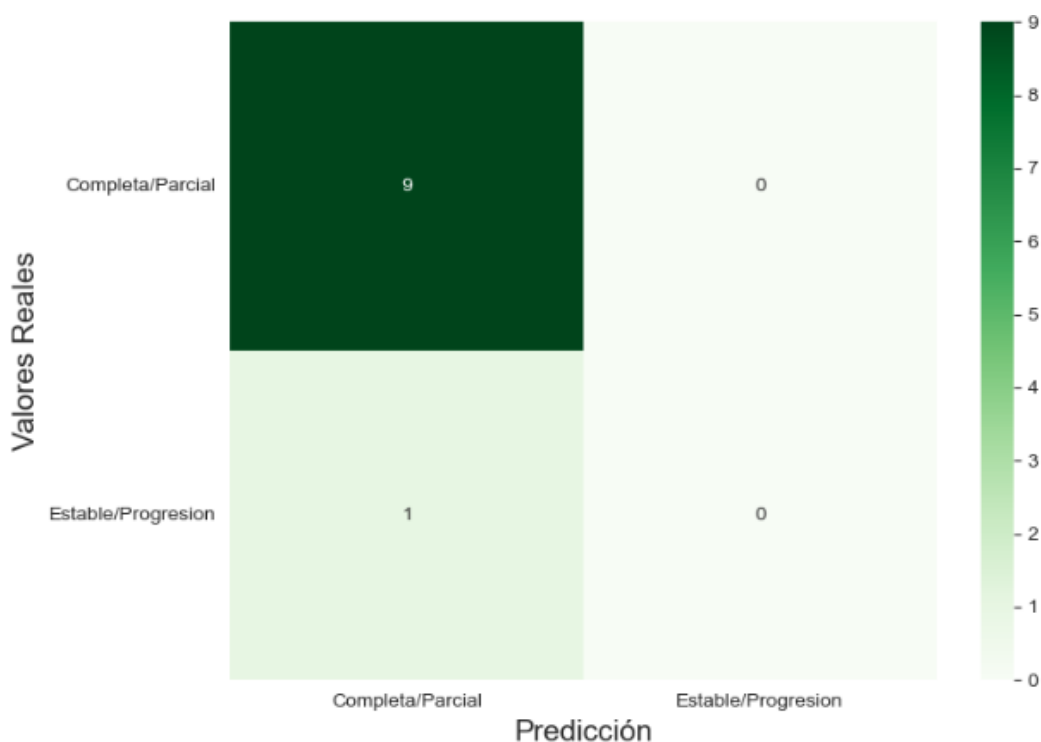


Figura 24: Matriz de confusión del modelo predictivo de Red Neuronal Artificial utilizando el conjunto de características seleccionadas mediante el método Random Forest. En la matriz de confusión, el eje Y representa las clasificaciones reales de

respuestas, mientras que el eje X muestra las predicciones correspondientes realizadas por el modelo.

La matriz de confusión revela que el modelo de red neuronal, entrenado con el conjunto de datos del método Random Forest, ha logrado acertar en 9 casos de la primera clase (Completa/Parcial), evidenciando una capacidad relativamente fuerte para identificar esta categoría. Sin embargo, llama la atención que, en 1 caso, originalmente clasificado como Estable/Progresión, el modelo lo haya asignado incorrectamente a la categoría de Completa/Parcial.

Este desajuste puede atribuirse al desequilibrio de clases, donde la predominancia de ejemplos de una clase puede influir en la predicción del modelo, favoreciendo la categoría mayoritaria.

En la siguiente tabla, se presentan las métricas de evaluación del modelo de Red Neuronal Artificial entrenado con el conjunto de características seleccionado mediante el método de Random Forest. Estas métricas ofrecen una visión detallada del rendimiento del modelo en la tarea de clasificación de respuestas postneoadyuvantes en metástasis de recto. Cada valor proporciona información valiosa sobre la capacidad predictiva y la precisión del modelo en diferentes aspectos de la clasificación, brindando una comprensión completa de su desempeño.

Tabla 2. Métricas de evaluación del modelo de Red Neuronal Artificial entrenado con el conjunto de características seleccionados mediante el método de Random Forest.

	Accuracy	Precision	Recall	F1 score
Red Neuronal Artificial	0,90	0,95	0,50	0,85

3.2.1.2 Segunda aplicación

En esta segunda sección, extendemos el análisis presentado anteriormente y ahora exploramos las curvas Receiver Operating Characteristic (ROC) para los conjuntos de características seleccionadas en la segunda aplicación. Al igual que en la primera sección, se han generado curvas ROC-AUC para cada clase individual y sus medias, permitiéndonos examinar la capacidad discriminativa de los modelos predictivos en el contexto específico de esta aplicación.

- **Método de selección: Random Forest.**

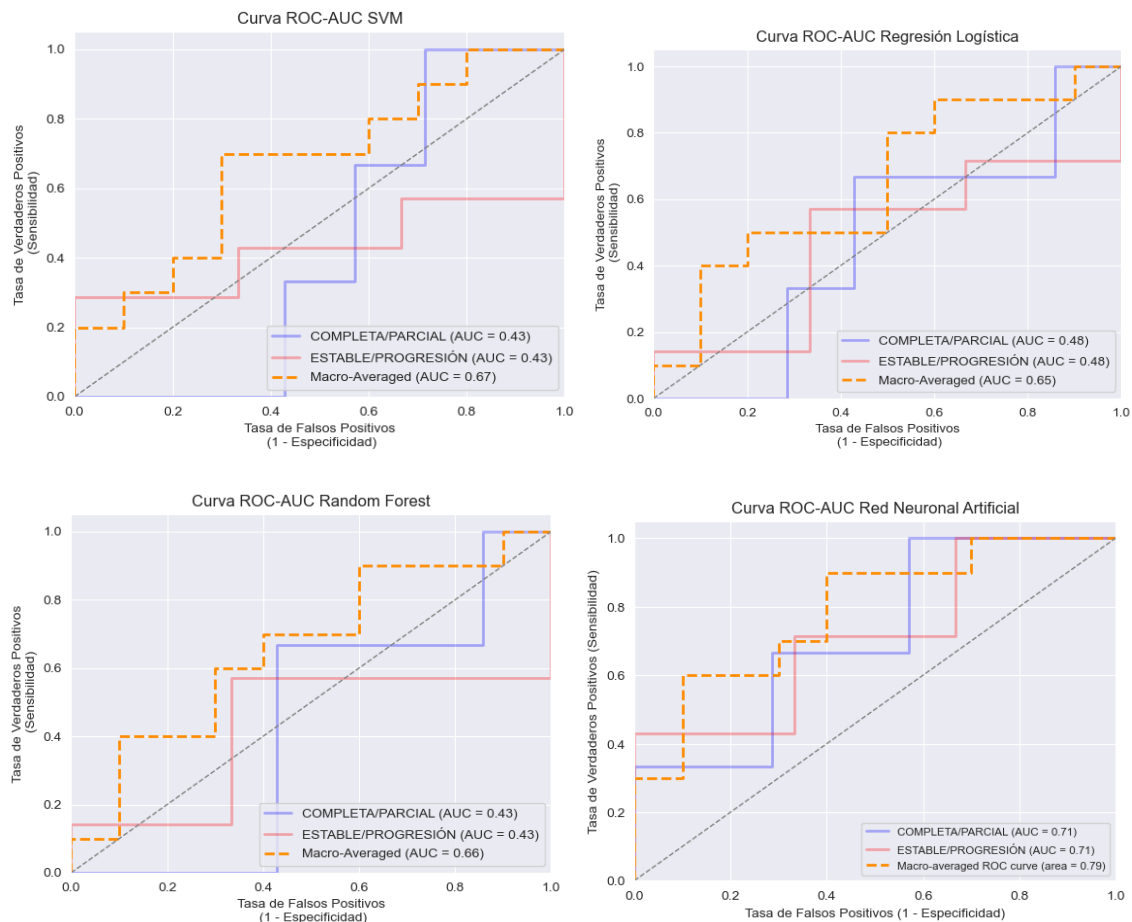


Figura 25: Curvas ROC-AUC de cada clase y media entrenadas por los modelos Random Forest, Regresión Logística, SVM y Red Neuronal Artificial con el conjunto de características seleccionadas mediante el método Random Forest.

- **SVM:** Mostró un AUC de 0.43 para ambas clases y una AUC media de 0.67, indicando una capacidad moderada de discriminación entre las categorías.
 - **Regresión Logística:** Presentó AUC de 0.48 para ambas clases y una AUC media de 0.65, mostrando una capacidad limitada para discriminar entre las respuestas "Completa" y "Parcial/Estable/Progresión".
 - **Random Forest:** Mostró AUC de 0.43 para ambas clases y una AUC media de 0.66, señalando una capacidad moderada de discriminación en este contexto.
 - **Red Neuronal Artificial:** Destacó con AUC de 0.71 para ambas clases y una AUC media de 0.79, indicando una sólida capacidad para distinguir entre "Completa" y "Parcial/Estable/Progresión".
- **Método de selección: Recursive Feature Elimination (RFE).**

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

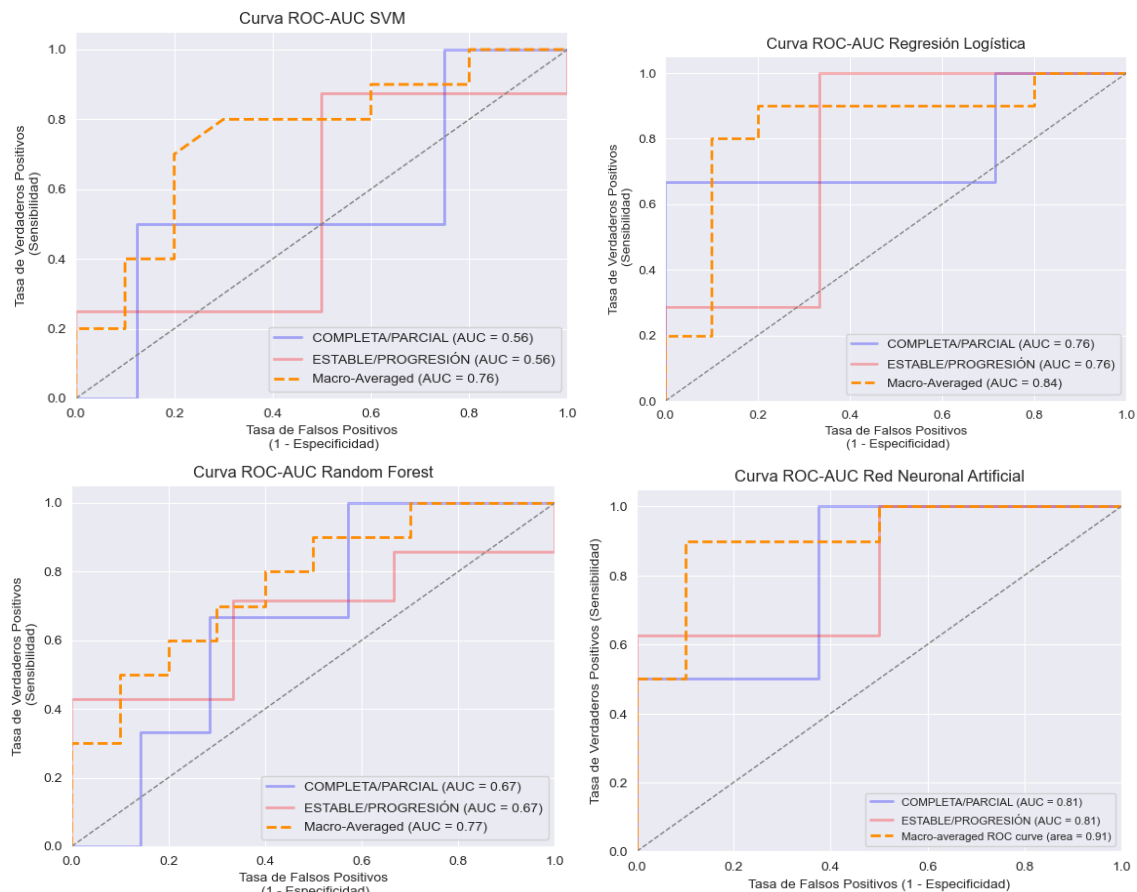


Figura 26: Curvas ROC-AUC de cada clase y media entrenadas por los modelos Random Forest, Regresión Logística, SVM y Red Neuronal Artificial con el conjunto de características seleccionadas mediante el método Recursive Feature Elimination (RFE).

- **SVM:** Registró AUC de 0.56 para ambas clases y una AUC media de 0.76, indicando una mejora en la capacidad de discriminación.
 - **Regresión Logística:** Exhibió fuerte mejora con AUC de 0.76 para ambas clases y una AUC media de 0.84, destacando un rendimiento más robusto en la discriminación de las respuestas.
 - **Random Forest:** Mantuvo un rendimiento consistente con AUC de 0.67 para ambas clases y una AUC media de 0.77, evidenciando estabilidad en la capacidad de clasificación.
 - **Red Neuronal Artificial:** Sobresalió con AUC de 0.81 para ambas clases y una AUC media de 0.91, indicando una notable mejora en la capacidad de distinguir entre "Completa" y "Parcial/Estable/Progresión".
- **Método de selección: Recursive Gradient Boosting.**

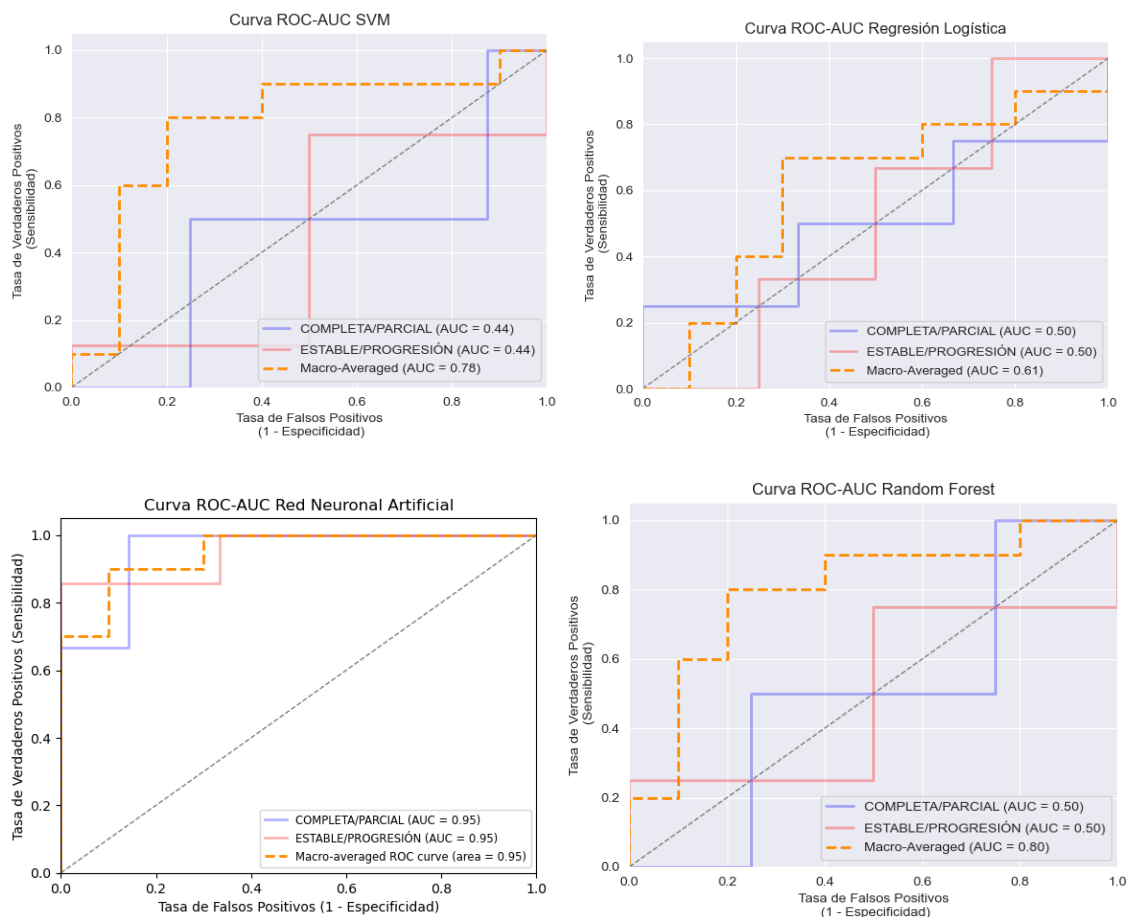


Figura 27. Curvas ROC-AUC de cada clase y media entrenadas por los modelos Random Forest, Regresión Logística, SVM y Red Neuronal Artificial con el conjunto de características seleccionadas mediante el método Gradient Boosting.

- **SVM:** Presentó AUC de 0.44 para ambas clases y una AUC media de 0.78, indicando un rendimiento moderado en la capacidad de discriminación.
- **Regresión Logística:** Alcanzó AUC de 0.50 para ambas clases y una AUC media de 0.61, mostrando una capacidad limitada en la discriminación de respuestas.
- **Random Forest:** Mantuvo un rendimiento constante con AUC de 0.50 para ambas clases y una AUC media de 0.80, evidenciando estabilidad en la capacidad de clasificación.
- **Red Neuronal Artificial:** Destacó con AUC de 0.95 para ambas clases y una AUC media de 0.95, indicando una capacidad excepcional para distinguir entre "Completa" y "Parcial/Estable/Progresión".

- **Método de selección: LASSO.**

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

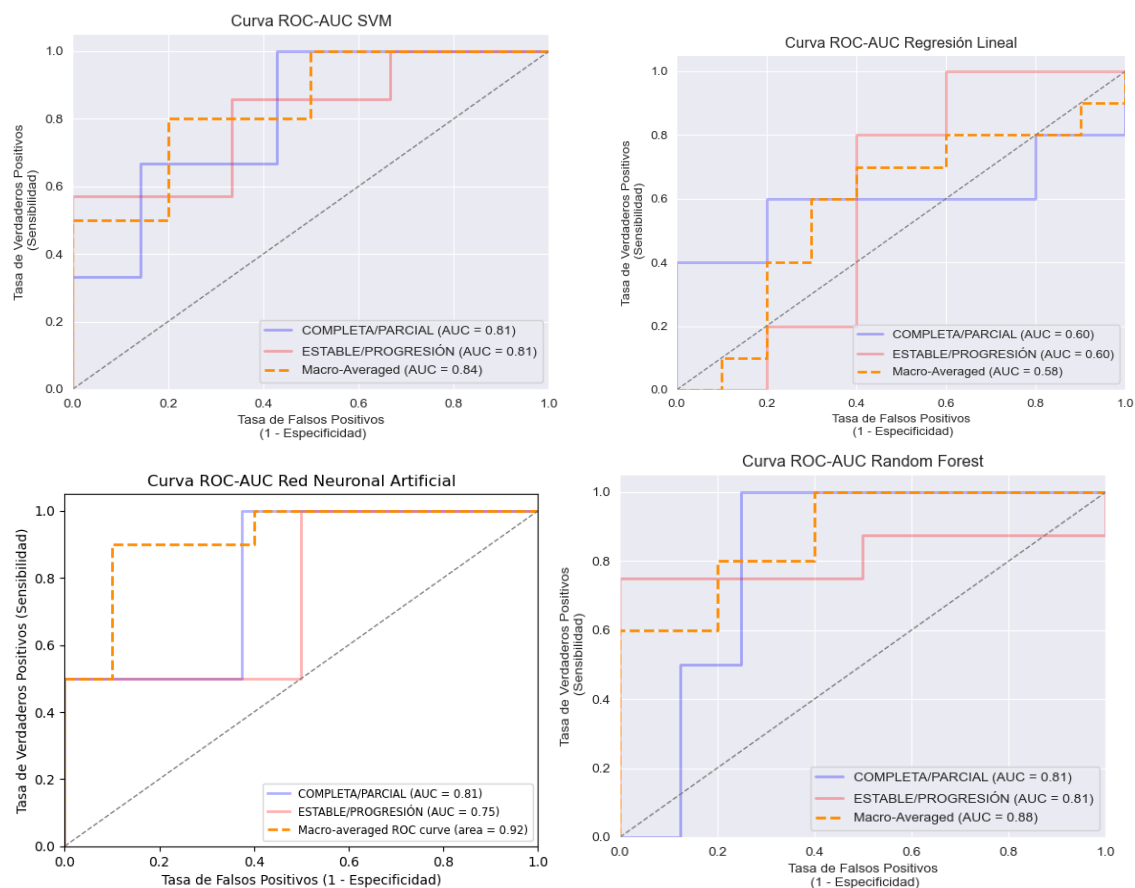


Figura 28: Curvas ROC-AUC de cada clase y media entrenadas por los modelos Random Forest, Regresión Logística, SVM y Red Neuronal Artificial con el conjunto de características seleccionadas mediante el método LASSO.

- **SVM:** Mostró AUC de 0.81 para ambas clases y una AUC media de 0.84, indicando un sólido rendimiento en la capacidad de discriminación.
- **Regresión Lineal:** Alcanzó AUC de 0.60 para ambas clases y una AUC media de 0.58, mostrando una capacidad limitada en la discriminación de respuestas.
- **Random Forest:** Destacó con AUC de 0.81 para ambas clases y una AUC media de 0.88, evidenciando una capacidad robusta y consistente en la clasificación.
- **Red Neuronal Artificial:** Mostró AUC de 0.81 para la primera clase, 0.75 para la segunda clase y una AUC media de 0.92, destacando por su capacidad para distinguir entre "Completa" y "Parcial/Estable/Progresión".

En la siguiente tabla se resumen los valores del Área bajo la Curva (AUC) obtenidos de las curvas Receiver Operating Characteristic (ROC) para cada modelo predictivo, considerando los diferentes métodos de selección de características radiómicas utilizados en ambas aplicaciones.

Tabla 3. Valores AUC de los 4 modelos de predicción en los 4 conjuntos de datos.

ROC - AUC	Red Neuronal Artificial	SVM	Random Forest	Regresión Logística
Random Forest	0,79	0,67	0,66	0,65
RFE	0,91	0,76	0,77	0,84
Gradient Boosting	0,95	0,78	0,80	0,61
LASSO	0,92	0,84	0,88	0,58

A continuación, se presenta la matriz de confusión correspondiente al modelo predictivo de Red Neuronal Artificial, utilizando el conjunto de características seleccionadas mediante el método Gradient Boosting, siendo el modelo con el valor AUC más elevado.

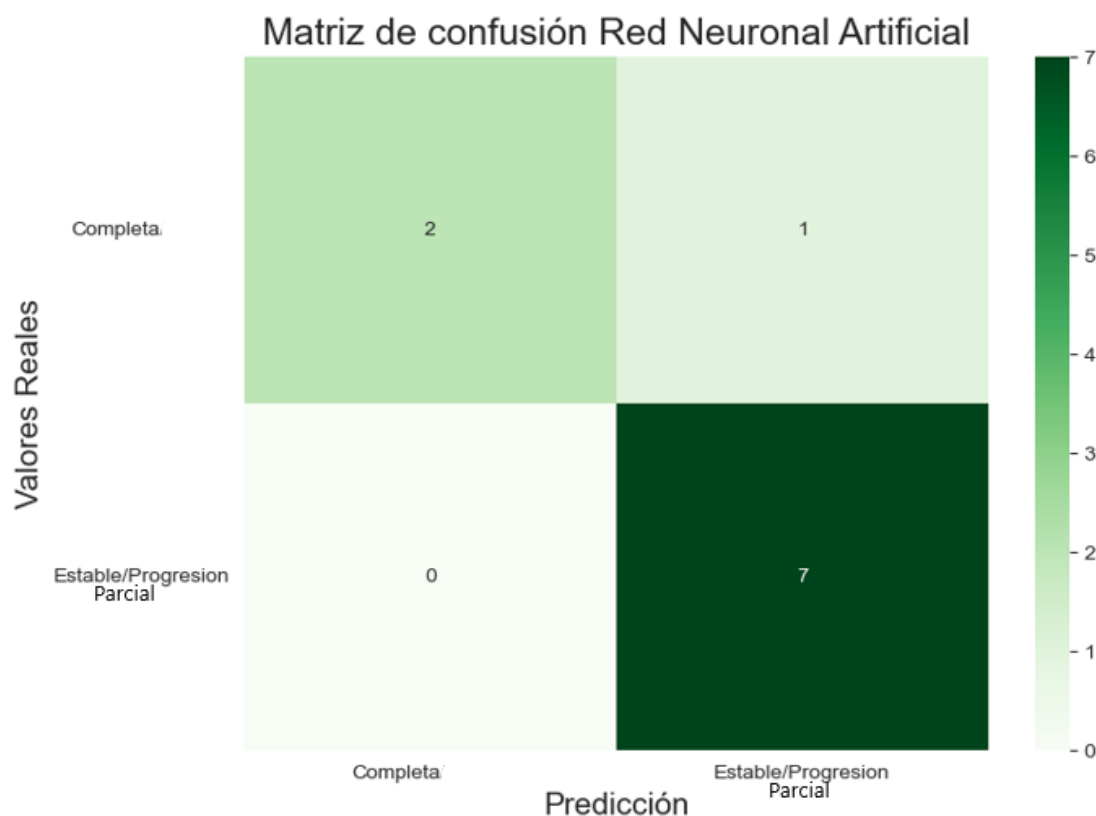


Figura 29: Matriz de confusión del modelo predictivo de Red Neuronal Artificial utilizando el conjunto de características seleccionadas mediante el método Gradient Boosting.. En la matriz de confusión, el eje Y representa las clasificaciones reales de respuestas, mientras que el eje X muestra las predicciones correspondientes realizadas por el modelo.

En la evaluación del modelo utilizando la red neuronal artificial con el conjunto de datos Gradient Boosting, se observa un rendimiento notable. Se lograron correctamente 2 casos de respuesta "Completa" y 7 casos de respuesta "Parcial/Estable/Progresión". Sin

embargo, se identificó un caso mal clasificado, en el cual una respuesta que debería haber sido etiquetada como "Completa" se clasificó erróneamente como "Parcial/Estable/Progresión". Estos resultados respaldan, en su mayoría, la efectividad del modelo en la clasificación precisa de las respuestas deseadas en el contexto del conjunto de datos Gradient Boosting.

En la siguiente tabla, se presentan las métricas de evaluación del modelo de Red Neuronal Artificial entrenado con el conjunto de características seleccionado mediante el método de Random Forest. Estas métricas ofrecen una visión detallada del rendimiento del modelo en la tarea de clasificación de respuestas postneoadyuvantes en metástasis de recto. Cada valor proporciona información valiosa sobre la capacidad predictiva y la precisión del modelo en diferentes aspectos de la clasificación, brindando una comprensión completa de su desempeño.

Tabla 4. Métricas de evaluación del modelo de Red Neuronal Artificial entrenado con el conjunto de características seleccionados mediante el método de Random Forest.

	Accuracy	Precision	Recall	F1 score
Red Neuronal Artificial	0,89	0,88	0,93	0,89

4 Discusión

En la presente discusión, se abordan los resultados derivados de las características seleccionadas, así como el rendimiento individual de cada modelo en la aplicación. Este análisis se dedica a examinar con detalle cómo los modelos han demostrado su capacidad predictiva, con una evaluación de las posibles aplicaciones prácticas en el entorno clínico.

En la evaluación de las características radiómicas seleccionadas por los diferentes métodos, se observa una consistencia notable en la presencia de ciertos tipos de características en múltiples conjuntos. Específicamente, las *First Order Features*, así como las características relacionadas con la textura, representadas por las clases *GLCM*, *GLDM* y *GLRLM*, emergen como las más repetidas en los conjuntos seleccionados por Random Forest (RF), Gradient Boosting (GB), Recursive Feature Elimination (RFE) y LASSO.

La repetición de estas categorías resalta la importancia de considerar tanto las características estadísticas fundamentales (*First Order Features*) como las que describen la textura y las propiedades espaciales de la imagen radiómica (*GLCM*, *GLDM*, *GLRLM*) en la evaluación de la respuesta postneoadyuvante en cáncer de recto. La presencia recurrente de estas características sugiere su papel crucial en la capacidad de los modelos para predecir patrones relevantes y, por ende, para mejorar la toma de decisiones clínicas.

En particular, las *First Order Features* proporcionan información fundamental sobre la distribución de intensidades y la variabilidad de niveles de gris en la imagen, siendo esenciales para entender la naturaleza del tejido tumoral. Por otro lado, las características de la matriz de co-ocurrencia de niveles de gris (*GLCM*) ofrecen detalles sobre la relación

espacial de las intensidades de píxeles, mientras que las características de la matriz de diferencia de niveles de gris (*GLDM*) y de la matriz de longitud de carrera de niveles de gris (*GLRLM*) complementan la información de textura al describir la variación en las intensidades de niveles de gris y las corridas de píxeles, respectivamente.

En la primera aplicación abordada en este proyecto, el propósito fundamental fue anticipar la respuesta al tratamiento neoadyuvante y distinguir entre dos categorías específicas. Se procedió a consolidar las respuestas completas y parciales en una clase, y las respuestas estables y progresivas en otra. De esta manera, el objetivo principal consiste en predecir si el paciente experimentará una respuesta positiva o no al tratamiento administrado. En el caso de predecir la clase II, se podría adaptar su tratamiento para evitar una toxicidad innecesaria, dado que no responde favorablemente al tratamiento.

Comenzando con la Red Neuronal Artificial (RNA), los resultados indican un buen rendimiento cuando se utiliza el conjunto seleccionado mediante Random Forest, con un valor de Curva ROC-AUC de 0,97. Esta métrica destaca la capacidad del modelo para capturar tanto la precisión como el equilibrio entre verdaderos positivos y verdaderos negativos. Sin embargo, una exploración más profunda revela una discrepancia en la Recall, con un valor de 0,50, sugiriendo una posible dificultad del modelo para identificar todos los casos positivos. Este hallazgo es especialmente crítico en el contexto clínico, donde la identificación precisa de respuestas postneoadyuvancia es esencial.

La precisión del 0,95 y el F1 Score de 0,85 subrayan la capacidad de la RNA para predecir correctamente la clase positiva, aunque con ciertas limitaciones. Esta dualidad en las métricas resalta la importancia de no depender exclusivamente de la precisión, ya que un modelo puede ser altamente preciso, pero pasar por alto casos cruciales. Es importante considerar el equilibrio entre precisión y recall para una evaluación más completa del rendimiento.

Al observar los conjuntos seleccionados mediante RFE, Gradient Boosting y Lasso, se evidencian variaciones en el rendimiento de la RNA. Estos resultados subrayan la influencia significativa de las características utilizadas en la capacidad predictiva del modelo. La consistencia en el rendimiento entre Random Forest y Lasso destaca la robustez de ciertas características radiómicas para la tarea de predicción.

Por otro lado, las Máquinas de Soporte Vectorial (SVM) exhiben variaciones considerables en la eficacia según el conjunto de características. La sensibilidad a la selección mediante RFE y la disminución del rendimiento con Lasso y Random Forest indican la necesidad de una cuidadosa consideración al elegir el conjunto de características para SVM. La robustez demostrada por Random Forest y Regresión Logística en general sugiere que estos modelos pueden ser opciones viables y consistentes en diversas configuraciones.

La interpretación de estos resultados tiene implicaciones clínicas significativas. La capacidad de predecir con alta precisión entre categorías como "Completa/Parcial" y "Estable/Progresiva" podría tener un impacto sustancial en la toma de decisiones clínicas.

Sin embargo, la variabilidad en el rendimiento entre modelos y conjuntos de características destaca la necesidad de una evaluación cuidadosa antes de la implementación clínica, reforzando la importancia de la interpretación y aplicación contextualizada de los resultados del modelo.

La importancia crítica de la selección de características se destaca aún más en la variabilidad del rendimiento. Las características radiómicas seleccionadas desempeñan un papel muy importante en la capacidad del modelo para generalizar y predecir con precisión la respuesta postneoadyuvancia. La consistencia en el rendimiento de ciertos conjuntos de características resalta la robustez de esas características específicas para la tarea de predicción.

En la fase de evaluación de la segunda aplicación, se realizó otro análisis con los mismos modelos de aprendizaje automático con el propósito de predecir respuestas completas y, por ende, evitar intervenciones quirúrgicas innecesarias en pacientes. Los resultados de cada modelo para conjuntos específicos de características son los siguientes:

En términos de Red Neuronal Artificial (RNA), se observó un rendimiento sólido en general. Se destacó un aumento notable en la precisión al emplear conjuntos seleccionados mediante Gradient Boosting (0,95) y Recursive Feature Elimination (RFE) (0,91). Estos resultados sugieren que las características identificadas por estos métodos son especialmente relevantes para la predicción de respuestas completas.

Las Máquinas de Soporte Vectorial (SVM) demostraron un desempeño aceptable, mostrándose más eficaces cuando se seleccionaron características mediante Lasso (0,84). Esto indica que la regularización Lasso puede ser crucial para la selección de características en SVM en este contexto específico.

El modelo Random Forest destacó con un rendimiento consistente y elevado en todos los conjuntos de características seleccionadas. Específicamente, alcanzó un AUC de 0,88 cuando se utilizaron características seleccionadas mediante Lasso, demostrando su capacidad robusta para diferentes conjuntos de características.

En cuanto a la Regresión Logística, se observó un rendimiento variable, siendo más efectiva cuando se seleccionaron características mediante RFE (0,84). No obstante, su desempeño fue más limitado en comparación con otros modelos, subrayando la importancia de una selección cuidadosa de características.

Desde una perspectiva clínica, estos resultados tienen implicaciones significativas al ofrecer una herramienta potencial para identificar pacientes con respuesta completa, evitando cirugías innecesarias y sus posibles complicaciones.

La variabilidad en el rendimiento entre modelos y conjuntos de características destaca la necesidad de una evaluación cuidadosa antes de la implementación clínica. La selección adecuada de características sigue siendo un componente crucial para la precisión del modelo.

No obstante, es esencial destacar algunas limitaciones identificadas en este estudio que podrían influir en la interpretación de los resultados. En primer lugar, la disponibilidad de datos limitados podría haber afectado la robustez de los modelos evaluados. La cantidad insuficiente de datos puede generar desafíos en la generalización de los resultados a una población más amplia.

Adicionalmente, se observó un desequilibrio significativo entre las clases en el conjunto de datos, lo cual podría introducir sesgos en el rendimiento de los modelos. La falta de equilibrio puede afectar la capacidad de los modelos para aprender de manera equitativa de todas las categorías, destacando la necesidad de abordar estratégicamente este desafío para mejorar la fiabilidad de las predicciones en todas las clases.

Otra consideración importante es la ausencia de características clínicas específicas en el conjunto de datos utilizado. La inclusión de información clínica adicional podría haber enriquecido la capacidad predictiva de los modelos al considerar variables más holísticas y contextualmente relevantes. La falta de estas características podría limitar la aplicabilidad clínica de los resultados, ya que factores importantes podrían no haberse tenido en cuenta durante el análisis.

En términos de perspectivas de investigaciones futuras, se sugiere la realización de estudios con conjuntos de datos más extensos y equilibrados para mejorar la validez externa de los modelos. Además, sería beneficioso explorar el impacto de la inclusión de características clínicas adicionales en la predicción de respuestas completas en pacientes con cáncer de recto.

Una dirección prometedora para futuras investigaciones sería abordar la limitación de la clasificación binaria utilizada en este estudio, considerando la posibilidad de llevar a cabo un análisis de cuatro clases para estudiar cada respuesta por separado. Esto permitiría una comprensión más detallada de las predicciones de los modelos en diferentes escenarios, proporcionando información valiosa para la toma de decisiones clínicas.

5 Conclusiones

En este estudio, se ha explorado la aplicación de la radiómica y el aprendizaje automático para predecir las respuestas post quimio-radioterapia (QT/RT) en pacientes con cáncer de recto sometidos a tratamiento neoadyuvante. La consistencia en las características radiómicas, como las First Order Features y las relacionadas con la textura, subraya su importancia en la evaluación de la respuesta al tratamiento.

En la primera aplicación, al anticipar la respuesta al tratamiento, la Red Neuronal Artificial (RNA) destacó con un rendimiento sólido, especialmente utilizando conjuntos seleccionados mediante Random Forest. Sin embargo, se identificó una discrepancia en la Recall, sugiriendo la necesidad de mejorar la identificación de casos positivos, aspecto crucial en el contexto clínico. La variabilidad entre modelos y conjuntos de características subraya la importancia de una selección cuidadosa. Las Máquinas de Soporte Vectorial (SVM) exhibieron variaciones, destacando la necesidad de considerar cuidadosamente el

conjunto de características, mientras que la consistencia entre Random Forest y Lasso resalta la robustez de ciertas características radiómicas.

En la segunda aplicación, al predecir respuestas completas, la Red Neuronal Artificial (RNA) destacó con Gradient Boosting y Recursive Feature Elimination (RFE), mientras que Random Forest demostró consistencia en diferentes conjuntos de características. La variabilidad en el rendimiento destaca la necesidad de evaluaciones cautelosas antes de la implementación clínica, enfatizando la importancia de la selección de características.

Estos hallazgos sugieren que, de lograrse una predicción precisa de la clase "completa" por parte del modelo, los pacientes podrían potencialmente evitar cirugías innecesarias, con aproximadamente un 30% de probabilidad de obtener esa respuesta según los resultados obtenidos. Esta posibilidad destaca el impacto práctico de la aplicación de modelos predictivos en la toma de decisiones clínicas, ofreciendo beneficios significativos tanto para los pacientes como para el sistema de atención médica.

Aunque este estudio presenta resultados prometedores, se identificaron limitaciones, como la disponibilidad limitada de datos y el desequilibrio entre las clases, lo que podría afectar la generalización de los modelos.

En términos de investigaciones futuras, se sugiere explorar conjuntos de datos más extensos y equilibrados, considerar la inclusión de características clínicas adicionales y abordar la limitación de la clasificación binaria. Estas mejoras podrían fortalecer la validez externa y la aplicabilidad clínica de los modelos.

En conclusión, este estudio destaca el potencial de la radiómica y el aprendizaje automático en la predicción de respuestas postneoadyuvantes en pacientes con cáncer de recto. A pesar de las limitaciones, los hallazgos proporcionan una base sólida para futuras investigaciones y sugieren la posibilidad de mejorar la toma de decisiones clínicas en este contexto.

6 Referencias consultadas

- [1] World Health Organization, "Colorectal cancer," Jul. 11, 2023. [En línea]. Disponible en: <https://www.who.int/es/news-room/fact-sheets/detail/colorectal-cancer>.
- [2] C. O. Decanini-Terán et al., "Cáncer de recto. Diagnóstico, estadificación y tratamiento," *Cir Cir*, vol. 79, no. 5, pp. 481-487, sep.-oct. 2011. Disponible en: <https://www.medigraphic.com/pdfs/circir/cc-2011/cc115o.pdf>.
- [3] "Cancer today". Global Cancer Observatory. [En línea]. Disponible: <https://www.iarc.who.int/>
- [4] López-Köstner, F., Carrillo G., K., Zárata C, A., Brien S, A., & Ladrón de Guevara H., D. (2012). Cáncer de recto: diagnóstico, estudio y estadificación. *Revista Chilena De Cirugia*, 64, 199-209.

- [5] P. H. Colin, "Concomitant chemotherapy and radiotherapy: theoretical basis and clinical experience," *Anticancer Research*, vol. 14, no. 6A, pp. 2357-2361, 1994.
- [6] G. Bannura C., "Terapia neoadyuvante total (TNT) en el manejo del cáncer de recto localmente avanzado," *Rev. cir.*, vol. 74, no. 5, Oct. 2022. DOI: 10.35687/s2452-454920220051572.
- [7] "American joint committee on cancer". ACS. [En línea]. Disponible: <https://www.facs.org/quality-programs/cancer-programs/american-joint-committee-on-cancer/>
- [8] M. Carles Fariña et al., "Modelo radiómico con PSMA-PET para la discriminación de pacientes con cáncer de próstata de alto riesgo," *Revista de Física Médica*, 2022.
- [9] Dubarrán et al., "Resultados oncológicos del manejo neoadyuvante con quimiorradioterapia seguido de cirugía en cáncer de recto localmente avanzado: en el Instituto Oncológico Nacional de Panamá," p. 2, 2023.
- [10] P. Priego et al., "Abordaje multidisciplinar en el tratamiento del cáncer de recto: eficacia del tratamiento neoadyuvante," *Revista Espanola De Enfermedades Digestivas*, vol. 100, pp. 393-399, 2008.
- [11] Lanzarin-Minero et al., "Predictores radiómicos F18-FDG PET/CT en la respuesta patológica completa a la quimioterapia neoadyuvante en pacientes con cáncer de mama," 2022. [Online]. Available: <https://www.semanticscholar.org/paper/afc4258a20e7b4daf4a9c0bf8480e6a23bcfff5c>.
- [12] M. Carles Fariña et al., "Modelo radiómico con PSMA-PET para la discriminación de pacientes con cáncer de próstata de alto riesgo," *Revista de Física Médica*, 2022.
- [13] Dubarrán et al., "Resultados oncológicos del manejo neoadyuvante con quimiorradioterapia seguido de cirugía en cáncer de recto localmente avanzado: en el Instituto Oncológico Nacional de Panamá," p. 2, 2023.
- [14] P. M. Cárcamo Ibarra et al., "Progress and current utility of radiomics in PET/CT study of non-metastatic breast cancer: A systematic review," *Rev Esp Med Nucl Imagen Mol (Engl Ed)*, vol. 42, no. 2, pp. 83-92, Mar-Apr 2023. doi: 10.1016/j.remnie.2022.11.001. Epub 2022 Nov 11. PMID: 36375751.
- [15] M. E. Mayerhoefer et al., "Introduction to Radiomics," *Journal of Nuclear Medicine*, vol. 61, no. 4, pp. 488-495, 2020, doi: 10.2967/jnumed.118.222893.
- [16] L. Pino, J. Galvis, M. Hidalgo, M. Maldonado, I. Triana, y C. Giedelmann, «Determinación de variables predictoras para tratamiento curativo con radioterapia en pacientes con adenocarcinoma de próstata en una cohorte del mundo real utilizando modelos de aprendizaje de máquina», *Rev.Col.Hematol.Oncol*, vol. 9, n.º Supl, pp. 195–197, jun. 2023.

- [17] S. Zhang et al., "Role of MRI-based radiomics in locally advanced rectal cancer (Review)," *Oncol Rep*, vol. 47, no. 2, p. 34, Feb. 2022, doi: 10.3892/or.2021.8245. Epub 2021 Dec 22. PMID: 34935061; PMCID: PMC8717123.
- [18] L. Shi et al., "Radiomics for Response and Outcome Assessment for Non-Small Cell Lung Cancer," *Technology in Cancer Research & Treatment*, vol. 17, pp. 1533033818782788, 2018, doi: 10.1177/1533033818782788.
- [19] P. M. Cárcamo Ibarra et al., "Progress and current utility of radiomics in PET/CT study of non-metastatic breast cancer: A systematic review," *Revista española de medicina nuclear e imagen molecular*, vol. 42, no. 2, pp. 83-92, 2023, doi: 10.1016/j.remnie.2022.11.001.
- [20] Dubarrán et al., "Resultados oncológicos del manejo neoadyuvante con quimiorradioterapia seguido de cirugía en cáncer de recto localmente avanzado: en el Instituto Oncológico Nacional de Panamá," p. 2, 2023.
- [21] N. Tirada et al., "Breast Cancer Tissue Markers, Genomic Profiling, and Other Prognostic Factors: A Primer for Radiologists," *Radiographics*, vol. 38, no. 7, pp. 1902-1920, 2018, doi: 10.1148/rg.2018180047.
- [22] Siemens Healthineers, "DICOM Standards," Siemens Healthineers, [Online]. Available: <https://www.siemens-healthineers.com/es/services/it-standards/dicom>. [Accessed: December 2023].
- [23] P. Lohmann, K. Bousabarah, M. Hoevels, and H. Treuer, "Radiomics in radiation oncology-basics, methods, and limitations," *Strahlenther Onkol*, vol. 196, no. 10, pp. 848-855, Oct. 2020. doi: 10.1007/s00066-020-01663-3.
- [24] SAS. "Estandarizar datos: Establecer opciones de estandarización," *SAS® Enterprise Guide®: Help Center*, [Online]. Available: <https://documentation.sas.com/doc/es/egdoccdc/8.1/egamotasks/n07jr5iuhmgfcun1dfq1udjressc.htm>. [Accessed: January 2024].
- [25] M. Frómeta Torres, "Método no supervisado para la selección de rasgos en problemas de regionalización," *La Habana*, agosto del 2020.
- [26] M. Sainin y R. Alfred, "A Genetic Based Wrapper Feature Selection Approach Using Nearest Neighbour Distance Matrix," 2011/07/20. DOI: 10.1109/DMO.2011.5976534.
- [27] M. Schonlau y R. Y. Zou, "The random forest algorithm for statistical learning," *The Stata Journal*, vol. 20, no. 1, pp. 3-29, 2020. DOI: 10.1177/1536867X20909688.
- [28] J. Z. Musoro et al., "Validation of prediction models based on lasso regression with multiply imputed data," *BMC Medical Research Methodology*, vol. 14, no. 1, p. 116, Oct. 2014. DOI: 10.1186/1471-2288-14-116.

[29] MathWorks. "Machine Learning - MathWorks," MathWorks, [En línea]. Disponible: <https://es.mathworks.com/discovery/machine-learning.html>. [Accedido: diciembre 2023].

[30] S. Liu y N. Jiang, "SVM parameters optimization algorithm and its application," en 2008 IEEE International Conference on Mechatronics and Automation, Takamatsu, 2008, pp. 509-513, doi: 10.1109/ICMA.2008.4798808.

[31] "Classification Model: SVM Classifier in Python with Example," Vitalflux, [En línea]. Disponible: <https://vitalflux.com/classification-model-svm-classifier-python-example/>. [Accedido: enero 2023].

[32] L. Alzubaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," Journal of Big Data, vol. 8, no. 1, p. 53, marzo 2021. DOI: 10.1186/s40537-021-00444-8.

[33] "Artificial Neural Networks and its Applications," GeeksforGeeks, [En línea]. Disponible: <https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/>. [Accedido: diciembre 2023].

[34] L. Jin et al., "Study on the Overfitting of the Artificial Neural Network Forecasting Model," Acta Meteorologica Sinica, vol. 19. [Recibido: 27 de septiembre de 2004; revisado: 6 de diciembre de 2004].

[35] W. Barrios, "Calculando la precisión en un modelo de clasificación multiclase," Medium, [En línea]. Disponible: <https://wbarriosb.medium.com/calculando-la-precisi%C3%B3n-en-un-modelo-de-clasificaci%C3%B3n-multiclase-224d96f52043>

7 Anexo: Código fuente

7.1 Selección de características

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import KFold
from sklearn.preprocessing import StandardScaler
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import GridSearchCV

datos = pd.read_excel('Base_Datos.xlsx')
datos = datos.iloc[:, 1:] #eliminar la columna ID

# Crear un mapeo para la columna Respuesta asignando números del 0 al 3 a las categorías Completa, Parcial, Estable y Progresion
datos["Respuesta"] = datos["Respuesta"].map({
    "Completa": 0,
    "Parcial": 1,
    "Estable": 2,
    "Progresion": 3
})
```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```
# Seleccionar todas las columnas excepto la última para las características (features)
features = datos.iloc[:, :-1]

# Seleccionar la última columna para la variable objetivo (target)
target = datos.iloc[:, -1]
```

```
# Escalar las características utilizando StandardScaler

# Importar la clase StandardScaler de scikit-Learn
from sklearn.preprocessing import StandardScaler

# Utilizar StandardScaler para escalar las características
scaled_features = StandardScaler().fit_transform(features)

# Crear un nuevo DataFrame con las características escaladas y mantener los nombres de las columnas
features = pd.DataFrame(scaled_features, columns=features.columns)
```

Random forest

```
##### SELECCIÓN DE CARACTERÍSTICAS
### IMPORTANCIA DE CARACTERÍSTICAS
# MODELOS DE APRENDIZAJE AUTOMÁTICO -- RANDOM FOREST

# Guardar las variables features y target en X e y
X = features
y = target

# Importar la biblioteca RandomForestClassifier de scikit-Learn
from sklearn.ensemble import RandomForestClassifier

# Crear un modelo RandomForestClassifier
modelo = RandomForestClassifier()

# Entrenar el modelo con los datos de entrada (X) y salida (y)
modelo.fit(X, y)

# Obtener las importancias de las características del modelo entrenado
importancias = modelo.feature_importances_

# Visualizar la importancia de cada característica en un gráfico de barras
plt.figure(figsize=(10, 6))
plt.bar(range(len(importancias)), importancias, tick_label=X.columns)
plt.xticks(rotation=90)
plt.title('Importancia de Características con Bosques Aleatorios')
plt.show()
```

```
# Importar la biblioteca RandomForestClassifier de scikit-Learn
from sklearn.ensemble import RandomForestClassifier

# Crear un modelo RandomForestClassifier
modelo = RandomForestClassifier()

# Entrenar el modelo con las características X y la variable y
modelo.fit(X, y)

# Obtener las importancias de las características del modelo entrenado
importancias = modelo.feature_importances_

# Establecer un umbral para la selección de características
umbral = 0.015

# Seleccionar características cuya importancia supere el umbral
caracteristicas_seleccionadas = X.columns[importancias > umbral]
importancias_seleccionadas = importancias[importancias > umbral]

# Visualizar solo las características seleccionadas en un gráfico de barras
plt.figure(figsize=(10, 6))
plt.bar(range(len(importancias_seleccionadas)), importancias_seleccionadas, tick_label=caracteristicas_seleccionadas)
plt.xticks(rotation=90)
plt.title('Importancia de Características con Bosques Aleatorios (Umbral: {:.2f})'.format(umbral))
plt.show()
```

```
# Seleccionar las características seleccionadas del DataFrame de características
df_seleccionado = features[caracteristicas_seleccionadas].copy()

# Agregar la columna de respuesta al DataFrame
df_seleccionado['Respuesta'] = target

# Guardar el DataFrame en un archivo Excel
df_seleccionado.to_excel('features_RandomForest.xlsx', index=False)
```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

Gradient Boosting

```
##### SELECCIÓN DE CARACTERÍSTICAS
### IMPORTANCIA DE CARACTERÍSTICAS
# MODELOS DE APRENDIZAJE AUTOMÁTICO -- Gradient Boosting

# Importar la biblioteca XGBoost
import xgboost as xgb

# Definir las variables de entrada (features) y salida (target)
X = features
y = target

# Crear un modelo XGBoost Classifier
modelo_xgboost = xgb.XGBClassifier()

# Entrenar el modelo con los datos
modelo_xgboost.fit(X, y)

# Obtener las importancias de las características
importancias_xgboost = modelo_xgboost.feature_importances_

# Crear un DataFrame con las características radiómicas y sus importancias
df_importancias = pd.DataFrame({'Característica': features.columns, 'Importancia': importancias_xgboost})

# Ordenar el DataFrame por importancia en orden descendente
df_importancias = df_importancias.sort_values(by='Importancia', ascending=False)

# Visualizar las características y sus importancias en un gráfico de barras
plt.figure(figsize=(10, 6))
plt.bar(range(len(df_importancias)), df_importancias['Importancia'], tick_label=df_importancias['Característica'])
plt.xticks(rotation=90)
plt.title('Importancia de Características con XGBoost (Gain)')
plt.show()
```

```
import xgboost as xgb
import matplotlib.pyplot as plt

# Inicializar el modelo XGBoost
modelo_xgboost = xgb.XGBClassifier()
modelo_xgboost.fit(X, y)

# Obtener las importancias de las características
importancias_xgboost = modelo_xgboost.feature_importances_

# Establecer un umbral para la importancia
umbral_importancia = 0.025

# Filtrar las características que superan el umbral
caracteristicas_seleccionadas = features.columns[importancias_xgboost > umbral_importancia]
importancias_seleccionadas = importancias_xgboost[importancias_xgboost > umbral_importancia]

# Visualizar solo las características seleccionadas
plt.figure(figsize=(10, 6))
plt.bar(range(len(importancias_seleccionadas)), importancias_seleccionadas, tick_label=caracteristicas_seleccionadas)
plt.xticks(rotation=90)
plt.title('Importancia de Características con XGBoost (Umbral: {:.2f})'.format(umbral_importancia))
plt.show()
```

```
# Crear un nuevo DataFrame con las características seleccionadas
df_seleccionado = features[caracteristicas_seleccionadas].copy()

# Agregar la columna 'Respuesta' al nuevo DataFrame
df_seleccionado['Respuesta'] = target

# Guardar el DataFrame en un archivo Excel sin incluir el índice
df_seleccionado.to_excel('features_GradientBoosting.xlsx', index=False)
```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

Recursive Feature Elimination (RFE)

```
##### SELECCIÓN DE CARACTERÍSTICAS
### IMPORTANCIA DE CARACTERÍSTICAS
# MODELOS DE APRENDIZAJE AUTOMÁTICO -- RFE

from sklearn.svm import SVC
from sklearn.feature_selection import RFE
from sklearn.model_selection import LeaveOneOut
import pandas as pd

# Crear el clasificador SVM
svm_classifier = SVC(kernel="linear", C=1)

# Inicializar LOOCV
loo = LeaveOneOut()

# Inicializar el RFE con el clasificador SVM y el número deseado de características a seleccionar
rfe = RFE(estimator=svm_classifier, n_features_to_select=10)

# Listas para almacenar predicciones y etiquetas reales
predictions_list = []
labels_list = []

# Iterar sobre los pliegues de LOOCV
for train_idx, test_idx in loo.split(features):

    # Seleccionar las características en el conjunto de entrenamiento y prueba
    X_train, X_test = features.iloc[train_idx], features.iloc[test_idx]
    y_train, y_test = target.iloc[train_idx], target.iloc[test_idx]

    # Ajustar el RFE al conjunto de entrenamiento
    rfe.fit(X_train, y_train)

    # Seleccionar las características en el conjunto de entrenamiento y prueba
    X_train_selected = rfe.transform(X_train)
    X_test_selected = rfe.transform(X_test)

    # Realizar predicciones en el conjunto de prueba
    svm_classifier.fit(X_train_selected, y_train)
    preds_selected = svm_classifier.predict(X_test_selected)

    # Guardar predicciones y etiquetas reales
    predictions_list.extend(preds_selected)
    labels_list.extend(y_test)

# Crear DataFrame con las características seleccionadas por RFE
caracteristicas_seleccionadas = features.columns[rfe.support_]
```

```
# Crear un nuevo DataFrame con las características seleccionadas
df_seleccionado = features[caracteristicas_seleccionadas].copy()

# Agregar la columna 'Respuesta' al nuevo DataFrame
df_seleccionado['Respuesta'] = target

# Guardar el DataFrame en un archivo Excel sin incluir el índice
df_seleccionado.to_excel('features_RFE.xlsx', index=False)
```

LASSO (Least Absolute Shrinkage and Selection Operator)

```
##### SELECCIÓN DE CARACTERÍSTICAS
### IMPORTANCIA DE CARACTERÍSTICAS
# MODELOS DE APRENDIZAJE AUTOMÁTICO -- LASSO

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import pandas as pd

# Dividir el conjunto de datos en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

# Inicializar el modelo de regresión logística con L1 regularization (LASSO)
modelo_lasso = LogisticRegression(penalty='l1', solver='liblinear', random_state=42)

# Ajustar el modelo al conjunto de entrenamiento
modelo_lasso.fit(X_train, y_train)

# Evaluar el rendimiento del modelo en el conjunto de prueba
precision = modelo_lasso.score(X_test, y_test)

# Obtener las características seleccionadas por LASSO
features_seleccionadas = features.columns[modelo_lasso.coef_[0] != 0]
features_seleccionadas
```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```
# Crear un DataFrame con Las características seleccionadas
df_caracteristicas_seleccionadas = features[features_seleccionadas].copy()

# Agregar la columna 'Respuesta' al nuevo DataFrame
df_caracteristicas_seleccionadas['Respuesta'] = target

# Guardar el DataFrame en un archivo Excel
nombre_archivo_excel = 'features_LASSO.xlsx'
df_caracteristicas_seleccionadas.to_excel(nombre_archivo_excel, index=False)

print(f"Características seleccionadas guardadas en {nombre_archivo_excel}")
```

Bosques Extra

```
##### SELECCIÓN DE CARACTERÍSTICAS
### IMPORTANCIA DE CARACTERÍSTICAS
# MODELOS DE APRENDIZAJE AUTOMÁTICO -- Bosques Extra (Extra Trees)

# Importar la clase ExtraTreesClassifier y la biblioteca matplotlib.pyplot
from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt

# Inicializar el clasificador de Bosques Extra con una semilla aleatoria para reproducibilidad
modelo_extra_trees = ExtraTreesClassifier(random_state=42)

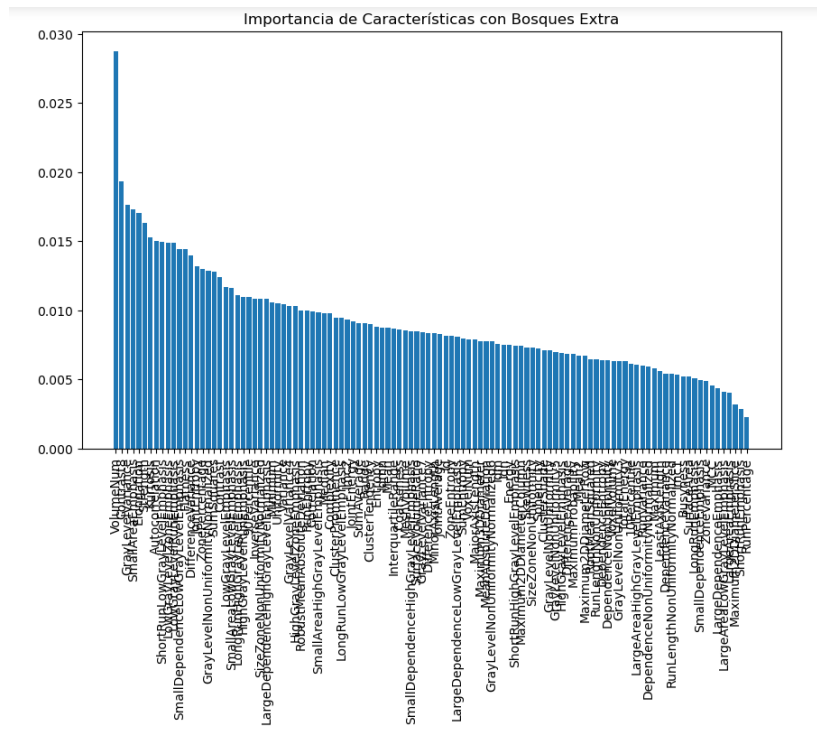
# Ajustar el modelo al conjunto de datos completo
modelo_extra_trees.fit(features, target)

# Obtener la importancia de las características
importancias_caracteristicas = modelo_extra_trees.feature_importances_

# Crear un DataFrame para visualizar las importancias
df_importancias = pd.DataFrame({'Caracteristica': features.columns, 'Importancia': importancias_caracteristicas})

# Ordenar el DataFrame por importancia en orden descendente
df_importancias = df_importancias.sort_values(by='Importancia', ascending=False)

# Visualizar las importancias en un gráfico de barras
plt.figure(figsize=(10, 6))
plt.bar(range(len(df_importancias)), df_importancias['Importancia'], tick_label=df_importancias['Caracteristica'])
plt.xticks(rotation=90)
plt.title('Importancia de Características con Bosques Extra')
plt.show()
```



Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```

from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt

# Inicializar el clasificador de Bosques Extra
modelo_extra_trees = ExtraTreesClassifier(random_state=42)

# Ajustar el modelo al conjunto de datos completo
modelo_extra_trees.fit(features, target)

# Obtener La importancia de Las características
importancias_caracteristicas = modelo_extra_trees.feature_importances_

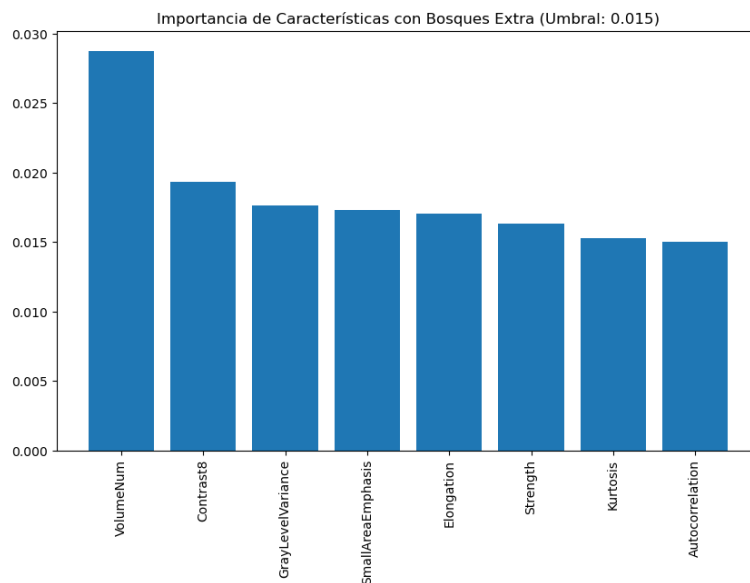
# Establecer un umbral de importancia
umbral_importancia = 0.015

# Filtrar Las características que superan el umbral
caracteristicas_seleccionadas = features.columns[importancias_caracteristicas > umbral_importancia]
importancias_seleccionadas = importancias_caracteristicas[importancias_caracteristicas > umbral_importancia]

# Crear un DataFrame para visualizar Las importancias seleccionadas
df_importancias_seleccionadas = pd.DataFrame({'Caracteristica': caracteristicas_seleccionadas, 'Importancia':
                                             importancias_seleccionadas})
df_importancias_seleccionadas = df_importancias_seleccionadas.sort_values(by='Importancia', ascending=False)

# Visualizar Las importancias seleccionadas
plt.figure(figsize=(10, 6))
plt.bar(range(len(df_importancias_seleccionadas)), df_importancias_seleccionadas['Importancia'],
        tick_label=df_importancias_seleccionadas['Caracteristica'])
|
plt.xticks(rotation=90)
plt.title('Importancia de Características con Bosques Extra (Umbral: {:.3f})'.format(umbral_importancia))
plt.show()

```



```

# Crear un DataFrame con Las características seleccionadas
df_caracteristicas_seleccionadas = features[caracteristicas_seleccionadas].copy()

# Agregar La columna "Respuesta" al nuevo DataFrame
df_caracteristicas_seleccionadas['Respuesta'] = target

# Guardar el DataFrame en un archivo Excel
nombre_archivo_excel = 'features_ExtraTrees.xlsx'
df_caracteristicas_seleccionadas.to_excel(nombre_archivo_excel, index=False)

print(f"Características seleccionadas guardadas en {nombre_archivo_excel}")

```

7.2 Construcción de modelos predictivos

En las dos aplicaciones mencionadas en el estudio, se realiza un mapeo de las clases de respuesta para adaptarse a la configuración específica del problema o el objetivo de la clasificación. Las dos aplicaciones distinguen las clases de respuesta y las agrupan de manera diferente.

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

Primera Aplicación:

En esta aplicación, se realiza un mapeo que agrupa las clases de respuesta de la siguiente manera:

- Clase 0: Completa y Parcial
- Clase 1: Estable y Progresión

Este mapeo está diseñado para distinguir dos grupos principales: uno que incluye las respuestas "Completa" y "Parcial" (Clase 0) y otro que incluye las respuestas "Estable" y "Progresión" (Clase 1).

Segunda Aplicación:

En esta aplicación, se realiza un mapeo diferente que agrupa las clases de respuesta de la siguiente manera:

- Clase 0: Completa
- Clase 1: Parcial, Estable y Progresión

En este caso, la Clase 0 está compuesta solo por la respuesta "Completa", mientras que la Clase 1 incluye las respuestas "Estable", "Parcial" y "Progresión".

7.2.1.1 Random Forest

```
import pandas as pd

# Cargar 'features_RandomForest.xlsx' en un DataFrame llamado features_RF
features_RF = pd.read_excel('features_RandomForest.xlsx')

df = features_RF

# Mapear la columna 'Respuesta':
# para la primera aplicación Completa/Parcial clase 0 y Estable/Progresion clase 1
# para la segunda aplicación Completa clase 0 y Parcial, Estable y Progresion clase 1
# df["Respuesta"] = df["Respuesta"].map({0: 0, 1: 1, 2: 1, 3: 1})

df["Respuesta"] = df["Respuesta"].map({0: 0, 1: 0, 2: 1, 3: 1})

# Importar la función train_test_split de scikit-learn
from sklearn.model_selection import train_test_split

# Separar las características (X) y la variable objetivo (y)
X = df.drop('Respuesta', axis=1) # Características
y = df['Respuesta'] # Variable objetivo

# Dividir los datos en conjuntos de entrenamiento y validación 80% - 20% de manera aleatoria
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

# Imprimir las dimensiones de los conjuntos resultantes
print("X_train shape:", X_train.shape)
print("X_val shape:", X_val.shape)
print("y_train shape:", y_train.shape)
print("y_val shape:", y_val.shape)

X_train shape: (40, 12)
X_val shape: (10, 12)
y_train shape: (40,)
y_val shape: (10,)
```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```
# Resetear Los índices de Los conjuntos de entrenamiento y validación  
features_train = X_train.reset_index(drop=True)  
features_val = X_val.reset_index(drop=True)  
target_train = y_train.reset_index(drop=True)  
target_val = y_val.reset_index(drop=True)
```

```
# Crear un nuevo DataFrame de entrenamiento (df_train) al concatenar características y variable objetivo de entrenamiento  
df_train = pd.concat([features_train, target_train], axis=1)  
  
features = features_train  
target = target_train
```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import KFold
import joblib
import pickle

# Inicializar el modelo RandomForestClassifier con Los hiperparámetros
rf_model = RandomForestClassifier(
    n_estimators=100,
    max_depth=None,
    min_samples_leaf=2,
    min_samples_split=2,
    random_state=42
)

# Configurar la validación cruzada de K-Fold con 5 splits
kfold = KFold(n_splits=5, shuffle=True, random_state=42)

# Lista para almacenar las predicciones de cada iteración de K-Fold
predictions_listRF = []

# Iterar a través de los splits de K-Fold
for train_idx, test_idx in kfold.split(features):

    # Dividir el conjunto de datos en conjuntos de entrenamiento y prueba
    train_features, test_features = features.iloc[train_idx], features.iloc[test_idx]
    train_target, test_target = target.iloc[train_idx], target.iloc[test_idx]

    # Entrenar el modelo RandomForest en el conjunto de entrenamiento
    rf_model.fit(train_features, train_target)

    # Realizar predicciones en el conjunto de prueba
    predsRF = rf_model.predict(test_features)

    # Crear un DataFrame con las respuestas reales y las predicciones de cada conjunto de prueba
    test_dataRF = df_train.loc[test_idx, :].copy()
    test_dataRF['Predictive Response'] = predsRF
    predictions_listRF.append(test_dataRF[['Respuesta', 'Predictive Response']])

# Concatenar todas las predicciones en un solo DataFrame
final_predictions_RF = pd.concat(predictions_listRF, ignore_index=True)

# Guardar el modelo entrenado RandomForest en un archivo pickle
with open('RF_Model_RF.pkl', 'wb') as file:
    pickle.dump(rf_model, file)
```

```
# Cargar el modelo RandomForest previamente guardado
loaded_model = joblib.load('RF_Model_RF.pkl')
```

```
from sklearn.metrics import accuracy_score, f1_score
import numpy as np
import pandas as pd

# Realizar predicciones utilizando el modelo cargado
predsRF = loaded_model.predict(features_val)

# Calcular la precisión del modelo
scoreRF = accuracy_score(target_val, predsRF)

# Crear una lista de 1 - puntuaciones de precisión para el cálculo de la desviación estándar
accuracy_scores = [1 - score for score in predsRF]
std_dev = np.std(accuracy_scores)

# Calcular el F1-score
f1RF = f1_score(target_val, predsRF, average='macro')

# Imprimir las métricas de evaluación
print(f'Accuracy: {scoreRF}')
print(f'Desviación estándar: {std_dev}')
print(f'F1_score: {f1RF}')

# Crear un DataFrame para las predicciones
predictionsRF = pd.DataFrame({
    'Respuesta': target_val,
    'Predictive Response': predsRF
})
```

```
Accuracy: 0.8
Desviación estándar: 0.0
F1_score: 0.4444444444444444
```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Definir un diccionario para mapear las clases
class_names_dict = {
    0: 'Completa/Parcial',
    1: 'Estable/Progresion',
}

# Calcular la matriz de confusión
conf_matrix = confusion_matrix(target_val, predsRF)

# Obtener los nombres de clase ordenados según el diccionario
class_names = [class_names_dict[i] for i in sorted(class_names_dict)]

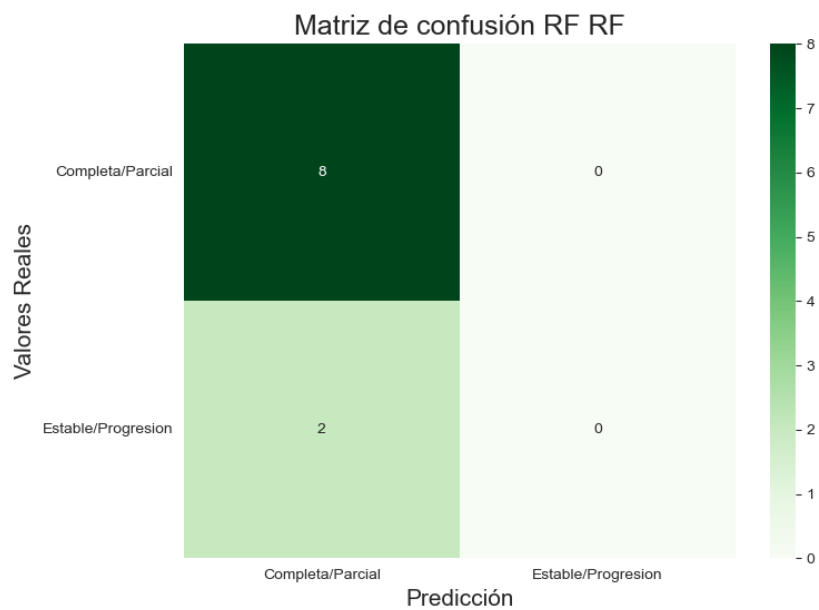
# Establecer el estilo y color de fondo
sns.set_style('darkgrid')
plt.figure(figsize=(8, 6))

# Crear el heatmap
heatmap = sns.heatmap(conf_matrix, annot=True, cmap='Greens', fmt='g')

# Añadir etiquetas a los ejes
heatmap.set_xticklabels(class_names, rotation=0, ha='center')
heatmap.set_yticklabels(class_names, rotation=0, va='center')

# Añadir títulos al gráfico
plt.xlabel('Predicción', fontsize=15)
plt.ylabel('Valores Reales', fontsize=15)
plt.title('Matriz de confusión RandomForest', fontsize=18)

# Mostrar el gráfico
plt.show()
```



Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```

from sklearn.metrics import roc_curve, roc_auc_score, auc
from sklearn.preprocessing import label_binarize
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Predice Las probabilidades
probas = loaded_model.predict_proba(features_val)

# Binarizar Las clases para calcular el puntaje ROC-AUC
y_test_bin = label_binarize(target_val, classes=np.unique(target_val))

# Si solo hay una clase, se apila la clase complementaria (binaria)
if y_test_bin.shape[1] == 1:
    y_test_bin = np.hstack([1 - y_test_bin, y_test_bin])

# Calcular La curva ROC-AUC y el puntaje para cada clase
fpr_rf = {}
tpr_rf = {}
roc_auc_rf = dict()

# Solo dos clases
for i in range(y_test_bin.shape[1]):
    fpr_rf[i], tpr_rf[i], _ = roc_curve(y_test_bin[:, i], probas[:, i])
    roc_auc_rf[i] = auc(fpr_rf[i], tpr_rf[i])

# Plotear Las curvas ROC-AUC para cada clase
plt.plot(fpr_rf[0], tpr_rf[0], color='blue', lw=2, alpha=0.3, label=f'COMPLETA/PARCIAL (AUC = {roc_auc_rf[0]:.2f})')
plt.plot(fpr_rf[1], tpr_rf[1], color='red', lw=2, alpha=0.3, label=f'ESTABLE/PROGRESIÓN (AUC = {roc_auc_rf[1]:.2f})')

# Calcular y plotear La curva ROC-AUC macro-averaged
fpr_rf["macro"], tpr_rf["macro"], thresholds = roc_curve(y_test_bin.ravel(), probas.ravel())
roc_auc_rf["macro"] = auc(fpr_rf["macro"], tpr_rf["macro"])

# Crear DataFrame con información de La curva ROC-AUC macro-averaged
data_rf_macro = {'FPR_Macro': fpr_rf["macro"],
                 'TPR_Macro': tpr_rf["macro"],
                 'AUC_Macro': [roc_auc_rf["macro"]] * len(fpr_rf["macro"])}

df_rf_macro = pd.DataFrame(data_rf_macro)

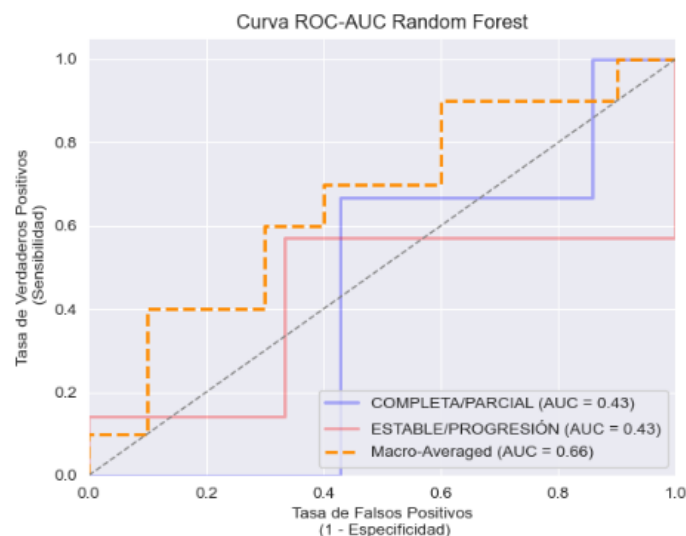
# Guardar el DataFrame en un archivo Excel
df_rf_macro.to_excel('informacion_rf_roc_auc_macro.xlsx', index=False)

# Plotear La curva ROC-AUC macro-averaged
plt.plot(fpr_rf["macro"], tpr_rf["macro"], color='darkorange', linestyle='--', lw=2, label=f'Macro-Averaged (AUC = {roc_auc_rf["macro"]:.2f})')

# Línea diagonal para referencia
plt.plot([0, 1], [0, 1], color='gray', lw=1, linestyle='--')

# Configurar Límites y etiquetas de Los ejes
plt.xlim([0, 1])
plt.ylim([0, 1.05])
plt.xlabel('Tasa de Falsos Positivos\n(1 - Especificidad)')
plt.ylabel('Tasa de Verdaderos Positivos\n(Sensibilidad)')
plt.title('Curva ROC-AUC Random Forest')
plt.legend(loc="lower right")
plt.show()

```



Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

7.2.1.2 SVM

```
import pandas as pd

features_LASSO = pd.read_excel('features_LASSO.xlsx')

df = features_LASSO

# Codificación de La variable Respuesta
df["Respuesta"] = df["Respuesta"].map({0: 0, 1: 1, 2: 1, 3: 1})

# Importar La función train_test_split de scikit-Learn
from sklearn.model_selection import train_test_split

# Separar Las características (X) y La variable objetivo (y)
X = df.drop('Respuesta', axis=1) # Características
y = df['Respuesta'] # Variable objetivo

# Dividir Los datos en conjuntos de entrenamiento y validación 80% - 20% de manera aleatoria
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2)

# Imprimir Las dimensiones de Los conjuntos resultantes
print("X_train shape:", X_train.shape)
print("X_val shape:", X_val.shape)
print("y_train shape:", y_train.shape)
print("y_val shape:", y_val.shape)
y_val

X_train shape: (40, 13)
X_val shape: (10, 13)
y_train shape: (40,)
y_val shape: (10,)

# Resetear Los índices de Los conjuntos de entrenamiento y validación
features_train = X_train.reset_index(drop=True)
features_val = X_val.reset_index(drop=True)
target_train = y_train.reset_index(drop=True)
target_val = y_val.reset_index(drop=True)

# Crear un nuevo DataFrame de entrenamiento (df_train) al concatenar características y variable objetivo de entrenamiento
df_train = pd.concat([features_train, target_train], axis=1)

features = features_train
target = target_train

# Importar Las bibliotecas necesarias
from sklearn.svm import SVC
from sklearn.model_selection import KFold
import joblib

# Crear un modelo SVM con kernel radial (rbf)
svm_model = SVC(kernel='rbf', C=1.0, gamma='scale', probability=True, random_state=42)

# Lista para almacenar Las predicciones de cada conjunto de prueba
predictions_listSVM = []

# Configurar La validación cruzada con KFold
kfold = KFold(n_splits=5, shuffle=True, random_state=42)

# Iterar a través de Los conjuntos de entrenamiento y prueba generados por KFold
for train_idx, test_idx in kfold.split(features):

    # Dividir el conjunto de datos en entrenamiento y prueba
    train_features, test_features = features.iloc[train_idx], features.iloc[test_idx]
    train_target, test_target = target.iloc[train_idx], target.iloc[test_idx]

    train_target = train_target.values.ravel()
    test_target = test_target.values.ravel()

    # Entrenar el modelo SVM en el conjunto de entrenamiento
    svm_model.fit(train_features, train_target)

    # Realizar predicciones en el conjunto de prueba
    predsSVM = svm_model.predict(test_features)

    # Crear un DataFrame con Las respuestas reales y Las predicciones
    test_dataSVM = df_train.loc[test_idx, :]
    test_dataSVM['Predictive Response'] = predsSVM
    predictions_listSVM.append(test_dataSVM[['Respuesta', 'Predictive Response']])

# Guardar el modelo entrenado SVM en un archivo
joblib.dump(svm_model, 'SVM_Model_LASSO.pkl')

['SVM_Model_LASSO.pkl']

# Cargar el modelo SVM
loaded_model = joblib.load('SVM_Model_LASSO.pkl')
```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```
# Importar Las bibliotecas necesarias
from sklearn.metrics import accuracy_score, f1_score
import numpy as np

# Realizar predicciones en el conjunto de validación
predsSVM = loaded_model.predict(features_val)

# Calcular La precisión del modelo
scoreSVM = accuracy_score(target_val, predsSVM)

# Crear una lista de 1 - puntuaciones de precisión para el cálculo de La desviación estándar
accuracy_scores = [1 - score for score in predsSVM]

# Calcular La desviación estándar de Las puntuaciones de precisión
std_devSVM = np.std(accuracy_scores)

# Calcular el F1-score
f1SVM = f1_score(target_val, predsSVM, average='macro')

# Imprimir métricas de rendimiento
print(f'Precisión: {scoreSVM}')
print(f'Desviación estándar: {std_devSVM}')
print(f'F1_score: {f1SVM}')

# Crear un DataFrame para Las predicciones
predictionsSVM = pd.DataFrame({
    'Respuesta': np.array(target_val).flatten(),
    'Predictive Response': np.array(predsSVM).flatten()
})
```

```
Precisión: 0.6
Desviación estándar: 0.30000000000000004
F1_score: 0.375
```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

class_names_dict = {
    0: 'Completa',
    1: 'Parcial, Estable y Progresion',
}

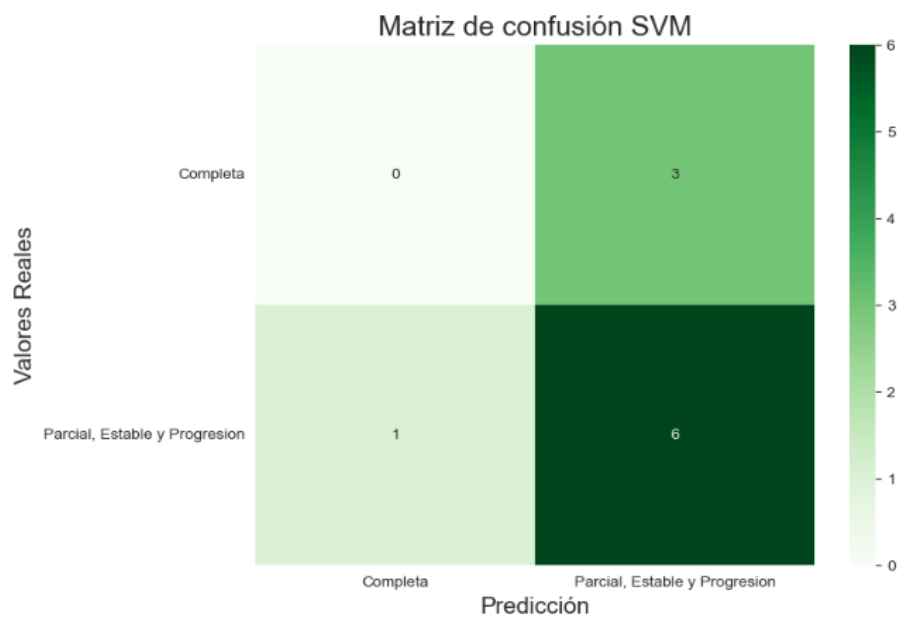
conf_matrix = confusion_matrix(target_val, predsSVM)
class_names = [class_names_dict[i] for i in sorted(class_names_dict)]

# Establecer estilo y color de fondo
sns.set_style('darkgrid')
plt.figure(figsize=(8, 6))

# Crear el heatmap con paleta de verdes
heatmap = sns.heatmap(conf_matrix, annot=True, cmap='Greens', fmt='g')

# Añadir etiquetas
heatmap.set_xticklabels(class_names, rotation=0, ha='center')
heatmap.set_yticklabels(class_names, rotation=0, va='center')

# Añadir títulos
plt.xlabel('Predicción', fontsize=15)
plt.ylabel('Valores Reales', fontsize=15)
plt.title('Matriz de confusión SVM', fontsize=18)
plt.show()
```



Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```
# Importar Las bibliotecas necesarias
from sklearn.metrics import roc_curve, roc_auc_score, auc
from sklearn.preprocessing import label_binarize
import matplotlib.pyplot as plt

# Predecir Las probabilidades
pred_probs = loaded_model.predict_proba(features_val)

# Binarizar Las Clases para calcular el puntaje ROC-AUC
y_test_bin = label_binarize(target_val, classes=np.unique(target_val))

# Ajustar Las clases si solo hay una columna en y_test_bin
if y_test_bin.shape[1] == 1:
    y_test_bin = np.hstack([1 - y_test_bin, y_test_bin])

# Calcular La curva ROC-AUC y el puntaje para cada clase
fpr = {}
tpr = {}
roc_auc = dict()

# Para cada clase en el conjunto de datos
for i in range(y_test_bin.shape[1]):
    fpr[i], tpr[i], _ = roc_curve(y_test_bin[:, i], pred_probs[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

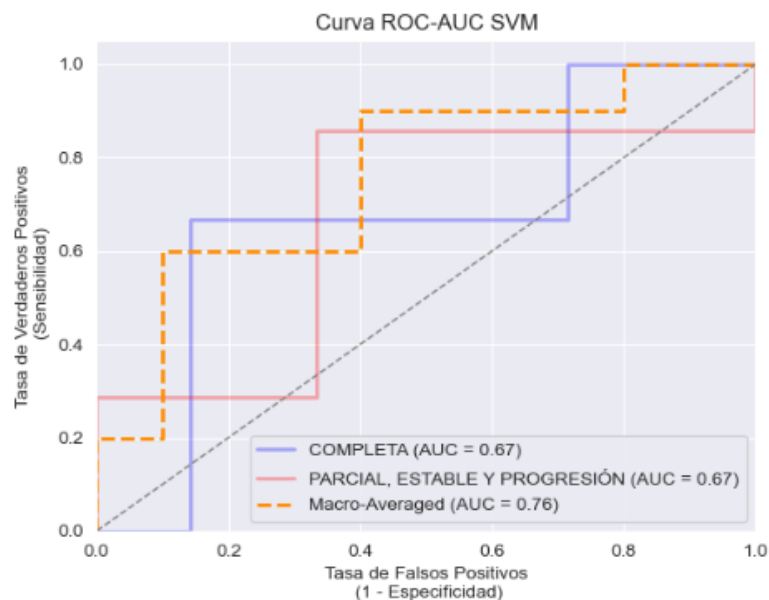
# Plotear Las curvas ROC-AUC para cada clase
plt.plot(fpr[0], tpr[0], color='blue', lw=2, alpha=0.3,
         label=f'COMPLETA (AUC = {roc_auc[0]:.2f})')

plt.plot(fpr[1], tpr[1], color='red', lw=2, alpha=0.3,
         label=f'PARCIAL, ESTABLE Y PROGRESIÓN (AUC = {roc_auc[1]:.2f})')

# Calcular y plotear La curva ROC-AUC macro-averaged
fpr["macro"], tpr["macro"], thresholds = roc_curve(y_test_bin.ravel(), pred_probs.ravel())
roc_auc["macro"] = auc(fpr["macro"], tpr["macro"])

plt.plot(fpr["macro"], tpr["macro"], color='darkorange', linestyle='--', lw=2,
         label=f'Macro-Averaged (AUC = {roc_auc["macro"]:.2f})')

# plot
plt.plot([0, 1], [0, 1], color='gray', lw=1, linestyle='--')
plt.xlim([0, 1])
plt.ylim([0, 1.05])
plt.xlabel('Tasa de Falsos Positivos (1 - Especificidad)')
plt.ylabel('Tasa de Verdaderos Positivos (Sensibilidad)')
plt.title('Curva ROC-AUC SVM')
plt.legend(loc="lower right")
plt.show()
```



Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

7.2.1.3 Regresión Logística

```
# Importar La biblioteca pandas con el alias pd
import pandas as pd

# Leer Los datos desde un archivo Excel y almacenarlos en df
df = pd.read_excel('features_LASSO.xlsx')

# Codificación de La variable Respuesta
df["Respuesta"] = df["Respuesta"].map({0: 0, 1: 1, 2: 1, 3: 1})

# Importar La función train_test_split de scikit-Learn
from sklearn.model_selection import train_test_split

# Separar Las características (X) y La variable objetivo (y)
X = df.drop('Respuesta', axis=1) # Características
y = df['Respuesta'] # Variable objetivo

# Dividir Los datos en conjuntos de entrenamiento y validación 80% - 20% de manera aleatoria
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2)

# Imprimir Las dimensiones de Los conjuntos resultantes
print("X_train shape:", X_train.shape)
print("X_val shape:", X_val.shape)
print("y_train shape:", y_train.shape)
print("y_val shape:", y_val.shape)
y_val

X_train shape: (40, 13)
X_val shape: (10, 13)
y_train shape: (40,)
y_val shape: (10,)

# Resetear Los índices de Los conjuntos de entrenamiento y validación
features_train = X_train.reset_index(drop=True)
features_val = X_val.reset_index(drop=True)
target_train = y_train.reset_index(drop=True)
target_val = y_val.reset_index(drop=True)

# Crear un nuevo DataFrame de entrenamiento (df_train) al concatenar características y variable objetivo de entrenamiento
df_train = pd.concat([features_train, target_train], axis=1)

features = features_train
target = target_train

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import KFold
import numpy as np
import joblib

kfold = KFold(n_splits=5, shuffle=True, random_state=42)

accuracy_scores = []
f1_scores = []

# Realizar La validación cruzada
for train_idx, test_idx in kfold.split(features):
    train_features, test_features = features.iloc[train_idx], features.iloc[test_idx]
    train_target, test_target = target.iloc[train_idx], target.iloc[test_idx]

    train_target = train_target.values.ravel()
    test_target = test_target.values.ravel()

    # Crear el modelo y entrenarlo
    logistic_model = LogisticRegression(max_iter=1000, multi_class='ovr')
    logistic_model.fit(train_features, train_target)

    # Realizar predicciones en el conjunto de prueba
    preds = logistic_model.predict(test_features)

# Guardar el modelo entrenado en un archivo
joblib.dump(logistic_model, 'modelo_regresion_logistica_LASSO.pkl')

# Cargar el modelo
loaded_model = joblib.load('modelo_regresion_logistica_LASSO.pkl')
```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```
# Importar Las bibliotecas necesarias
from sklearn.metrics import accuracy_score, f1_score
import numpy as np

# Realizar predicciones en el conjunto de validación
predsRLM = loaded_model.predict(features_val)

# Calcular La precisión del modelo
scoreRLM = accuracy_score(target_val, predsRLM)

# Crear una lista de 1 - puntuaciones de precisión para el cálculo de la desviación estándar
accuracy_scores = [1 - score for score in predsRLM]

# Calcular La desviación estándar de Las puntuaciones de precisión
std_devRLM = np.std(accuracy_scores)

# Calcular el F1-score
f1RLM = f1_score(target_val, predsRLM, average='macro')

# Imprimir métricas de rendimiento
print(f'Precisión: {scoreRLM}')
print(f'Desviación estándar: {std_devRLM}')
print(f'F1_score: {f1RLM}')

# Crear un DataFrame para Las predicciones
predictionsRLM = pd.DataFrame({
    'Respuesta': np.array(target_val).flatten(),
    'Predictive Response': np.array(predsRLM).flatten()
})
```

Precisión: 0.7
 Desviación estándar: 0.30000000000000004
 F1_score: 0.6000000000000001

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

class_names_dict = {
    0: 'Completa',
    1: 'Parcial, Estable y Progresion',
}

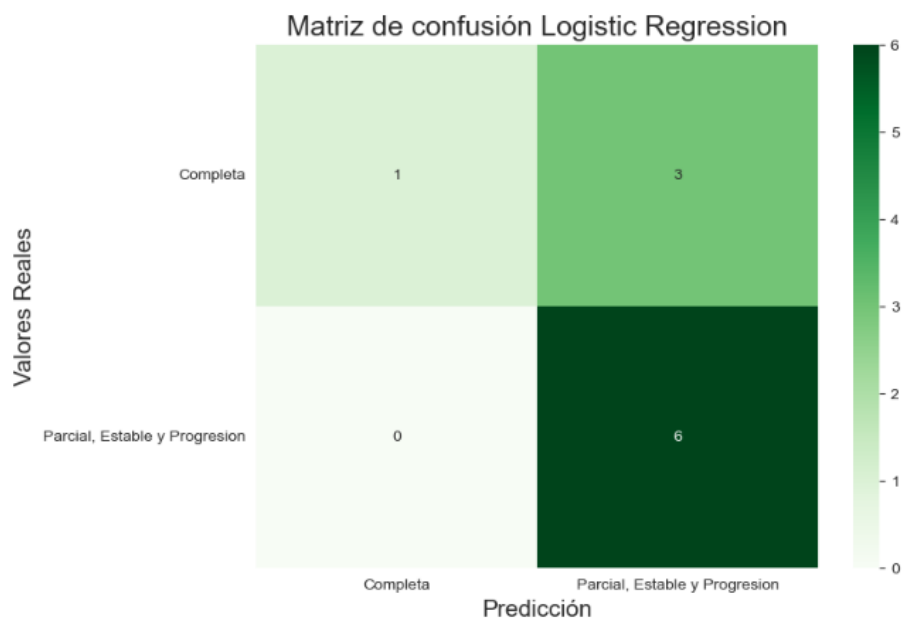
conf_matrix = confusion_matrix(target_val, predsRLM)
class_names = [class_names_dict[i] for i in sorted(class_names_dict)]

# Establecer estilo y color de fondo
sns.set_style('darkgrid')
plt.figure(figsize=(8, 6))

# Crear el heatmap con paleta de verdes
heatmap = sns.heatmap(conf_matrix, annot=True, cmap='Greens', fmt='g')

# Añadir etiquetas
heatmap.set_xticklabels(class_names, rotation=0, ha='center')
heatmap.set_yticklabels(class_names, rotation=0, va='center')

# Añadir títulos
plt.xlabel('Predicción', fontsize=15)
plt.ylabel('Valores Reales', fontsize=15)
plt.title('Matriz de confusión Logistic Regression', fontsize=18)
plt.show()
```



Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```

# Importar Las bibliotecas necesarias
from sklearn.metrics import roc_curve, roc_auc_score, auc
from sklearn.preprocessing import label_binarize
import matplotlib.pyplot as plt

# Predecir Las probabilidades
pred_probs = loaded_model.predict_proba(features_val)

# Binarizar Las clases para calcular el puntaje ROC-AUC
y_test_bin = label_binarize(target_val, classes=np.unique(target_val))

# Ajustar las clases si solo hay una columna en y_test_bin
if y_test_bin.shape[1] == 1:
    y_test_bin = np.hstack([1 - y_test_bin, y_test_bin])

# Calcular La curva ROC-AUC y el puntaje para cada clase
fpr_rf = {}
tpr_rf = {}
roc_auc_rf = dict()

# Para cada clase en el conjunto de datos
for i in range(y_test_bin.shape[1]):
    fpr_rf[i], tpr_rf[i], _ = roc_curve(y_test_bin[:, i], pred_probs[:, i])
    roc_auc_rf[i] = auc(fpr_rf[i], tpr_rf[i])

# Plotear Las curvas ROC-AUC para cada clase
plt.plot(fpr_rf[0], tpr_rf[0], color='blue', lw=2, alpha=0.3,
         label=f'COMPLETA (AUC = {roc_auc_rf[0]:.2f})')

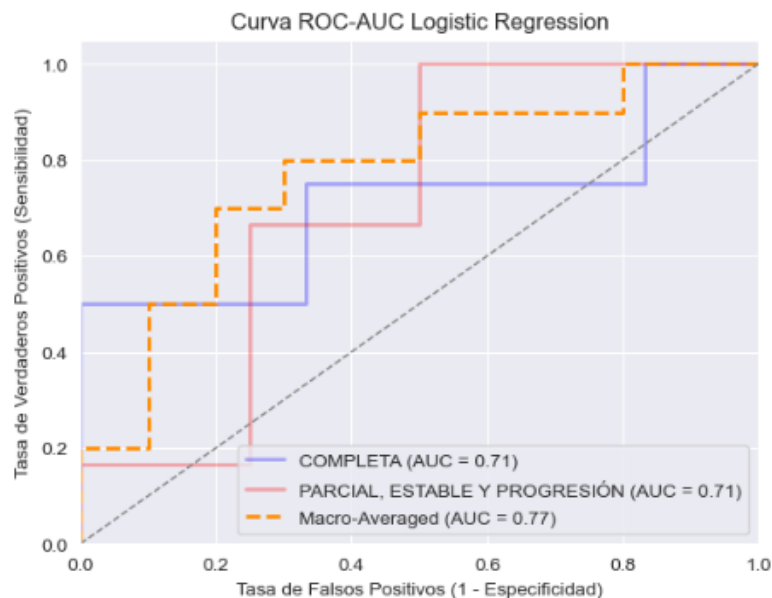
plt.plot(fpr_rf[1], tpr_rf[1], color='red', lw=2, alpha=0.3,
         label=f'PARCIAL, ESTABLE Y PROGRESIÓN (AUC = {roc_auc_rf[1]:.2f})')

# Calcular y plotear La curva ROC-AUC macro-averaged
fpr_rf["macro"], tpr_rf["macro"], thresholds = roc_curve(y_test_bin.ravel(), pred_probs.ravel())
roc_auc_rf["macro"] = auc(fpr_rf["macro"], tpr_rf["macro"])

plt.plot(fpr_rf["macro"], tpr_rf["macro"], color='darkorange', linestyle='--', lw=2,
         label=f'Macro-Averaged (AUC = {roc_auc_rf["macro"]:.2f})')

# plot
plt.plot([0, 1], [0, 1], color='gray', lw=1, linestyle='--')
plt.xlim([0, 1])
plt.ylim([0, 1.05])
plt.xlabel('Tasa de Falsos Positivos (1 - Especificidad)')
plt.ylabel('Tasa de Verdaderos Positivos (Sensibilidad)')
plt.title('Curva ROC-AUC Logistic Regression')
plt.legend(loc="lower right")
plt.show()

```



Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

7.2.1.4 Red Neuronal Artificial

```
import pandas as pd
```

```
# Leer Los datos desde un archivo Excel y almacenarlos en df  
df = pd.read_excel('features_RandomForest.xlsx')
```

```
# Codificación de La variable Respuesta  
df["Respuesta"] = df["Respuesta"].map({0: 0, 1: 0, 2: 1, 3: 1})
```

```
# Importar la función train_test_split de scikit-Learn  
from sklearn.model_selection import train_test_split
```

```
# Separar las características (X) y la variable objetivo (y)  
X = df.drop('Respuesta', axis=1) # Características  
y = df['Respuesta'] # Variable objetivo
```

```
# Dividir los datos en conjuntos de entrenamiento y validación 80% - 20% de manera aleatoria  
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2)
```

```
# Imprimir las dimensiones de los conjuntos resultantes  
print("X_train shape:", X_train.shape)  
print("X_val shape:", X_val.shape)  
print("y_train shape:", y_train.shape)  
print("y_val shape:", y_val.shape)
```

```
X_train shape: (40, 12)  
X_val shape: (10, 12)  
y_train shape: (40,)  
y_val shape: (10,)
```

```
# Resetear los índices de los conjuntos de entrenamiento y validación  
features_train = X_train.reset_index(drop=True)  
features_val = X_val.reset_index(drop=True)  
target_train = y_train.reset_index(drop=True)  
target_val = y_val.reset_index(drop=True)
```

```
# Crear un nuevo DataFrame de entrenamiento (df_train) al concatenar características y variable objetivo de entrenamiento  
df_train = pd.concat([features_train, target_train], axis=1)
```

```
features = features_train  
target = target_train
```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```

from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.optimizers import Adam
from keras.callbacks import EarlyStopping
from sklearn.model_selection import LeaveOneOut
import numpy as np
import pandas as pd

target = target_train
features = features_train

# Definir modelo ANN
model = Sequential()
model.add(Dense(28, activation='relu', input_shape=(features_train.shape[1],))) # Input Layer with ReLU activation
model.add(Dense(16, activation='relu', kernel_regularizer='l2')) # Intermediate Layer with ReLU activation and fewer neurons
model.add(Dropout(0.3)) # Keep dropout rate as 0.3
model.add(Dense(1, activation='sigmoid')) # Output Layer with Sigmoid activation for binary classification

# Compilar el modelo
optimizer = Adam(learning_rate=0.001)
model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])

# Inicializar el objeto LOOCV
loo = LeaveOneOut()

# Definir Early Stopping
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

# Listas para almacenar resultados
predictions = []
true_labels = []
model_history = []

# Entrenamiento y validación con LOOCV
for train_index, test_index in loo.split(features_train):
    x_train, x_test = features_train.iloc[train_index], features_train.iloc[test_index]
    y_train, y_test = target_train.iloc[train_index], target_train.iloc[test_index]

    y_train = y_train.values.ravel()
    y_test = y_test.values.ravel()

    # Entrenar el modelo en este fold
    history = model.fit(x_train, y_train, epochs=50, batch_size=32, verbose=0,
                       validation_data=(x_test, y_test), callbacks=[early_stopping])

    # Guardar la historia de entrenamiento
    model_history.append(history)

    # Predecir y almacenar los resultados
    preds = model.predict(x_test)

    predictions = pd.DataFrame({
        'Respuesta': y_test,
        'Predictive Response': (preds > 0.5).astype(int).ravel() # Adjust predictions for binary output
    })
    true_labels.append(predictions)

# Guardar el modelo y los pesos
model.save('modelo_ANN_RF.h5')
model.save_weights('pesos_ANN_RF.h5')

# Guardar las predicciones en un DataFrame
predictions_df = pd.concat(true_labels)
predictions_df.to_excel('PredictionsANN_RF.xlsx', index=False)

```

```

import numpy as np
import matplotlib.pyplot as plt

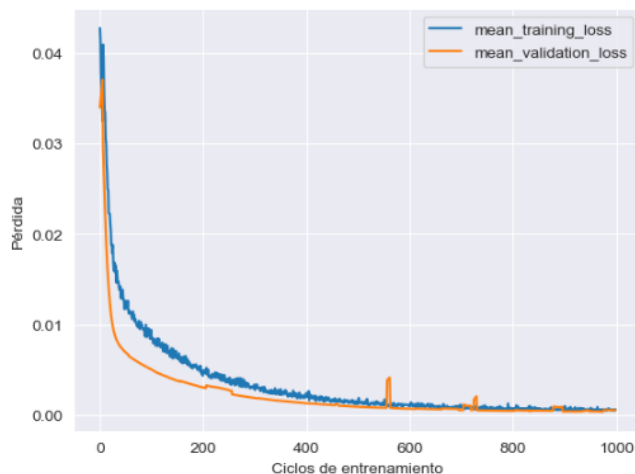
# Lista que contiene las historias de entrenamiento de varios modelos
training_losses = [history.history['loss'] for history in model_history]
validation_losses = [history.history['val_loss'] for history in model_history]

# Calcular las pérdidas promedio para el entrenamiento y la validación en cada ciclo
mean_training_loss = np.mean(training_losses, axis=0, dtype='object')
mean_validation_loss = np.mean(validation_losses, axis=0, dtype='object')

# Plotear las pérdidas promedio
plt.plot(mean_training_loss)
plt.plot(mean_validation_loss)
plt.xlabel('Ciclos de entrenamiento')
plt.ylabel('Pérdida')
plt.legend(['mean_training_loss', 'mean_validation_loss'])
plt.show()

```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.



```
from keras.models import load_model

# Cargar el modelo de Red Neuronal Artificial (RNA)
modelRN = load_model('modelo_ANN_RF.h5')

# Cargar Los pesos del modelo
modelRN.load_weights('pesos_ANN_RF.h5')

from sklearn.metrics import accuracy_score, f1_score
import pandas as pd

target_val = target_val.values.ravel()
preds_val = modelRN.predict(features_val)

# Calcular precisión y F1 score
accuracy = accuracy_score(target_val, np.argmax(preds_val, axis=1))
f1 = f1_score(target_val, np.argmax(preds_val, axis=1), average='weighted')

# Imprimir precisión y F1 score
print(f'Accuracy: {accuracy}')
print(f'F1 Score: {f1}')

# Crear un DataFrame para las predicciones
predictions_val = pd.DataFrame({
    'Respuesta': target_val,
    'Predictive Response': np.argmax(preds_val, axis=1)
})

# Guardar las predicciones en un archivo Excel
predictions_val.to_excel('PredictionsANN_validation.xlsx', index=False)

1/1 [=====] - 0s 153ms/step
Accuracy: 0.9
F1 Score: 0.8526315789473685
```

Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```

from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import label_binarize
import matplotlib.pyplot as plt

# Binarizar las clases para calcular ROC-AUC
nuevo_testy_bin = label_binarize(target_val, classes=np.unique(target_val))

if nuevo_testy_bin.shape[1] == 1:
    nuevo_testy_bin = np.hstack([1 - nuevo_testy_bin, nuevo_testy_bin])

# Calcular la curva ROC-AUC y el puntaje para cada clase
fpr = {}
tpr = {}
roc_auc = {}

# Solo dos clases
for i in range(nuevo_testy_bin.shape[1]):
    fpr[i], tpr[i], umbral = roc_curve(nuevo_testy_bin[:, i], preds_val[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

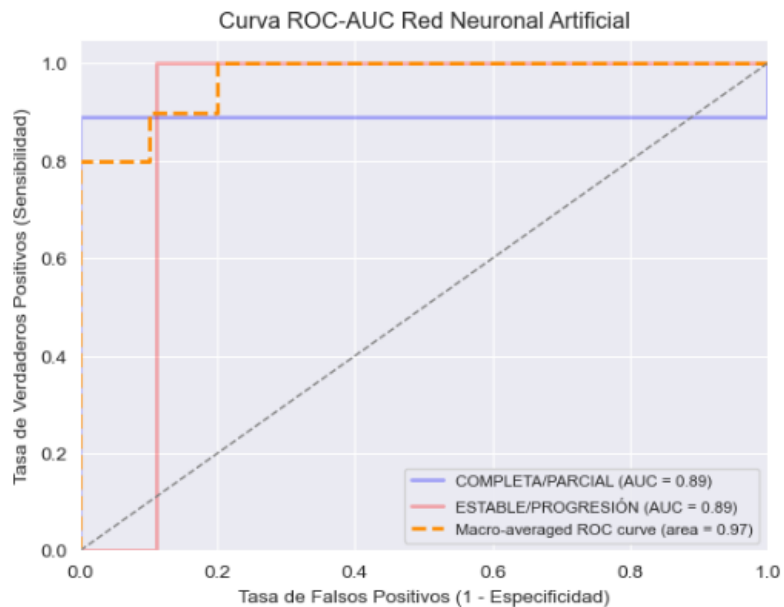
# Plotear las curvas ROC-AUC para cada clase
plt.figure()
plt.plot(fpr[0], tpr[0], color='blue', lw=2, alpha=0.3, label=f'COMPLETA/PARCIAL (AUC = {roc_auc[0]:.2f})')
plt.plot(fpr[1], tpr[1], color='red', lw=2, alpha=0.3, label=f'ESTABLE/PROGRESIÓN (AUC = {roc_auc[1]:.2f})')

# Calcular y plotear la curva ROC-AUC macro-averaged
fpr["macro"], tpr["macro"], thresholds = roc_curve(nuevo_testy_bin.ravel(), preds_val[:, :2].ravel())
roc_auc["macro"] = auc(fpr["macro"], tpr["macro"])

plt.plot(fpr["macro"], tpr["macro"], color='darkorange', linestyle='--', lw=2,
        label=f'Macro-averaged ROC curve (area = {roc_auc["macro"]:.2f})')

# plot
plt.plot([0, 1], [0, 1], color='gray', lw=1, linestyle='--')
plt.xlim([0, 1])
plt.ylim([0, 1.05])
plt.xlabel('Tasa de Falsos Positivos (1 - Especificidad)')
plt.ylabel('Tasa de Verdaderos Positivos (Sensibilidad)')
plt.title('Curva ROC-AUC Red Neuronal Artificial')
plt.legend(loc="lower right", fontsize='small', frameon=True)
plt.show()

```



Desarrollo de modelos predictivos basados en inteligencia artificial y radiómica para asistir en las decisiones clínicas para pacientes con cáncer de recto.

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Definir diccionario de clases
class_names_dict = {
    0: 'Completa/Parcial',
    1: 'Estable/Progresion',
}

# Calcular la matriz de confusión
conf_matrix = confusion_matrix(target_val, np.argmax(preds_val, axis=1))
class_names = [class_names_dict[i] for i in sorted(class_names_dict)]

# Establecer estilo y color de fondo
sns.set_style('darkgrid')
plt.figure(figsize=(8, 6))

# Crear el heatmap
heatmap = sns.heatmap(conf_matrix, annot=True, cmap='Greens', fmt='g')

# Añadir etiquetas
heatmap.set_xticklabels(class_names, rotation=0, ha='center')
heatmap.set_yticklabels(class_names, rotation=0, va='center')

# Añadir títulos
plt.xlabel('Predicción', fontsize=15)
plt.ylabel('Valores Reales', fontsize=15)
plt.title('Matriz de confusión ANN RF', fontsize=18)
plt.show()
```

