

Juan Pedro Fernández Oliver

Radioterapia ablativa con intención curativa para el tratamiento de las metástasis en pacientes con cáncer: radiómica e inteligencia artificial para asistir en las decisiones clínicas.

**Trabajo Final de Grado
dirigido por Dr. Hernández y Dra. Arenas**

Grado en Ingeniería Biomédica



UNIVERSITAT ROVIRA I VIRGILI

**Tarragona
2023**

Resumen

La radioterapia corporal ablativa estereotáctica (SABR) es una novedosa técnica de radioterapia que utiliza altas dosis de radiación para tratar tumores y destruir las células cancerosas, minimizando los daños a los tejidos circundantes. Los recientes estudios realizados durante los últimos años, han proporcionado resultados relevantes sobre el uso de la radiómica en estudios clínicos para posteriores aplicaciones médicas prácticas. Se ha comprobado que los modelos combinados por características radiómicas y variables clínicas son la clave para la predicción del seguimiento de los tratamientos radioterapéuticos SABR.

Mediante esta premisa, el presente estudio, parte del uso de la radiómica junto con algoritmos de *Machine Learning* para la predicción de la respuesta del cáncer metastásico en pacientes oligometastásicos tratados con SABR, con la finalidad de asistir en las decisiones clínicas. Se han estudiado un total de 128 metástasis distintas proveídas por 93 pacientes tratados en la unidad de radioterapia del Hospital Universitario Sant Joan de Reus (HUSJR). De esta forma, se exploran diversos métodos con el fin de obtener la mejor predicción de la respuesta metastásica, comparando distintos modelos de selección de características y enfoques de predicción.

Finalmente, los resultados obtenidos demuestran, a pesar de las limitaciones propias al estudio, la viabilidad de una aplicación mediante la implementación de una Red Neuronal Artificial capaz de abordar el problema mencionado.

Abstract

Stereotactic ablative body radiation therapy (SABR) is a novel radiation therapy technique that uses high doses of radiation to treat tumors and destroy cancer cells, minimizing damage to surrounding tissues. Recent studies conducted in recent years have provided relevant results on the use of radiomics in clinical studies for subsequent practical medical applications. Combining radiological models and clinical variables have been shown to be the key to predicting follow-up of SABR radiation therapy treatments.

Using this premise, the present study uses radiomics together with Machine Learning algorithms to predict the response of metastatic cancer in oligometastatic patients treated with SABR, in order to assist in clinical decisions. A total of 128 different metastases provided by 93 patients treated in the radiation therapy unit of the Hospital Universitario Sant Joan de Reus (HUSJR) have been studied. In this way, various methods are explored in order to obtain the best prediction of the metastatic response, comparing different characteristic selection models and prediction approaches.

Finally, the results obtained demonstrate, despite the limitations inherent to the study, the feasibility of an application by implementing an Artificial Neural Network capable of addressing the aforementioned problem.

Índice

1	Introducción.....	1
1.1	Nomenclatura del volumen tumoral.....	4
2	Hipótesis y Objetivos.....	5
3	Materiales y Métodos	6
3.1	Metodología	6
3.2	Descripción de la base de datos.....	7
3.2.1	Descripción clínica.....	9
3.3	Extracción de características.....	12
3.4	Reducción y selección de características	17
3.4.1	Random Forest (RF).....	18
3.4.2	Minimum Redundancy and Maximum Relevance (mRMR)	18
3.4.3	Relief	18
3.5	Construcción de modelos predictivos.....	19
3.5.1	Random Forest	20
3.5.2	Support Vector Machine	20
3.5.3	Red Neuronal Artificial	20
4	Resultados	23
4.1	Selección de variables.....	23
4.2	Entrenamiento y validación	26
5	Discusión	33
6	Conclusiones	36
7	Bibliografía.....	37
8	Anexos.....	39
8.1	Anexo 1: Código del programa	39
8.1.1	Preprocesado de datos	39
8.1.2	Selección de características.....	40
8.1.3	Modelos predictivos.....	42
8.2	Anexo 2: Imágenes adicionales	50
8.2.1	Conjunto Global	50
8.2.2	Conjunto Metástasis Óseas	51
8.2.3	Conjunto Metástasis Pulmonares	52

1 Introducción

La metástasis es el evento más complejo y mortal; el reto más amenazante al cáncer. Según la CTCA (*Cancer Treatment Centers of America*) el cáncer metastásico representaba en 2021 hasta el 90% de las muertes por cáncer cada año en los Estados Unidos [1]. Poniendo en contexto, se estima que la tasa de supervivencia a 5 años en pacientes con cáncer de mama metastásico es del 29% para las mujeres y del 22% para los hombres [2].

En la actualidad, el cáncer en etapa de metástasis sigue siendo el caso con mayor dificultad a tratar, entre otras razones, por las limitaciones que presentan los tratamientos actuales como la extirpación quirúrgica, la cual se ve delimitada por la zona a tratar y a expensas de dejar células malignas circundantes, con la posibilidad de la reaparición del tumor.

Según un informe reportado en 2022 por la SEOM (Sociedad Española de Oncología Médica) los cánceres que más frecuentemente se diagnostican son de mama, pulmón, colorrectal y próstata respectivamente [3] y con mayor probabilidad de derivar metástasis en hueso, pulmón, hígado y glándulas suprarrenales, entre otras.

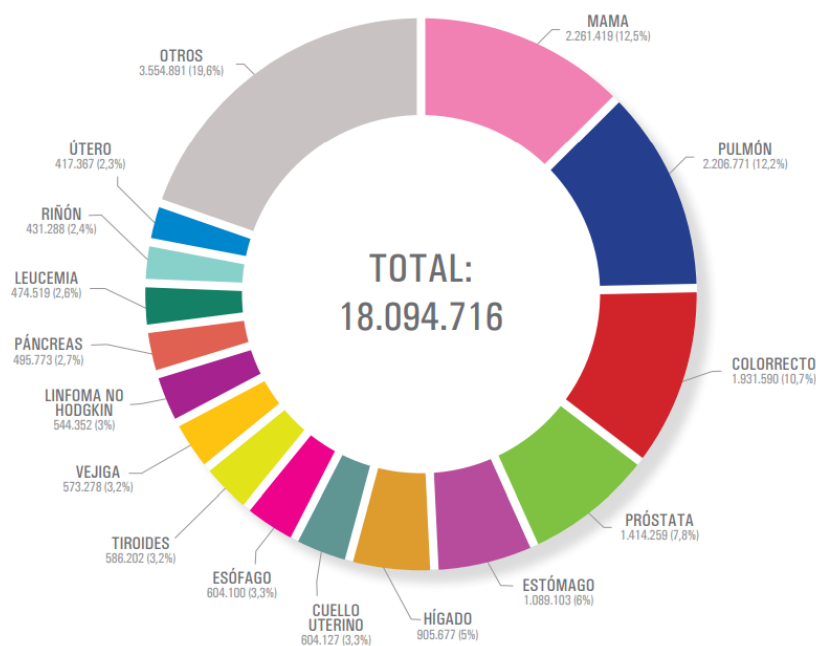


Figura 1. Tumores más frecuentemente diagnosticados para ambos sexos en el mundo para el año 2022 [3].

Un paciente oligometastásico (“oligos” del griego “poco” o “escaso”), hace referencia a aquella persona que padece de cáncer en estado avanzado, dando lugar a un número limitado de metástasis, generalmente de uno a cinco, en reducidos sitios más allá de su ubicación original.

La radioterapia corporal ablativa estereotáctica (SABR o SBRT), del inglés *Stereotactic ablative body radiotherapy*, es una técnica de radioterapia empleada para eliminar los tumores malignos en una zona concreta del cuerpo de un paciente utilizando dosis biológicamente efectiva de alta energía radioactiva, mayores a las sesiones de radioterapia convencionales.

La unidad de medida de la dosis administrada en radioterapia es el Gray (Gy), que corresponde a la energía absorbida por unidad de masa. Esta tecnología se caracteriza por administrar en pocas sesiones y con gran precisión, dosis intensas de radiación concentrada, limitando la dosis en los órganos circundantes, de forma que, mientras en tratamientos comunes de 25 a 30 sesiones se utilizan 1.8 y 2 Gy por sesión, la dosis por sesión en SABR, debe ser superior a 4 Gy, alcanzando los 30 y 40 Gy de dosis total en algunos casos, durante las 2 a 5 sesiones que puede llegar a durar el tratamiento [4].

Se lleva a cabo en casos de pacientes con cánceres difíciles de tratar o tumores avanzados de volumen limitado para quienes la cirugía puede no ser un tratamiento óptimo. Al ser un método altamente focalizado, tiene diversas ventajas como su alta efectividad y precisión y la capacidad de preservar el tejido sano mientras se intensifica la dosis de radiación [5].

Esta metodología de radioterapia hipofraccionada de alta dosis precisa, logra excelentes tasas de control local para tumores primarios o metástasis. Al dirigirse específicamente a las áreas afectadas, esta técnica puede lograr una mayor destrucción de las células cancerosas, reduciendo el tamaño de las lesiones y mejorando los síntomas asociados. Dentro de estos términos, se considera un tratamiento apto para pacientes con enfermedad oligometastásica, donde es necesario centrar y dirigir la dosis radioactiva a la zona metastásica, evitando los tejidos y órganos circundantes, haciendo competencia al tratamiento radioterapéutico habitual [6].

Por otro lado, la radiómica es una disciplina de la medicina basada en la extracción de datos y análisis de patrones cuantificables que componen las imágenes radiológicas como tomografías computarizadas (TAC) o resonancias magnéticas (RM) dentro del campo de la radiología o radiodiagnóstico [7], donde el ojo humano es incapaz de percibir a simple vista. La radiómica puede proporcionar un análisis profundo y longitudinal de la heterogeneidad de todo el tumor y de los tejidos circundantes in vivo [8], mejorando el diagnóstico mediante una evaluación más precisa y detallada del tumor o de la metástasis.

El desarrollo de esta técnica, se basa en la selección de una región de interés (ROI), la cual queda aislada del resto de la imagen médica mediante técnicas de segmentación. Es en este punto donde se procede a analizar los píxeles contenido en esta región, obteniendo datos sobre el tamaño, la forma, intensidad, textura, así como valores máximos y mínimos de tonos de grises, varianza, entropía, etc.

Existen diversos estudios que resaltan la importancia del manejo de la radiómica en los tratamientos radioterapéuticos. Se intuye que, mediante el uso combinado de la radiómica junto con modelos de inteligencia artificial y algoritmos de aprendizaje automático (*Machine Learning*), es posible desarrollar programas capaces de actuar y asistir en las decisiones clínicas, mejorando el diagnóstico y permitiendo al oncólogo determinar el tratamiento más favorable en su debido caso, llegando incluso a predecir la respuesta al mismo. Hasta el momento, se ha comprobado que estos modelos radiómicos se favorecen con la inclusión de características médicas o datos clínicos, formando así un modelo combinado de características radiómicas y variables clínicas que mejoran en la evaluación del seguimiento de las enfermedades cancerígenas y en tratamientos SABR [9].

Otros recientes estudios demuestran la eficacia de la radiómica como una herramienta con un gran valor potencial para contribuir a la gestión y toma de decisiones en diversos tratamientos no solo radioterapéuticos, sino también en quimioterapia y posoperatorios, proporcionando información de referencia para tratamientos precisos. Por ejemplo, el departamento de Medicina Nuclear del Hospital de la Universidad de Keimyung (República de Corea), desarrolló con éxito un modelo radiómico basado en aprendizaje automático para la

predicción de metástasis de ganglios linfáticos en el cáncer de mama que podría ayudar a la predicción individualizada preoperatoria de estos pacientes [10].

Sin embargo, el uso de la radiómica como aplicación clínica todavía se encuentra estancada, en un estado incipiente, a la espera de nuevos análisis e investigaciones que avalen y respalden su importancia clínica y terapéutica.

Con este fin, a lo largo del proyecto se plantea un nuevo estudio retrospectivo donde se solidarice el uso de la radiómica mediante la recreación de modelos predictivos de aprendizaje automático, a partir de tomografías computarizadas y sus respectivos volúmenes tumorales, para la predicción de la respuesta metastásica, siguiendo el criterio RECIST.

El criterio RECIST (*Response Evaluation Criteria In Solid Tumors*), es un modelo estándar de clasificación utilizado para medir la respuesta o el modo en que un paciente cancerígeno responde al tratamiento [11].

De esta manera, se puede clasificar la respuesta tumoral en 4 tipos de observaciones:

- **Respuesta Completa** o "Complete Response" (CR): Desaparición de toda la lesión tumoral. No hay evidencia del tumor.
- **Respuesta Parcial** o "Partial Response" (PR): Reducción de más de un 30% en el tamaño total del tumor. El tumor es reducido, pero aún persiste.
- **Respuesta Estable** o "Stable Disease" (SD): Reducción de menos del 30% o aumento del 20% del tamaño total del tumor. El tumor resiste al tratamiento.
- **Respuesta Progressiva** o "Progressive Disease" (PD): Aumento de más del 20% o aparición de nuevos tumores. No hay respuesta al tratamiento.

Seguidamente, la Figura 1 ilustra un diagrama explicativo que detalla el uso de la radiómica en este estudio, presentando un resumen de los pasos realizados:

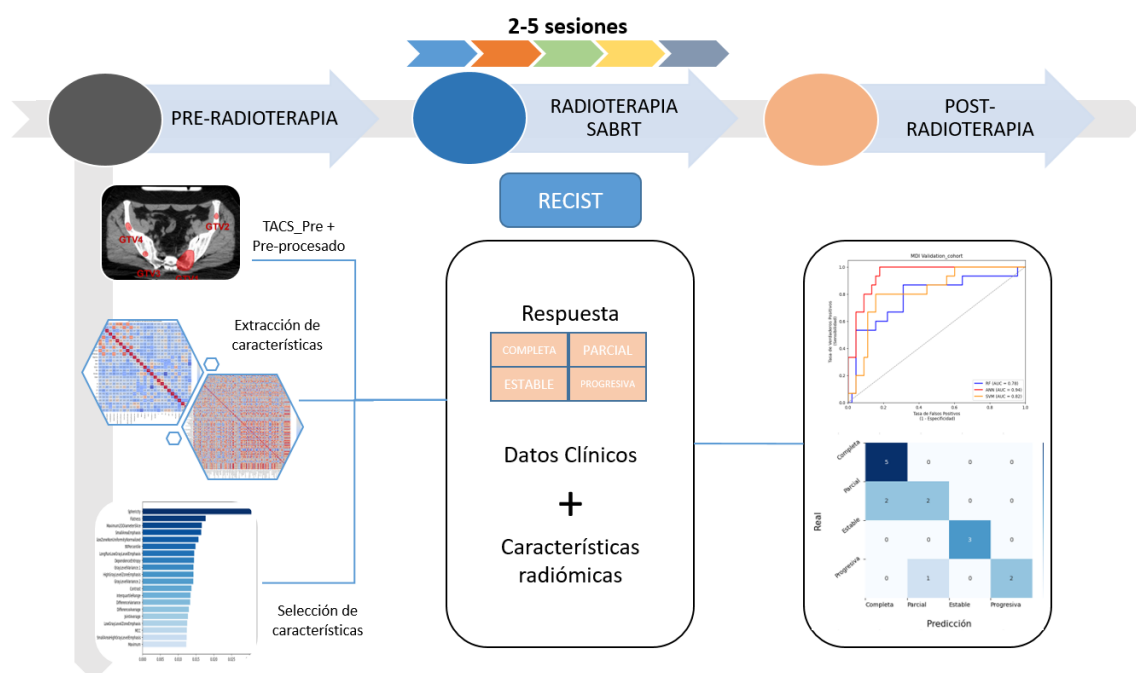


Figura 1. Diagrama explicativo del uso de la radiómica y modelos de aprendizaje automático para predecir la respuesta metastásica, basado en el análisis de tomografías computarizadas y volúmenes tumorales, siguiendo el criterio RECIST.

1.1 Nomenclatura del volumen tumoral

Para determinar la selección de las imágenes metastásicas y sus respectivos volúmenes de contorno, es necesario conocer los distintos tipos de volúmenes o ROIs, definidos por la Comisión Internacional de Unidades Radiológicas (ICRU) [12]. Estos pueden derivar según:

- **Gross Tumor Volume (GTV):** Región o volumen de enfermedad macroscópica visible del tumor. Es el volumen requerido y utilizado para la descarga de imágenes tumorales y su posterior extracción de variables.
- **Clinical Target Volume (CTV):** Volumen que contiene enfermedad microscópica o subclínica que debe tratarse, además del mencionado GTV.
- **Internal Target Volume (ITV):** Región generada en caso de que el volumen anterior no se mantenga fijo a lo largo de la simulación, ya sea a causa de la respiración o al movimiento interno de los propios órganos
- **Planning Target Volume (PTV):** Volumen que tiene en cuenta los posibles errores de posicionamiento en el paciente debido a incertidumbres en la colocación o imposibilidades geométricas durante la simulación.

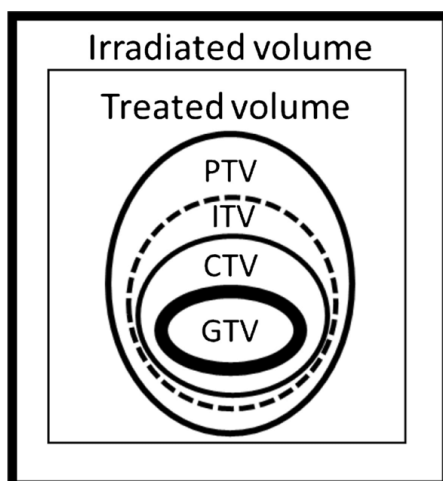


Figura 2. Definiciones esquemáticas de GTV, CTV, PTV e ITV (basadas en la Comisión Internacional de Unidades y Medidas de Radiación, informe 62 - ICRU-62) [12].

2 Hipótesis y Objetivos

La hipótesis planteada en el presente estudio resalta en la integración de la radiómica sobre la práctica clínica, combinada con tecnologías avanzadas como la inteligencia artificial y algoritmos de aprendizaje automático, para mejorar significativamente el diagnóstico e identificar una mayor precisión de las características que afectan al pronóstico y en la predicción de la respuesta al tratamiento. De esta manera, ayudar a mejorar en la atención y en la toma de decisiones clínicas basadas en la evidencia.

Por consiguiente, este proyecto de estudio retrospectivo, tiene como objetivos la obtención de biomarcadores radiómicos para la elaboración y desarrollo de un modelo de inteligencia artificial con la capacidad de predecir la respuesta tumoral y/o metastásica de los pacientes oligometastásicos tratados con SABR. De esta manera podemos diferenciar tres diferentes fines:

1. Creación de base de datos de variables clínicas y obtención de las imágenes TAC de los pacientes de cáncer con metástasis tratados con radioterapia curativa en el Hospital Universitario Sant Joan de Reus.
2. Extracción de las características radiómicas de las imágenes TAC descritas en el apartado anterior.
3. Desarrollo de modelos de inteligencia artificial para predecir la respuesta clínica en relación con el pronóstico y la respuesta al tratamiento de estos pacientes. Caracterizar la capacidad predictiva de estos modelos y evaluar posibles aplicaciones prácticas.

3 Materiales y Métodos

3.1 Metodología

La metodología utilizada en este trabajo se fundamenta en una serie de pasos para analizar y predecir la respuesta metastásica en pacientes a partir de las características radiómicas extraídas de los volúmenes GTVs definidos como la región de interés en las tomografías computarizadas de simulación.

Primero, las características radiómicas de mayor relevancia, que son datos cuantitativos de imágenes médicas, se recopilan con el objetivo de obtener información detallada sobre las propiedades de las metástasis. Estas características se consiguen mediante el software de Slicer3D, capacitado para realizar técnicas de procesamiento de imágenes y análisis radiómico.

A continuación, se lleva a cabo un proceso de selección de características, donde se identifican las características radiómicas más importantes en relación con la variable objetivo, siendo esta la respuesta metastásica. Además de las características radiómicas, se incorporan otras variables clínicas relevantes que pueden influir en la respuesta metastásica. Estas variables clínicas pueden incluir información demográfica, biomarcadores, historial médico u otros datos del paciente que puedan afectar al modelo final.

Por último, se utilizan técnicas de inteligencia artificial y modelos de aprendizaje automático, como algoritmos de regresión o clasificación, para desarrollar un modelo predictivo. Este modelo se entrena utilizando las características radiómicas seleccionadas junto a variables clínicas, y finalmente, evaluar su capacidad para predecir la respuesta metastásica en función de la clasificación RECIST mencionada anteriormente.

La elección de los métodos utilizados se escogió siguiendo distintos modelos de artículos académicos donde se destaca el uso de la radiómica con técnicas de *Machine Learning*, basadas en características radiómicas para el seguimiento de enfermedades cancerígenas [9] [19] [20].

A continuación, se muestra un plan esquemático de los procedimientos llevados a práctica, para cumplir con el propósito inicial y los objetivos finales:

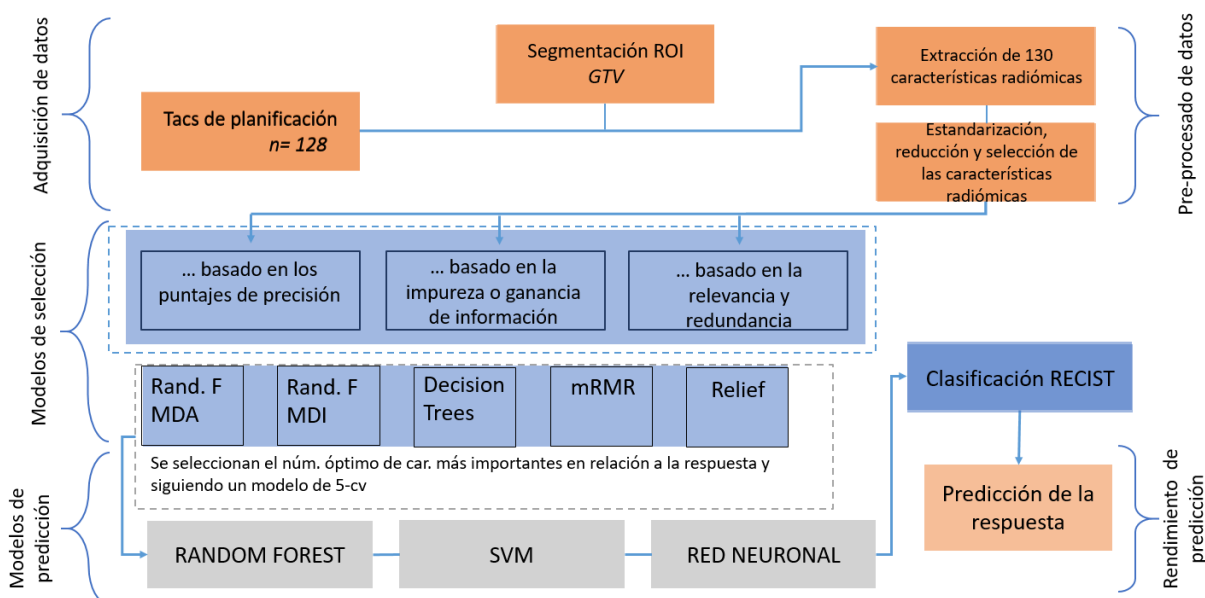


Figura 3. Plan esquemático para el desarrollo de un modelo predictivo a partir del uso de la radiómica junto a modelos de inteligencia artificial y algoritmos de aprendizaje automático.

3.2 Descripción de la base de datos

Se dispone de una base de datos creada por un conjunto de estudiantes de medicina de la Universidad Rovira i Virgili en el Hospital Universitario Sant Joan de Reus (HUSJR). En ella, se destacan una multitud de datos clínicos de los distintos pacientes con cáncer oligometastásico, que han sido tratados con SABR en el servicio de Oncología Radioterápica de la misma entidad hospitalaria desde el año 2019 hasta el 2022.

Para la obtención de las tomografías se utilizó el entorno de Aria de la empresa Varian, donde se dispone de todo el historial de visitas y sesiones de tratamiento realizadas en el mismo servicio oncológico. En este entorno, mediante el ID del paciente, se accede a su información clínica, así como a las diferentes tomografías ejecutadas durante el tiempo en que el paciente fue tratado.

Es en este mismo marco donde a partir de los tacs de planificación, se descargan estas imágenes DICOM, (Digital Imaging and Communication In Medicine), que es un estándar internacional para imágenes médicas, puesto que permite el intercambio de imágenes médicas y la información relacionada de forma independiente del fabricante de equipos de imagen, para facilitar la conectividad de dispositivos y sistemas médicos [13]. Donde conviene enfatizar, es en la descarga de estos archivos de forma anónima cumpliendo con la protección de datos del hospital y el acuerdo de confidencialidad pactado.

De esta forma, contamos con la selección de un total de 93 pacientes oligometastásicos, de los cuales podemos dividir en cuatro grupos (Mama, Colorrectal, Próstata y Pulmón) según el cáncer primario o de origen de las metástasis desarrolladas a posteriori.

De este modo, de los 93 pacientes seleccionados, 8 corresponden al cáncer de Mama, 14 de Colorrectal, 26 de Próstata y los restantes 45 pacientes de Pulmón.

Un aspecto crucial reside en optimizar el rendimiento del modelo y aprovechar enteramente los datos, especialmente en los casos de pacientes con múltiples metástasis. Se concretó por abordar cada metástasis de manera individualizada ejemplificando a pacientes distintos.

Siguiendo esta especificación, para los 93 pacientes metastásicos, se obtuvieron un total de 128 metástasis, de los cuales 11 corresponden a cáncer primario de Mama, 17 de Colorrectal, 42 de Próstata y 58 de Pulmón.

Tumor Principal	Pacientes	Metástasis
Mama	8	11
CCR	14	17
Próstata	26	42
Pulmón	45	58
Total	93	128

Figura 4. Clasificación de los pacientes y sus respectivas metástasis en función del tumor principal.

Cabe destacar que las localizaciones de la mayoría de metástasis se encuentran en el hueso, a excepción de este último grupo, dónde la gran mayoría tratan de metástasis pulmonares.

No obstante, también nos podemos encontrar con metástasis ganglionares, cerebrales, hepáticas y otras. Siguiendo esta división nos encontramos con 71 metástasis óseas, 46 pulmonares, 4 ganglionares, 3 cerebrales, 1 hepática y 3 últimas clasificadas como "otros". Al disponer de una variación bastante grande de ejemplares, estos últimos grupos (ganglionares, cerebrales y hepáticas) se agrupan en el conjunto de "otros" sumando un total de 11 metástasis.

En la siguiente tabla, se encuentra descrita la clasificación en relación al número de metástasis estudiadas, con respecto a su localización y tumor primario respectivamente, así como imagen representativa sobre la base de datos descrita:

Tabla 1. Representación y clasificación de las metástasis estudiadas según su localización y tumor primario.

LOCALIZACIÓN/TUMOR PRIMARIO	MAMA	CCR	PRÓSTATA	PULMÓN	TOTAL
ÓSEA	11	7	38	15	71
PULMONAR	-	6	-	40	46
OTROS	-	4	4	3	11
TOTAL:	11	17	42	58	128

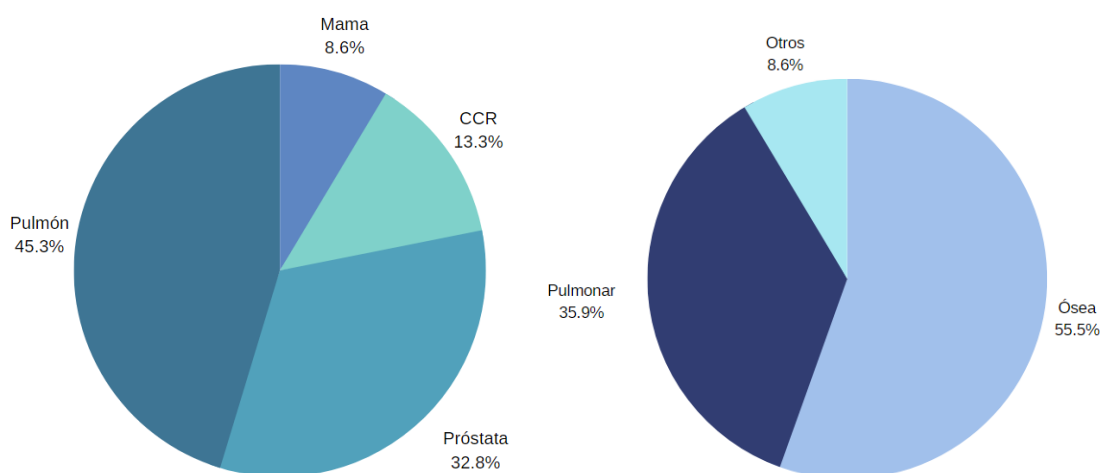


Figura 5. Diagramas circulares de la clasificación de la base de datos utilizada en base al tumor primario y la localización metastásica respectivamente. Diseño creado con *Canva*.

Por otro lado, siguiendo la base de datos y en relación a la respuesta, nos encontramos con que, de estas 128 metástasis estudiadas, 32 desarrollaron una respuesta completa, 34 respuesta parcial, 36 respuesta estable y 26 una respuesta progresiva. A continuación, se muestran una serie de tablas con la información representada:

Tabla 2. Representación y clasificación de las respuestas metastásicas según su localización.

LOCALIZACIÓN/ RESPUESTA	COMPLETA	PARCIAL	ESTABLE	PROGRESIVA	TOTAL
ÓSEA	25	18	14	14	71
PULMONAR	3	15	18	10	46
OTROS	4	1	4	2	11
TOTAL:	32	34	36	26	128

Tabla 3. Representación y clasificación de las respuestas metastásicas según el tumor primario.

TUMOR PRIMARIO/RESP UESTA	COMPLETA	PARCIAL	ESTABLE	PROGRESIVA	TOTAL
MAMA	3	5	3	-	11
CCR	1	2	4	10	17
PRÓSTATA	21	8	6	7	42
PULMÓN	7	19	23	9	58
TOTAL:	32	34	36	26	128

3.2.1 Descripción clínica

Siguiendo con la descripción de la base de datos, se emplearon las siguientes variables clínicas:

- Variables de interés común:
 - Edad.
 - Sexo.
 - Tabaquismo.
 - Índice de Masa Corporal (IMC): Peso y estatura.
- Variables de características médicas:
 - Tumor Primario.
 - Grado histológico.
 - T: Tamaño y extensión del tumor.
 - N: Número de ganglios linfáticos cercanos cancerosos.
 - M: Metástasis distante.
 - Localización metastásica.
 - AP: Anatomía patológica.
 - Dosis total de radiación (Gy)
 - Número de sesiones totales
- Variables secundarias:
 - Tratamientos previos a SABR: Quimioterapia (QT), Hormonoterapia (HT).
 - Estado Actual.
 - Seguimiento: seguimiento en meses durante y después del proceso radioterapéutico.

- Diseminación metastásica: Metacrónico (propagación del tumor posterior al diagnóstico inicial) o Sincrónico (propagación en etapa temprana al diagnóstico inicial).
- Variables principales:
 - Respuesta metastásica.

A continuación, se describe detalladamente en las siguientes tablas, una descripción más precisa de la muestra de estudio:

Tabla del análisis de estudio de los pacientes tratados

Total N= 93					
<i>Interés común</i>					
Edad	Mean (SD)	62.2 (11.7)	Tabaquismo	SI	43 (46.3%)
				NO	50 (53.7%)
Sexo	M	69 (74.2%)	IMC	Mean (SD)	25.8 (3.9)
	F	24 (25.8%)			
<i>Secundarias</i>					
QT			Estado Actual	Vivo sin enfermedad	15 (16.1%)
				NO	64 (68.8%)
HT				Vivo con progresión	15 (16.2%)
				NO	24 (25.8%)
Seguimiento	Mean (SD)	14.5 (12.8)	M1 DX	SI = Sincrónico	33 (35.5%)
				NO = Metacrónico	60 (64.5%)

Tabla 4. Muestra de los pacientes oligometastásicos estudiados. Por lo que a las siglas se refiere QT hace referencia al tratamiento pre-SABR de quimioterapia; HT: tratamiento pre-SABR de hormonoterapia; M1 DX: Diseminación metastásica.

Tabla del análisis de estudio del tumor y cáncer metastásico tratados

Total					
N= 128					

Características médicas

Tumor Primario	Mama	11 (8.6%)	N	NX	13 (10.2%)	
	Colorrectal	17 (13.3%)		N0	66 (51.6%)	
	Próstata	42 (32.8%)		N1	29 (22.6%)	
	Pulmón	58 (45.3%)		N2	16 (12.5%)	
Grado	Grado I	18 (14.1%)	M	N3	4 (3.1%)	
	Grado II	66 (51.5%)		MX	25 (19.5%)	
	Grado III	38 (29.7%)		M0	14 (11%)	
	Grado IV	6 (4.7%)		M1	89 (69.5%)	
T	TX	8 (6.3%)	Localización	Ósea	71 (57.8%)	
	T1	38 (29.6%)		Pulmonar	46 (36%)	
	T2	30 (23.4%)		O T R O S	Cerebral	3 (2.34%)
	T3	39 (30.5%)			Hepática	1 (0.8%)
	T4	13 (10.2%)			Ganglionar	4 (3.12%)
					Otra	3 (2.34%)
DT [Gy]	Mean (SD)	35.6 (12.5)	Sesiones Totales	Mean (SD)	4.1 (3.1)	

Principales

Respuesta Metastásica	Completa	32 (25%)	Estable	36 (28.2%)
	Parcial	34 (26.5%)	Progresiva	26 (20.3%)

Tabla 5. Muestra de las metástasis tratadas con SABR y utilizadas para el presente estudio. Por lo que a las siglas se refiere T hace referencia al tamaño y extensor del tumor principal, de tal forma TX: no es posible medir el tumor primario; T1, T2, T3, T4: indica el tamaño o la extensión del tumor; N: número de ganglios linfáticos cercanos cancerosos; NX: no es posible medir el cáncer en los ganglios linfáticos cercanos; N0: no hay cáncer en los ganglios linfáticos cercanos; N1, N2, N3: número y ubicación de los ganglios linfáticos con presencia de cáncer, a mayor número, se refleja una mayor cantidad de ganglios cancerosos; M: presencia de metástasis distante; MX: no es posible medir la metástasis; M0: cáncer no diseminado; M1: cáncer diseminado a otras partes del cuerpo; DT: dosis total de radiación [Gy].

3.3 Extracción de características

Para la extracción de las características radiómicas se utilizó el software o programa "3DSlicer" caracterizado por ser una de las mejores herramientas para la visualización y procesamiento de imágenes médicas con un enfoque en aplicaciones clínicas y biomédicas, permitiendo la extracción de cientos de segmentos por imagen [14].

En la siguiente sección se muestran una serie de imágenes utilizadas en el programa mencionado, como ejemplo de las distintas metástasis tratadas por SABR en el servicio de Oncología Radioterápica del hospital Sant Joan de Reus:

Paciente ID_02:

Paciente con cáncer de mama como tumor primario que dispone de cuatro metástasis óseas bien diferenciadas.

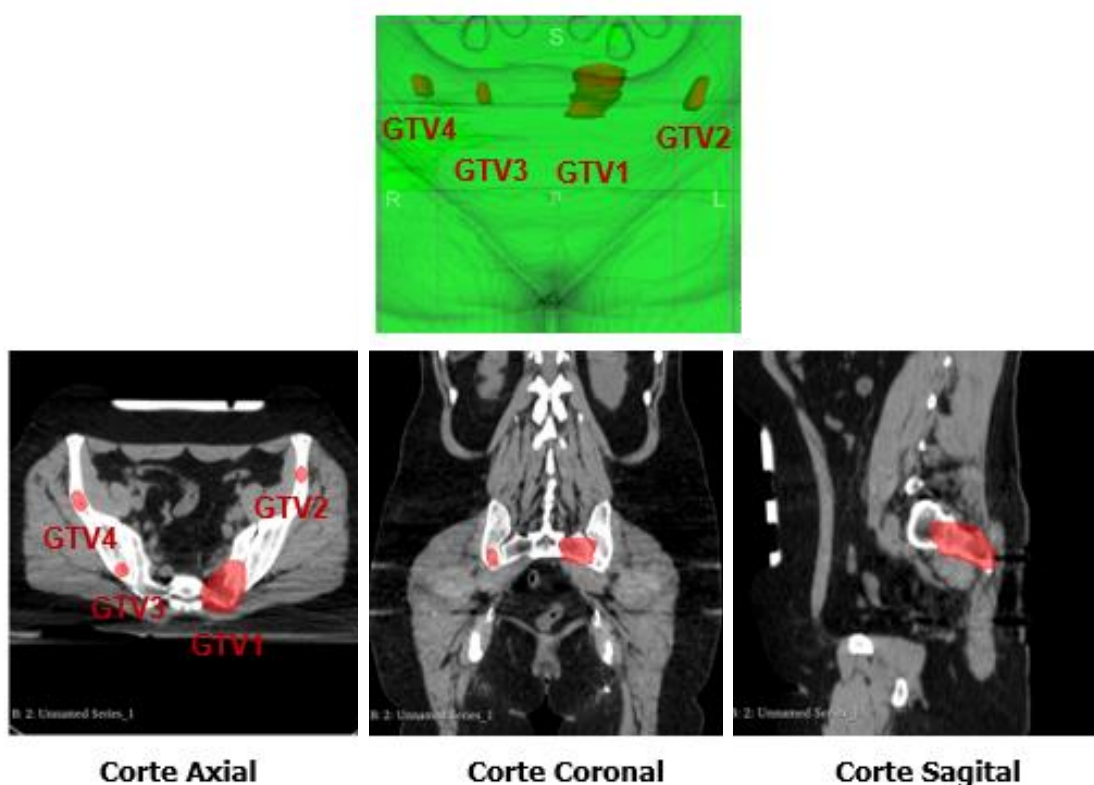


Figura 6. Imagen TAC y sus respectivos cortes axial, coronal y sagital de un paciente de mama con cuatro metástasis óseas a la altura de la cadera representadas por el volumen de GTV correspondiente. Imágenes extraídas por el programa Slicer 3D.

Paciente ID_05:

Paciente que experimenta cáncer de próstata con metástasis ósea.

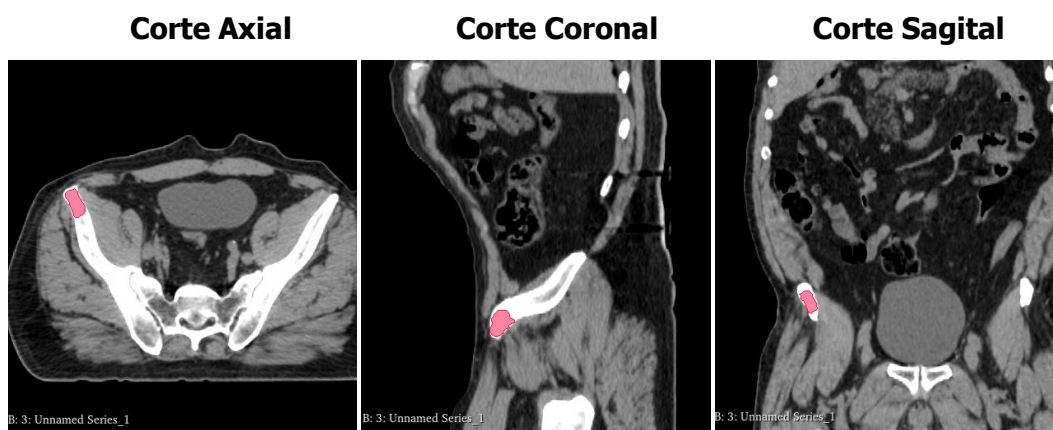


Figura 7. Imagen TAC y sus respectivos cortes axial, coronal y sagital de un paciente de próstata con metástasis ósea representada por el volumen de GTV correspondiente. Imágenes extraídas por el programa Slicer 3D.

Paciente ID_35:

Paciente enfermo de cáncer colorrectal con metástasis ósea sobre la zona baja del hombro.

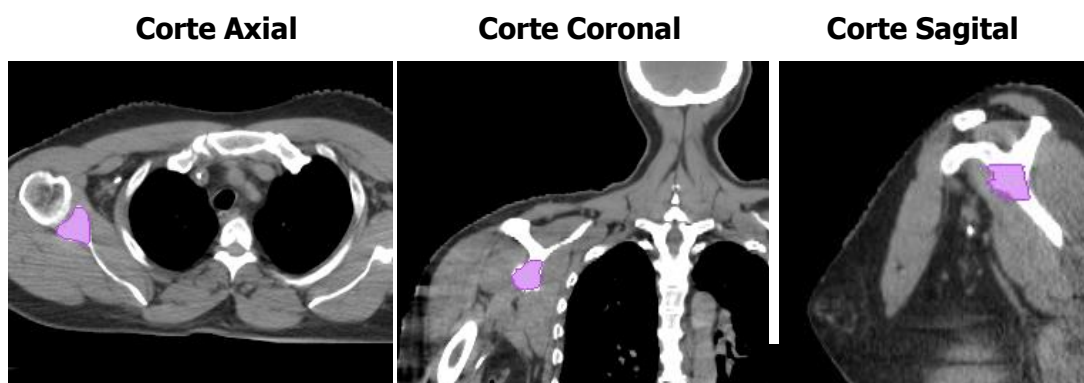


Figura 8. Imagen TAC y sus respectivos cortes axial, coronal y sagital de un paciente de cáncer colorrectal con metástasis ósea representada por el volumen de GTV correspondiente. Imágenes extraídas con Slicer 3D.

Paciente ID_39:

Paciente con tumor primario pulmonar que sufre de una metástasis ósea sobre la costilla inferior del pulmón derecho y otra pequeña metástasis pulmonar en la zona superior izquierda.

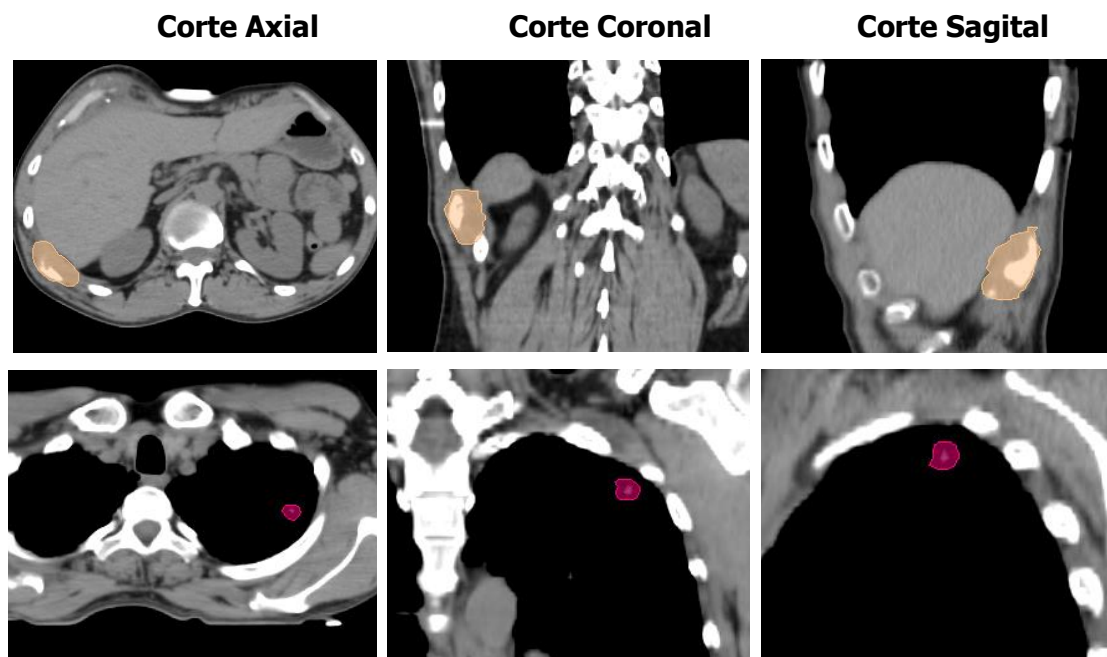


Figura 9. Imagen TAC de un paciente de cáncer pulmonar. En la zona superior se representan sus respectivos cortes axial, coronal y sagital de la metástasis ósea, y en la parte inferior los de la metástasis pulmonar, ambas representadas por el volumen de GTV correspondiente. Imágenes extraídas con Slicer 3D.

Cabe mencionar que, para finalmente obtener las características radiómicas de dichas metástasis, es necesario la instalación de las extensiones *SlicerRT* que nos permite delimitar automáticamente los volúmenes de las imágenes DICOM descargadas y *SlicerRadiomics* que permite extraer las características del volumen seleccionado en una tabla en formato CSV.

De esta forma, se pudo obtener un total de 130 características radiómicas, que es el máximo que nos permite este software, divididas en 8 grupos [15]:

a) First Order Features.

Las características de Primer Orden describen la distribución de las intensidades de cada píxel volumétrico (o vóxel) dentro de la región de la imagen definida por la máscara, es decir, las propiedades estadísticas de cada unidad cúbica que componen la región del tumor. Sus variables son:

- Energy
- Total Energy
- Entropy
- Minimum
- 10th percentile
- 90th percentile
- Maximum
- Mean
- Median
- Interquartile Range
- Range
- Mean Absolute Deviation (MAD)
- Robust Mean Absolute Deviation (rMAD)
- Root Mean Squared (RMS)
- Uniformity
- Skewness
- Kurtosis
- Variance

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

Se tratan de características de la matriz de coocurrencia de nivel de grises, calculan la frecuencia en que se producen pares de píxeles específicos en una orientación espacial específica y miden la textura del tumor, es decir, cómo varía la intensidad en diferentes partes del tumor [15]. Obtenemos los siguientes parámetros:

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

c) Gray Level Dependence Matrix (GLDM) Features.

Cuantifica las dependencias del nivel de gris en una imagen. Una dependencia de nivel de gris se define como el número de vóxeles conectados dentro una distancia determinada que depende del píxel volumétrico central [15]. Se dividen en:

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

También llamada matriz de longitud de ejecución de nivel de gris, cuantifica los niveles de gris de la región medida, es decir, el número de píxeles consecutivos con un mismo nivel de gris [16].

Podemos ver las siguientes variables:

- Short Run Emphasis (SRE)
- Long Run Emphasis (LRE)
- Run Percentage (RP)
- Run Variance (RV)
- Run Entropy (RE)
- Run Length Non-Uniformity (RLN)
- Run Length Non-Uniformity Normalized (RLNN)
- Gray Level Non-Uniformity (GLN)
- Gray Level Non-Uniformity Normalized (GLNN)
- Gray Level Variance (GLV)
- 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.

Esta matriz cuantifica las zonas de nivel de gris, definidas como el número de píxeles volumétricos conectados que contienen la misma intensidad de nivel de gris. Miden la homogeneidad, la uniformidad y otras propiedades texturales del tumor en diferentes escalas.

Podemos destacar:

- Small Area Emphasis (SAE)
- Large Area Emphasis (LAE)
- Gray Level Variance (GLV)
- Zone Percentage (ZP)
- Zone Variance (ZV)
- 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)
- 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.

Una matriz de diferencia de tonos de grises, cuantifica la relación entre un valor de gris determinado y el valor promedio de sus alrededores dentro de una distancia ' δ ' [17]. Las características que se obtienen son:

- Coarseness
- Contrast
- Busyness
- Complexity
- Strength

g) Shape 2D Features.

Estas características son independientes de la distribución de intensidad del nivel de gris en el ROI; solo se calculan en las imágenes y máscaras no derivadas. Estas características describen el tamaño y la forma bidimensional en toda el área de la imagen.

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

h) Shape 3D Features.

De igual forma que el anterior grupo, estas características son independientes del ROI y se calculan en la toda el área de la imagen [17]. Describen el tamaño, como el diámetro o el área de superficie del tumor y la forma tridimensional con las siguientes características:

- | | |
|--------------------------------|--------------------------------|
| - 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 | - Minor Axis Length |
| - Flatness | - Least Axis Length |

De esta manera, tras un primer filtrado manual como parte del preprocesado de datos, se eliminaron aquellas características de tipo diagnóstico; que incluyen variables propias del programa especificado (*Versions, Configuration, Image-Original...*) que son de poco uso y valor informativo. Finalmente, se conservaron las restantes 107 características radiómicas de tipo *original*, con las que se empezaron a desarrollar los modelos de reducción y predicción con inteligencia artificial.

3.4 Reducción y selección de características

La reducción y selección de características, previo a la creación del modelo predictivo, es un paso necesario e indispensable cuando hablamos de modelos de inteligencia artificial. Si se utilizaran todas las variables obtenidas, el modelo tendría dimensiones demasiado grandes de variables características y sería demasiado complejo, lo que daría como resultado un ajuste excesivo y poco generalizable [18].

Existen tres diferentes métodos de selección de variables [19]:

- **Filter Method** (Método de filtro): selecciona las características de mayor peso a partir de medidas estadísticas como la correlación, ganancia de información o la prueba estadística como el chi-cuadrado.
- **Wrapper Method** (Método de envoltura): selecciona el mejor subconjunto de variables con algoritmos de aprendizaje específico. Es un método computacionalmente costoso debido a la evaluación repetida del modelo con diferentes subconjuntos de datos, por esta razón es recomendable ser utilizado en problemas con un conjunto de datos pequeño.
- **Embedded Method** (Método incrustado): reduce el conjunto de datos a las características de mayor importancia a la vez que realiza el entrenamiento del modelo.

La programación desarrollada fue llevada a cabo a través de la versión 3.10.9 de Python y a partir de la plataforma de software de *Jupyter Notebook*, tanto para la selección de características como para la creación de modelos predictivos. Entre las librerías utilizadas, se pueden destacar *Numpy, Pandas* y *Scikit-learn* o *Sklearn*.

Finalmente, se utilizaron distintos métodos de selección, entre ellos, *Random Forest* (RF) como método de filtrado, y otros dos métodos de envoltura: mRMR (*Minimum Redudancy and Maximum Relevance*) y Relief.

3.4.1 Random Forest (RF)

Los bosques aleatorios, del inglés *random forest*, se forman combinando diferentes árboles de decisión independientes, y se producen divisiones aleatorias de características donde cada árbol constituye a una predicción. Actualmente, la librería de *sci-kit learn* ofrece dos maneras de algoritmos de selección de variables [20] (1) *Mean Decrease Impurity* (MDI); que mide la diferencia de impureza o la ganancia de información en cada árbol de decisión, (2) *Mean Decrease Accuracy* (MDA); de forma similar, pero en este caso mide la importancia de las variables según la precisión del modelo.

Para MDI se hizo uso del atributo "feature_importances_", el cual indica la importancia de las variables que se calculan como la media y la desviación estándar de la acumulación de la disminución de impurezas dentro de cada árbol [21].

En el caso de MDA, se calcula la precisión global del conjunto total de variables y posteriormente, el mismo cálculo excluyendo una característica determinada. Restando ambos resultados, conseguimos la diferencia de precisión que corresponde a la precisión que aporta dicha característica, representando su importancia en relación a la variable objetivo. Al realizar este procedimiento para cada variable, conseguimos la importancia de cada característica en relación a la respuesta.

Destacar que, los parámetros óptimos fueron obtenidos por la técnica de *GridSearchCV*, que se caracteriza por encontrar los mejores datos para un modelo definiendo los distintos valores posibles dentro de una cuadrícula. Además, ambos métodos, fueron entrenados mediante un "5-fold cross validation", utilizando el 80% del *dataset* como entrenamiento y el 20% restante como prueba, es decir, cada modelo se entrena realizando validación cruzada durante 5 iteraciones independientes. Al realizar múltiples iteraciones, se asegura que cada dato tenga la oportunidad de formar parte tanto del conjunto de entrenamiento como del conjunto de prueba.

Finalmente, siguiendo esta especificación, las variables radiómicas de mayor peso fueron seleccionadas calculando la media total de los puntajes individuales conseguidos en cada una de las cinco iteraciones para ambos métodos.

3.4.2 Minimum Redundancy and Maximum Relevance (mRMR)

El método mRMR busca identificar el subconjunto de datos óptimos, que maximice la relevancia con la variable objetivo (la respuesta en este caso) y minimice la redundancia de la misma con las características determinadas.

La relevancia se puede medir utilizando métricas estadísticas como la correlación, información mutua o la ganancia de información. Por otro lado, la redundancia, se refiere a la relación entre dos características, por lo que, si dos características son similares, tendrán una redundancia mayor. Un modelo con alta redundancia podría llevar a obtener resultados inconcluyentes y poco interpretables o a un modelo sobre ajustado [22].

Se utilizó el paquete *mrmr* de *GitHub* [23], el cual se caracteriza por disponer de dos métodos mRMR en función del problema a tratar. Por un lado, "mrmr_regression" para problemas donde la variable objetivo es numérica, y por el otro lado "mrmr_classif", que fue el utilizado en el presente trabajo y se caracteriza para problemas de categorización.

3.4.3 Relief

Este método se caracteriza por ser utilizado comúnmente para la selección de características en problemas de clasificación. Su idea central se basa en, comparar las instancias cercanas con las de la variable objetivo, es decir, a medida que el algoritmo recorre

las instancias de los datos, actualiza los pesos de las distintas variables en función de su capacidad para distinguir entre las clases [24].

En este contexto, una instancia se refiere a una agrupación de datos. Poniendo un ejemplo, en una base de datos médica, cada instancia correspondería a un paciente, incluyendo sus datos clínicos como edad, sexo, altura, etc.

De las distintas librerías que incluyen este método, se utilizó el algoritmo de "ReliefF", que se caracteriza por ser una extensión del método Relief permitiendo tratar con problemas de clasificación multiclase.

3.5 Construcción de modelos predictivos

Existen una multitud de modelos de inteligencia artificial a partir de algoritmos de aprendizaje automático. Escoger el modelo que más se ajuste al problema especificado es una de las tareas más importantes antes de su desarrollo, en este trabajo se construyeron tres modelos con capacidad predictiva para el problema de clasificación mencionado.

Primero de todo, destacar que se dividió todo el conjunto de datos en tres subconjuntos bien diferenciados para poder entrenar y validar los modelos como corresponden, quedando así la siguiente estructura de datos:

- **Conjunto de entrenamiento:** En este proyecto, corresponde al 80% de datos utilizados como conjunto encargado del entrenamiento del modelo.
- **Conjunto de prueba:** Corresponde al 20% de datos utilizados como conjunto encargado de la prueba o test. Con este conjunto se adecuarán los parámetros óptimos de los modelos entrenados.
- **Conjunto de validación:** Se adecua al conjunto encargado de validar el modelo correspondiente, a partir de datos no entrenados ni vistos previamente. Este conjunto, indicaría una utilidad real del modelo.

80% 20%

Por otro lado, el desarrollo de un modelo predictivo consta de los siguientes pasos:

1. **Preprocesamiento de datos:** consiste en transformar y limpiar los datos para el modelado posterior. En el modelo desarrollado se realizó una estandarización o normalización de datos mediante la función *StandardScaler* permitiendo obtener todos los datos en una misma escala eliminando cualquier sesgo o efecto debido a las diferencias de las unidades de medida.

Con respecto al preprocesamiento de datos clínicos, se utilizó una codificación binaria para aquellas variables categóricas binarias, como el tabaquismo, el sexo y otras. En cuanto a las demás variables categóricas no binarias, como es el caso de la respuesta metastásica, se utilizó una codificación multiclase representada con números enteros.

2. **Selección del modelo:** elegir el tipo de modelo óptimo según el problema que se intenta resolver, como un modelo de regresión lineal o un modelo de árboles de decisiones.
3. **Entrenamiento del modelo:** ajustar los parámetros del modelo utilizando los datos del conjunto de entrenamiento.
4. **Validación del modelo:** evaluar el rendimiento final del modelo mediante el conjunto de validación, donde los datos no fueron entrenados durante el entrenamiento del modelo.

- 5. Optimización del modelo:** finalmente ajustar el modelo y sus parámetros para mejorar su rendimiento.

De esta forma, se desarrollaron tres modelos de predicción basados en modelos supervisados: un modelo basado en bosques aleatorios o árboles de decisión (RF), otro modelo basado en vectores de soporte (SVM), y por último un modelo de red neuronal artificial (ANN).

3.5.1 Random Forest

Como bien se ha introducido anteriormente, el método de bosques aleatorios, funciona a través de la creación de distintos árboles de decisión. De esta forma, puede actuar tanto en problemas de selección y reducción de características, así como método de predicción.

De igual forma que para la selección de variables radiómicas, se programó el método a partir de los parámetros óptimos obtenidos por la misma función de *GridSearchCV*. Con estas características, el modelo estaba formado por 138 árboles de decisión indicada por la variable de "n_estimators", un factor de aleatoriedad de 42 mediante la variable "random_state", el cual debe permanecer constante para evitar resultados inconcluyentes, ya que permite que cada vez que se ejecute el modelo se obtengan los mismos resultados en términos de división de datos, "min_samples_leaf" de 1 y "min_samples_split" de 5, que indican el número de muestras mínimas y requeridas por nodo, y por último, una profundidad máxima de 5 que ayuda a controlar el sobre ajuste y mejorar la generalización del modelo.

Recalcar que, el entrenamiento del modelo se realizó siguiendo una validación cruzada de 5 iteraciones o un "5-cross validation", esto debido a la escasez de datos finales y al intento de reducir un posible sobreajuste en el modelo.

3.5.2 Support Vector Machine

Las máquinas de soporte de vectores o *Support Vector Machines* (SVM), son métodos que utilizan algoritmos de aprendizaje supervisado, útil tanto en problemas de regresión como en clasificación [25]. El objetivo principal es encontrar la mejor línea o hiperplano que divida mejor el conjunto de datos. Originalmente se creó para tratar problemas de clasificación binaria, pero actualmente se utilizan técnicas de *one-vs-one* o *one-vs-rest* para tratar problemas de clasificación multiclase como es el caso.

En el modelo elaborado se utiliza la técnica de *one-vs-one* para abordar con el problema de la clasificación multiclase. Además, de igual forma que con el método anterior, se entrenó el modelo siguiendo una validación cruzada de 5 iteraciones.

3.5.3 Red Neuronal Artificial

Una red neuronal artificial, también llamada ANN, por sus siglas en inglés *Artificial Neural Network*, es un modelo que funciona a través de diferentes capas que contienen diversas neuronas o nodos, las cuales memorizan en base a unos datos iniciales que se van ajustando a medida que van aprendiendo con más datos de entrenamiento.

Una red neuronal está formada mínimamente por una capa de entrada y una capa de salida, y puede disponer de diversas capas intermedias, llamadas ocultas que proporcionan grados de libertad a la red. En el modelo desarrollado se utiliza una capa de entrada con 28 neuronas, que corresponden al conjunto de variables radiómicas y datos clínicos, una capa oculta con 36 y una capa de salida con 4 neuronas correspondiendo a los cuatro tipos de respuestas finales que se requiere predecir. Destacar también el uso de la librería de *Keras*, que proporciona una interfaz de Python para redes neuronales artificiales.

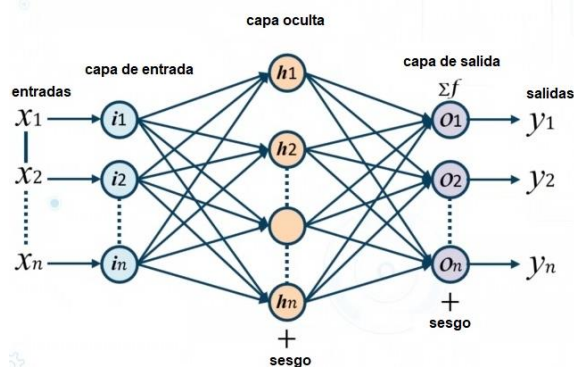


Figura 10. Esquema de una red neuronal artificial. Se puede ver representada las diferentes capas que componen la red y sus respectivas entradas y salidas [26].

En una red neuronal, los sesgos y los pesos son parámetros ajustables que determinan la influencia de cada neurona en la salida de la red. Los pesos son valores que se aprenden durante el proceso de entrenamiento de la red neuronal. Inicialmente, los pesos se establecen de forma aleatoria y luego se ajustan iterativamente para minimizar la pérdida. Por otro lado, los sesgos son otras constantes aleatorias y diferentes para cada neurona, que se suman a estas entradas ponderadas [26]. Por último, tenemos la función de activación, que es una función matemática aplicada a la suma ponderada de las entradas y los sesgos de una neurona, para obtener la salida final de la red neuronal.

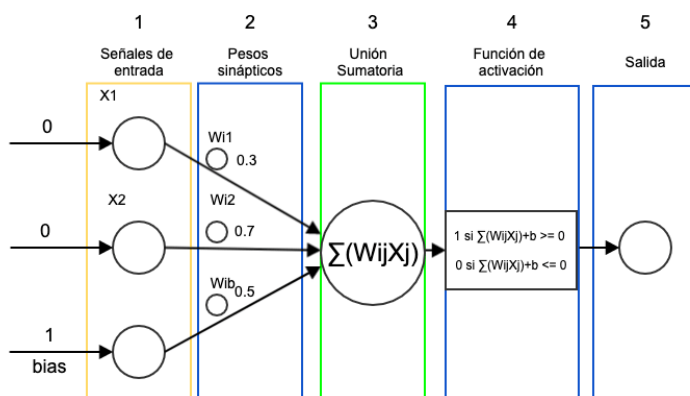


Figura 11. Estructura básica del diseño de una red neuronal de forma simplificada [28].

Existen diferentes funciones de activación, en este modelo se utiliza una función de activación de *ReLU*, por sus siglas en inglés *Rectified Linear Unit*, que se caracteriza por ser una función de transformación no lineal que permite aprender relaciones complejas de datos. Funciona activando la neurona para valores positivos mayores a 0, es decir todas aquellas entradas o *inputs* que no sean negativas. Se utiliza comúnmente en capas ocultas para ayudar a la red a capturar patrones y características más complejas en los datos de entrada. También se utilizó la función de activación *Softmax* utilizada comúnmente en la capa de salida para problemas de clasificación, ya que comprime las salidas de las neuronas en probabilidades de 0 a 1, de tal forma que la suma de las salidas sea igual a 1 [29]. Además, se emplea un optimizador *Adam* con un valor muy bajo para poder ajustar los pesos y sesgos, permitiendo que el ajuste sea lo más preciso posible.

Para el entrenamiento se emplea la técnica de *Batch mode*, donde los pesos de cada conjunto como muestra de entrada, se van ajustando y actualizando para cada ciclo de entrenamiento o *epoch*.

Asimismo, se emplea la precisión como métrica de acierto para evaluar el entrenamiento del modelo, y se utiliza la métrica de pérdida denominada *sparse_categorical_crossentropy*

diseñada en problemas de clasificación multiclase donde las etiquetas son representadas como números enteros, y no como *one-hot vectors*, lo cual es aplicable en el contexto de este estudio.

Para obtener resultados concluyentes utilizando una red neuronal es importante reducir lo máximo posible el sobre ajuste también llamado *Overfitting*. Este estado viene derivado por que el modelo se ha ajustado o ceñido demasiado a los datos de entrenamiento y por tanto no es capaz de generar nuevos datos, lo que provoca una mala generalización.

Se hallan distintas formas utilizadas con la finalidad de reducir el sobre ajuste, entre ellas se destacan:

- Aumento de datos de la base de datos original, permitiendo a la red un conjunto de entrenamiento mayor teniendo más opciones de generalizar nuevos datos.
- Simplificar el conjunto de datos, reduciendo la complejidad y dimensionalidad del modelo o reduciendo las variables y características extraídas. En este punto es donde se centran los modelos de selección de características explicados con anterioridad.
- Distintas técnicas de regularización como:
 - Reducir el tamaño de la red utilizando menos capas neuronales o minimizando el número de neuronas en cada una de ellas.
 - Regularizar los pesos utilizando funciones *drop-outs* que funcionan apagando un porcentaje indicado de neuronas en cada capa, o regularizadores *L2* que suavizan las conexiones entre las neuronas de mayor peso, para evitar un peso dominante que impida la generalización del modelo.
 - Deteniendo el entrenamiento en un punto indicado, un ejemplo es la función *early-stopping* que consiste en detener el entrenamiento de la red cuando se observa que el rendimiento deja de mejorar o comienza a empeorar [30]. En este punto, se toma el modelo almacenado en la iteración anterior como la mejor versión del modelo y se detiene el entrenamiento.

En el presente estudio, debido a la limitación de disponibilidad de un aumento de datos de entrenamiento, se optó por utilizar técnicas de regularización para mejorar el rendimiento del modelo.

Se implementó la técnica de *drop-outs* con una tasa del 0.2, lo que implica que el 20% de las neuronas se desactivan aleatoriamente en la capa oculta durante el entrenamiento. Además, se agregó una regularización *L2* en esta misma capa y se empleó la técnica de *early stopping* con una paciencia de 20 ciclos, lo que permite detener el entrenamiento en el punto óptimo antes de que la precisión comience a empeorar. Estas estrategias de regularización se aplicaron para evitar el sobreajuste y lograr un mejor rendimiento en la validación final del modelo entrenado.

4 Resultados

Se obtuvieron resultados de los distintos modelos entrenados para tres conjuntos o agrupaciones de la base de datos original. Es decir, debido a la complejidad, número de muestras y al desequilibrio entre clases, como es el número de metástasis óseas en comparación al resto, se decidió estudiar los resultados con la siguiente clasificación de datos.

1. Conjunto **GLOBAL**, caracterizado por disponer las 128 metástasis. En este conjunto se seleccionan aleatoriamente 10 metástasis óseas y 10 metástasis pulmonares como conjunto de validación.
2. Conjunto único de **METÁSTASIS ÓSEAS**, formado exclusivamente por las 71 metástasis óseas. Se utilizan 15 metástasis óseas como conjunto de validación.
3. Conjunto único de **METÁSTASIS PULMONARES**, de forma similar, solo cuenta con las 46 metástasis pulmonares descritas. Al ser el grupo con menor cantidad de datos, solo se utilizan 10 metástasis pulmonares en la validación.

4.1 Selección de variables

Después de experimentar con varios números de variables óptimas, se procedió a reducir y seleccionar hasta un total de 10 características radiómicas para cada caso. Comentar que, para el entrenamiento posterior, se añadieron las demás variables clínicas mencionadas en apartados anteriores. Ver la sección 3.2.1.

Tanto para el conjunto global como para el conjunto de metástasis óseas, en la reducción de características se obtuvieron mejores resultados mediante el método de *Random Forest* a partir de la diferencia de impureza (MDI), por el contrario, para el caso de metástasis pulmonares se acabó seleccionando las características calculadas con el método de mínima redundancia y máxima relevancia (mRMR). Ver anexo 8.2.

Así, en el primer conjunto se redujeron las siguientes 10 características radiómicas: dos de Primer Orden, una de Shape2D, una de tipo GLDM, tres de tipo GLCM y tres de tipo GLSZM:

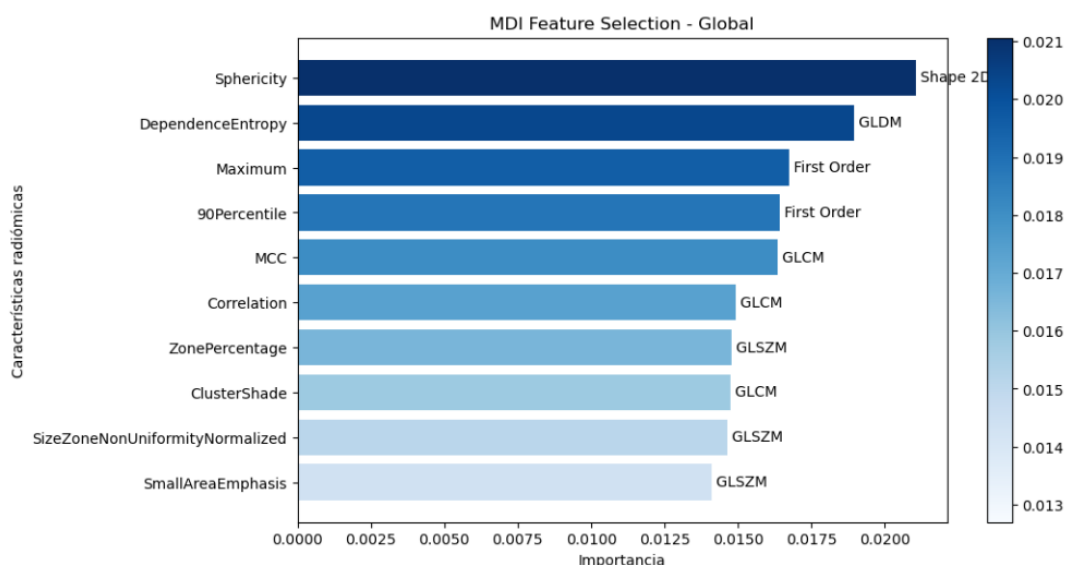


Figura 12. Importancia de las 10 características reducidas para el caso global con el método MDI.

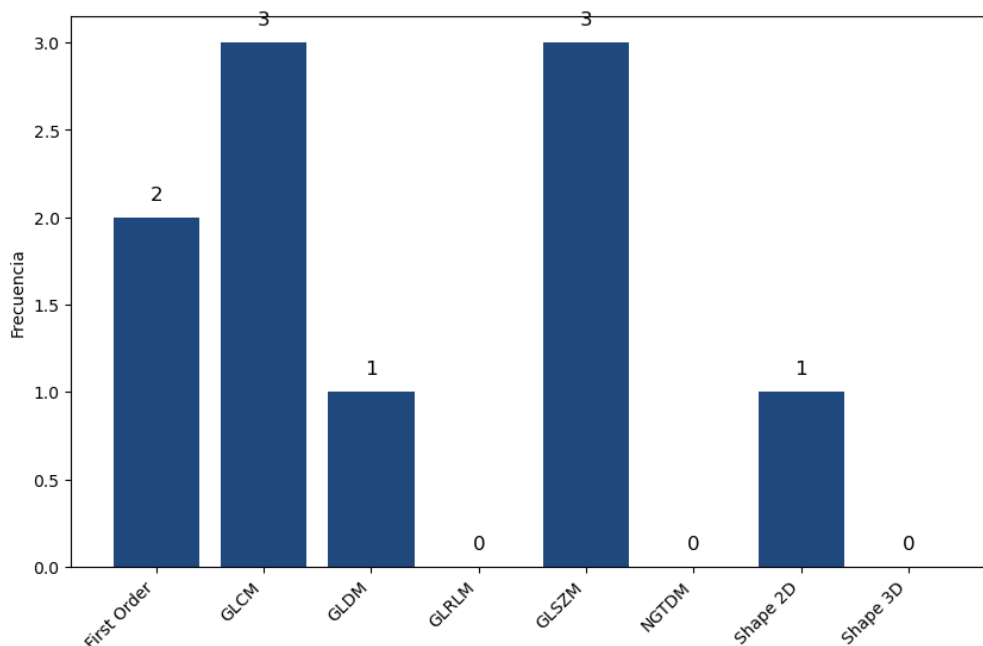


Figura 13. Representación por gráfico de barras de la frecuencia de los tipos de características reducidas para el conjunto global mediante el método de reducción MDI.

En el caso del estudio exclusivo de metástasis óseas, se eligieron las siguientes 10 características formadas por: tres de Primer Orden, una de Shape2D, dos de tipo NGTDM, y las últimas cuatro de tipo GLCM:

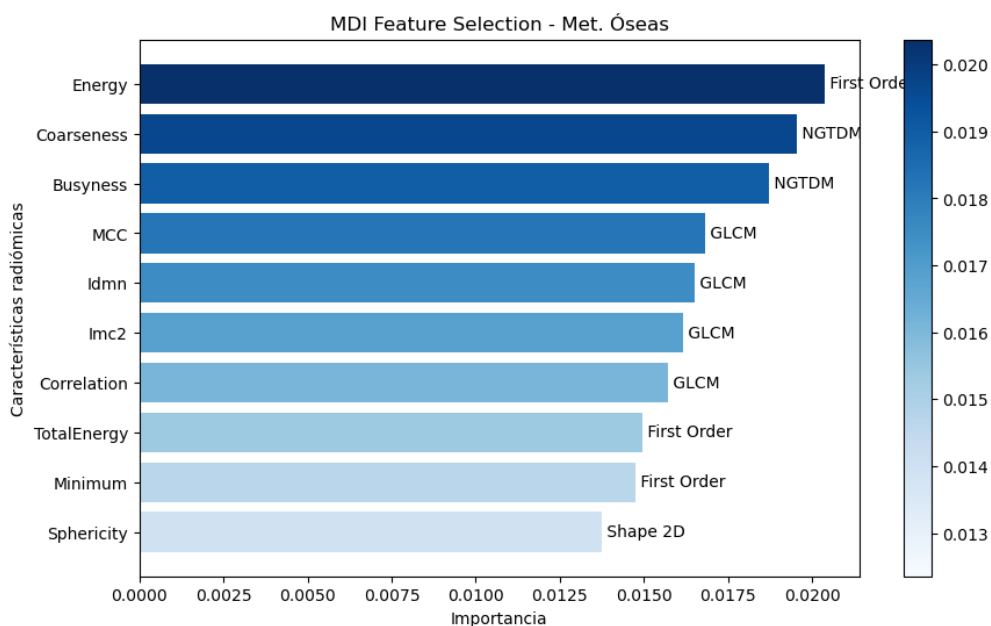


Figura 14. Importancia de las 10 características reducidas para el caso de metástasis óseas con el método de MDI.

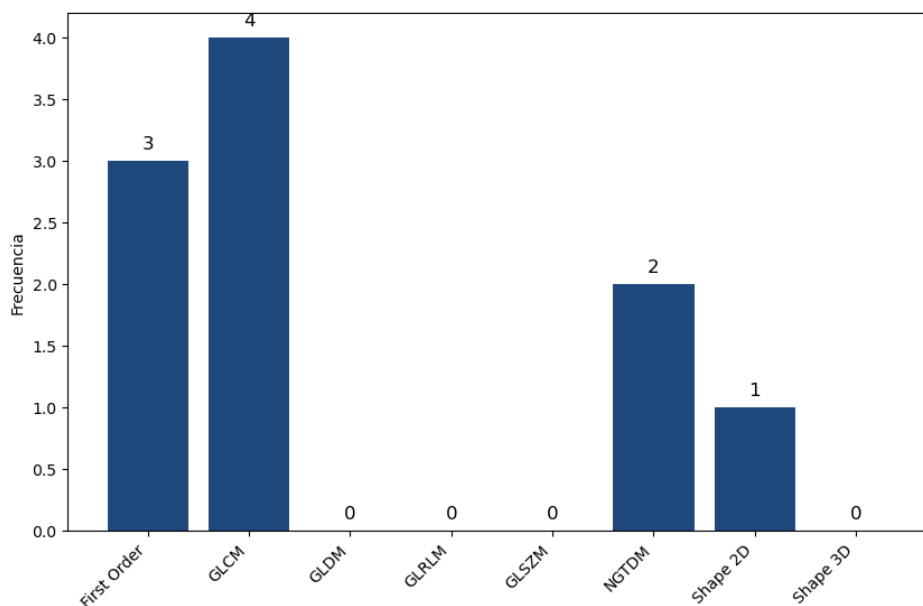


Figura 15. Gráfico de barras donde se muestra la frecuencia de cada tipo de característica obtenidas en el grupo de metástasis óseas mediante el método MDI.

Finalmente, para el caso específico de metástasis pulmonares se identificaron las siguientes 10 características radiómicas: una de Primer Orden, dos de tipo GLDM, una de tipo GLCM, una de tipo Shape2D, tres de tipo GLRLM y dos de tipo GLSZM:

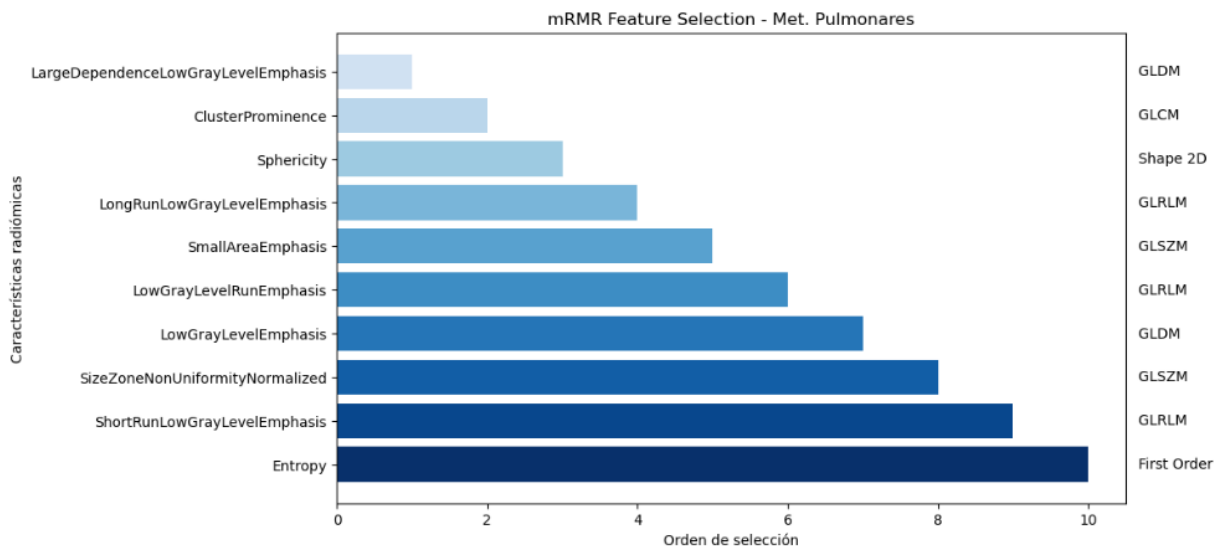


Figura 16. Orden de selección de las 10 características de mayor relevancia a través del enfoque mRMR, aplicado para el conjunto exclusivo de metástasis pulmonares. Las posiciones reflejan la jerarquía de importancia, siendo el puesto de orden 1 el más significativo.

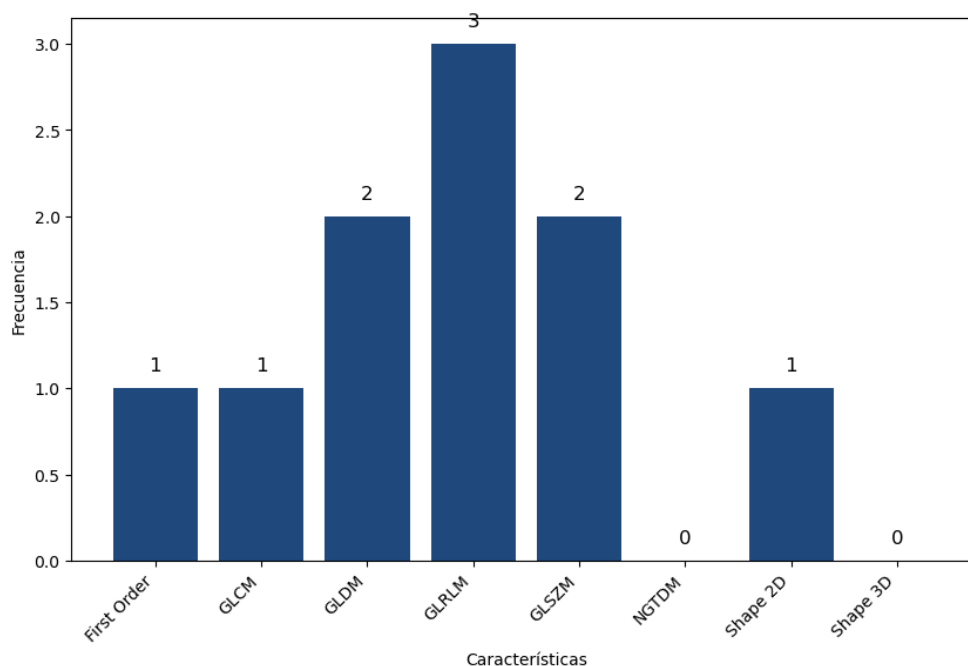


Figura 17. Gráfico de barras donde se muestra la frecuencia de cada tipo de característica mediante mRMR para el grupo de metástasis pulmonares.

4.2 Entrenamiento y validación

Para la comparación de los resultados de entrenamiento, se decidió evaluar cada modelo mediante diferentes métricas de validación como la **exactitud** (*accuracy*), que corresponde al número de aciertos sobre el total, el **F1-score**, que mide en una escala de 0 a 1, el equilibrio entre la capacidad de predecir correctamente ejemplos positivos (**Sensibilidad o Recall**) y negativos (**Precisión**), y finalmente la curva **ROC** y su área bajo la curva o **AUC**, que representa como de eficiente es el modelo para discriminar entre clases y que sus predicciones sean más confiables, midiendo la proporción de acierto a partir de la tasa de verdaderos positivos y la tasa de falsos positivos.

Especificar que al tratarse de un problema de clasificación multiclase, para la elaboración de la curva ROC-AUC, se utilizó la técnica de *one-vs-rest* o una contra el resto, la cual consiste en, calcular la ROC-AUC de cada clase de forma binarizada, frente a la combinación del resto. Poniendo un ejemplo, teniendo las clases A, B, C y D, si se busca recrear la curva ROC de la clase A, este método tratará esta clase como positiva o afirmativa, y las demás clases B, C y D, es decir, no-A, se tratarán como negativas.

En la siguiente sección se encuentran las distintas curvas ROC y sus áreas bajo la curva AUC, durante la validación para cada modelo. También se puede observar las curvas ROC-AUC medias y para cada clase, según el conjunto de datos trabajados.

- Conjunto GLOBAL:

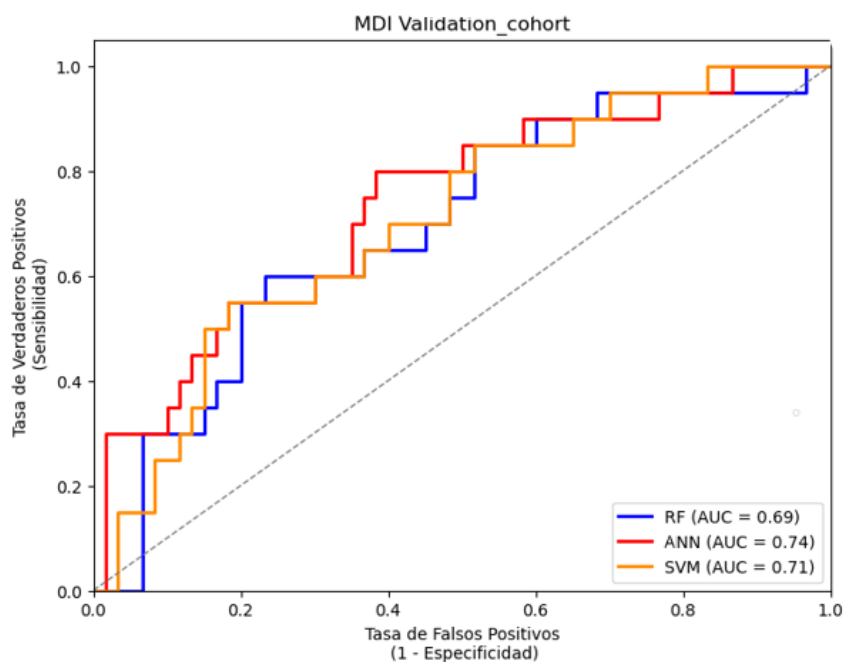


Figura 18. Curvas ROC-AUC medias del conjunto de validación de los modelos predictivos entrenados (RF, SVM y ANN) para el conjunto de datos global a partir de la reducción de características mediante RF MDI.

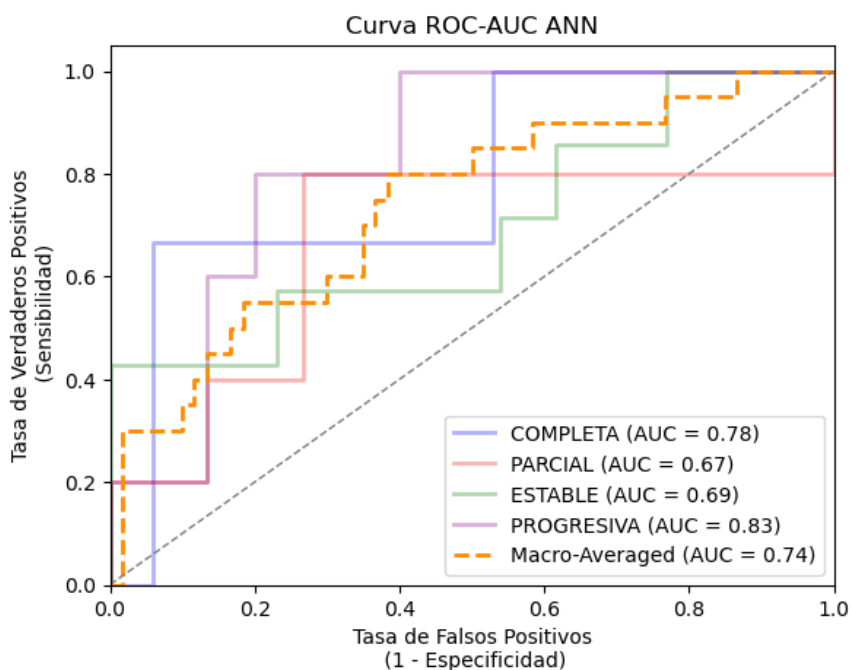


Figura 19. Curvas ROC-AUC de cada clase y media entrenadas por el modelo de red neuronal del conjunto de validación para el conjunto de datos global.

- Conjunto MET. ÓSEAS:

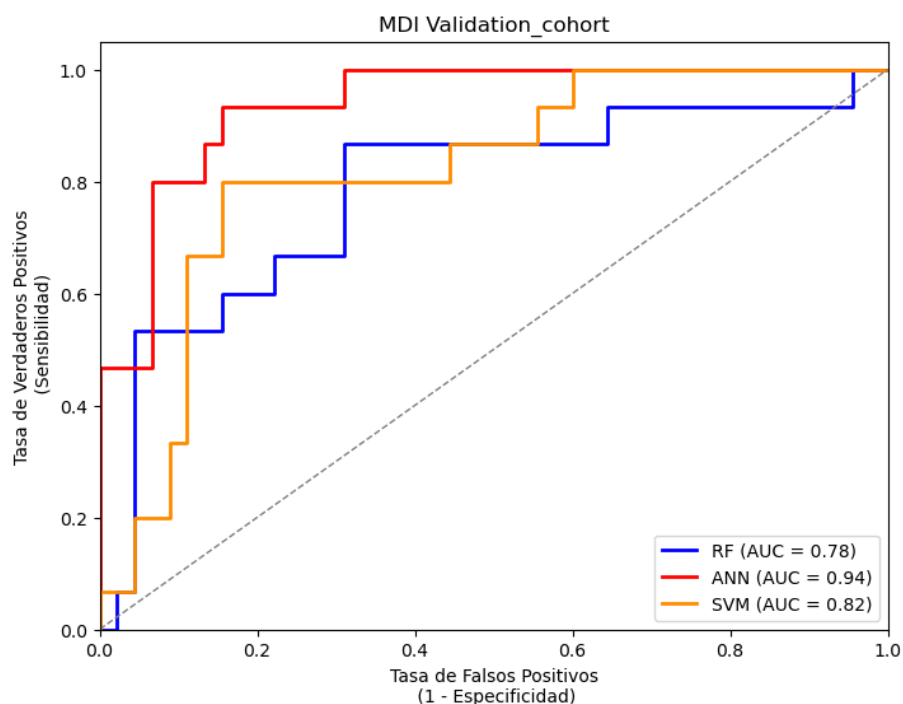


Figura 20. Curvas ROC-AUC medias del conjunto de validación de los modelos predictivos entrenados (RF, SVM y ANN) para el conjunto de datos único de metástasis óseas a partir de la reducción de características mediante RF MDI.

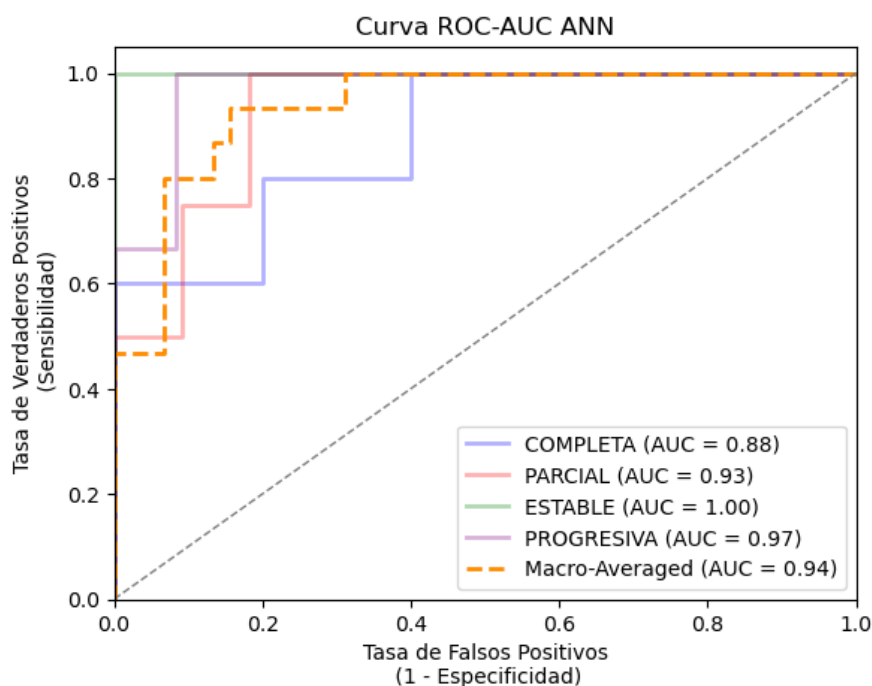


Figura 21. Curvas ROC-AUC de cada clase y media entrenadas por el modelo de red neuronal del conjunto de validación para el conjunto de datos único de metástasis óseas.

- Conjunto MET. PULMONARES:

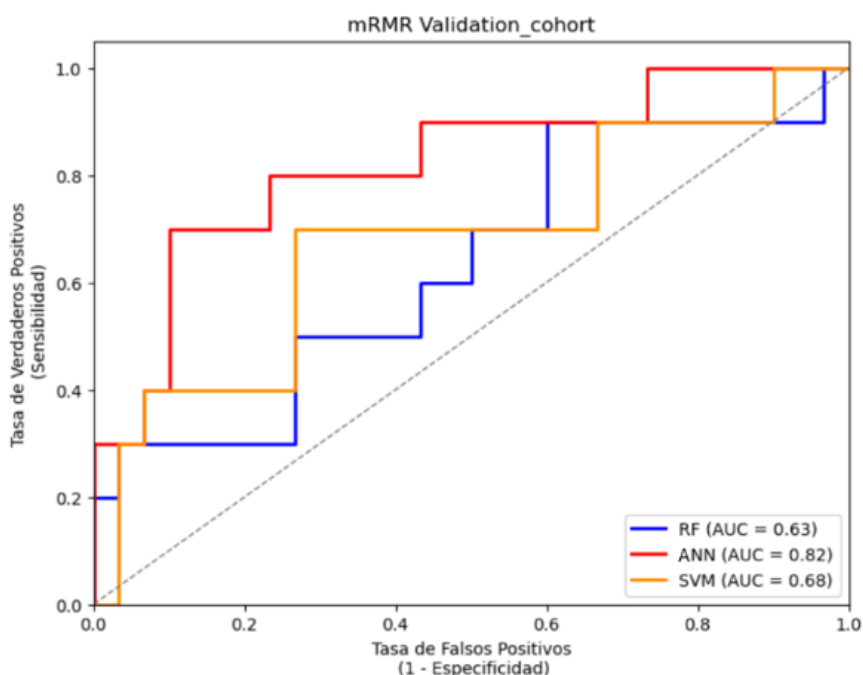


Figura 22. Curvas ROC-AUC medias del conjunto de validación de los modelos predictivos entrenados (RF, SVM y ANN) para el conjunto de datos único de metástasis pulmonares a partir de la reducción de características mediante mRMR.

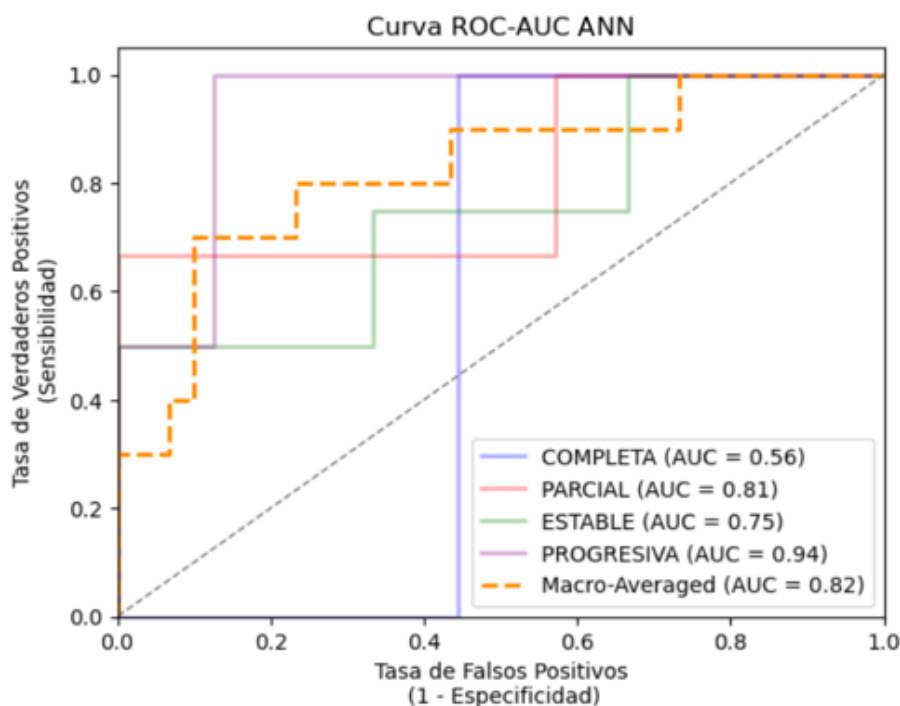


Figura 23. Curvas ROC-AUC de cada clase y media entrenadas por el modelo de red neuronal del conjunto de validación para el conjunto de datos único de metástasis pulmonares.

De esta manera, tras obtener el mejor modelo para cada conjunto, se decide estudiar el punto óptimo o el mejor *threshold* de la curva ROC para obtener la matriz de confusión y finalmente calcular los valores de sensibilidad, precisión y f1-score.

Para la determinación del punto óptimo se eligió el *threshold* de mayor f1-score, es decir, el umbral en que el f1-score es máximo.

Al tratarse de un problema de clasificación multiclase, los valores Verdaderos Positivos (TP), Verdaderos Negativos (TN), Falsos Positivos (FP) y Falsos Negativos (FN) se obtienen de la siguiente manera:

MATRIZ DE CONFUSIÓN - $MC(i, j)$ de $N \times N$

		CLASE I					CLASE N-I				
		Predicted = 1	Predicted = 2	Predicted = ...	Predicted = N-1	Predicted = N	Predicted = 1	Predicted = 2	Predicted = ...	Predicted = N-1	Predicted = N
Real = 1	TP	FN	FN	FN	FN	TN	TN	TN	FP	TN	
Real = 2	FP	TN	TN	TN	TN	TN	TN	TN	FP	TN	
Real = ...	FP	TN	...	TN	TN	TN	TN	...	FP	TN	
Real = N-1	FP	TN	TN	TN	TN	FN	FN	FN	TP	FN	
Real = N	FP	TN	TN	TN	TN	TN	TN	TN	FP	TN	

Figura 24. Matriz de Confusión para un clasificador multiclase [31].

El cálculo de la sensibilidad, precisión y f1-score se pueden ver dado por las siguientes fórmulas. Ver la ecuación (1), ecuación (2) y ecuación (3) respectivamente:

$$\text{Recall o Sensibilidad} = \frac{TP}{TP + FN} \tag{1}$$

$$\text{Precisión} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{F1score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

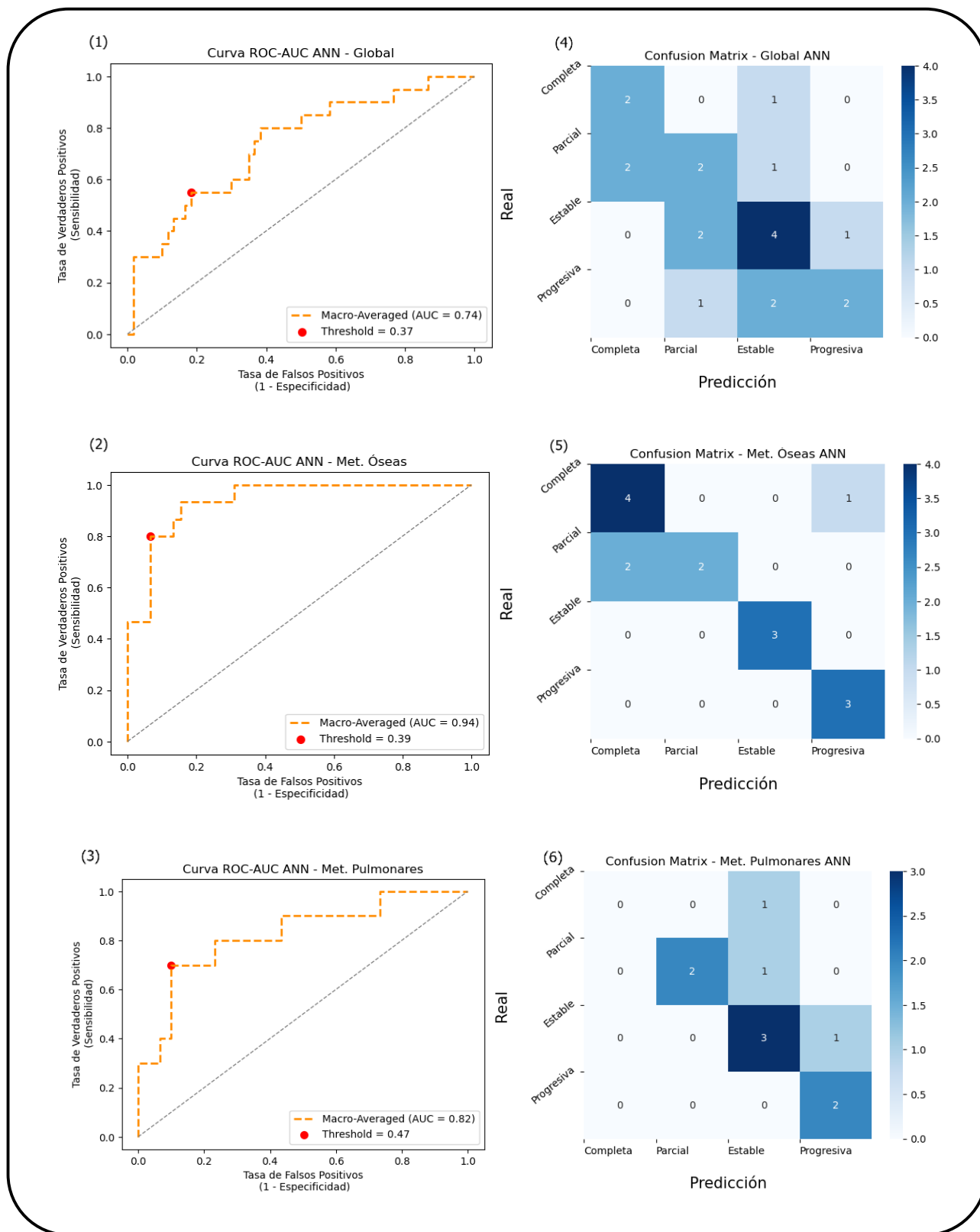


Figura 25. (1) (2) (3) Representación del punto o umbral óptimo de la curva ROC para los conjuntos Global, Metástasis Óseas y Metástasis Pulmonares respectivamente. (4) (5) (6) Matrices de confusión correspondientes utilizando el mismo umbral para los conjuntos mencionados respectivamente. En las matrices de confusión, el eje Y representa las clasificaciones de respuestas verdaderas mientras que el eje X muestra las predicciones correspondientes.

Finalmente, a continuación se muestra una tabla comparativa con los resultados de las distintas métricas de evaluación mencionadas:

Tabla 6. Métricas de evaluación de los distintos conjuntos entrenados.

ANN	<i>Accuracy</i>	<i>AUC</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-score</i>
GLOBAL	50%	0.74	0.51	0.49	0.50
MET. ÓSEAS	80%	0.94	0.82	0.85	0.81
MET. PULMONARES	69%	0.82	0.60	0.57	0.57

5 Discusión

En esta sección se discutirán los resultados obtenidos caracterizando la capacidad predictiva de los modelos y evaluando posibles aplicaciones prácticas en el ámbito clínico.

Primeramente, en relación a la selección de las características radiómicas, se han obtenido similitudes para cada conjunto en características de grupo *GLCM*, *First Order* y *Shape2D*. Estas características se relacionan por describir la textura, tamaño, forma y otras propiedades estadísticas del vóxel que delimita la zona tumoral. Se entiende, por lo tanto, que estas variables son las que tienen la capacidad de describir la metástasis radiómicamente, y por ende, tienen mayor relación con la respuesta metastásica en primer lugar. Entre este grupo de variables se pueden destacar la Energía, Esfericidad, Correlación y otras variables de medida del nivel de gris.

Al analizar los conjuntos de forma individual, se observan diferencias que sugieren que las metástasis pulmonares están mejor representadas radiómicamente en términos de respuesta cuando se utilizan variables de medición de píxeles agrupados con niveles de gris similares, específicamente representados por la clase *GLRLM*. En cambio, las metástasis óseas parecen describirse mejor por la textura del tumor, representadas por la clase *GLCM* que miden la variación de la intensidad en diferentes partes del cáncer.

En segundo lugar, siguiendo con la discusión de resultados, se puede ver una mejora clara en los modelos de predicción con el uso de la Red Neuronal Artificial. A partir de la curva ROC y su área bajo la curva AUC, se observa una notoria mejoría mediante *ANN* en los tres diferentes conjuntos entrenados. Tanto para el modelo de *RF* como para el modelo de *SVM*, se obtienen curvas ROC similares, con poca diferencia de valor AUC, siendo *SVM* un poco más favorable que *RF*, pero francamente inferiores en cuanto a resultados en comparación con la red neuronal.

Observando los diferentes conjuntos individualmente, es claro que al entrenar los datos de forma conjunta se presentan dificultades para conseguir predicciones correctas. Estudiando las metástasis por separado en función de su localización metastásica, según sea ósea o pulmonar, la mejora en la predicción de resultados aumenta considerablemente consiguiendo resultados notablemente superiores para el conjunto exclusivo de metástasis óseas llegando a obtener una exactitud del 80% y un F1-score del 0.81 para la validación del modelo.

Aun cuando la exactitud del conjunto único de metástasis pulmonares es de casi el 70%, se puede ver que tanto su sensibilidad como F1-score son relativamente bajos, por lo que no se considera que el modelo sea capaz, con los datos entrenados, de predecir correctamente la respuesta para las metástasis pulmonares. La razón de obtener una mejor predicción para metástasis óseas podría deberse al desequilibrio de datos, teniendo mayor número de localizaciones metastásicas en hueso para el entrenamiento que en el caso del cáncer metastásico pulmonar.

Centrándose en el conjunto de metástasis óseas, como se ilustra en la figura 25, se ha identificado el punto óptimo de la curva ROC donde se obtiene el mayor valor posible de F1-score del modelo. Según esta evaluación, el punto óptimo seleccionado es aquel en que no solo permite al modelo predecir con precisión, sino también diferenciar de manera justa entre las cuatro clasificaciones de respuesta dadas. A nivel de este estudio, este punto se considera óptimo. Sin embargo, en un contexto clínico, la elección de este punto podría plantar dudas sobre el acierto en la asistencia al tratamiento clínico.

Teniendo en consideración lo mencionado, si se considera el punto óptimo aquel en que la sensibilidad es alta, siempre que la especificidad no sea muy baja, este umbral tendría un mayor valor en términos de toma de decisiones médicas. El hecho de obtener una sensibilidad menor podría llevar a tener consecuencias significativas en las decisiones clínicas. Por ejemplo,

programar un tratamiento radioterapéutico basado en una predicción errónea o con falsos negativos, en la respuesta al procedimiento, supondría acarrear efectos adversos en lugar de los resultados deseados.

De esta forma, observando las curvas ROC para cada una de las respuestas dadas en este mismo conjunto, tal como se muestra en la figura 21, se contempla una sensibilidad del 100% para la respuesta estable y progresiva, con una especificidad de aproximadamente el 90% para la respuesta progresiva. Estos puntos, indican que el modelo es capaz de predecir casi con certeza una respuesta negativa al tratamiento. En este contexto, se considera como respuesta negativa aquella en que la metástasis no prospera o empeora.

Por lo tanto, las posibles aplicaciones prácticas derivadas del modelo utilizado en la red neuronal podrían incluir la mejora en la toma de decisiones clínicas para el tratamiento de SABR en pacientes oligometastásicos. Esto permitiría la obtención de predicciones de la respuesta metastásica y proporcionaría al médico oncólogo una evaluación más objetiva durante la fase de planificación del tratamiento.

En tercer lugar, en comparación a los recientes estudios realizados [9] [10] durante los últimos años y que han servido como ejemplo y parte de guía a lo largo de este trabajo, se consigue corroborar el uso de la radiómica para mejorar el seguimiento de las metástasis tratadas con radioterapia SABR. Si bien, el punto principal de estos estudios resalta en la importancia del uso de modelos combinados de características radiómicas junto a variables clínicas, en el presente trabajo se intenta ir más allá y lograr, aunque a priori y a falta de más pruebas de estudio con mayores datos de muestra, un modelo capaz de predecir la respuesta de las metástasis con localizaciones óseas tratadas con SABR.

Por otro lado, aun cuando los hallazgos presentados anteriormente ofrecen una visión clara del uso de la radiómica conjuntamente a modelos de aprendizaje automático para la predicción de la respuesta metastásica en pacientes oligometastásicos tratados con SABR, es imprescindible abordar las restricciones y limitaciones que han influido en la investigación e interpretación de este estudio.

La principal limitación de este estudio, se encuentra en la disponibilidad limitada de datos y la imposibilidad de ampliar la muestra de pacientes, debido a la ausencia de un número suficiente de casos adicionales. Aunque el sobreajuste fue controlado, el tamaño de la muestra fue relativamente pequeña, con solo 128 metástasis tratadas. Además, los datos se encuentran desequilibrados en cuanto al tumor primario y a la localización metastásica se refiere. No siendo el caso de la respuesta, donde el conjunto se encuentra prácticamente equilibrado.

En vista de esto, la red neuronal enfrenta dificultades al entrenar el análisis de todos los tipos de datos estudiados, ya sea debido a la complejidad de este enfoque o la limitada capacidad de la red para discernir y aprender patrones intrínsecos. Como respuesta al problema señalado, en el transcurso del estudio, se optó a separar los datos en función de su localización metastásica. Se espera que, con un conjunto de datos suficientemente grande, la red neuronal sea capaz de efectuar predicciones precisas sin necesidad de recurrir a esta separación de datos.

Además, al tratarse de un estudio retrospectivo, sólo puede analizarse a base de datos existentes, pero en el futuro se puede ampliar incorporando variables y datos nuevos de más pacientes y estudiarlos de manera prospectiva. Otra limitación a añadir, se debe a que la adquisición de los datos proviene únicamente del mismo hospital. Se debería de agregar datos de distintos hospitales para una validación externa mejorando así la repetibilidad del modelo.

Finalmente, como perspectivas de investigación futuras, resulta necesario expandir de forma considerable la cantidad de muestras con la finalidad de evaluar el modelo en un contexto de aplicación real. Este aumento en la cantidad de muestras, permitiría validar la

efectividad del modelo en situaciones más complejas y variadas. Con este avance, proporcionar una herramienta o aplicación clínica mediante la cual se permita determinar la respuesta a la presencia del tumor metastásico tras un TAC inicial de simulación o preparación.

Siguiendo esta perspectiva, resultaría beneficioso adaptar y transformar la programación para la selección de características radiómicas en Slicer3D a un único código en Python, posibilitando compilar y generar eficientemente los datos radiómicos requeridos a medida que se planifican los tratamientos SABR. Además, brindaría al oncólogo médico una aplicación simple, para una evaluación rápida y efectiva de la respuesta metastásica ante el procedimiento clínico.

Adicionalmente, se vislumbra la posibilidad de llevar a cabo un análisis de estudio posterior que involucre la unificación de las respuestas completas y parciales, comparándolas con las respuestas estables y progresivas. El objetivo sería transformar el enfoque actual, continuando con el objetivo de predecir la respuesta metastásica, ahora de tipo binaria, orientada hacia la supervivencia del paciente como criterio de valoración del pronóstico. Esta evolución podría resultar esencial en la asistencia clínica para tratamientos radioterapéuticos con SABR sobre pacientes oligometastásicos. Un análisis integral de este tipo ofrecería una valiosa perspectiva sobre la efectividad de estos enfoques en la mejora de la supervivencia y calidad de vida de los pacientes.

6 Conclusiones

En este estudio se ha destacado el uso de la radiómica junto con algoritmos de aprendizaje automático para la elaboración de modelos capaces de predecir la respuesta metastásica en pacientes oligometastásicos tratados con radioterapia SABR. De esta manera, se han abordado los objetivos propuestos inicialmente, cumpliendo con la creación de base de datos de las metástasis y pacientes tratados como muestra para el entrenamiento de los modelos, la descarga y extracción de características radiómicas a partir de las imágenes TAC de planificación y, finalmente, la elaboración de modelos de inteligencia artificial con capacidad predictiva para evaluar el seguimiento del cáncer metastásico.

En resumen, tras evaluar modelos de entrenamiento de aprendizaje automático supervisados, se destaca un significativo mejor desempeño en la Red Neuronal Artificial para la predicción de la respuesta para casos de metástasis con localizaciones óseas, lo cual indica que parece entrenar con mayor efectividad a partir de la separación de los datos muestreados en base a su localización metastásica.

La aplicabilidad clínica de estos resultados es prometedora, permitiendo una aplicación clínica como herramienta para la toma de decisiones en el tratamiento de pacientes oligometastásicos con SABR. Sin embargo, es importante tener en cuenta la disponibilidad limitada de datos en este estudio. Sería necesario recopilar más datos de entrenamiento y muestra como estudio para poder ser de utilidad y participar como punto de apoyo en la asistencia a las decisiones clínicas.

En conclusión, se ha conseguido cumplir con los objetivos iniciales del trabajo, aún a falta de más datos y pruebas de estudio, los resultados indican que el modelo entrenado representa un paso significativo hacia la integración de la radiómica junto a modelos de inteligencia artificial para la asistencia y mejora en la toma de decisiones médicas.

7 Bibliografía

- [1] «Doctors, researchers re-thinking metastatic cancer treatments», *City of Hope*, 21 de enero de 2021. <https://www.cancercenter.com/community/blog/2021/02/newsletter-metastatic-cancer>.
- [2] «Cáncer de mama metastásico - Estadísticas», *Cancer.Net*, 20 de julio de 2017. <https://www.cancer.net/es/tipos-de-c%C3%A1ncer/c%C3%A1ncer-de-mama-metast%C3%A1sico/estad%C3%ADsticas>.
- [3] «LAS_CIFRAS_DEL_CANCER_EN_ESPANA_2022.pdf». [En línea]. Disponible en: https://seom.org/images/LAS_CIFRAS_DEL_CANCER_EN_ESPANA_2022.pdf
- [4] G. Földi *et al.*, «[Stereotactic ablative body radiotherapy (SABRT) for locally advanced pancreatic cancer. Case report and review of literature]», *Magy Onkol*, vol. 61, n.º 4, pp. 387-392, dic. 2017.
- [5] «Stereotactic Ablative Radiotherapy (SABR/SBRT)». <https://stanfordhealthcare.org/medical-treatments/s/stereotactic-body-radiation-therapy.html>.
- [6] D. A. Palma *et al.*, «Stereotactic ablative radiotherapy for the comprehensive treatment of 4-10 oligometastatic tumors (SABR-COMET-10): study protocol for a randomized phase III trial», *BMC Cancer*, vol. 19, n.º 1, p. 816, ago. 2019, doi: 10.1186/s12885-019-5977-6.
- [7] F. Vernuccio, R. Cannella, A. Comelli, G. Salvaggio, R. Lagalla, y M. Midiri, «[Radiomics and artificial intelligence: new frontiers in medicine.]», *Recenti Prog Med*, vol. 111, n.º 3, pp. 130-135, mar. 2020, doi: 10.1701/3315.32853.
- [8] A. Singh *et al.*, «Development of a robust radiomic biomarker of progression-free survival in advanced non-small cell lung cancer patients treated with first-line immunotherapy», *Sci Rep*, vol. 12, p. 9993, jun. 2022, doi: 10.1038/s41598-022-14160-7.
- [9] Yang H, Wang L, Shao G, Dong B, Wang F, Wei Y, Li P, Chen H, Chen W, Zheng Y, He Y, Zhao Y, Du X, Sun X, Wang Z, Wang Y, Zhou X, Lai X, Feng W, Shen L, Qiu G, Ji Y, Chen J, Jiang Y, Liu J, Zeng J, Wang C, Zhao Q, Yang X, Hu X, Ma H, Chen Q, Chen M, Jiang H, Xu Y. A combined predictive model based on radiomics features and clinical factors for disease progression in early-stage non-small cell lung cancer treated with stereotactic ablative radiotherapy. *Front Oncol*. 2022 Aug 2;12:967360. doi: 10.3389/fonc.2022.967360. PMID: 35982975; PMCID: PMC9380646.
- [10] B.-I. Song, «A machine learning-based radiomics model for the prediction of axillary lymph-node metastasis in breast cancer», *Breast Cancer*, vol. 28, n.º 3, pp. 664-671, may 2021, doi: 10.1007/s12282-020-01202-z.
- [11] J. Cervera Deval, «RECIST y el radiólogo», *Radiología*, vol. 56, n.º 3, pp. 193-205, may 2014, doi: 10.1016/j.rx.2012.03.010.
- [12] A. Bertolet, «Proyecto Fin de Máster: Algoritmo de planificación basado en capas (SBAP) para tratamientos de radioterapia de intensidad modulada con Philips Pinnacle», 2018. doi: 10.13140/RG.2.2.25882.08643.
- [13] P. Mildemberger, M. Eichelberg, y E. Martin, «Introduction to the DICOM standard», *Eur Radiol*, vol. 12, n.º 4, pp. 920-927, abr. 2002, doi: 10.1007/s003300101100.
- [14] A. Fedorov *et al.*, «3D Slicer as an image computing platform for the Quantitative Imaging Network», *Magnetic Resonance Imaging*, vol. 30, n.º 9, pp. 1323-1341, nov. 2012, doi: 10.1016/j.mri.2012.05.001.
- [15] «Van Griethuysen et al. - 2017 - Computational Radiomics System to Decode the Radio.pdf». [En línea]. Disponible en: https://pyradiomics.readthedocs.io/_/downloads/en/2.0.1/pdf/
- [16] «Análisis de textura empleando la matriz de coocurrencia de nivel de gris (GLCM) - MATLAB & Simulink - MathWorks España». <https://es.mathworks.com/help/images/texture-analysis-using-the-gray-level-co-occurrence-matrix-glcm.html>.
- [17] «TFM_JAVIER_GUILLEN_CANO.pdf». [En línea]. Disponible en: https://oa.upm.es/72062/3/TFM_JAVIER_GUILLEN_CANO.pdf

- [18] Y.-P. Zhang *et al.*, «Artificial intelligence-driven radiomics study in cancer: the role of feature engineering and modeling», *Mil Med Res*, vol. 10, p. 22, may 2023, doi: 10.1186/s40779-023-00458-8.
- [19] R. Shaikh, «Feature Selection Techniques in Machine Learning with Python», *Medium*, 28 de octubre de 2018. <https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e>.
- [20] R. Chang, S. Qi, Y. Yue, X. Zhang, J. Song, y W. Qian, «Predictive Radiomic Models for the Chemotherapy Response in Non-Small-Cell Lung Cancer based on Computerized-Tomography Images», *Front Oncol*, vol. 11, p. 646190, jul. 2021, doi: 10.3389/fonc.2021.646190.
- [21] «Feature importances with a forest of trees», *scikit-learn*. https://scikit-learn/stable/auto_examples/ensemble/plot_forest_importances.html.
- [22] C. Ding y H. Peng, «Minimum redundancy feature selection from microarray gene expression data», *J Bioinform Comput Biol*, vol. 3, n.º 2, pp. 185-205, abr. 2005, doi: 10.1142/s0219720005001004.
- [23] «GitHub - smazzanti/mrmm: mRMR (minimum-Redundancy-Maximum-Relevance) for automatic feature selection at scale.» <https://github.com/smazzanti/mrmm>.
- [24] M. Robnik-Šikonja y I. Kononenko, «Theoretical and Empirical Analysis of ReliefF and RReliefF», *Machine Learning*, vol. 53, n.º 1, pp. 23-69, oct. 2003, doi: 10.1023/A:1025667309714.
- [25] L. Wang, *Support Vector Machines: Theory and Applications*. Springer Science & Business Media, 2005.
- [26] SheetalSharma, «Artificial Neural Network (ANN) in Machine Learning - DataScienceCentral.com», *Data Science Central*, 8 de agosto de 2017. <https://www.datasciencecentral.com/artificial-neural-network-ann-in-machine-learning/>.
- [27] A. Bhandari, J. Koppen, y M. Agzarian, «Convolutional neural networks for brain tumour segmentation», *Insights Imaging*, vol. 11, p. 77, jun. 2020, doi: 10.1186/s13244-020-00869-4.
- [28] U. García, «Introducción a las Redes Neuronales», *Future Lab*, 9 de febrero de 2020. <https://medium.com/futurelabmx/introducci%C3%B3n-a-las-redes-neuronales-53abbc3abfbf>.
- [29] P. Panda, S. A. Aketi, y K. Roy, «Toward Scalable, Efficient, and Accurate Deep Spiking Neural Networks With Backward Residual Connections, Stochastic Softmax, and Hybridization», *Front Neurosci*, vol. 14, p. 653, jun. 2020, doi: 10.3389/fnins.2020.00653.
- [30] L. Prechelt, «Automatic early stopping using cross validation: quantifying the criteria», *Neural Netw*, vol. 11, n.º 4, pp. 761-767, jun. 1998, doi: 10.1016/s0893-6080(98)00010-0.
- [31] W. B. Bustamante, «Calculando la precisión en un modelo de Clasificación Multiclase», *Medium*, 3 de abril de 2021. <https://wbarriosb.medium.com/calculando-la-precisi%C3%B3n-en-un-modelo-de-clasificaci%C3%B3n-multiclase-224d96f52043>.

8 Anexos

8.1 Anexo 1: Código del programa

En el siguiente apartado se introducirán los códigos utilizados para la elaboración del trabajo. Los ejemplos mostrados forman parte del conjunto exclusivo de metástasis óseas.

8.1.1 Preprocesado de datos

```
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
```

```
import os
ruta = os.path.abspath('C:/Users/USUARI/Desktop/Juanpe Fernandez/UNI/5T0/TFG/Final/GLOBAL107.xlsx')
df=pd.read_excel(ruta)
df.shape
```

```
(128, 125)
```

```
# Codificación de las distintas variables en forma binaria y categórica
df["Response"] = df["Response"].map({"COMPLETA": 0, "PARCIAL": 1, "ESTABLE": 2, "PROGRESIVA": 3})
df["Sexo"] = df["Sexo"].map({"M": 1, "F": 0})
df["Tabaco"] = df["Tabaco"].map({"SI": 1, "NO": 0})
df["QT"] = df["QT"].map({"SI": 1, "NO": 0})
df["HT"] = df["HT"].map({"SI": 1, "NO": 0})
df["M1 DX"] = df["M1 DX"].map({"SI": 1, "NO": 0})

df["Primario"] = df["Primario"].map({"CCR": 1, "MAMA": 2, "PROSTATA": 3, "PULMON": 4})
df["T"] = df["T"].map({"T1": 1, "T2": 2, "T3": 3, "T4": 4, "Tx": 5})
df["N"] = df["N"].map({"N0": 0, "N1": 1, "N2": 2, "N3": 3, "Nx": 5})
df["M"] = df["M"].map({"M0": 0, "M1": 1, "M2": 2, "M3": 3, "Mx": 5})
df["Estado Actual"] = df["Estado Actual"].map({
    "VIVO SIN ENFERMEDAD": 1,
    "VIVO CON ENFERMEDAD ESTABLE": 2,
    "VIVO CON PD": 3,
    "VIVO CON ENFERMEDAD": 4,
    "MUERTO POR ENFERMEDAD": 5,
    "MUERTO OTRAS CAUSAS CON ENFERMEDAD": 6,
    "MUERTO POR CAUSA DESCONOCIDA": 7,
    "EXITUS": 8
})
df["Localizacion"] = df["Localizacion"].map({"OSEA": 1, "PULMONAR": 2, "OTROS": 3})
df["AP"] = df["AP"].map({"ADC": 0, "ESCAMOSO": 1, "CPNCP":2, "OTROS":3})
```

```
# Definir características (únicamente las radiómicas) y objetivo, tratando el caso excusivo de met. oseas
features = df[df['Localizacion'] == 1].drop(['ID', 'Response', 'Edad', 'Sexo', 'Tabaco', 'IMC',
    'Primario', 'Grado', 'T', 'N', 'M', 'QT', 'HT', 'M1 DX',
    'Seguimiento', 'Estado Actual', 'Localizacion', 'AP'], axis=1)
target = df[df['Localizacion'] == 1]['Response']
```

```
# Estandarización de las características
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)
features = pd.DataFrame(scaled_features, columns=features.columns)
```

8.1.2 Selección de características

8.1.2.1 Random Forest

```
# Validación cruzada
kfold = KFold(n_splits=5, shuffle=True, random_state=42)
```

GridSearchCV para obtener los mejores parámetros para el random forest

```
# Definir Los parámetros y sus valores posibles
n_estimators = [int(x) for x in np.linspace(start= 10, stop =300, num=10)]
param_grid = {
    'n_estimators': n_estimators,
    'max_depth': [None, 5, 10],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
}

rf_model = RandomForestClassifier(random_state=42)

# Búsqueda de Los mejores parámetros con validación cruzada
grid_search = GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=kfold)
grid_search.fit(features, target)

best_params = grid_search.best_params_
print("Mejores parámetros: ", best_params)
```

Mejores parámetros: {'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 106}

8.1.2.1.1 Mean Decrease Accuracy (MDA)

RANDOM FOREST CON MDA (Mean Decrease Accuracy): Calcula diferencia de precisión que aporta cada característica

```
#RANDOM FOREST MDA
rf_model = RandomForestClassifier(n_estimators=106, max_depth=None, min_samples_leaf=2, min_samples_split=5, random_state=42)

# Calcular La precisión inicial del modelo
initial_score = []
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]
    rf_model.fit(train_features, train_target)
    score = rf_model.score(test_features, test_target)
    initial_score.append(score)
initial_accuracy = np.mean(initial_score)

# Calcular mean decrease accuracy
feature_importances = []
for feature in features.columns:
    scores = []
    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]

        # Excluir La característica actual del conjunto de características
        reduced_train_features = train_features.drop([feature], axis=1)
        reduced_test_features = test_features.drop([feature], axis=1)

        # Entrenar modelo en el entrenamiento reducido
        rf_model.fit(reduced_train_features, train_target)

        # Calcular precisión en el test reducido
        score = rf_model.score(reduced_test_features, test_target)
        scores.append(score)

    # Calcular La diferencia media en La precisión
    decrease_accuracy = initial_accuracy - np.mean(scores)
    feature_importances.append(decrease_accuracy)
```

```
# Crear DataFrame con Las importancias de Las características
importances_df = pd.DataFrame({'Feature': features.columns, 'Mean Decrease Accuracy': feature_importances})
importances_df = importances_df.sort_values('Mean Decrease Accuracy', ascending=False)

# Obtener Las características más importantes y añadir Las variables clinicas para un posterior uso
caracteristicas = list(importances_df['Feature'][:10])
caracteristicas.extend(["ID", "Response", "Edad", "Sexo", "Tabaco", "INC", "Primario", "Grado",
                        "T", "N", "M", "QT", "HT", "M1 DX", "Seguimiento", "Estado Actual", "Localizacion", "AP"])

df_global = df.loc[:, caracteristicas]
df_global.to_excel('caract_RF_MDA.xlsx', index=False)
```

8.1.2.1.2 Mean Decrease Impurity (MDI)

RANDOM FOREST CON MDI (Mean Decrease Impurity): Calcula la importancia de cada carac. con "feature_importances_"

```
#RANDOM FOREST MDI
rf_model = RandomForestClassifier(n_estimators=100, max_depth=None, min_samples_leaf=2, min_samples_split=5, random_state=42)

feature_importances = []
scores = []
for train_idx, test_idx in kfold.split(features):
    # Obtener datos 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 modelo
    rf_model.fit(train_features, train_target)

    # Calcular importancia de características con el atributo feature_importances_
    importances = rf_model.feature_importances_
    feature_importances.append(rf_model.feature_importances_)

    # Calcular rendimiento en conjunto de prueba
    score = rf_model.score(test_features, test_target)
    scores.append(score)
```

```
# Crear DataFrame con las características más importantes
importances_df = pd.DataFrame(feature_importances, columns=features.columns)
importances_mean = importances_df.mean().sort_values(ascending=True)

# Ordenar características por importancia
importances_mean = importances_df.mean().sort_values(ascending=False)

# Seleccionar las 10 características más importantes y añadir las variables clínicas para un posterior uso
caracteristicas = list(importances_mean.index[:10])
caracteristicas.extend(["ID", "Response", "Edad", "Sexo", "Tabaco", "IMC", "Primario", "Grado", "T", "N", "M", "QT", "HT",
                       "M1 DX", "Seguimiento", "Estado Actual", "Localizacion", "AP"])

df_global = df.loc[:, caracteristicas]
df_global.to_excel('caract_RF_MDI.xlsx', index=False)
```

8.1.2.2 Minimum Redundancy and Maximum Relevance (mRMR)

mRMR (Minimum Redundancy and Maximum Relevance)

```
import mrmr
from sklearn.datasets import make_classification

# select top 10 features using mRMR
from mrmr import mrmr_classif
selected_features = mrmr_classif(X=features, y=target, K=10)
```

```
# Seleccionar las 10 características más importantes y añadir las variables clínicas para un posterior uso
caracteristicas=[]
caracteristicas = selected_features
caracteristicas.extend(["ID", "Response", "Edad", "Sexo", "Tabaco", "IMC", "Primario", "Grado", "T", "N", "M", "QT",
                       "HT", "M1 DX", "Seguimiento", "Estado Actual", "Localizacion", "AP"])

df_global = df.loc[:, caracteristicas]
df_global.to_excel('caract_mRMR.xlsx', index=False)
```

8.1.2.3 Relief

Relief: mediante el uso del algoritmo "ReliefF"

```
from skfeature.function.similarity_based import reliefF

# Aplicar el método Relief
scores = reliefF.reliefF(features.values, target.values)
top_features_indices = np.argsort(scores)[::-1][:10]

# Obtener las características seleccionadas
top_features = list(features.columns[top_features_indices])
```

```
# Seleccionar las 10 características más importantes
caracteristicas=[]
caracteristicas = top_features
caracteristicas.extend(["ID", "Response", "Edad", "Sexo", "Tabaco", "IMC", "Primario", "Grado", "T", "N", "M", "QT", "HT",
                       "M1 DX", "Seguimiento", "Estado Actual", "Localizacion", "AP"])

df_global = df.loc[:, caracteristicas]
df_global.to_excel('caract_Relief.xlsx', index=False)
```

8.1.3 Modelos predictivos

```
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.preprocessing import MinMaxScaler
from sklearn.model_selection import GridSearchCV
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.optimizers import Adam
from sklearn.model_selection import KFold
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, f1_score
from sklearn.metrics import roc_curve, roc_auc_score, auc
from sklearn.preprocessing import label_binarize
from sklearn.model_selection import cross_val_score
import joblib
import os
```

Utilizamos las 10 car. radiómicas de mayor importancia, calculadas en el proceso de selección.

```
# ruta1 = os.path.abspath('C:/Users/USUARI/Desktop/Juanpe Fernandez/UNI/5T0/TFG/Final/Feature Selection/caract_RF_MDA.xlsx')
ruta1 = os.path.abspath('C:/Users/USUARI/Desktop/Juanpe Fernandez/UNI/5T0/TFG/Final/Feature Selection/caract_RF_MDI.xlsx')
# ruta1 = os.path.abspath('C:/Users/USUARI/Desktop/Juanpe Fernandez/UNI/5T0/TFG/Final/Feature Selection/caract_mRMR.xlsx')
# ruta1 = os.path.abspath('C:/Users/USUARI/Desktop/Juanpe Fernandez/UNI/5T0/TFG/Final/Feature Selection/caract_Relief.xlsx')

df=pd.read_excel(ruta1)
df.shape

(128, 28)
```

```
class_names_dict = {
    0: 'Completa',
    1: 'Parcial',
    2: 'Estable',
    3: 'Progresiva'
}
```

Se divide el conjunto de entrenamiento/test y validación (caso específico del conjunto de met.Óseas)

```
df_global = df[df["Localizacion"] == 1].sample(n=56, random_state=42)
df_global.to_excel("dfGlobal108.xlsx", index=False)
```

```
df_validacion = df[(df["Localizacion"] == 1) & ~(df.index.isin(df_global.index))].head(15)
df_validacion.to_excel("dfValidacion20.xlsx", index=False)
```

```
features = df_global.drop(['Response', 'ID'], axis=1)
target = df_global['Response']

df_global = df_global.reset_index(drop=True)
```

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

8.1.3.1 Random Forest

```
rf_model = RandomForestClassifier(n_estimators=138, max_depth=None, min_samples_leaf=1, min_samples_split=5, random_state=42)
predictions_listRF = []

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

    rf_model.fit(train_features, train_target)
    predsRF = rf_model.predict(test_features)

    # Crear DataFrame con Las predicciones y Las respuestas verdaderas de cada conjunto de test
    test_dataRF = df_global.loc[test_idx, :]
    test_dataRF['Predictive Response'] = predsRF
    predictions_listRF.append(test_dataRF[['ID', 'Primario', 'Localizacion', 'Response', 'Predictive Response']])

# Guardamos el modelo RF
joblib.dump(rf_model, 'RF_Model.pk1')

['RF_Model.pk1']
```

VALIDACIÓN

```
import joblib
modelRF = joblib.load('C:/Users/USUARI/Desktop/Juanpe Fernandez/UNI/5T0/TFG/FINAL/Predictive Models/RF_Model.pkl')
```

```
import os
ruta = os.path.abspath('C:/Users/USUARI/Desktop/Juanpe Fernandez/UNI/5T0/TFG/Final/Predictive Models/dfValidacion20.xlsx')
df=pd.read_excel(ruta)
df.shape
```

(15, 28)

VALIDACIÓN CON EL MODELO DE RANDOM FOREST (Precisión, Matriz de Confusión y Curva ROC-AUC)

```
from sklearn.metrics import accuracy_score, f1_score
nuevo_testx = features
nuevo_testy = target

predsRF = modelRF.predict(nuevo_testx)

scoreRF = accuracy_score(nuevo_testy, predsRF)
std_dev = np.std([1 - score for score in predsRF])
f1RF=f1_score(nuevo_testy, predsRF, average='macro')

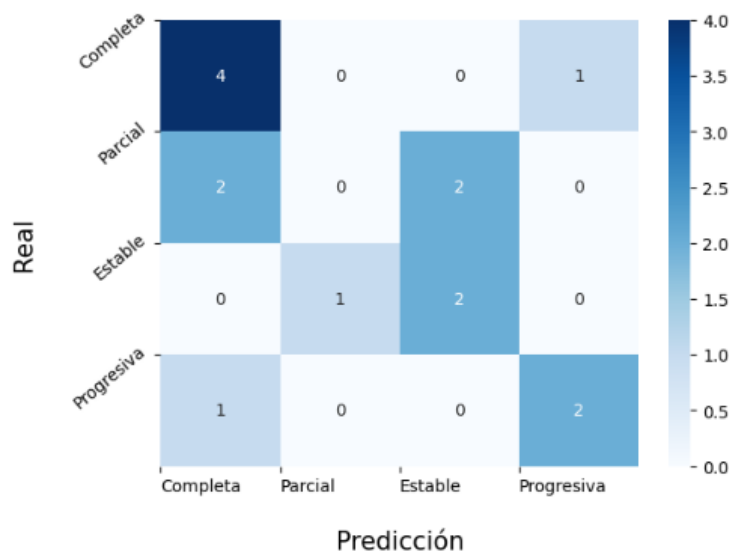
print(f'Precisión: {scoreRF}')
print(f'Desviación estándar: {std_dev}')
print(f'F1_score: {f1RF}')

predictionsRF = pd.DataFrame({
    'ID': df['ID'].values,
    'Primario': df['Primario'].values,
    'Localizacion':df['Localizacion'].values,
    'Response': df['Response'].values,
    'Predictive Response': predsRF
})

predictionsRF.to_excel('predictionsoPruebaRandomForestValidacion.xlsx', index=False)
```

Precisión: 0.5333333333333333
 Desviación estándar: 1.2220201853215573
 F1_score: 0.5333333333333333

```
conf_matrix = confusion_matrix(nuevo_testy, predsRF)
class_names = [class_names_dict[i] for i in sorted(class_names_dict)]
sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='g')
plt.xticks(ticks=range(len(class_names)), labels=class_names, rotation=0, ha='left', va='top', rotation_mode='anchor')
plt.yticks(ticks=range(len(class_names)), labels=class_names, rotation=40, ha='right', va='bottom', rotation_mode='anchor')
plt.xlabel('\nPredicción', fontsize=15)
plt.ylabel('\nReal\n', fontsize=15)
plt.show()
```



```
from sklearn.metrics import roc_curve, roc_auc_score, auc
from sklearn.preprocessing import label_binarize

# Predecir Las probabilidades
# Como tratamos con un problema de clasificación multiclase, se utiliza la técnica de one vs rest, o una contra el resto.
probas = modelRF.predict_proba(nuevo_testX)

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

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

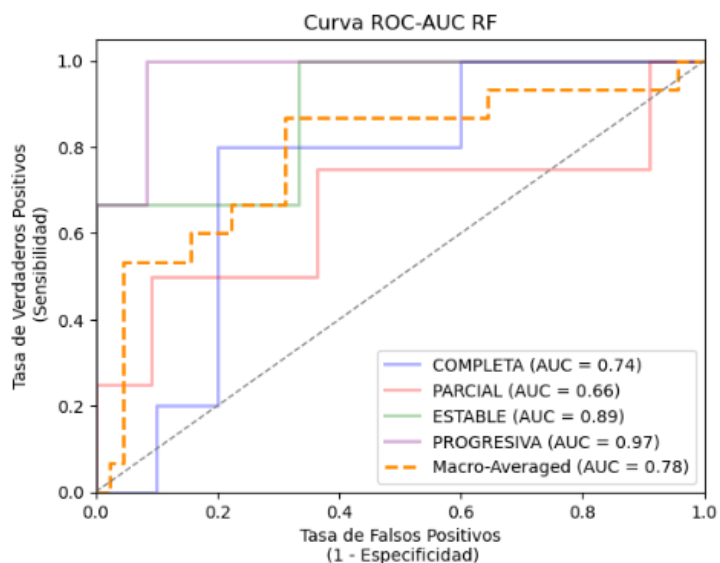
for i in range(len(np.unique(nuevo_testy))):
    fpr_rf[i], tpr_rf[i], _ = roc_curve(y_test_bin[:, i], probas[:, i])
    roc_auc_rf[i] = auc(fpr_rf[i], tpr_rf[i])

# plot
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 (AUC = {roc_auc_rf[1]:.2f})')
plt.plot(fpr_rf[2], tpr_rf[2], color='green', lw=2, alpha=0.3, label=f'ESTABLE (AUC = {roc_auc_rf[2]:.2f})')
plt.plot(fpr_rf[3], tpr_rf[3], color='purple', lw=2, alpha=0.3, label=f'PROGRESIVA (AUC = {roc_auc_rf[3]:.2f})')

# Calcular y graficar La curva ROC-AUC media
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"])

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

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\n(1 - Especificidad)')
plt.ylabel('Tasa de Verdaderos Positivos\n(Sensibilidad)')
plt.title('Curva ROC-AUC RF')
plt.legend(loc="lower right")
plt.show()
```



8.1.3.2 Support Vector Machines (SVM)

SVM - Support Vector Machine

```
from sklearn.svm import SVC
svm_model = SVC(kernel='rbf', C=1.0, gamma='scale', probability=True, random_state=42)

predictions_listSVM = []

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

    svm_model.fit(train_features, train_target)
    predsSVM = svm_model.predict(test_features)

    # Crear DataFrame con Las predicciones y Las respuestas verdaderas de cada conjunto de test/prueba
    test_dataSVM = df_global.loc[test_idx, :]
    test_dataSVM['Predictive Response'] = predsSVM
    predictions_listSVM.append(test_dataSVM[['ID', 'Primario', 'Localizacion', 'Response', 'Predictive Response']])

# Guardar el modelo SVM
joblib.dump(svm_model, 'SVM_Model.pkl')
```

VALIDACIÓN

```
# Cargar el modelo SVM entrenado
modelSVM = joblib.load('C:/Users/USUARI/Desktop/Juanpe Fernandez/UNI/5T0/TFG/FINAL/Predictive Models/SVM_Model.pkl')
```

VALIDACION CON EL MODELO DE SVM (Precisión...)

```
from sklearn.metrics import accuracy_score, f1_score
nuevo_testX = features
nuevo_testy = target

predsSVM = modelSVM.predict(nuevo_testX)

scoresSVM = accuracy_score(nuevo_testy, predsSVM)
std_devSVM = np.std([1 - score for score in predsSVM])
f1SVM=f1_score(nuevo_testy, predsSVM, average='macro')

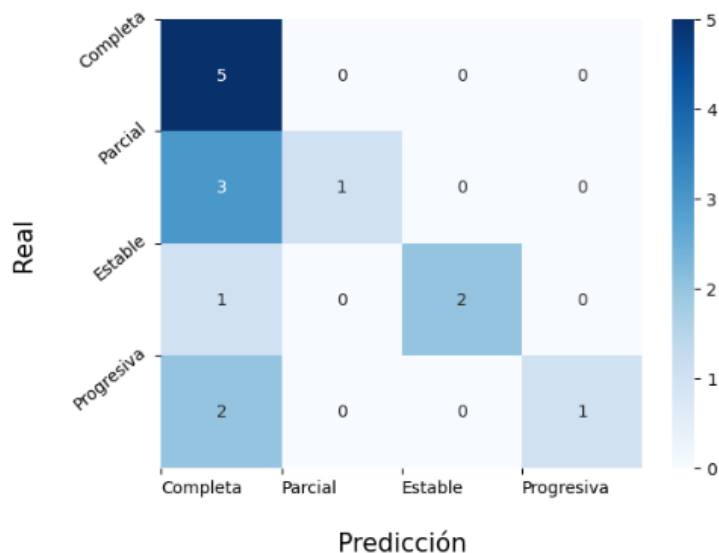
# Imprimir resultados de validación cruzada
print(f'Precisión: {scoresSVM}')
print(f'Desviación estándar: {std_devSVM}')
print(f'F1_score: {f1SVM}')

predictionsRF = pd.DataFrame({
    'ID': df['ID'].values,
    'Primario': df['Primario'].values,
    'Localizacion':df['Localizacion'].values,
    'Response': df['Response'].values,
    'Predictive Response': predsSVM
})

predictionsRF.to_excel('prediccionsoPruebaSVMValidacion.xlsx', index=False)
```

Precisión: 0.6
Desviación estándar: 0.9568466729604884
F1_score: 0.58125

```
conf_matrix = confusion_matrix(nuevo_testy, predsSVM)
class_names = [class_names_dict[i] for i in sorted(class_names_dict)]
sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='g')
plt.xticks(ticks=range(len(class_names)), labels=class_names, rotation=0, ha='left', va='top', rotation_mode='anchor')
plt.yticks(ticks=range(len(class_names)), labels=class_names, rotation=40, ha='right', va='bottom', rotation_mode='anchor')
plt.xlabel('\nPredicción', fontsize=15)
plt.ylabel('\nReal\n', fontsize=15)
plt.show()
```



```
# Predecir Las probabilidades
# Como tratamos con un problema de clasificación multicalse, se utiliza la técnica de one vs rest, o una contra el resto.
probasSVM = modelSVM.predict_proba(nuevo_testX)

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

# Calcular La curva ROC-AUC y el puntaje para cada clase
fpr_svm = {}
tpr_svm = {}
roc_auc_svm = dict()

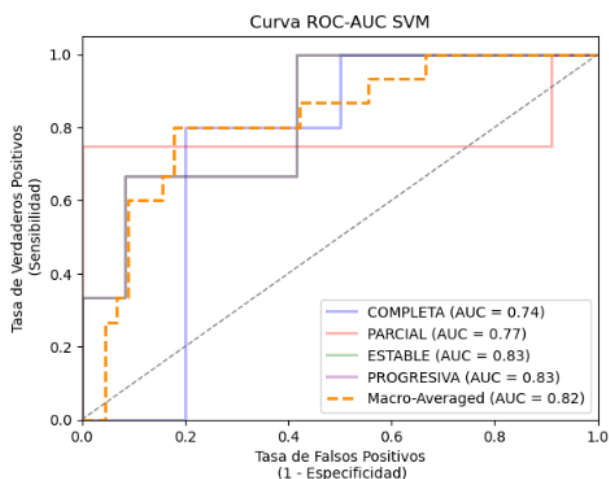
for i in range(len(np.unique(nuevo_testy))):
    fpr_svm[i], tpr_svm[i], thresholds = roc_curve(y_test_bin[:, i], probasSVM[:, i])
    roc_auc_svm[i] = auc(fpr_svm[i], tpr_svm[i])

# plot
plt.plot(fpr_svm[0], tpr_svm[0], color='blue', lw=2, alpha=0.3, label=f'COMPLETA (AUC = {roc_auc_svm[0]:.2f})')
plt.plot(fpr_svm[1], tpr_svm[1], color='red', lw=2, alpha=0.3, label=f'PARCIAL (AUC = {roc_auc_svm[1]:.2f})')
plt.plot(fpr_svm[2], tpr_svm[2], color='green', lw=2, alpha=0.3, label=f'ESTABLE (AUC = {roc_auc_svm[2]:.2f})')
plt.plot(fpr_svm[3], tpr_svm[3], color='purple', lw=2, alpha=0.3, label=f'PROGRESIVA (AUC = {roc_auc_svm[3]:.2f})')

# Calcular y graficar La curva ROC-AUC media
fpr_svm["macro"], tpr_svm["macro"], _ = roc_curve(y_test_bin.ravel(), probasSVM.ravel())
roc_auc_svm["macro"] = auc(fpr_svm["macro"], tpr_svm["macro"])

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

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\n(1 - Especificidad)')
plt.ylabel('Tasa de Verdaderos Positivos\n(Sensibilidad)')
plt.title('Curva ROC-AUC SVM')
plt.legend(loc="lower right")
plt.show()
```



8.1.3.3 Artificial Neuronal Network (ANN)

RED NEURONAL ARTIFICIAL (ANN)

```
from keras import regularizers
from keras.callbacks import EarlyStopping, Callback
# Definir modelo de red neuronal
model = Sequential()
#Utilizo capas Dense, donde todas las salidas de las neuronas se conectan a la entrada de las neuronas de la siguiente capa
model.add(Dense(28, activation='relu')) #capa de entrada con activacion ReLU
model.add(Dense(36, activation='relu', kernel_regularizer=regularizers.l2(0.01))) #capa intermedia con activacion ReLU
model.add(Dropout(0.2))
model.add(Dense(4, activation='softmax')) #capa de salida con activacion Softmax

# Compilar modelo
optimizer = Adam(lr=0.001)
model.compile(loss='sparse_categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])

# Definir Early Stopping
early_stopping = EarlyStopping(monitor='val_loss', min_delta=0.0001, patience=20, verbose=True, mode='auto',
                               baseline=None, restore_best_weights=True)

predictions_list = []
entrenamientos = []
for i, (train_idx, test_idx) in enumerate(kfold.split(features)):
    X_train, X_test = features[train_idx], features[test_idx]
    y_train, y_test = target.iloc[train_idx], target.iloc[test_idx]

    entrenamiento = model.fit(X_train, y_train, validation_data=(X_test, y_test), batch_size=32, epochs=300,
                              verbose=False, callbacks=[early_stopping])
    entrenamientos.append(entrenamiento)
    preds = model.predict(X_test)

    # Crear DataFrame con las predicciones y las respuestas verdaderas de cada conjunto de test/prueba
    test_data = df_global.loc[test_idx, :]
    predictions = pd.DataFrame({
        'ID': test_data['ID'].values,
        'Primario': test_data['Primario'].values,
        'Localizacion': test_data['Localizacion'].values,
        'Response': test_data['Response'].values,
        'Predictive Response': np.argmax(preds, axis=1)
    })
    predictions_list.append(predictions)
```

```
# Guardamos la red
model.save('RedNeuronal.h5')
model.save_weights("PesosCNN.h5")

# Guardar DataFrame con las predicciones y las respuestas verdaderas en un archivo Excel
predictions_df = pd.concat(predictions_list)
predictions_df.to_excel('PredictionsCNN.xlsx', index=False)
```

```
#GRAFICO LA MEDIA DE TODOS LOS KFOLDS para visualizar si hay sobreajuste
plt.plot(np.mean([entrenamiento.history['loss'] for entrenamiento in entrenamientos], axis=0))
plt.plot(np.mean([entrenamiento.history['val_loss'] for entrenamiento in entrenamientos], axis=0))
plt.xlabel('Ciclos de entrenamiento')
plt.ylabel('Pérdida')
plt.legend(['Pérdida de entrenamiento', 'Pérdida de test'])
plt.show()
```

VALIDACIÓN

```
from keras.models import load_model
modelRN = load_model('C:/Users/USUARI/Desktop/CAPA OCULTA 36/MDI OSEAS/RedNeuronal.h5')
modelRN.load_weights('C:/Users/USUARI/Desktop/CAPA OCULTA 36/MDI OSEAS/PesosCNN.h5')
```

VALIDACIÓN CON EL MODELO DE RED NEURONAL (Precisión, Matriz de Confusión y Curva ROC-AUC)

```
nuevo_testX = features
nuevo_testy = target

preds = modelRN.predict(nuevo_testX)
score = modelRN.evaluate(nuevo_testX, nuevo_testy, verbose=False)
```

```
# Imprimir resultados de validación cruzada
print(f'Exactitud: {score[1]}')
print(f'Desviación estándar: {np.std(score)}')

predictions = pd.DataFrame({
    'ID': df['ID'].values,
    'Primario': df['Primario'].values,
    'Localizacion':df['Localizacion'].values,
    'Response': df['Response'].values,
    'Predictive Response': np.argmax(preds, axis=1)
})

predictions.to_excel('predictionsoPruebaRedNeuronalValidacion.xlsx', index=False)
```

```
1/1 [=====] - 0s 43ms/step
Exactitud: 0.80000011920929
Desviación estándar: 0.07391193509101868
```

```
# Binarizar Las clases
nuevo_testy_bin = label_binarize(nuevo_testy, classes=np.unique(nuevo_testy))

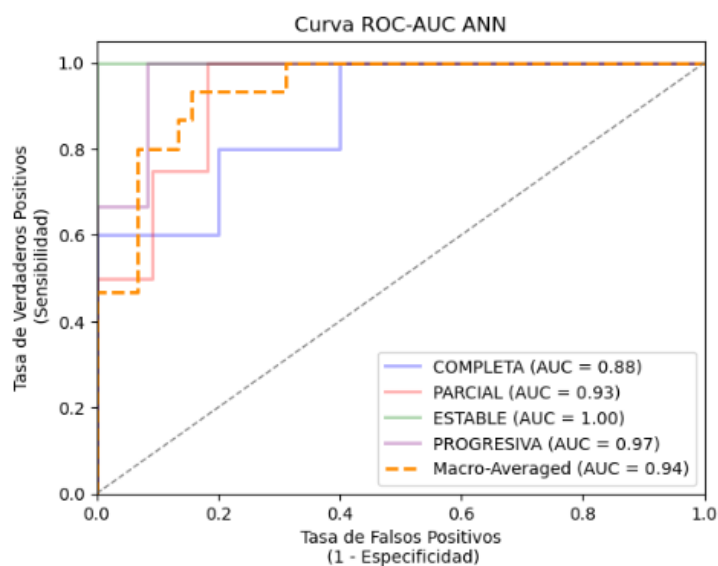
# Calcular La curva ROC-AUC y el puntaje para cada clase
fpr = {}
tpr = {}
roc_auc = dict()

i=0
for i in range(len(np.unique(nuevo_testy))):
    fpr[i], tpr[i], thresholds = roc_curve(nuevo_testy_bin[:, i], preds[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

#plot
plt.figure()
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 (AUC = {roc_auc[1]:.2f})')
plt.plot(fpr[2], tpr[2], color='green', lw=2, alpha=0.3, label=f'ESTABLE (AUC = {roc_auc[2]:.2f})')
plt.plot(fpr[3], tpr[3], color='purple', lw=2, alpha=0.3, label=f'PROGRESIVA (AUC = {roc_auc[3]:.2f})')

# Calcular y graficar La curva ROC media
fpr["macro"], tpr["macro"], thresholds = roc_curve(nuevo_testy_bin.ravel(), preds.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})')

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\n(1 - Especificidad)')
plt.ylabel('Tasa de Verdaderos Positivos\n(Sensibilidad)')
plt.title('Curva ROC-AUC ANN')
plt.legend(loc="lower right")
plt.show()
```



```
print(thresholds)
```

```
[1.9164152 0.9164153 0.68115705 0.61571515 0.39078978 0.27775848
0.27713498 0.27667555 0.24198972 0.17410491 0.14756812 0.00249933]
```

```
from sklearn.metrics import f1_score, recall_score, precision_score

nuevo_testy_bin = label_binarize(nuevo_testy, classes=np.unique(nuevo_testy))

best_threshold = None
best_f1_score = 0.0
best_precision = 0.0
best_recall = 0.0

for threshold in thresholds:
    preds_bin = (preds >= threshold).astype(int)

    # Calcular las métricas de precisión, recall y f1_score
    precision = precision_score(nuevo_testy_bin, preds_bin, average='macro', zero_division=1)
    recall = recall_score(nuevo_testy_bin, preds_bin, average='macro', zero_division=1)
    f1 = f1_score(nuevo_testy_bin, preds_bin, average='macro', zero_division=1)

    print(f"Threshold: {threshold:}, Precision: {precision:.6f}, Recall: {recall:.6f}, F1-score: {f1:.6f}")

    if f1 >= best_f1_score:
        best_threshold = threshold
        best_f1_score = f1
        best_precision = precision
        best_recall = recall

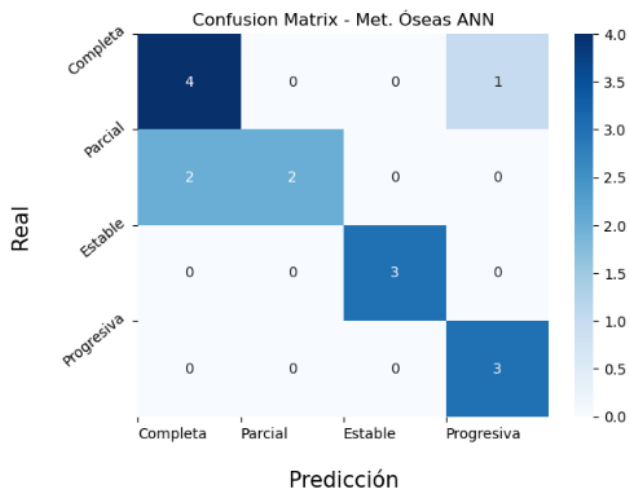
print("\nMejor Threshold:")
print(f"Threshold: {best_threshold:}, Precision: {best_precision:.6f}, Recall: {best_recall:.6f}, F1-score: {best_f1_score:.6f}")
```

```
Threshold: 1.9164152145385742, Precision: 1.000000, Recall: 0.000000, F1-score: 0.000000
Threshold: 0.916415274143219, Precision: 1.000000, Recall: 0.050000, F1-score: 0.083333
Threshold: 0.6811570525169373, Precision: 1.000000, Recall: 0.462500, F1-score: 0.612500
Threshold: 0.6157151460647583, Precision: 0.816667, Recall: 0.462500, F1-score: 0.541667
Threshold: 0.39078977704048157, Precision: 0.854167, Recall: 0.825000, F1-score: 0.812771
Threshold: 0.27775847911834717, Precision: 0.709524, Recall: 0.825000, F1-score: 0.747024
Threshold: 0.2771349847316742, Precision: 0.730357, Recall: 0.887500, F1-score: 0.791667
Threshold: 0.2766755521297455, Precision: 0.712500, Recall: 0.887500, F1-score: 0.778846
Threshold: 0.24198971688747406, Precision: 0.726389, Recall: 0.937500, F1-score: 0.803571
Threshold: 0.17410491406917572, Precision: 0.513636, Recall: 0.937500, F1-score: 0.656250
Threshold: 0.14756812155246735, Precision: 0.530303, Recall: 1.000000, F1-score: 0.689583
Threshold: 0.0024993272963911295, Precision: 0.250000, Recall: 1.000000, F1-score: 0.396930
```

```
Mejor Threshold:
Threshold: 0.39078977704048157, Precision: 0.854167, Recall: 0.825000, F1-score: 0.812771
```

```
# Calcular la matriz de confusión
conf_matrix = confusion_matrix(nuevo_testy, np.argmax(preds_bin, axis=1))

class_names = [class_names_dict[i] for i in sorted(class_names_dict)]
sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='g')
plt.xticks(ticks=range(len(class_names)), labels=class_names, rotation=0, ha='left', va='top', rotation_mode='anchor')
plt.yticks(ticks=range(len(class_names)), labels=class_names, rotation=40, ha='right', va='bottom', rotation_mode='anchor')
plt.xlabel('\nPredicción', fontsize=15)
plt.ylabel('\nReal\n', fontsize=15)
plt.title('Confusion Matrix - Met. Óseas ANN')
plt.show()
```



8.2 Anexo 2: Imágenes adicionales

En la siguiente sección se presentan las curvas ROC-AUC correspondientes a cada método de predicción, destacando las diferencias entre los cuatro enfoques de selección de características (MDA, MDI, mRMR y Relief). Esta comparación ha sido determinante en la elección del método de selección más adecuado para cada conjunto.

8.2.1 Conjunto Global

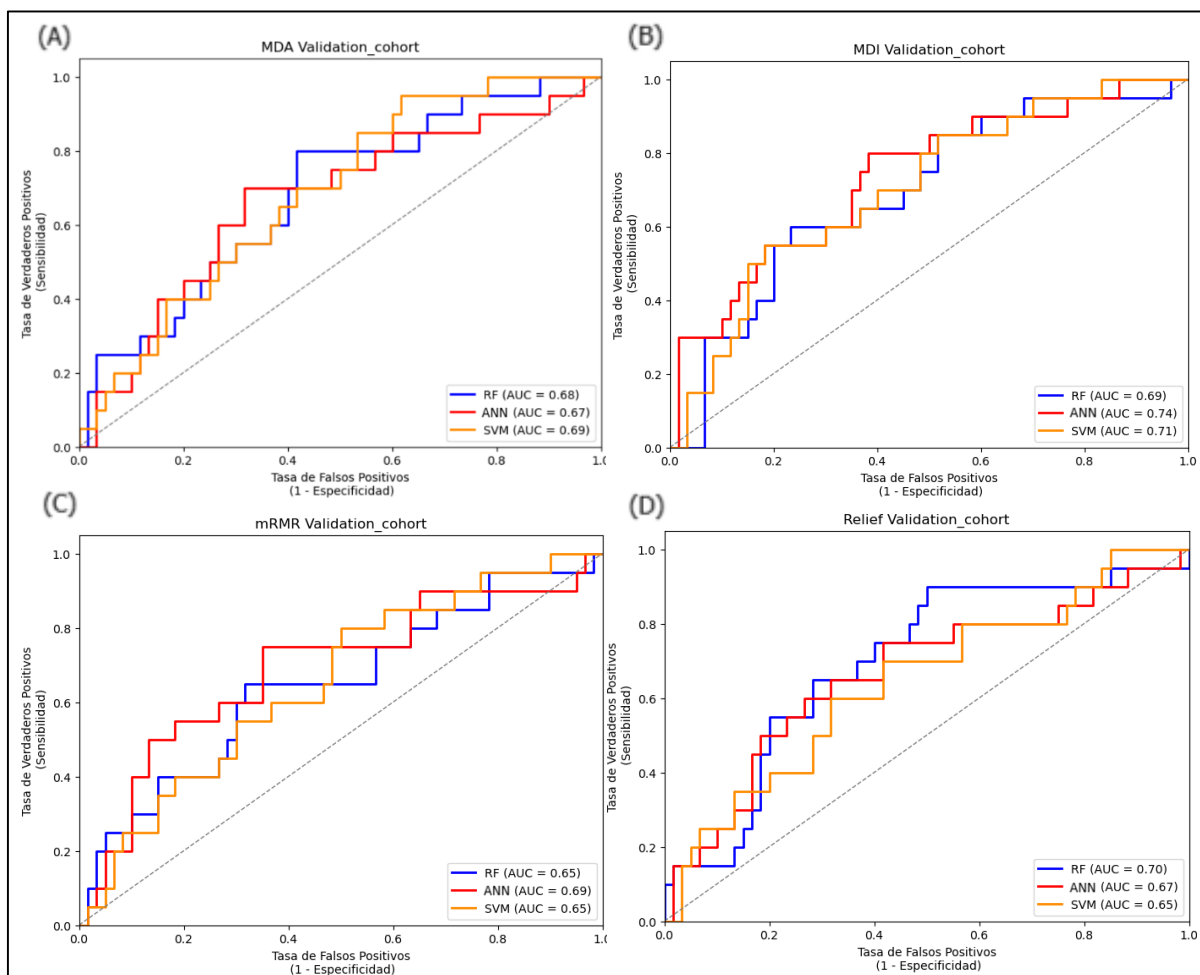


Figura 26. Representación de las curvas ROC-AUC para cada método de selección de características para el conjunto global. (A) *Mean Decrease Accuracy* (MDA); (B) *Mean Decrease Impurity* (MDI); (C) *Minimum Redundancy and Maximum Relevance* (mRMR); (D) *Relief*.

8.2.2 Conjunto Metástasis Óseas

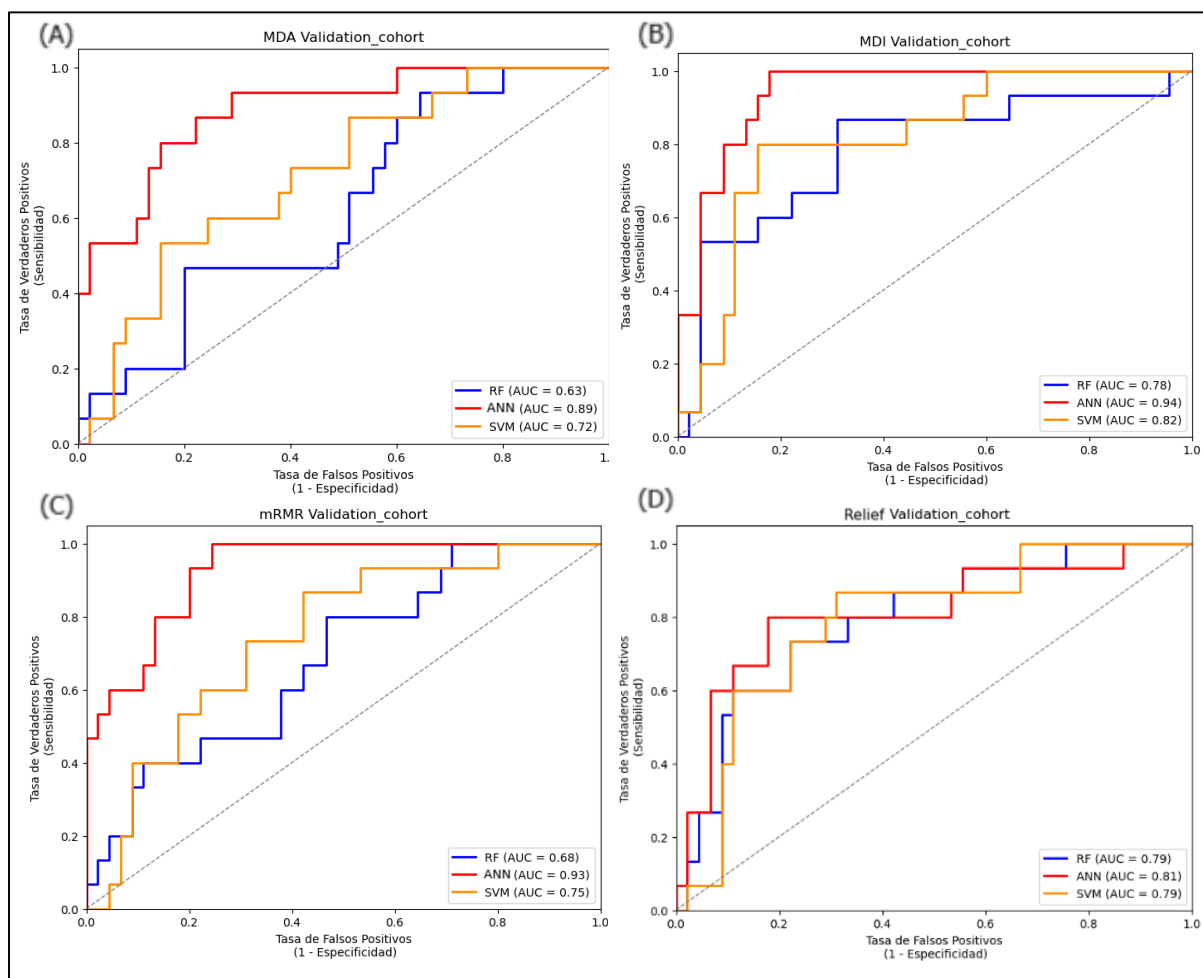


Figura 27. Representación de las curvas ROC-AUC para cada método de selección de características para el conjunto exclusivo de metástasis óseas. (A) *Mean Decrease Accuracy* (MDA); (B) *Mean Decrease Impurity* (MDI); (C) *Minimum Redundancy and Maximum Relevance* (mRMR); (D) *Relief*.

8.2.3 Conjunto Metástasis Pulmonares

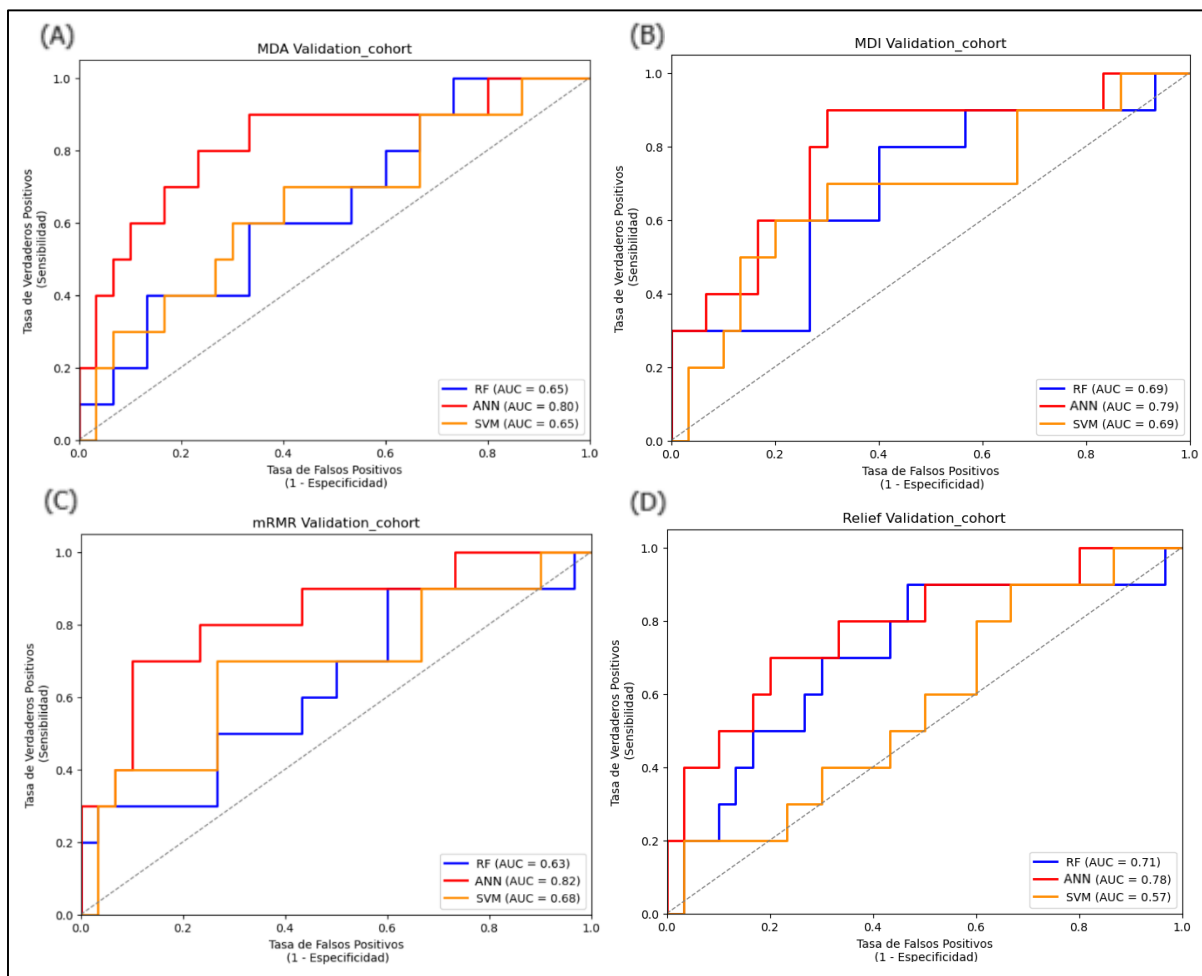


Figura 28. Representación de las curvas ROC-AUC para cada método de selección de características para el conjunto exclusivo de metástasis pulmonares. (A) *Mean Decrease Accuracy* (MDA); (B) *Mean Decrease Impurity* (MDI); (C) *Minimum Redundancy and Maximum Relevance* (mRMR); (D) *Relief*.