Paula Maixé Brull

Desenvolupament de models predictius basats en intel·ligència artificial i radiòmica sobre la resposta als tractaments de radioteràpia a dosis baixes en pacients amb patologia artrodegenerativa

TREBALL DE FI DE GRAU

dirigit pel Dr. Víctor Hernández Masgrau
i la Dra. Meritxell Arenas Prat

Grau d'Enginyeria Biomèdica

UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2024

RESUM

Aquest treball de final de grau se centra en l'estudi de l'ús de la radioteràpia a baixes dosis (RTDB) per al tractament de pacients amb artrosi a les mans. L'objectiu és desenvolupar models predictius per predir la resposta al tractament. Per aquest propòsit, s'han recopilat i analitzat dades de 92 pacients tractats amb RTDB, extraient un total de 1689 característiques radiòmiques per pacient. Aquestes característiques han estat analitzades utilitzant diversos mètodes per identificar-ne les més rellevants. Mitjançant tècniques d'aprenentatge automàtic, com ara Random Forest, Support Vector Machine (SVM) amb kernel Radial Basis Function (RBF), i Xarxes Neuronals Convolucionals (CNN), s'han construït models capaços de predir l'eficàcia del tractament.

Aquest estudi destaca per la seva aplicació innovadora de la radiòmica, combinada amb l'aprenentatge automàtic, amb l'objectiu de personalitzar el tractament i millorar la presa de decisions clíniques en pacients amb artrosi. La metodologia emprada, com l'automatització de diferents processos del projecte, proporciona una base sòlida per a futures investigacions en aquest camp.

ABSTRACT

This undergraduate thesis focuses on the study of low-dose radiotherapy (LDRT) for the treatment of patients with hand osteoarthritis. The objective is to develop predictive models to anticipate the response to treatment. For this purpose, data from 92 patients treated with LDRT were collected and analyzed, extracting a total of 1689 radiomic features per patient. These features were analyzed using various methods to identify the most relevant ones. Using machine learning techniques, such as Random Forest, Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel, and Convolutional Neural Networks (CNN), models were built to predict treatment efficacy.

This study stands out for its innovative application of radiomics combined with machine learning, aiming to personalize treatment and improve clinical decision-making in patients with osteoarthritis. The methodology employed, including the automation of various project processes, provides a solid foundation for future research in this field.

AGRAÏMENTS

Primerament, vull expressar el meu més sincer agraïment a l'Hospital Sant Joan de Reus per oferir-me l'oportunitat de realitzar aquest estudi. Gràcies a l'accés als recursos i al suport proporcionat, he pogut formar-me, desenvolupar-me professionalment i adquirir noves competències que seran de gran valor en la meva futura trajectòria.

Voldria agrair especialment al Dr. Víctor Hernández, a la Dra. Meritxell Arenas, a la Marta Canela i a la Raquel García pel seu suport constant durant tot el procés de realització d'aquest Treball de Final de Grau. Així mateix, vull donar les gràcies a l'enginyer Alberto Martínez per la seva valuosa ajuda en la creació dels codis necessaris per a aquest projecte.

En segon lloc, vull expressar el meu agraïment als meus professors i professores per la seva dedicació i per haver-me guiat en aquest camí d'aprenentatge, així com als meus companys i companyes de classe per fer que aquesta etapa universitària hagi estat única i inoblidable.

Finalment, vull donar les gràcies a la meva família i als meus amics i amigues pel seu suport incondicional i per estar en tot moment al meu costat durant aquests anys.

Índex

1. Introducció 1
   1.1 Contextualització de l'Artrosi 1
   1.2. Radiòmica 2
   1.3. Artrosi Tractada amb RTDB 2
   1.4 Predicció de la Resposta al Tractament RTDB 3
   1.2. Models Predictius i la Presa de Decisions Clíniques 4
      1.2.1. Random Forest 4
      1.2.2. SVM amb Kernel RBF 5
      1.2.3. CNN 5
   1.3. Procés Radioteràpic 6
2. Objectius 8
3. Materials i Mètodes 9
   3.1. Metodologia 9
   3.2. Descripció de la Base de Dades 9
   3.3. Extracció de Característiques 11
      3.3.1. DICOM a NIFTI: 12
      3.3.2. Extracció de Característiques Radiòmiques : 12
   3.4. Reducció i Selecció de Característiques 13
      3.4.1. Correlació: 13
      3.4.2. Informació Mútua (MI): 14
      3.4.3. Selecció Supervisada amb RF: 14
      3.4.4. ANOVA: 14
      3.4.5. LassoCV: 15
      3.4.6. Taula de Presència-Absència: 15
   3.5. Creació de Models 15
      3.5.1. Random Forest 16
      3.5.2. SVM amb Kernel RBF 16
      3.5.3. CNN 16
   3.6. Validació dels Models 17
4. Resultats 20
   4.1. Característiques Seleccionades 20
   4.2. Models Predictius 22
5. Discussió 26
6. Conclusions 30

7. Referències 31
Annex I 34
Annex II 35
Annex III 37
Annex IV 40

1. Introducció

1.1 Contextualització de l'Artrosi

L'artrosi es una artropatia degenerativa consistent en l'expressió d'un grup heterogeni de patologies d'etiologia multifactorial amb manifestacions biològiques, morfològiques i clíniques similars, que afecta a les articulacions mòbils. Es caracteritza per estrès cel·lular i degradació de la matriu extracel·lular del cartílag, induït inicialment per micro- i macro-lesions que activen respostes de reparació, entre les qual s'hi inclouen les vies pro inflamatòries de la immunitat innata. Més tard, sorgeixen altres alteracions com la degradació del cartílag, remodelació òssia, inflamació articular i pèrdua de funció articular normal, alteracions que expliquen dos de les característiques més importants d'aquesta malaltia: dolor i disfunció articular [1].

Figura 1: Escala de valoració Kellgren i Lawrence [2].

L'artrosi es la malaltia reumàtica més prevalent i més freqüent en la població. Acostuma a afectar les persones grans, tot i que pot aparèixer a qualsevol edat. A Espanya, el 29,3% de la població major de 20 anys pateix artrosis en una o més articulacions, encara que no sempre produeix molèsties ni símptomes. La prevalença d'aquesta malaltia augmenta amb l'edat, especialment a partir de la cinquena dècada de vida. Es calcula que fins el 70% de la població de més de 50 anys té signes radiològics d'artrosi en alguna de les articulacions del cos. A l'arribar als 75 anys, l'artrosi afecta al 80% d'aquestes persones. El pic de prevalença es dona al voltant dels 70-75 anys [3]. A Catalunya hi ha més d'un milió de persones que pateixen artrosi i l'envelliment de la població provoca que la xifra vagi en augment [4]. Per tant, es tracta d'una malaltia que afecta a una part molt extensa de la població, impactant tant a l'esperança de vida com a la qualitat i expectativa de vida.

Dins els factors de risc no modificables per patir aquesta malaltia trobem la susceptibilitat genètica, el sexe, sent més prevalent i greu en dones, i l'edat elevada. En quant als factors modificables, destaca l'obesitat, l'ocupació i activitat laboral, la pràctica d'esport professional i la densitat mineral òssia [2].

L'artrosi es tracta amb una combinació de mesures farmacològiques, no farmacològiques, i en alguns casos, tractaments quirúrgics. Els analgèsics com el paracetamol i els antiinflamatoris no esteroides (AINEs) s'utilitzen per alleujar el dolor i la inflamació [4]. A nivell no farmacològic, l'exercici regular, la fisioteràpia i la pèrdua de pes són essencials per millorar la mobilitat i reduir la càrrega sobre les articulacions

[5]. Les infiltracions intraarticulars amb corticosteroides o àcid hialurònic també poden ser útils per reduir el dolor i la inflamació [6]. En casos més greus, es pot recórrer a la cirurgia, com l'artroplàstia, per reemplaçar l'articulació danyada [7].

El diagnòstic i graduació de la gravetat de l'artrosi es realitzen normalment a partir de dades clíniques (simptomatologia del pacient) i de la imatge radiològica, el qual determinarà el pronòstic de la malaltia. El tractament de radioteràpia a dosis baixes (RTDB), té com a objectiu millorar la simptomatologia (reduir el dolor i millorar la funció) i evitar l'evolució de l'artrosi [1].

1.2. Radiòmica

La radiòmica és un camp emergent dins de la imatge mèdica que se centra en l'extracció d'informació quantitativa d'imatges mèdiques, com ara la tomografia axial computada (TAC), la ressonància magnètica (MRI) i la tomografia per emissió de positrons (PET). Aquesta informació, anomenada característiques radiòmiques, inclou descriptors de la textura, la forma, el contrast de píxels i altres propietats de les lesions i teixits. La radiòmica es basa en el concepte que les imatges mèdiques contenen molt més que informació visual: també poden codificar indicadors biològics subjacents que poden ser útils per a la diagnosi, pronòstic i planificació del tractament en diverses malalties [8] [9].

En el context de l'artrosi, l'ús de la radiòmica pot oferir noves perspectives per al tractament i seguiment dels pacients. Tradicionalment, l'artrosi es diagnostica i es monitoritza mitjançant imatges radiològiques convencionals, però aquestes no proporcionen informació quantitativa que pugui predir amb precisió la progressió de la malaltia o la resposta al tractament de radioteràpia. La radiòmica permet una anàlisi més profunda de les imatges, identificant patrons complexos que poden estar associats amb la gravetat de l'artrosi i la resposta als tractaments [8].

La integració de la radiòmica en el tractament de l'artrosi amb RTDB pot proporcionar una eina valuosa per avaluar la resposta dels pacients, permetent una personalització del tractament. Les característiques radiòmiques extretes de les imatges TAC podrien ajudar a identificar biomarcadors d'eficàcia terapèutica i a predir quins pacients podrien beneficiar-se més d'aquest tipus de tractament [10].

1.3. Artrosi Tractada amb RTDB

La RTDB actua a dosis molt més baixes que les utilitzades per tractar càncers. Les dosis de RTDB habituals oscil·len entre 0,3 i 1,5 Gy per sessió, depenent de la condició que es tracti. Per a condicions agudes, es poden administrar entre 3 i 5 Gy en total, mentre que per condicions cròniques es pot arribar fins a 12 Gy en total. Les sessions se solen distribuir al llarg de la setmana amb una freqüència de 1-3 vegades per setmana, i les sessions es poden repetir segons la resposta del pacient i la condició tractada.

L'ús de la RTDB en el tractament de l'artrosi ha estat objecte d'estudi i aplicació durant més d'un segle, tot i que el seu ús ha estat limitat en comparació amb altres tractaments

més convencionals. La radioteràpia, tot i estar tradicionalment associada al tractament de tumors, s'ha demostrat que a baixes dosis té efectes antiinflamatoris que poden ser eficaços en la reducció del dolor i la millora de la funcionalitat en pacients amb artrosi, especialment en aquells que no responen adequadament als tractaments estàndard [11].

Malgrat els beneficis potencials, l'ús de la RTDB en l'artrosi no està exempt de debat. La preocupació ètica més gran es centra en el risc potencial de càncer induït per la radiació, una por que ha limitat la seva adopció en molts països, incloent els Estats Units, on la radioteràpia com a tractament per malalties benignes es va abandonar en gran mesura durant els anys 1980. No obstant això, estudis recents suggereixen que aquest risc ha estat sobreestimulat, especialment quan s'utilitzen dosis molt baixes, comparables a les d'un TAC [12].

En països com Alemanya, aquesta teràpia és una pràctica establerta, amb milers de pacients tractats anualment sense que s'hagin registrat casos de càncer induït per radiació, la qual cosa reforça la seva seguretat quan s'aplica correctament [11] [12].

Actualment, són pocs els hospitals que ofereixen aquest tractament per a l'artrosi. A més d'Alemanya, la Clínica Cleveland als Estats Units ha reiniciat l'ús de la RTDB per a l'artrosi, amb un enfocament cautelós i estandarditzat per assegurar-se que els pacients reben el màxim benefici amb el mínim risc. Des de l'inici del programa a principis de 2023, es van tractar una dotzena de pacients, amb resultats prometedors en termes de reducció del dolor i millora de la funcionalitat articular [11].

A l'Hospital Sant Joan de Reus, s'aplica la RTDB per al tractament de l'artrosi i altres malalties benignes. Això permet oferir un tractament pioner als pacients i també brinda la possibilitat d'estudiar de primera mà els efectes i beneficis d'aquest tractament, contribuint així al coneixement i avanç en aquest camp terapèutic.

1.4 Predicció de la Resposta al Tractament RTDB

Podem trobar diferents estudis on s'utilitza la radiòmica per analitzar diferents aspectes del tractament d'artrosi. Un estudi ha investigat l'ús de la radiòmica per analitzar imatges de TAC de genolls en pacients que han passat per una artroplàstia total del genoll. L'objectiu de l'estudi era predir la satisfacció del pacient després de la cirurgia. Les característiques radiòmiques extretes de les imatges van demostrar ser útils per identificar factors que podrien influir en la satisfacció postoperatòria, suggerint que la radiòmica pot proporcionar informació addicional que no es pot obtenir a partir de l'anàlisi visual convencional [13].

Un altre estudi s'ha centrat en l'ús de radiòmica en imatges de MRI per identificar i classificar l'artrosi del genoll. Aquest treball ha utilitzat tècniques de segmentació automàtica i l'extracció de característiques radiòmiques per millorar la precisió en la detecció de l'artrosi així com en la determinació de la seva gravetat. Aquest enfocament

ofereix un mètode més quantitatiu per avaluar la malaltia en comparació amb les tècniques tradicionals d'imatge [14].

Així doncs, la radiòmica s'ha utilitzat per l'artrosi, especialment en l'anàlisi de les imatges mèdiques per a la identificació i classificació de la malaltia, així com per a la predicció de la satisfacció dels pacients després de procediments quirúrgics. No obstant això, fins ara, no s'ha aplicat la radiòmica per predir la resposta al tractament de radioteràpia en pacients amb artrosi. Encara que la predicció de la resposta al tractament amb radioteràpia s'ha explorat àmpliament en el context del càncer, aquesta és una àrea inexplorada en el camp de l'artrosi. Aquest treball representa el primer intent de realitzar aquesta predicció en pacients amb artrosi, obrint una nova via de recerca en la integració de la radiòmica i la radioteràpia en malalties no oncològiques.

1.2. Models Predictius i la Presa de Decisions Clíniques

Els models predictius són eines matemàtiques i computacionals que permeten predir esdeveniments futurs basant-se en dades històriques. En l'àmbit clínic, aquests models s'utilitzen per ajudar els professionals de la salut a prendre decisions informades, millorant així el diagnòstic, el tractament i els resultats dels pacients. Aquestes eines es poden aplicar en diverses àrees com el diagnòstic mèdic, la predicció de resultats, la gestió de recursos i la prevenció de reingressos. Els models predictius poden ajudar a identificar malalties en fases primerenques, com el càncer, analitzant dades i imatges mèdiques o històries clíniques. També es poden utilitzar per estimar la resposta d'un pacient a un tractament específic, cosa que permet personalitzar les teràpies segons les necessitats individuals. A més, en hospitals, aquests models poden ajudar a preveure la demanda de recursos com llits, personal i equipament, millorant l'eficiència operativa [15].

Els models que es van entrenar en aquest estudi son Random Forest, Support Vector Machine (SVM) amb kernel Radial Basis Function (KBF) i Xarxa Neuronal (CNN).

1.2.1. Random Forest

Random Forest és un mètode d'aprenentatge automàtic que utilitza una combinació de múltiples arbres de decisió per millorar la precisió i la capacitat predictiva del model. Cada arbre es forma a partir d'una mostra aleatòria del conjunt de dades, i les prediccions finals es fan a través de la mitjana o la moda de les prediccions dels arbres individuals. Aquesta agregació fa el model robust i eficient per a tasques de classificació i regressió, ja que ajuda a reduir la variància i a evitar el problema del sobreajustament, un fenomen en el qual el model s'ajusta massa als detalls i al soroll del conjunt de dades d'entrenament, fent que perdi la capacitat de generalitzar i predir correctament sobre dades noves [16]. A més, Random Forest introdueix variabilitat addicional en cada arbre seleccionant aleatòriament un subconjunt de les característiques disponibles per a la divisió en cada node, la qual cosa millora la diversitat dels arbres i, per tant, la capacitat del model per generalitzar a noves dades [17].

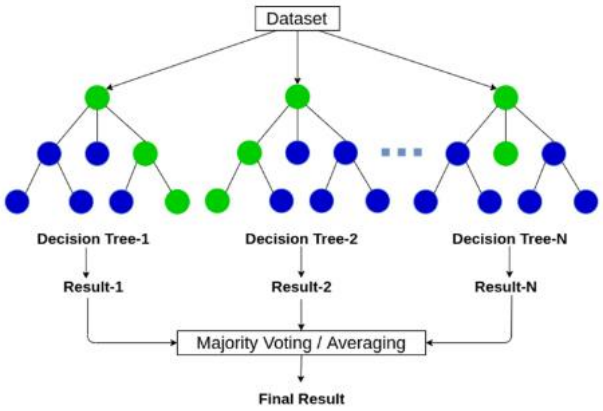

Figura 2: Esquema general del model Random Forest. [18]

1.2.2. SVM amb Kernel RBF

El model RBF és un dels kernels més potents i populars utilitzats en SVM, RBF és una extensió de l'SVM linear que pot gestionar dades que no són linealment separables. Utilitza un kernel RBF, una funció de base radial per projectar les dades a un espai de dimensionalitat superior, on es poden separar linealment. Aquest model és especialment útil per a la classificació de dades complexes on les relacions entre variables no són lineals, com ara l'anàlisi d'imatges mèdiques o la classificació de seqüències genètiques [19] [20].

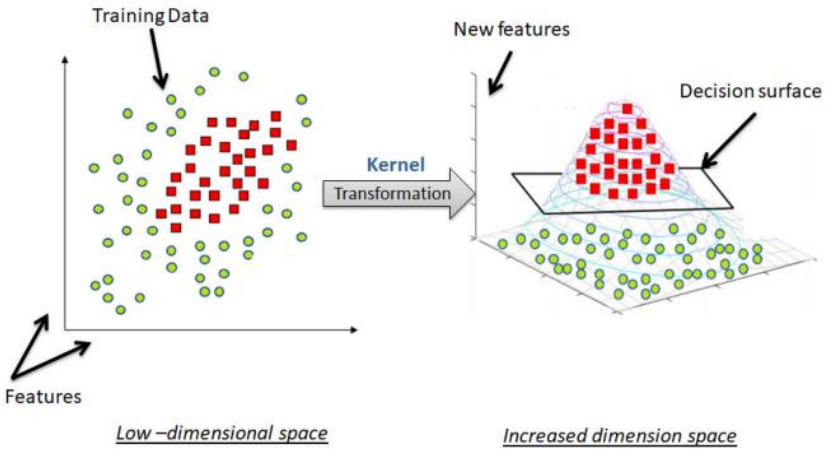

Figura 3: Esquema general del model SVM amb kernel RBF [21].

1.2.3. CNN

Les CNN són un tipus de xarxa neuronal dissenyada específicament per processar dades que tenen una estructura de graella, com les imatges. A través de diverses capes convolucionals, les CNN poden extreure característiques complexes de les imatges, com patrons de textura o formes, que són utilitzades per a la classificació o detecció d'objectes [22]. Aquestes xarxes neuronals s'utilitzen en l'anàlisi d'imatges mèdiques, com la

detecció automàtica de tumors en escàners o la classificació de radiografies i respostes als tractaments [23], i també en el diagnòstic assistit per ordinador, ajudant els metges a identificar anomalies que podrien passar desapercebudes en una revisió manual [24].

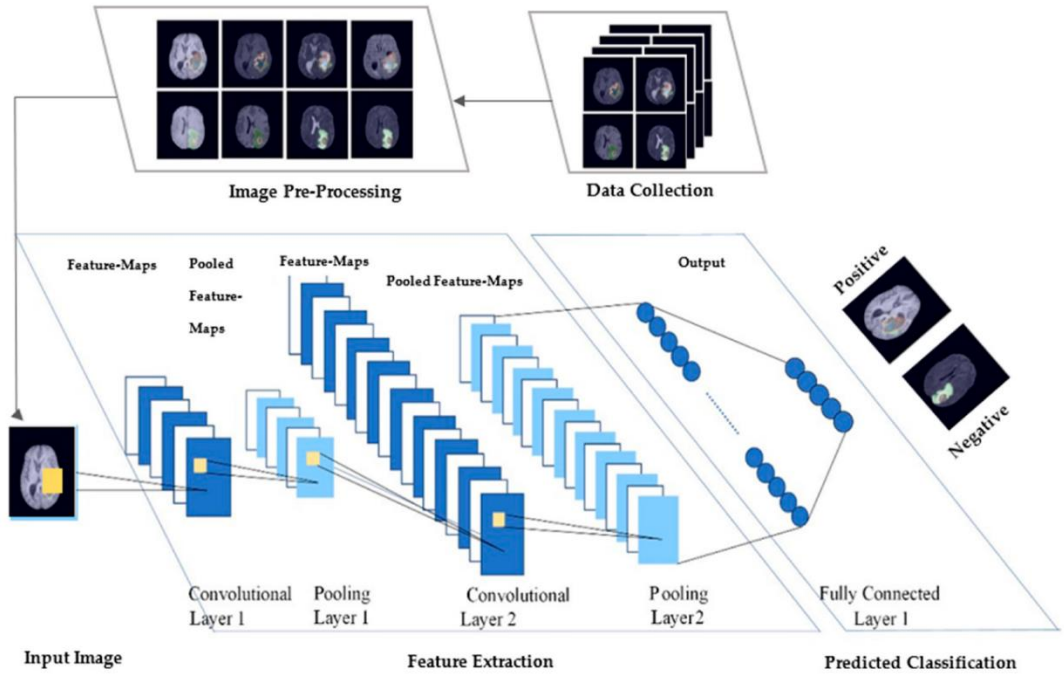

Figura 4: Esquema general del model CNN [25].

1.3. Procés Radioteràpic

La planificació de tractament en radioteràpia conté varies etapes. Primer de tot es fa un TAC de simulació amb l'objectiu d'obtenir un marc de referència del pacient sobre el que poder realitzar el pla terapèutic posterior. En aquest TAC, s'utilitzen projeccions de làsers i diferents immobilitzadors segons la localització anatòmica, els quals s'han d'incloure durant tot el tractament, ja que l'objectiu primordial es repetir la simulació de referència en cada sessió per tal de ser el màxim precisos possible. A l'inici de cada sessió, s'obté una imatge de Tomografia Computaritzada de Feix Cònic del pacient per tal de verificar que es troba col·locat igual en cada una d'elles.

El segon pas és la delimitació de volums, una tasca que duen a terme els metges. En aquest procés es tenen en compte tan les zones afectades com els òrgans crítics. El metge ha d'establir la dosis de tractament sobre l'articulació afectada i les restriccions o '*constraints'* sobre el teixit sa crític. Es contornegen els següents volums o ROIs (Region of Interest), definits per la Comissió Internacional d'Unitats Radiològiques. [26]

- **Gross Tumor Volume (GTV)**: delimita el tumor macroscòpic, el que podem veure a la imatge TAC.
- **Clinical Target Volume (CTV)**: delimita tant el tumor macroscòpic com el microscòpic, ja que tenim cèl·lules que a les imatges TAC no s'aprecien i s'han de tenir en compte pel tractament.

- **Internal Target Volume (ITV)**: s'aplica en zones que la posició varia inconscientment degut a la respiració o al moviment intern dels òrgans, per exemple, els pulmons al respirar, el cor al bombejar o si la bufeta esta plena o buida.
- **Planning Target Volume (PTV)**: té en compte el possible error de col·locació del pacient.

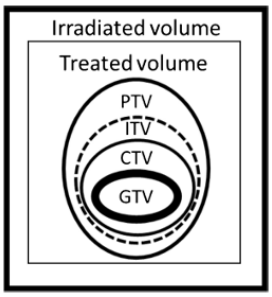

Figura 5: Definició esquemàtica de la delimitació de volums. [26]

Un cop tenim els volums contornejats i les dosis preinscrites, un físic mèdic s'encarrega de dissenyar tot el pla de tractament i la dosimetria. En general, això inclou escollir l'energia dels feixos i la seva disposició angular, la forma dels camps de radiació, etc. En aquest procés, els físics s'ajuden de programes informàtics avançats i tècniques com VMAT (Volumetric Modulated Arc Therapy) per l'optimització del tractament.

Finalment, el pla de tractament s'avalua i s'aprova tant pel metge com pel físic. Hi ha casos en que, si és necessari, es verifica la dosimetria sobre maniquins abans que el tractament sigui administrat al pacient.

2. Objectius

L'objectiu principal d'aquest treball de final de grau és el desenvolupament i validació d'una metodologia innovadora per a l'anàlisi de dades que permeti identificar factors clau que puguin predir la resposta dels pacients amb artrosi tractats amb RTDB. A través d'aquesta investigació es contribuirà en la personalització i optimització de les estratègies de tractament. D'aquesta manera podem definir els següents objectius específics.

1. **Creació d'una base de dades clínica:** Recopilar i estructurar les dades clíniques dels pacients tractats, incloent informació sobre l'historial mèdic i les respostes al tractament.
2. **Extracció de característiques radiòmiques:** Processar les imatges de TAC dels pacients per extreure característiques radiòmiques.
3. **Selecció de característiques:** Aplicar mètodes estadístics i d'aprenentatge automàtic per seleccionar les característiques radiòmiques més rellevants.
4. **Desenvolupament de models predictius:** Crear models predictius que integrin les característiques radiòmiques seleccionades per predir la resposta.
5. **Validació dels models:** Validar els models predictius creats mitjançant tècniques de validació creuada i altres mètodes per quantificar la seva robustesa i capacitat de generalització a nous pacients.

3. Materials i Mètodes

3.1. Metodologia

La metodologia utilitzada en aquest treball es fonamenta en una sèrie de passos per analitzar i predir la resposta dels pacients amb artrosi tractats amb radioteràpia a baixes dosis, a partir de les característiques radiòmiques extretes dels volums PTV, definits com la regió d'interès en les tomografies computades de simulació.

Primer de tot es van extreure les característiques radiòmiques de cada imatge TAC dels pacients inclosos en l'estudi. Aquestes característiques es van obtenir mitjançant un codi de Python, la qual cosa és important a destacar ja que es va automatitzar aquest procés, sense la necessitat de plataformes externes o intervenció manual.

El segon pas va ser la reducció de dimensionalitat i la selecció de característiques rellevants per a la predicció de la resposta. Es van utilitzar cinc mètodes diferents i es van triar aquelles variables seleccionades en els cinc mètodes, per tal d'assegurar que s'estaven seleccionant correctament les variables més rellevants.

Al tercer pas, es van utilitzar tècniques d'intel·ligència artificial i models d'aprenentatge automàtic per desenvolupar un model predictiu. Aquest model es va entrenar amb les característiques seleccionades i, finalment, es va avaluar la seva capacitat de predir la resposta.

3.2. Descripció de la Base de Dades

Per realitzar aquest estudi es va crear una base de dades amb pacients amb artrosi tractats amb RTDB al Servei d'Oncologia Radioteràpia de l'Hospital Universitari Sant Joan de Reus. En un inici era una base de dades única que incloïa tots els casos d'artrosi de mans, columna i genolls. Seguidament, es va decidir que per fer un estudi més precís es tractarien els casos per separat, ja que cada localització anatòmica té unes propietats diferents que ens podrien afectar en la predicció de la resposta. Així doncs, finalment, agafant la mostra més gran de pacients en funció de la localització, es va crear una base de dades dels pacients tractats d'artrosi a les mans, amb la qual es va realitzar la selecció de característiques i la creació de models predictius d'aquest projecte.

Aquesta base de dades es va crear a partir d'informació extreta de l'historial clínic dels pacients, concretament de les plataformes SAP i HNet de l'Hospital Universitari Sant Joan de Reus. Les variables clíniques recollides son les següents: l'edat del pacient en el moment del tractament; el sexe, codificat com a 1 per femení i 2 per masculí; i la data d'inici i de finalització del tractament. També es va especificar la localització específica de la zona tractada, que en el cas de les mans parlem de casos unilaterals o bilaterals.

En un principi la variable de la resposta es va dividir segons la resposta de dolor i funcionalitat, ja que hi havia pacients que milloraven significament en un aspecte però amb l'altre no. Així doncs, es va classificar el dolor en no respon o sí respon, i la funcionalitat com no respon, resposta parcial o resposta completa. Més endavant, parlant amb els metges, es va veure que tenir una única predicció de resposta els facilita la feina, per tant es va definir una resposta combinada de dolor i funcionalitat:

		Funcionalitat		
		No respon	**Resposta parcial**	**Resposta completa**
Dolor	**No respon**	Classe 0	Classe 0	Classe 1
	Sí respon	Classe 1	Classe 1	Classe 1

Taula 1: Taula de resposta combinada per a dolor i funcionalitat.

En aquest estudi es busca identificar els pacients que han tingut una resposta positiva i els que han tingut una resposta negativa. Degut al desequilibri de respostes, es va definir una resposta positiva al tractament (1) si aquesta és completa o parcial i una resposta negativa (0) si aquesta és sense resposta. La mostra en el cas de les mans consta de 92 pacients tractats, dels quals 64 han tingut una resposta positiva i 28 una resposta negativa. Per tant, s'observa que es disposa d'una mostra desequilibrada amb una alta prevalença de uns, és a dir, de pacients que responen positivament al tractament RTDB.

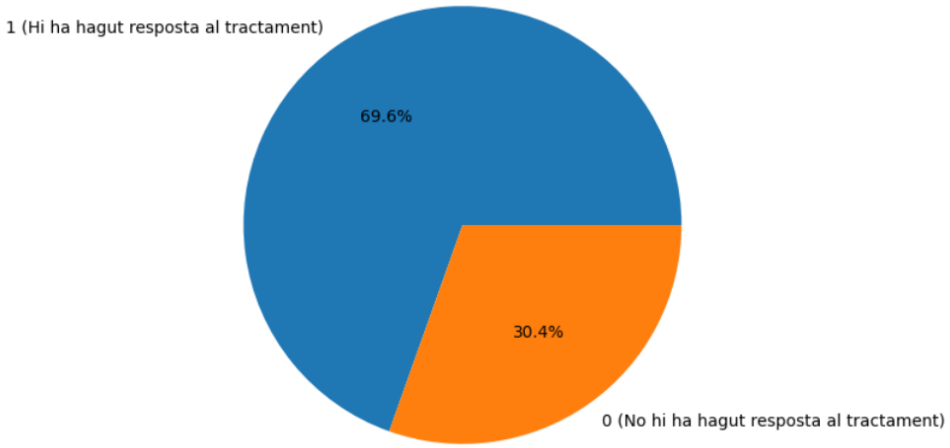

Figura 6: Gràfica circular que representa el percentatge de respostes positives i negatives de pacients amb artrosi a les mans.

Per tal de realitzar el codi per al processament de dades, és essencial la transformació de dades categòriques a valors numèrics, com en el cas de la variable 'Sexe', perquè els models de Machine Learning puguin treballar amb les dades de manera eficient.

Inicialment, en el codi referenciat en l'Annex IV, es van importar les dades utilitzant la llibreria *pandas*, que permet carregar les dades des d'un fitxer Excel amb tota la informació clínica recollida dels pacients i amb la resposta al tractament indicada. Es va utilitzar la funció *set_index()* per establir la columna *NumTAC* com a índex, i posteriorment es va fer ús de *dropna()* per eliminar qualsevol fila amb valors mancants.

En la fase següent, es va abordar el desequilibri en les classes mitjançant l'aplicació de la tècnica *SMOTE* (Synthetic Minority Over-sampling Technique), proporcionada per la llibreria *imblearn*. La funció *SMOTE()* es va utilitzar per generar mostres sintètiques per a la classe minoritària. Aquesta operació ens va permetre equilibrar la distribució de les classes abans d'entrenar els models.

Després de l'aplicació de *SMOTE*, les dades remostrades es van convertir en un nou DataFrame utilitzant *pandas*.

3.3. Extracció de Característiques

Les característiques radiòmiques es van obtenir a partir d'imatges TAC, on esta delimitat el volum d'interès PTV (volum de tractament planificat), el qual s'utilitza per dissenyar el tractament. Per la obtenció de les tomografies es va utilitzar l'entorn del programa Aria de l'empresa Varian, on a partir del ID del pacient, s'accedeix a la seva informació clínica i a les diferents tomografies executades durant el temps de tractament. Per visualitzar les imatges, un cop ja exportades, es va utilitzar el programa 3DSlicer 5.2.2, un programari gratuït de codi obert per a la visualització, processament, segmentació, registre i anàlisi d'imatges i malles mèdiques, biomèdiques i altres imatges i malles 3D [27].

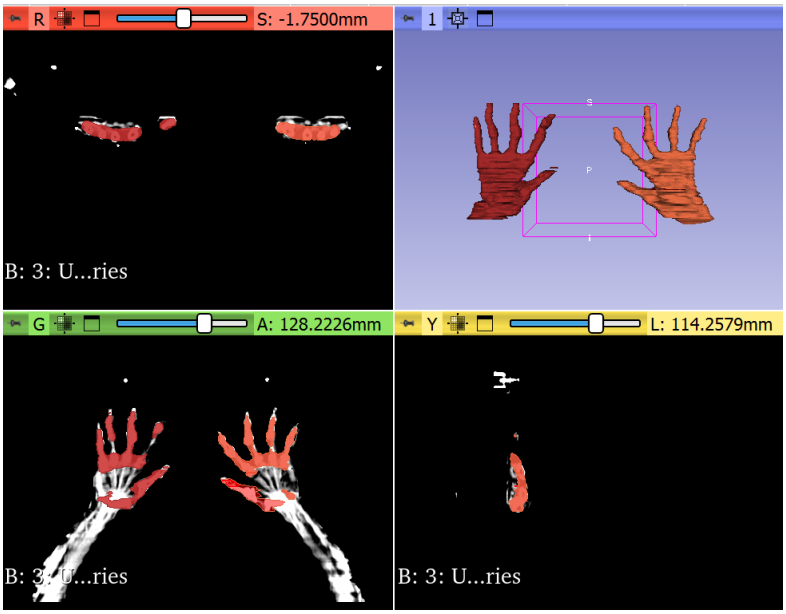

Figura 7: TAC amb PTV de pacient amb artrosi a les mans visualizat amb 3DSlicer.

Les dades d'aquest projecte van ser pseudonimitzades seguint els estàndards legals a nivell del Reglament General de Protecció de Dades 2016/679 (RGPD) de la Unió Europea i la Llei Orgànica de Protecció de Dades Personals i garantia dels drets digitals 3/2018 (LOPDGDD) d'Espanya, així com la Llei d'Investigació Biomèdica 14/2007.

Els TACs i les seves estructures es van exportat en format DICOM (Digital Imaging and Communication In Medicine), que és un estàndard internacional per a imatges mèdiques, ja que permet l'intercanvi d'imatges mèdiques i la informació relacionada de forma independent del fabricant dels equips d'imatge, per tal de facilitar la connectivitat entre dispositius i sistemes mèdics [28].

Per a l'extracció de característiques es van convertir les imatges DICOM a format NIfTI (Neuroimaging Informatics Technology Initiative), un format de fitxer utilitzat principalment per a l'emmagatzematge d'imatges mèdiques tridimensionals, que permet l'emmagatzematge de múltiples volums d'imatge en un sol fitxer, així com la informació associada, cosa que facilita el processament i l'anàlisi de les dades [29].

Pel canvi de format dels fitxers i l'extracció de característiques radiòmiques es va utilitzar Python, creant així un procés automatitzat que extreu un total de 1689 característiques radiòmiques de cada imatge de TAC.

3.3.1. DICOM a NIFTI:

Es va desenvolupar un codi Python, referenciat en l'Annex II, per convertir imatges mèdiques en format DICOM a format NIfTI. Es van utilitzar diverses llibreries clau per dur a terme aquest procés. *DicomRTTool* va proporcionar eines per llegir, processar i manipular els fitxers DICOM. La classe *DicomReaderWriter* va facilitar la navegació pels fitxers DICOM i l'extracció de regions d'interès (ROI), com el PTV. Es va utilitzar *SimpleITK* per llegir les imatges DICOM i posteriorment guardar-les en format NIfTI. La llibreria *os* es va emprar per gestionar les operacions del sistema de fitxers, com la navegació per directoris i la creació de carpetes per emmagatzemar els fitxers convertits.

El codi va recórrer automàticament les carpetes que contenien les imatges DICOM dels pacients, va extreure les imatges i les màscares de les ROI, i després va guardar aquestes imatges en format NIfTI en un directori especificat. Aquesta automatització va assegurar que les imatges es convertissin i s'emmagatzemessin correctament sense necessitat d'intervenció manual.

3.3.2. Extracció de Característiques Radiòmiques :

El codi referenciat en l'Annex III, va utilitzar diverses llibreries de Python essencials per al processament d'imatges i l'anàlisi de dades. *Pandas* es va emprar per organitzar i manipular les dades en estructures de DataFrame, facilitant un maneig senzill i eficient de la informació extreta. Per a les operacions numèriques, es va utilitzar *Numpy*, mentre que *Nibabel* va gestionar la càrrega i manipulació de les imatges mèdiques en format NIfTI.

En aquest context, una de les eines clau va ser *PyRadiomics*, que va permetre extreure característiques radiòmiques rellevants a partir de les regions d'interès (ROI) definides en les imatges mèdiques. A més, la llibreria *os* va servir per gestionar fitxers i directoris, facilitant la interacció amb el sistema operatiu per a la navegació i manipulació dels arxius.

El procés va començar amb la càrrega de les imatges en format NIfTI des d'un directori específic utilitzant *Nibabel*. Aquestes imatges es van processar per extreure'n les característiques radiòmiques, amb *PyRadiomics* obtenint dades que descrivien les propietats texturals i altres aspectes rellevants de les imatges.

Un cop extretes, les característiques radiòmiques es van emmagatzemar en un DataFrame, proporcionant una estructura fàcilment manipulable per a l'anàlisi posterior. Finalment, aquest DataFrame es va desar en un fitxer Excel per permetre una revisió i anàlisi més accessible.

Aquest codi va demostrar ser una eina poderosa per a l'extracció de característiques radiòmiques de les imatges NIfTI, automatitzant el procés i garantint una major eficiència i precisió en la gestió de grans volums de dades mèdiques, essencial en l'àmbit de la radiòmica.

3.4. Reducció i Selecció de Característiques

La selecció de característiques és crucial en la construcció de models predictius perquè millora la precisió, evita el sobreajustament degut al soroll, redueix la complexitat computacional, fa que el model sigui més interpretatiu i prevé l'impacte negatiu de característiques irrellevants o redundants [30].

En aquest projecte es va dur a terme un filtratge, a partir del codi referenciat en l'Annex IV, utilitzant cinc mètodes estadístics i computacionals diferents per seleccionar les característiques rellevants. Cada un d'aquests mètodes va proporcionar una llista de les característiques seleccionades, que es van comparar en una taula de presència-absència. Finalment, es van seleccionar aquelles característiques que havien estat escollides per tots cinc mètodes.

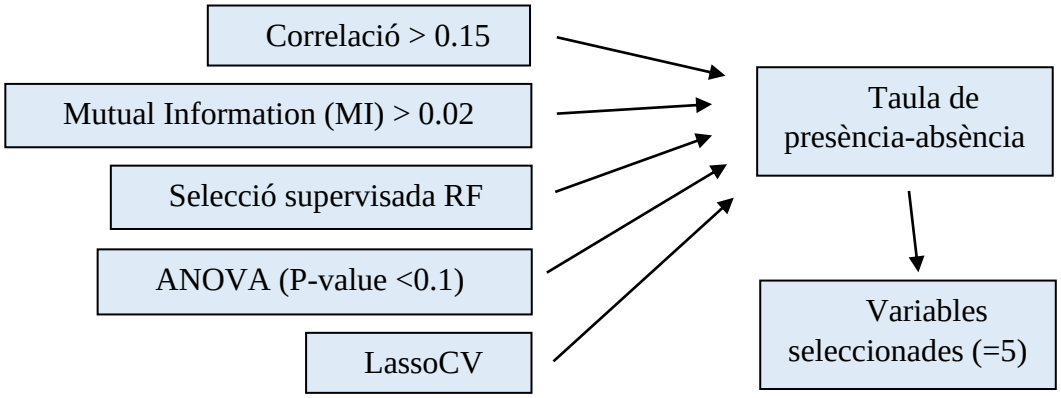

Figura 8: Esquema general del procés seguit per a la selecció de característiques.

3.4.1. Correlació:

La correlació és una mesura estadística que indica el grau de relació entre dues variables. En concret, la correlació lineal serveix per determinar com de correlacionades linealment estan dues variables diferents. Dues variables estan relacionades quan al variar els valors d'una variable també canvien els valors de l'altra variable.

$$\rho X, Y = \frac{Cov(X, Y)}{\sigma X \sigma Y}$$

Es calcula utilitzant el coeficient de Pearson, un valor que varia entre -1 i 1, sent -1 la relació negativa perfecta i 1 la relació positiva perfecta [31].

En el codi es va calcular la correlació absoluta entre cada característica i la variable objectiu 'y' utilitzant el coeficient de Pearson. Finalment, es van seleccionar aquelles característiques amb una correlació superior al llindar 0.15, un punt de tall utilitzat per filtrar les característiques en les que hi ha una relació estadística positiva entre les dues variables, considerant que aquesta correlació era suficientment forta per ser considerada rellevant.

3.4.2. Informació Mútua (MI):

La informació mútua o MI és una mesura de la dependència mútua entre variables, indica la quantitat d'informació que una variable conté sobre una altra. A diferència de la correlació, aquest mètode permet treballar relacions no lineals. No pot prendre valors negatius, així sent 0 quan les variables comparades son totalment independents [32].

$$MI(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y)\, log(\frac{p(x,y)}{p(x)p(y)})$$

En el codi es va utilitzar la eina *mutual_info_regression* de la llibreria *sklearn.feature_selection* de Python. Es van seleccionar totes les variables que prenen un valor de MI més gran de 0.02 respecte a la variable a predir (y). La selecció d'aquest llindar va ser empírica i sovint es va justificar per millorar la precisió i reduir el soroll en processos de selecció de característiques [27].

3.4.3. Selecció Supervisada amb RF:

Random Forest crea múltiples arbres de decisió i calcula quines característiques són més útils per dividir les dades en aquests arbres. Les característiques que tenen un impacte més gran en la reducció de la impuresa dels nodes es consideren més importants. Aquestes es seleccionen per formar part del model, mentre que les característiques menys rellevants es descarten, permetent simplificar el model i millorar-ne l'eficiència sense perdre precisió. Aquesta tècnica és molt útil perquè és robusta contra soroll i característiques irrellevants, i no requereix un preprocessament extens de les dades. [33]

En el codi es va utilitzar la eina de *RandomForestClassifier* de la llibreria *sklearn.ensemble* de Python.

3.4.4. ANOVA:

L'ANOVA (Anàlisi de la Variància) és una tècnica estadística que es basa en la distribució F de *Fisher-Snedecor*, utilitzant els graus de llibertat corresponents per determinar si les mitjanes de diferents grups són significativament diferents [34]. Per calcular el valor F s'utilitza la següent formula:

$$F = \frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2}$$

Inicialment, el codi va eliminar les característiques constants, és a dir, aquelles que no tenien variabilitat en les dades, utilitzant la funció *VarianceThreshold*.

A continuació, es va aplicar l'ANOVA a les característiques restants. L'ANOVA va calcular els valors F i els p-values per a cada característica, indicant la seva rellevància per distingir entre les classes de la variable objectiu. Valors F elevats i p-values baixos van suggerir que la característica tenia un impacte significatiu en la predicció.

Amb aquests resultats, el codi va utilitzar la funció *SelectKBest* per seleccionar les millors característiques basant-se en els valors F obtinguts. Finalment, el codi va filtrar les característiques seleccionades amb p-values inferiors a un llindar determinat, en

aquest cas 0.1, un llindar que va relaxar els criteris d'inclusió de variables, especialment en estudis exploratoris o quan es volien identificar tendències potencials que podrien no ser estadísticament significatives amb un llindar més estricte com el 0.05.

3.4.5. LassoCV:

Lasso (Least Absolute Shrinkage and Selection Operator) és un tipus de regressió que, a més d'ajustar un model lineal, introdueix una penalització en la funció de cost que fa que alguns dels coeficients de les característiques es redueixin a zero. Això té l'efecte de seleccionar automàticament les característiques més rellevants i descartar les menys importants, ja que les característiques amb coeficients zero no contribueixen al model.

LassoCV en particular, és una extensió de Lasso que incorpora validació creuada per trobar automàticament el millor valor del paràmetre de regularització, que controla la força de la penalització. [35]

El codi va començar creant el model i ajustant paràmetres com el nombre màxim d'iteracions i la tolerància per assegurar la seva convergència. Aquest model es va entrenar amb les dades escalades per trobar els coeficients associats a cada característica.

Un cop ajustat el model, es van seleccionar les característiques que tenien un coeficient diferent de zero, ja que aquestes eren les que el model considerava més rellevants per a la predicció.

3.4.6. Taula de Presència-Absència:

Un cop es van extreure les característiques més rellevants amb cada mètode, es van organitzar en una taula de presència-absència. En aquesta taula es pot observar per quin mètode va estar seleccionada cada característica. Cada columna correspon a un dels mètodes estadístics emprats, mentre que a les files s'hi troben les característiques. A la intersecció de cada fila i columna s'indica amb un 1 si la característica va estar seleccionada com a important per aquell model o un 0 si va ser descartada.

Aquesta taula ajuda a reconèixer les característiques reconegudes com importants per als diferents mètodes utilitzats. D'aquesta forma, es va fer la selecció final de característiques agafant aquelles que van ser identificades en els cinc mètodes diferents.

3.5. Creació de Models

En aquest treball es van utilitzar els models Random Forest, SVM amb kernel RBF i CNN, degut a la seva capacitat d'abordar problemes de classificació complexos amb dades multidimensionals, combinant robustesa en l'extracció de característiques (Random Forest), optimització en separació no lineal (SVM) i alta eficàcia en el processament d'imatges (CNN).

En tots els models, es va començar dividint el conjunt de dades en dos conjunts: un per entrenar el model (X_train i y_train) i un altre per provar-lo i validar-lo (X_test i y_test). Per a aquesta divisió, es va utilitzar la funció *train_test_split* de la llibreria *sklearn.model_selection*, la qual va permetre separar les dades de manera aleatòria en

percentatges definits. En aquest cas, es va assignar un 80% de les dades per a l'entrenament i un 20% per a la prova.

A continuació, per millorar el rendiment del model, es van escalar les dades utilitzant un *StandardScaler* de la llibreria *sklearn.preprocessing*. Aquest pas va ser crucial, ja que va permetre normalitzar les dades, fent que totes les característiques tinguessin una distribució similar, fet que va contribuir a millorar la convergència del model durant l'entrenament.

Finalment, es va aplicar la validació creuada (*cross-validation*) amb *StratifiedKFold* per avaluar la robustesa dels models. Aquesta tècnica va dividir les dades d'entrenament en múltiples subgrups, assegurant que el model no depengués massa d'un conjunt específic de dades, cosa que va permetre obtenir una avaluació més fiable.

3.5.1. Random Forest

Un cop les dades van estar preparades, es va procedir a ajustar el model de Random Forest. Es va utilitzar la tècnica *RandomizedSearchCV* de la llibreria *sklearn.model_selection*, que va permetre provar automàticament un conjunt aleatori de diverses combinacions d'hiperparàmetres.

Un cop trobada la millor combinació d'hiperparàmetres, es va entrenar el model amb les dades d'entrenament ajustades, obtenint així el model òptim..

3.5.2. SVM amb Kernel RBF

La divisió de les dades en entrenament i prova, així com l'escalat de les dades, es va fer de manera similar al que es va descriure en el model de Random Forest.

Un cop les dades van estar llestes, es va procedir a l'ajust del model *SVM* amb kernel *RBF*. Igualment, es va utilitzar la tècnica de *RandomizedSearchCV* per optimitzar els hiperparàmetres, concretament provant diverses combinacions de *C* i *gamma*. Aquesta optimització va ajudar a trobar la configuració que oferia el millor rendiment, evitant el sobreajustament.

Després d'identificar la millor combinació d'hiperparàmetres, es va entrenar el model amb les dades d'entrenament ajustades.

3.5.3. CNN

Pel que fa al model de xarxes neuronals convolucionals (CNN), les dades es van remodelar en tres dimensions per adaptar-les a l'entrada esperada per la CNN. Es va definir el model utilitzant l'API *Sequential* de *tensorflow.keras*. Aquest model constava de capes convolucionals (*Conv1D*) per extreure característiques rellevants, capes de *max-pooling* (*MaxPooling1D*) per reduir la dimensionalitat, i capes d'entrada i sortida (*Dense*), amb una activació *sigmoid* per realitzar la classificació binària.

El model es va compilar amb l'optimitzador *Adam* i la funció de pèrdua *binary_crossentropy*. L'entrenament es va dur a terme durant 20 *epochs*, monitoritzant el rendiment en el conjunt de prova i guardant la història d'entrenament per a la seva anàlisi posterior.

3.6. Validació dels Models

La validació dels models és un procés fonamental en la ciència de dades i l'aprenentatge automàtic, ja que permet avaluar la qualitat i el rendiment d'un model abans d'aplicar-lo en situacions reals. Dins d'aquest procés, hi ha diverses mètriques que ajuden a mesurar la precisió del model, sent la sensibilitat i l'especificitat dues de les més importants.

La sensibilitat es defineix com la capacitat del model per identificar correctament els pacients que responen al tractament, les instàncies positives. En altres paraules, ens diu quin percentatge de les mostres positives reals són identificades correctament pel model com a positives. Una alta sensibilitat indica que el model és molt bo a detectar els casos positius, minimitzant els falsos negatius.

$$Sensibilitat = \frac{Veritables\ Positius}{Veritables\ Positius + Falsos\ Negatius}$$

D'altra banda, l'especificitat mesura la capacitat del model per identificar correctament els pacients que no han respost efectivament al tractament, les instàncies negatives. Això vol dir que ens indica quin percentatge de les mostres negatives reals són classificades correctament com a negatives pel model. Una alta especificitat indica que el model és eficient a evitar falsos positius, és a dir, a no classificar erròniament com a positives les instàncies que realment són negatives.

$$Especificitat = \frac{Veritables\ Negatius}{Veritables\ Negatius + Falsos\ Positius}$$

Principalment es va utilitzar la sensibilitat i la especificitat pel seu ampli ús en aplicacions mèdiques, però també es van estudiar altres mètriques per mesurar la qualitat i el rendiment del model.

La precisió és la proporció de prediccions positives correctes respecte a totes les prediccions positives que ha fet el model. És a dir, mesura la qualitat de les prediccions positives. La precisió òptima s'assoleix quan el model fa la màxima quantitat de prediccions positives correctes amb el mínim nombre de falsos positius.

L'F1-score és la mitjana harmònica entre la precisió i la sensibilitat. Aquesta mètrica combina les dues mesures en un únic valor, proporcionant un balanç entre les dues.

L'exactitud (accuracy) és la proporció de prediccions correctes (tant positives com negatives) respecte al total de prediccions fetes pel model. Mesura l'eficiència global del model. L'exactitud òptima és el valor màxim d'aquesta mètrica, i indica que el model ha classificat correctament el major nombre possible d'instàncies.

Per a la validació del model, es van avaluar diverses tècniques i eines, incloent-hi la matriu de confusió, la corba ROC (Receiver Operating Characteristic) i la corba d'aprenentatge.

La matriu de confusió es tracta d'una taula de doble entrada que permet comparar les prediccions fetes pel model amb les classes reals, organitzant-les en quatre categories:

- Veritables Positius (TP): Nombre d'exemples correctament classificats com a positius.
- Falsos Positius (FP): Nombre d'exemples incorrectament classificats com a positius.
- Veritables Negatius (TN): Nombre d'exemples correctament classificats com a negatius.
- Falsos Negatius (FN): Nombre d'exemples incorrectament classificats com a negatius.

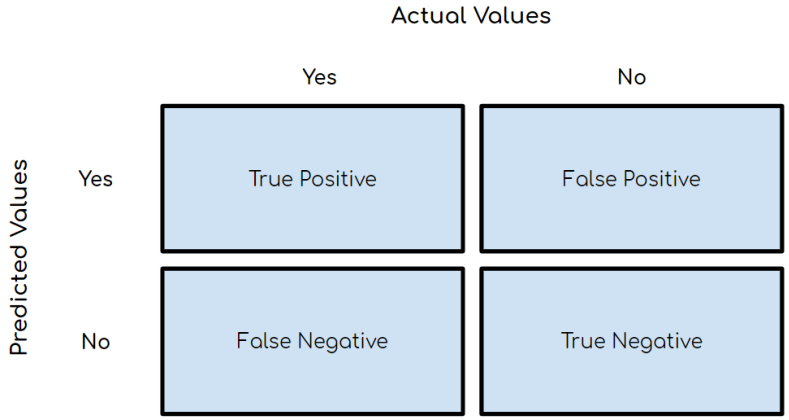

Figura 9: Estructura de la matriu de confusió [36].

A més, a partir d'aquesta matriu, es poden derivar diverses mètriques de rendiment, com ara la precisió (precision), la sensibilitat (recall), l'especificitat, i l'F1-score.

La corba ROC (Receiver Operating Characteristic) representa la relació entre la taxa de falsos positius (1-especificitat) i la taxa de veritables positius (sensibilitat) a mesura que es varia el llindar de decisió. L'Àrea sota la corba ROC (AUC) és un valor que indica la capacitat del model per distingir entre les classes positives i negatives. Un AUC més proper a 1 indica un model amb una millor capacitat de discriminació. Un AUC de 0.5 indica que el model no té capacitat de discriminació, equivalent a fer una classificació aleatòria [37].

La corba d'aprenentatge és un gràfic que representa l'evolució del rendiment d'un model de Machine Learning a mesura que es proporciona més dades d'entrenament. Aquesta corba es compon de dues línies principals:

1. Rendiment en les dades d'entrenament: Es mostra com de bé el model s'ajusta a les dades que ha vist durant l'entrenament. Quan es comença amb poques dades, el rendiment pot ser baix, però a mesura que augmenta la quantitat de dades, aquesta línia tendeix a pujar, indicant que el model està aprenent i millorant la seva capacitat de predicció en les dades conegudes.
2. Rendiment en les dades de validació o prova: Es mostra com de bé el model generalitza a noves dades que no ha vist abans. Idealment, aquesta línia també hauria de pujar a mesura que augmenta la mida de l'entrenament, indicant que el model no només s'ajusta bé a les dades d'entrenament, sinó que també és capaç de fer bones prediccions en dades noves.

Si ambdues línies es mantenen properes i amb valors elevats, això indica que el model està aprenent de manera efectiva i generalitza bé. Però si la línia d'entrenament és alta mentre que la de validació es queda baixa, es mostra un possible sobreajustament, on el model ha après massa bé els detalls de les dades d'entrenament, però falla en generalitzar. Al contrari, si ambdues línies es mantenen baixes, això indica un infraajustament, suggerint que el model és massa simple per captar els patrons adequats de les dades [38].

Aquestes eines conjuntament proporcionen una visió completa del rendiment del model, permetent avaluar-lo i millorar-lo abans d'aplicar-lo a dades no vistes o en aplicacions del món real.

En el codi referenciat en l'Annex IV, es va avaluar el model amb les dades de prova (X_test). Es va generar la corba ROC utilitzant la funció *roc_curve* de *sklearn.metrics*, la qual va ajudar a determinar el llindar òptim per a les prediccions. La funció *auc*, també de *sklearn.metrics*, es va utilitzar per calcular l'àrea sota la corba ROC, una mètrica que va proporcionar una visió general del rendiment del model. La corba ROC i l'AUC es van visualitzar amb *matplotlib.pyplot*.

La sensibilitat (o *recall*), l'especificitat, la precisió, l'F1-score i l'exactitud es van obtenir utilitzant funcions com *precision_score*, *accuracy_score* i *f1_score* de *sklearn.metrics*. A més, es va generar una matriu de confusió utilitzant *confusion_matrix*.

4. Resultats

4.1. Característiques Seleccionades

Després d'obtenir la taula de presència-absència, es van escollir com a rellevants aquelles característiques que s'identificaven en cada un dels cinc mètodes utilitzats per la selecció de característiques.

	Correlació	Informació mútua	Random Forest	ANOVA	LassoCV
original_shape_Elongation	0	1	1	0	0
original_shape_Flatness	0	0	0	0	0
original_shape_LeastAxisLength	0	0	0	0	0
original_shape_MajorAxisLength	0	1	1	0	0
original_shape_Maximum2DDiameterColumn	0	1	1	0	0
original_shape_Maximum2DDiameterRow	0	0	0	0	0
original_shape_Maximum2DDiameterSlice	0	0	0	0	0
original_shape_Maximum3DDiameter	0	0	1	0	0
original_shape_MeshVolume	0	1	0	0	0
original_shape_MinorAxisLength	0	1	0	0	0
original_shape_Sphericity	0	1	1	0	0
original_shape_SurfaceArea	0	1	0	0	0

Taula 2: Taula de presència-absència per algunes de les característiques radiòmiques.

Així doncs, es van reduir a un total de 12 característiques importants.

```
Variables seleccionades: ['original_glszm_GrayLevelVariance', 'wavelet-LLH_firstor
der_Median', 'wavelet-LHL_firstorder_Maximum', 'wavelet-HLL_glcm_ClusterProminenc
e', 'wavelet-HHH_gldm_DependenceNonUniformityNormalized', 'square_glrlm_RunEntrop
y', 'squareroot_firstorder_Kurtosis', 'exponential_glcm_Imc1', 'gradient_ngtdm_Str
ength', 'lbp-2D_glszm_ZoneVariance', 'lbp-3D-m1_glszm_ZoneVariance', 'lbp-3D-m2_gl
szm_ZoneVariance']
Nombre de variables seleccionades: 12
```

Figura 10: Característiques seleccionades.

1. **original_glszm_GrayLevelVariance**: Mesura la variabilitat en els nivells de gris dins de les zones de l'estructura estudiada. Un valor alt indica una gran variabilitat en els nivells de gris.
2. **wavelet-LLH_firstorder_Median**: És el resultat d'aplicar una transformada wavelet a l'imatge, que descompon la imatge en diferents freqüències. LLH es refereix a una de les bandes resultant. *Median* és la mediana de les intensitats dels píxels en aquesta banda.
3. **wavelet-LHL_firstorder_Maximum**: Similar a l'anterior, però aquesta vegada es refereix a la banda LHL de la transformada wavelet. *Maximum* representa el valor màxim d'intensitat en aquesta banda.
4. **wavelet-HLL_glcm_ClusterProminence**: És una mesura del grau d'asimetria en la distribució de la matriu de concurrència de nivells de grisos (GLCM) en la banda HLL. Això proporciona informació sobre la textura de l'estructura.
5. **wavelet-HHH_gldm_DependenceNonUniformityNormalized**: Mesura la uniformitat en la distribució de les dependències de píxels dins de la imatge, normalitzat per la transformada wavelet HHH.
6. **square_glrlm_RunEntropy**: Mesura l'entropia de la longitud de les sèries consecutives de píxels amb la mateixa intensitat dins de la imatge transformada quadrada. L'entropia alta indica una textura complexa.
7. **squareroot_firstorder_Kurtosis**: És una mesura estadística que descriu la distribució d'intensitats en la imatge, en aquest cas després d'haver aplicat una transformació d'arrel quadrada.
8. **exponential_glcm_Imc1**: És una mesura derivada de la GLCM que quantifica la correlació entre parelles de nivells de gris en la imatge després d'aplicar una transformació exponencial.
9. **gradient_ngtdm_Strength**: És una característica del model de dependència de to de grisos de veïnatge (NGTDM), que mesura la força de les transicions de nivell de grisos en la imatge basada en el gradient.
10. **lbp-2D_glszm_ZoneVariance**: Mesura la variabilitat de les zones dins de la imatge processada amb el mètode Local Binary Pattern (LBP) 2D.
11. **lbp-3D-m1_glszm_ZoneVariance**: Similar a l'anterior, però aplicada a imatges en 3D amb un model específic m1.
12. **lbp-3D-m2_glszm_ZoneVariance**: Similar a l'anterior, però amb un model m2.

[39][40][41]

Es va calcular la correlació de les característiques seleccionades com a rellevants amb la resposta:

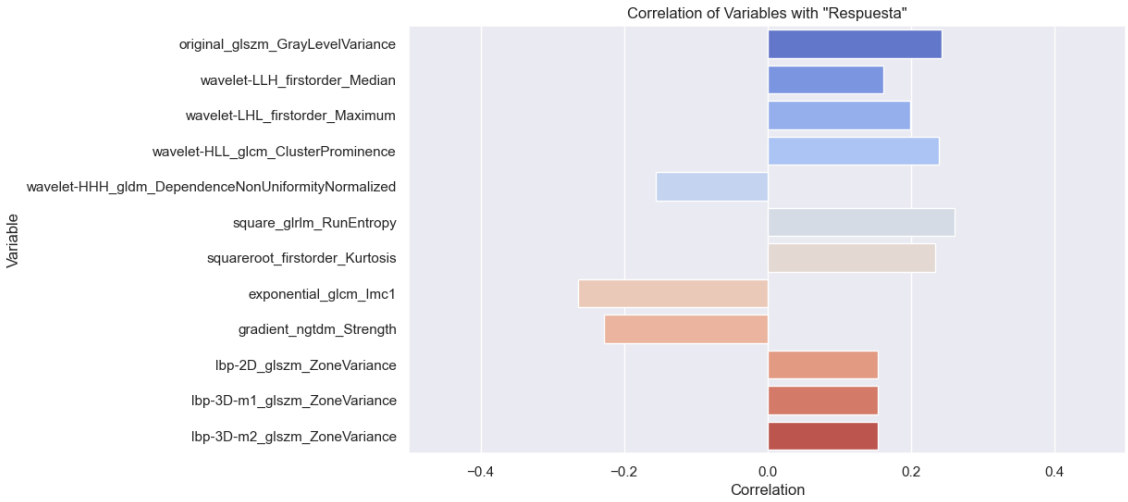

Figura 11: Correlació de les dades seleccionades amb la variable resposta.

Aquest gràfic mostra la correlació entre les característiques radiòmiques seleccionades i la resposta dels pacients al tractament de radioteràpia a dosis baixes. En aquest cas, cada barra del gràfic representa la força i la direcció de la correlació entre una característica radiòmica i la probabilitat que un pacient respongui (1) o no respongui (0) al tractament.

Les barres que van direcció cap a la dreta, indiquen una correlació positiva. Això vol dir que com més alt sigui el valor de la característica en qüestió, més probabilitats hi ha que el pacient respongui positivament al tractament. Per exemple, una característica com "*original_glszm_GrayLevelVariance*" mostra una correlació positiva significativa, suggerint que els pacients amb valors més alts d'aquesta característica tenen més probabilitats de veure's beneficiats pel tractament.

Les barres que van direcció cap a l'esquerra indiquen una correlació negativa. Això significa que com més alta sigui aquesta característica, menys probable és que el pacient respongui al tractament. Característiques com "*lbp-3D-m2_glszm_ZoneVariance*" tenen una correlació negativa destacada, la qual cosa suggereix que un pacient amb valors elevats d'aquesta característica podria no respondre bé al tractament.

4.2. Models Predictius

A través de l'aplicació de diferents tècniques de Machine Learning, com el Random Forest, SVM amb kernel RBF, i CNN, es va buscar determinar quina és la metodologia més efectiva per predir amb precisió els resultats del tractament. Per això, es van comparar els models mitjançant mètriques com la corba ROC, AUC i la matriu de confusió per determinar la metodologia més precisa.

Seguidament es poden veure les diferents corbes ROC:

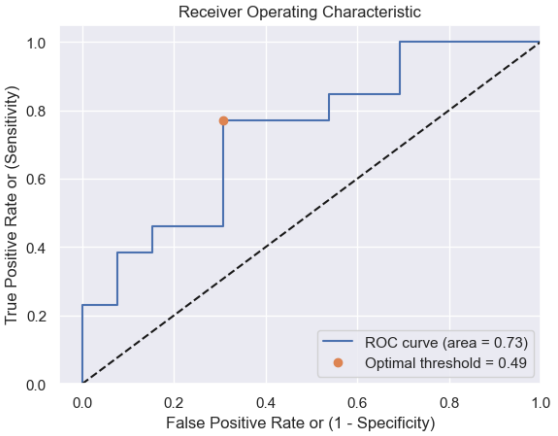

Figura 12: Corba ROC del model Random Forest.

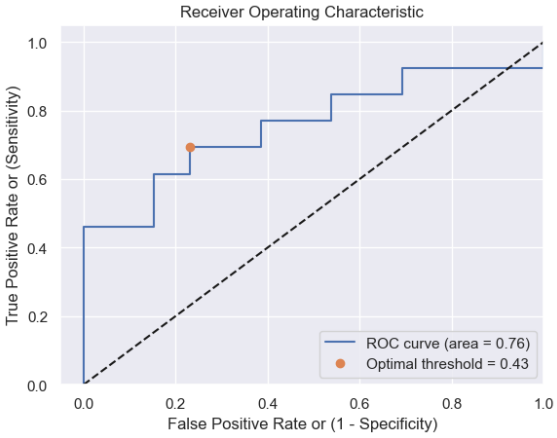

Figura 13: Corba ROC del model SVM amb kernel RBF.

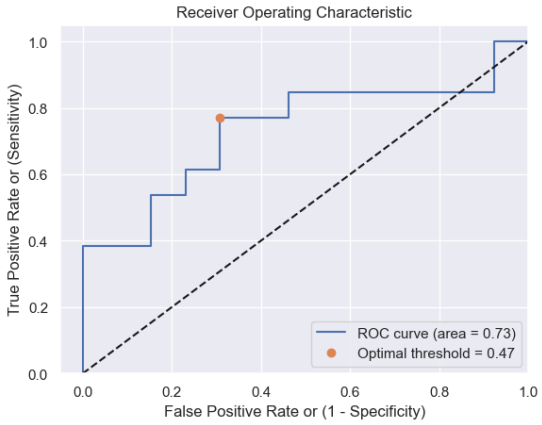

Figura 14: Corba ROC del model CNN.

En revisar les corbes ROC i les AUC dels tres models es van observar algunes diferències en la seva capacitat per distingir entre les classes. El model Random Forest va presentar una AUC de 0,73, la qual cosa va indicar que tenia una capacitat raonable per discriminar entre les classes positives i negatives, tot i que no era excepcional. Això es va reflectir en la corba ROC, que no s'apropava gaire a l'ideal, situada a la cantonada superior esquerra.

D'altra banda, el model SVM amb kernel RBF va millorar lleugerament, amb una AUC de 0,76. Aquesta lleugera millora va indicar que aquest model era una mica més eficaç en distingir entre les classes, cosa que es va manifestar en una corba ROC més propera a la cantonada ideal. Això va suggerir una major capacitat per mantenir una bona relació entre la taxa de vertaders positius i la de falsos positius.

Finalment, el model CNN va mostrar una AUC de 0,73. Tot i treballar amb una quantitat de dades limitada, el CNN va aconseguir una discriminació entre classes igual a la de Random Forest.

En conjunt, les corbes ROC i les AUC van mostrar que l'SVM amb kernel RBF era el més eficient a l'hora de discriminar entre classes. El model CNN i el model Random

Forest també van oferir bons resultats tot i quedar-se lleugerament per darrere en aquesta tasca específica.

Després es va decidir estudiar el punt òptim o el millor valor llindar (threshold) de la corba ROC per obtenir la matriu de confusió i, finalment, calcular els valors de diferents mètriques com la sensibilitat i l'especificitat. Per determinar el punt òptim, es va seleccionar el valor llindar amb el qual s'aconseguia un f1-score més alt, és a dir, el valor en què aquest índex era màxim. En aquestes matrius es van poder veure els casos que un model havia predit correctament (vertaders positius i negatius) i quants havia predit incorrectament (falsos positius i negatius).

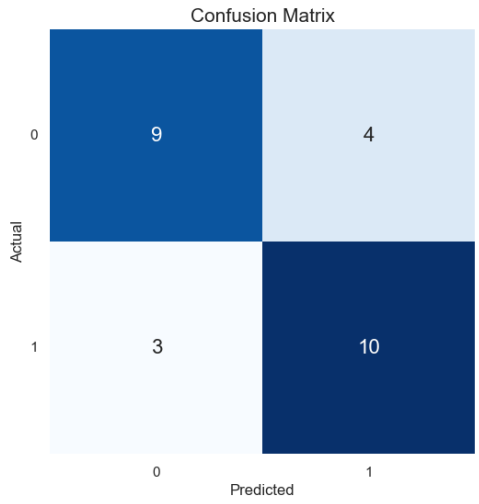

Figura 15: Matriu de confusió del model Random Forest.

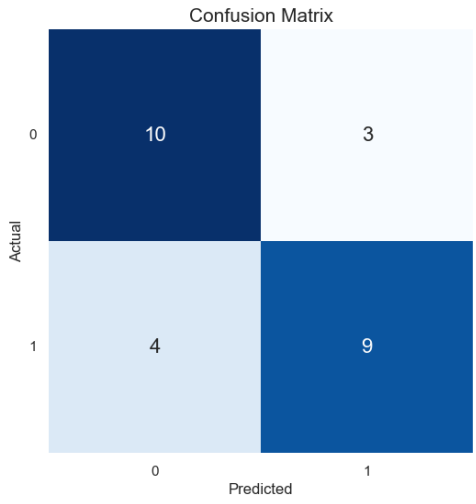

Figura 16: Matriu de confusió del model SVM amb kernel RBF.

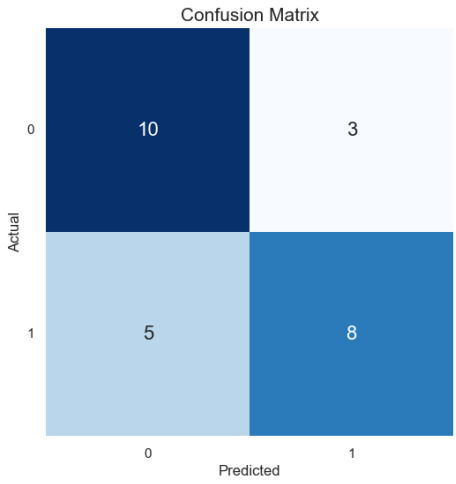

Figura 18: Matriu de confusió del model CNN.

En analitzar les matrius de confusió dels tres models, Random Forest, SVM amb kernel RBF, i CNN, es van observar algunes similituds i diferències notables en els seus rendiments. Aquestes matrius avaluaven els casos del conjunt de validació, és a dir, de les dades independents no vistes durant l'entrenament.

El model Random Forest va mostrar un bon equilibri entre la capacitat de predir correctament tant la classe positiva com la negativa. Va aconseguir identificar correctament 10 casos de la classe 1 i 9 de la classe 0. No obstant això, va cometre 4 errors en predir la classe 0 com a classe 1, i 3 errors en predir la classe 1 com a classe 0. Això va indicar una lleugera tendència a confondre la classe 0 amb la classe 1.

El model SVM amb kernel RBF va presentar un rendiment molt similar al Random Forest. Va identificar correctament 9 casos de la classe 1 i 10 de la classe 0. Va cometre 3 errors en predir la classe 0 com a classe 1 i 4 errors en predir la classe 1 com a classe 0. Tot i que el comportament era gairebé idèntic, es podria dir que aquest model va mostrar una lleugera millora en la predicció de la classe 0 en comparació amb Random Forest.

Pel que fa al model CNN, també va presentar un rendiment molt semblant als altres dos models. Va identificar correctament 8 casos de la classe 1 i 10 de la classe 0, cometent 3 errors en predir la classe 0 com a classe 1 i 5 errors en predir la classe 1 com a classe 0. Aquest resultat va suggerir que, tot i que el CNN és conegut per necessitar grans quantitats de dades per rendir de manera òptima, amb les dades disponibles va aconseguir mantenir un rendiment comparable als altres models.

En resum, les matrius de confusió van mostrar que els tres models tenien un rendiment molt similar, amb un lleuger biaix a l'hora de confondre la classe 0 amb la classe 1. Cap dels models es va destacar clarament per sobre dels altres només basant-se en aquestes matrius, i per això es va considerar prudent tenir en compte altres mètriques.

Així doncs, per acabar de validar els models es van calcular les següents mètriques:

	Random Forest	**SVM RBF**	**CNN**
Threshold	0.489	0.432	0.469
Sensibilitat (Recall)	0.769	0.692	0.615
Especificitat	0.692	0.769	0.769
Precisió	0.714	0.750	0.727
F1-score	0.741	0.720	0.667
Exactitud (Accuracy)	0.731	0.731	0.692
ROC AUC	0.734	0.757	0.734

Taula 3: Resultats per als diferents models predictius.

5. Discussió

En aquest treball s'han avaluat diferents models d'intel·ligència artificial, basats en radiòmica, per predir la resposta de pacients amb artrosi a les mans que es tracten amb radioteràpia a baixes dosis.

Aquest treball és el primer estudi sobre la predicció de la resposta de la radioteràpia a dosis baixes en pacients amb artrosi, marcant un avenç pioner en aquesta àrea de la medicina, així oferint una base de coneixement crucial que pot guiar futures investigacions.

Un dels factors destacables d'aquest treball és la creació d'un codi que ha automatitzat completament el procés d'extracció de característiques radiòmiques utilitzant Python, sense necessitat de plataformes externes o intervenció manual. Aquesta automatització no només ha accelerat el procés d'anàlisi, sinó que també ha assegurat una major consistència i precisió en la selecció de característiques, contribuint de manera significativa a la robustesa dels models predictius desenvolupats.

Cal destacar que la selecció de característiques radiòmiques ha estat especialment efectiva gràcies a la combinació de cinc mètodes: correlació, informació mútua, Random Forest, ANOVA i LassoCV. Cada mètode ha aportat avantatges únics: la correlació ha permès identificar les relacions lineals més fortes, mentre que la informació mútua ha capturat relacions no lineals importants. El Random Forest ha ajudat a destacar les característiques més rellevants per a la classificació, l'ANOVA ha assegurat que les variables seleccionades tinguessin una relació estadísticament significativa amb la resposta, i el LassoCV ha contribuït a evitar el sobreajustament mantenint el model simple i eficient. Aquesta combinació ha resultat en una selecció de característiques precisa i robusta, millorant la capacitat predictiva del model i assegurant una millor generalització a noves dades.

Els resultats de la selecció de característiques radiòmiques reflecteix un enfocament detallat en la captura de la complexitat textural i estructural de les imatges mitjançant tècniques avançades com la transformada Wavelet i els patrons binaris locals (LBP). La diversitat de mesures, que inclouen variabilitat, entropia i asimetria, subratlla la importància de modelar tant la distribució d'intensitats com la textura de les zones imatges. Això pot proporcionar un avantatge significatiu en la predicció clínica, ja que aquestes característiques capturen informació que les variables clíniques per si soles podrien passar per alt.

La no selecció de variables clíniques com l'edat i el sexe com a rellevants podria ser atribuïda tant a un desequilibri en la distribució del sexe com a una limitada variabilitat en les edats dels pacients, factors que redueixen la capacitat del model per identificar aquestes variables com a influents en la predicció. Això subratlla la importància de considerar la distribució de les dades i la disposició d'una major varietat de dades clíniques.

Un cop creat els models, els resultats obtinguts a partir de les matrius de confusió mostren que tots tres models tenen un rendiment similar, amb petites variacions que reflecteixen les seves fortaleses i limitacions.

El Random Forest ha mostrat una gran capacitat per identificar correctament els casos positius, destacant per la seva alta sensibilitat amb un valor de 0.769, el més alt dels tres models, el que el fa especialment útil quan és crucial maximitzar la detecció de pacients que respondran positivament al tractament. També té un bon equilibri en termes de F1-score, aconseguint un valor de 0.741, que el fa molt fiable en prediccions globals. Tot i això, també ha presentat una lleugera tendència a confondre alguns casos negatius, amb una especificitat de 0.692, fet que indica una àrea de millora en aquest aspecte. Aquest model ofereix un bon equilibri entre sensibilitat i precisió, sent adequat per captar el màxim nombre de respostes positives al tractament.

El model SVM amb kernel RBF ha demostrat ser el més equilibrat dels tres, amb una especificitat de 0.769, la més alta juntament amb el model CNN, i una precisió de 0.750, la qual cosa significa que quan prediu sol tenir raó amb més freqüència que els altres models. Aquest balanç entre alta especificitat i precisió consistent el converteix en una opció robusta per a contextos clínics on és important mantenir un bon equilibri entre la detecció de casos positius i la minimització de falsos positius. Amb una ROC AUC de 0.757, que és el millor indicador global de la seva capacitat de discriminació entre classes positives i negatives, aquest model podria ser preferible en situacions on la reducció d'errors de classificació és fonamental.

El model CNN ha mostrat un rendiment decent en precisió, amb un valor de 0.727, la qual cosa indica que el model sol fer prediccions correctes amb freqüència. No obstant això, el seu F1-score, amb un valor de 0.667, reflecteix que la combinació entre precisió i sensibilitat no és tan elevada com en els altres models, la qual cosa podria ser atribuïda a la petita mida del conjunt de dades. Els models CNN solen requerir grans volums de dades per entrenar-se de manera efectiva, fet que podria limitar el seu rendiment en aquest cas. Malgrat això, el model ha aconseguit una especificitat alta de 0.769, indicant que és efectiu en la identificació de la classe negativa. La seva ROC AUC, amb un valor de 0.734, similar al del model Random Forest, suggereix que té una bona capacitat per distingir entre classes, tot i que lleugerament inferior al model SVM amb kernel RBF (0.757). Aquesta capacitat pot no ser tan consistent o eficient com la d'altres models, però el CNN ofereix un bon equilibri en termes de discriminació entre classes.

Així doncs, si es busca un model equilibrat i fiable en diverses situacions, l'SVM amb kernel RBF sembla ser la millor opció, especialment perquè maximitza la precisió (0.750) i té una ROC AUC més alta (0.757), indicant una bona capacitat global de discriminació entre classes. Si l'objectiu principal és captar el màxim nombre de positius correctes, el Random Forest, amb la seva sensibilitat òptima de 0.769, seria preferible. D'altra banda, el CNN, tot i tenir una ROC AUC com la de Random Forest, ofereix un bon equilibri en termes de precisió (0.727) i F1-score (0.667), però no arriba a igualar el rendiment de l'SVM. Encara que el CNN té un bon comportament en termes d'especificitat i discriminació entre classes, no és tan adequat com l'SVM per a aquesta tasca específica, donat que els altres models aconsegueixen millors resultats globals.

En cas d’aplicar aquests models en la pràctica clínica, seria crucial prioritzar aquells models que minimitzin el nombre de falsos positius (FP). Un fals positiu implica que el model prediu incorrectament que un pacient respondrà al tractament, quan en realitat no ho farà. Això pot portar a que pacients siguin sotmesos a un tractament innecessari,

exposant-los als possibles efectes secundaris i costos associats sense cap benefici real. Aquesta situació és especialment preocupant perquè, a més de ser ineficaç, també pot retardar l'inici d'altres tractaments que podrien ser més adequats per a aquests pacients.

D'altra banda, tot i que els falsos negatius (FN) també són un problema, la seva gravetat pot ser menor en aquest context clínic. Un fals negatiu significa que el model no prediu una resposta positiva quan en realitat aquesta podria existir. En aquests casos, el pacient podria rebre altres opcions terapèutiques sota el criteri dels metges, i si aquests tractaments alternatius no fossin efectius, sempre es podria reconsiderar la radioteràpia com una opció. Així doncs, el risc d'un FN es pot gestionar més fàcilment, ja que hi ha l'oportunitat de replantejar el tractament en base a l'evolució del pacient.

Per tant, basant-se en aquesta consideració, el model SVM amb kernel RBF emergeix com el candidat preferit, ja que ofereix un bon equilibri entre la precisió i l'especificitat, reduint així la probabilitat de falsos positius. Aquesta característica el converteix en una opció robusta i fiable per a la predicció de la resposta al tractament, assegurant que només els pacients que tenen més probabilitats de beneficiar-se de la radioteràpia siguin sotmesos a aquest tractament. Així, es maximitza l'eficàcia clínica i es minimitzen els riscos d'intervencions innecessàries.

Els resultats de l'estudi permeten personalitzar el tractament, millorant-ne l'eficàcia i reduint els efectes secundaris. Aquesta personalització ajuda a assegurar que cada pacient rebi la teràpia més adequada a les seves característiques específiques, incrementant les possibilitats d'èxit. A més, la nostra investigació contribueix significativament a la medicina de precisió, obrint noves vies per comprendre millor els mecanismes biològics que influeixen en la resposta a la radioteràpia a baixes dosis. Aquest avenç no només té el potencial de millorar la qualitat de vida dels pacients, sinó que també optimitza l'ús dels recursos sanitaris, fent-los més eficients i efectius.

En l'estudi s'han presentat algunes limitacions. Una de les principals limitacions d'aquest estudi és la mida reduïda de la mostra, que pot limitar la generalització dels resultats. Per superar aquesta limitació, seria essencial disposar de mostres més grans, possibilitant col·laboracions multicèntriques que facilitin el compartiment de dades. No obstant això, aquesta estratègia presenta reptes importants, ja que les dades mèdiques són altament privades i estan protegides per lleis estrictes, complicant el seu intercanvi i compartiment segur entre institucions.

S'ha observat que el conjunt de dades amb què s'ha treballat no estava ben balancejat. Això ha fet necessari utilitzar mètodes per compensar aquest desequilibri i permetre entrenar els models de manera consistent i uniforme en tots els casos. Tanmateix, aquest desequilibri també té un aspecte positiu, ja que reflecteix que, en la majoria dels casos d'artrosi a les mans, el tractament ha tingut una resposta positiva.

Es planteja la necessitat de millorar el procés de recollida d'informació sobre l'estat de la malaltia dels pacients, ja que s'ha detectat una manca d'escales objectives i qüestionaris estandarditzats per avaluar l'evolució de l'artrosi durant i després del tractament. Aquesta absència d'eines d'avaluació objectiva pot limitar la capacitat de mesurar amb precisió la resposta dels pacients. Per abordar aquesta mancança, es proposa la introducció de mesures d'avaluació objectiva i sistemàtica, com escales específiques per a l'artrosi i

qüestionaris validats que permetin una valoració detallada del dolor, la funcionalitat i la qualitat de vida dels pacients. Aquestes eines proporcionaran dades més consistents i fiables, facilitant així un millor anàlisi dels resultats i una presa de decisions més informada i personalitzada.

No obstant això, cal destacar que encara queda molt per investigar en aquest àmbit. És essencial continuar amb estudis que incorporin dades més objectives i variades, augmentant el nombre de dades clíniques estudiades i combinar-les amb altres tipus de característiques, com per exemple metabolòmiques. Aquestes dades complementàries podrien enriquir encara més la nostra capacitat predictiva i oferir una millor predicció de l'efectivitat del tractament.

6. Conclusions

Aquest treball ha abordat de manera exhaustiva el potencial de l'anàlisi radiòmica a partir d'imatges TAC per predir la resposta dels pacients amb artrosi que s'han tractat amb RTDB, un àmbit poc explorat fins ara. A través de l'anàlisi de diferents models predictius basats en tècniques de Machine Learning i la selecció de característiques radiòmiques, s'ha avançat en la comprensió de com aquests enfocaments poden contribuir a millorar la presa de decisions clíniques.

Després d'avaluar diversos models de Machine Learning, s'ha identificat un rendiment superior en el model SVM amb kernel RBF per a la predicció de la resposta al tractament. Aquest, és un model automàtic i escalable que ha demostrat una major efectivitat en la classificació precisa dels pacients segons la seva resposta al tractament, destacant per la seva capacitat d'equilibrar la sensibilitat i l'especificitat, fet que el fa especialment adequat per a contextos clínics.

L'aplicabilitat clínica d'aquests resultats és prometedora, ja que aquest model es pot considerar una eina útil per a la presa de decisions en el tractament de pacients amb artrosi. Tot i els resultats prometedors, cal reconèixer que aquest estudi pioner en el seu camp, presenta algunes limitacions, com la mida reduïda de la mostra, la variabilitat en la resposta al tractament i el desequilibri en el conjunt de dades. Aquestes limitacions indiquen la necessitat de conjunts de dades més amplis, incorporant altres tipus de dades més variades per millorar la capacitat predictiva dels models.

Finalment, és important continuar investigant en el camp de la RTDB, ja que tot i el risc de càncer induït, nombrosos estudis han demostrat l'efectivitat del tractament. Aquest estudi mostra que la radiòmica permet predir millor la resposta al tractament, assegurant-nos que el pacient realment en rebrà un benefici abans d'irradiar. Seguir explorant aquest camp permetrà oferir als pacients un tractament que millori la seva qualitat de vida d'una manera segura i efectiva.

7. Referències

[1] Artrosis - Farreras Rozman. Medicina Interna - ClinicalKey Student [Internet]. Disponible a: https://www.clinicalkey.com/student/content/book/3-s2.0-B9788413824864001207

[2] Conoce los grados de la artrosis de rodilla [Internet]. Disponible a: https://www.clinicacellus.cl/estos-son-los-grados-de-la-artrosis-de-rodilla/

[3] Artrosis | Causas, Síntomas y Tratamiento | PortalCLÍNIC [Internet]. Disponible a: https://www.clinicbarcelona.org/asistencia/enfermedades/artrosis

[4] Hochberg MC, Altman RD, April KT, Benkhalti M, Guyatt G, McGowan J, et al. American College of Rheumatology 2012 recommendations for the use of nonpharmacologic and pharmacologic therapies in osteoarthritis of the hand, hip, and knee. Arthritis Care Res (Hoboken). abril 2012;64(4):465-74.

[5] Hunter DJ, Bierma-Zeinstra S. Osteoarthritis. Lancet [Internet]. 27 abril 2019;393(10182):1745-59. Disponible a: https://pubmed.ncbi.nlm.nih.gov/31034380/

[6] An update on the epidemiology of knee and hip osteoarthritis with a view to prevention [Internet]. Disponible a: https://onlinelibrary.wiley.com/doi/epdf/10.1002/1529-0131%28199808%2941%3A8%3C1343%3A%3AAID-ART3%3E3.0.CO%3B2-9

[7] Moskowitz R, Altman R, Hochberg M, Buckwalter J, Goldberg V. Chapter 21B: Lower Extremity Considerations: Knee. Osteoarthritis: Diagnosis and Medical/surgical Management [Internet]. 4th Editio. 2007;394-414. Disponible a: https://books.google.com/books/about/Osteoarthritis.html?id=YfFj8Gbq5H0C

[8] Aerts HJWL, Rios Velazquez E, Leijenaar RTH, Parmar C, Grossmann P, Carvalho S, et al. ARTICLE Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. 2014; Disponible a: www.nature.com/naturecommunications

[9] Zwanenburg A, Vallières M, Abdalah MA, Aerts HJWL, Andrearczyk V, Apte A, et al. The image biomarker standardization initiative: Standardized quantitative radiomics for high-throughput image-based phenotyping. Radiology [Internet]. 1 maig 2020;295(2):328-38. Disponible a: https://doi.org/10.1148/radiol.2020191145

[10] Haider SP, Burtness B, Yarbrough WG, Payabvash S. Applications of radiomics in precision diagnosis, prognostication and treatment planning of head and neck squamous cell carcinomas. Disponible a: https://doi.org/10.1186/s41199-020-00053-7

[11] Dove APH, Cmelak A, Darrow K, Mccomas KN, Chowdhary M, Beckta J, et al. The Use of Low-Dose Radiation Therapy in Osteoarthritis: A Review. Int J Radiat Oncol Biol Phys [Internet]. 2022;114:203-20. Disponible a: https://doi.org/10.1016/j.ijrobp.2022.04.029

[12] Reviving Low-Dose Radiation Therapy for Osteoarthritis - PRIMR [Internet]. Disponible a: https://www.primrmed.com/blog-post/the-untapped-potential-of-a-century-old-treatment-reviving-low-dose-radiation-therapy-for-osteoarthritis

[13] Tian R, Duan · Xudong, Fangze Xing ·, Zhao Y, Liu C, Li · Heng, et al. Computed tomography radiomics in predicting patient satisfaction after robotic-assisted total knee arthroplasty. Int J Comput Assist Radiol Surg [Internet]. 2024; Disponible a: https://doi.org/10.1007/s11548-024-03192-1

[14] Xue Z, Wang L, Sun Q, Xu J, Liu Y, Ai S, et al. Radiomics analysis using MR imaging of subchondral bone for identification of knee osteoarthritis. J Orthop Surg Res [Internet]. 1 desembre 2022;17(1). Disponible a: https://pubmed.ncbi.nlm.nih.gov/36104732/

[15] Modelos predictivos ¿Qué son y para qué se usan? - Nuclio School [Internet]. Disponible a: https://nuclio.school/blog/modelos-predictivos-que-son-y-usos/

[16] Overfitting. Qué es, causas, consecuencias y cómo solucionarlo | Grupo Atico34 [Internet]. Disponible a: https://protecciondatos-lopd.com/empresas/overfitting/#Que_es_el_overfitting_en_el_aprendizaje_automatico

[17] Breiman L. Random Forests. Mach Learn [Internet]. 1 octubre 2001; 45(1):5-32. Disponible a: https://typeset.io/papers/random-forests-z67p5gbtbb

[18] Anas Brital | Random Forest Algorithm Explained. [Internet]. Disponible a: https://anasbrital98.github.io/blog/2021/Random-Forest/

[19] Schölkopf B, Smola AJ. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. 7 desembre 2001; Disponible a: https://direct.mit.edu/books/monograph/1821/Learning-with-KernelsSupport-Vector-Machines

[20] The RBF kernel in SVM: A Complete Guide - PyCodeMates [Internet]. Disponible a: https://www.pycodemates.com/2022/10/the-rbf-kernel-in-svm-complete-guide.html

[21] Tychola KA, Kalampokas T, Papakostas GA. Quantum Machine Learning—An Overview. Electronics (Switzerland). 1 juny 2023;12(11).

[22] Lecun Y, Bengio Y, Hinton G. Deep learning. Nature [Internet]. 27 maig 2015;521(7553):436-44. Disponible a: https://www.researchgate.net/publication/277411157_Deep_Learning

[23] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. Med Image Anal. 1 desembre 2017;42:60-88.

[24] Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, et al. Dermatologist-level classification of skin cancer with deep neural networks HHS Public Access. Nature [Internet]. 2017;542(7639):115-8. Disponible a: www.nature.com/reprints.

[25] Salehi AW, Khan S, Gupta G, Alabduallah BI, Almjally A, Alsolai H, et al. A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope. Sustainability 2023, Vol 15, Page 5930 [Internet]. 29 març 2023;15(7):5930. Disponible a: https://www.mdpi.com/2071-1050/15/7/5930/htm

[26] Bertolet A. Algoritmo de planificación basado en capas (SBAP) para tratamientos de radioterapia de intensidad modulada con Philips Pinnacle. 2018; Disponible a: https://www.researchgate.net/publication/332304798_Proyecto_Fin_de_Master_Algoritmo_de_planificacion_basado_en_capas_SBAP_para_tratamientos_de_radioterapia_de_intensidad_modulada_con_Philips_Pinnacle

[27] Mutual information [Internet]. Disponible a: https://nlp.stanford.edu/IR-book/html/htmledition/mutual-information-1.html

[28] Mildenberger P, Eichelberg M, Martin E. Introduction to the DICOM standard. Eur Radiol [Internet]. 1 abril 2002;12(4):920-7. Disponible a: https://pubmed.ncbi.nlm.nih.gov/11960249/

[29] (PDF) A (sort of) new image data format standard: NiFTI-1 [Internet]. Disponible a: https://www.researchgate.net/publication/233741159_A_sort_of_new_image_data_format_standard_NiFTI-1

[30] Guyon I, De AM. An Introduction to Variable and Feature Selection André Elisseeff. Journal of Machine Learning Research. 2003;3:1157-82.

[31] Correlación: qué es, tipos, fórmula, interpretación,... [Internet]. Disponible a: https://www.probabilidadyestadistica.net/correlacion/?utm_content=cmp-true

[32] Información mutua _ AcademiaLab [Internet]. Disponible a: https://academia-lab.com/enciclopedia/informacion-mutua/

[33] Una guía práctica para implementar un Random Forest Classifier en Python | by Tute Gomez E | Medium [Internet]. Disponible a: https://tutegomeze.medium.com/una-gu%C3%ADa-pr%C3%A1ctica-para-implementar-un-random-forest-classifier-en-python-5e38c290ae03

[34] Análisis de varianza (ANOVA) con Python [Internet]. Disponible a: https://cienciadedatos.net/documentos/pystats09-analisis-de-varianza-anova-python.html

[35] LassoCV — scikit-learn 1.5.1 documentation [Internet]. Disponible a: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoCV.html

[36] ¿Qué son las Matrices de Confusión? [Internet]. Disponible a: https://birowebs.com/que-son-las-matrices-de-confusion/

[37] The ROC Curve, Explained - Sharp Sight [Internet]. Disponible a: https://www.sharpsightlabs.com/blog/roc-curve-explained/

[38] Anzanello MJ, Fogliatto FS. Learning curve models and applications: Literature review and research directions. Int J Ind Ergon [Internet]. setembre 2011;41(5):573-83. Disponible a:

http://sitiobigdata.com/2019/12/24/una-introduccion-suave-a-las-curvas-de-aprendizaje-para-diagnosticar-el-rendimiento-del-modelo-de-aprendizaje-automatico/

[39] Zwanenburg A, Vallières M, Abdalah MA, Aerts HJWL, Andrearczyk V, Apte A, et al. The image biomarker standardization initiative: Standardized quantitative radiomics for high-throughput image-based phenotyping. Radiology [Internet]. 1 maig 2020;295(2):328-38. Disponible a: https://doi.org/10.1148/radiol.2020191145

[40] Du K, Allen Li X, Parmar chintan C, W L Aerts hugo HJ, Parmar C, Grossmann P, et al. radiomic Machine-learning classifiers for Prognostic Biomarkers of head and neck cancer. Oncol [Internet]. 2015;5:272. Disponible a: www.frontiersin.org

[41] Van Griethuysen JJM, Fedorov A, Parmar C, Hosny A, Aucoin N, Narayan V, et al. Focus on Computer Resources Computational Radiomics System to Decode the Radiographic Phenotype. Disponible a: http://cancerres.aacrjournals.org/

[42] Arenas M, Gil F, Gironella M, Hernández V, Jorcano S, Biete A, et al. Anti-inflammatory effects of low-dose radiotherapy in an experimental model of systemic inflammation in mice. Int J Radiat Oncol Biol Phys. 2006;66(2):560-7. doi:10.1016/j.ijrobp.2006.06.004.

[43] Seegenschmiedt MH, Keilholz L, Katalinic A, Makoski H, Haase W, Storck M. Antiinflammatory effects of low-dose radiotherapy: indications, dose-response, and radiobiological mechanisms. Strahlenther Onkol. 2012;188(11):975-81. doi:10.1007/s00066-012-0170-4.

Annex I

En aquest annex es presenten les corbes d'aprenentatge dels tres models de Machine Learning que s'han utilitzat per predir la resposta al tractament RTDB en pacients amb artrosi a les mans: Random Forest, SVM amb kernel RBF i CNN.

Aquestes corbes d'aprenentatge proporcionen una visió clara del rendiment dels models analitzats, facilitant la comprensió de la seva capacitat de generalització i el seu potencial per a la presa de decisions clíniques.

Figura 19: Corba d'aprenentatge del model Random Forest

Figura 20: Corba d'aprenentatge del model SVM amb kernel RBF

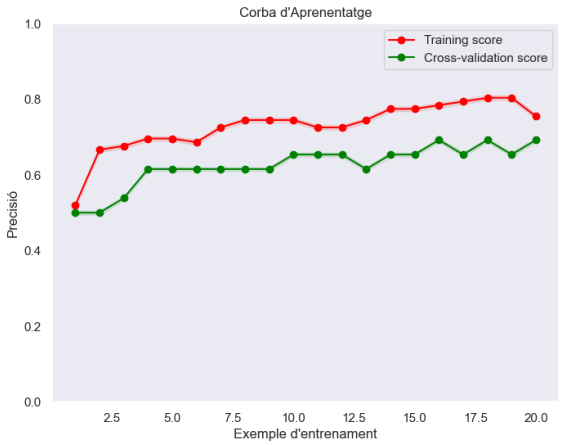

Figura 21: Corba d'aprenentatge del model CNN

Annex II

En aquest annex es presenta el codi utilitzat per convertir les imatges de format DICOM a format NIfTI.

```
  from DicomRTTool.ReaderWriter import DicomReaderWriter,
  ROIAssociationClass
  # file mangagment
  import os

  # array manipulation and plotting
  import numpy as np
  import pandas as pd
  import matplotlib.pyplot as plt

  # medical image manipulation
  import SimpleITK as sitk
  from DicomRTTool.ReaderWriter import DicomReaderWriter
  import os
  import SimpleITK as sitk

  # Main directory containing DICOM data
  main_directory = "C:/Users/Paula/Desktop/RADIOMICA/Pacients/Manos"
  output_directory =
  "C:/Users/Paula/Desktop/RADIOMICA/Pacients_nii/Nifti_Data/Manos"

  # Instantiate the DicomReaderWriter
  Dicom_reader = DicomReaderWriter(description='Images', arg_max=True)

  # Iterate through each sub-directory in the main directory
  for subdir, dirs, files in os.walk(main_directory):
      for dir in dirs:
          current_path = os.path.join(subdir, dir)
          print(f"Processing {current_path}...")

          # Process DICOM data
          Dicom_reader.walk_through_folders(current_path)
          all_rois = Dicom_reader.return_rois(print_rois=False)

  Dicom_reader.set_contour_names_and_associations(contour_names=['ptv1'])
#
```

```
Check to see which indexes have all of the rois we want

        indexes = Dicom_reader.which_indexes_have_all_rois() # Check
      to
see which indexes have all of the rois we want, now we can see indexes
```

```
        if indexes:
pt_indx = indexes[-1] # Assuming you want to process the last
index with all ROIs
            Dicom_reader.set_index(pt_indx)
            Dicom_reader.get_images_and_mask()

            # Prepare output subdirectory
            subfolder_name = os.path.basename(current_path)
output_subdir = os.path.join(output_directory, subfolder_name)
            print(f"Saving images in {output_subdir}")

            if not os.path.exists(output_subdir):
                os.makedirs(output_subdir)

            image = Dicom_reader.ArrayDicom # image array
            mask = Dicom_reader.mask # mask array
            dicom_sitk_handle = Dicom_reader.dicom_handle # SimpleITK
image handle
mask_sitk_handle = Dicom_reader.annotation_handle # SimpleITK
mask handle


            # Save the NIfTI images and masks
sitk.WriteImage(dicom_sitk_handle, os.path.join(output_subdir,
'Image.nii'))
sitk.WriteImage(mask_sitk_handle, os.path.join(output_subdir,
'Mask.nii'))

print(f"Saved Image and Mask in {output_subdir}")
        else:
  print(f"No suitable indexes found in {current_path}.
Skipping...")

print("Processing completed.")
```

Codi 1. Codi per convertir les imatges de format DICOM a format NIfTI.

Annex III

En aquest annex es presenta el codi utilitzat per extreure les característiques radiòmiques de les imatges en format NIfTI.

```
  import sys
  print(sys.version)

  import os
  import pandas as pd
  import radiomics
  dataDir =
"C:/Users/Paula/Desktop/RADIOMICA/Pacients_nii/Nifti_Data/Manos"
  print ("dataDir, relative path:", dataDir)
  print ("dataDir, absolute path:", os.path.abspath(dataDir))

  paramPath = os.path.join(os.getcwd(), "Params.yaml")
  print ("paramPath, relative path:", paramPath)
  print ("Parameter file, absolute path:", os.path.abspath(paramPath))

  print(os.listdir(dataDir))
  def radiomic_features_extraction(dataDir, paramPath):
      df = pd.DataFrame()

      count = 0
      for path in os.listdir(dataDir):
          if os.path.isdir(os.path.join(dataDir, path)):
              count += 1

      for i in range(1, count + 1):
  imagePath = os.path.join(dataDir, str(i).zfill(3) + "/" +
  "Image.nii")
  labelPath = os.path.join(dataDir, str(i).zfill(3) + "/" +
  "Mask.nii")

          print(imagePath)
          print(labelPath)


  extractor = featureextractor.RadiomicsFeatureExtractor(paramPath)

          extractor.enableAllFeatures()
```

```
        print ("Extraction parameters:\n\t", extractor.kwargs)
        print ("Enabled filters:\n\t", extractor.inputImages)
print ("Enabled features:\n\t", extractor.enabledFeature)
```

```
    try:
        result = extractor.execute(imagePath, labelPath)

        # Add the folder number to the result
        result['folder_number'] = i

        df = df.append(result, ignore_index=True)
    except Exception as e:
        print(f"Error processing case {i}: {e}")

for i in df.columns.values:
    if "diagnostics" in i:
        df.drop([i], axis=1, inplace=True)

return df
def radiomic_features_extraction(dataDir, params):
    df = pd.DataFrame()

    count = 0
    for path in os.listdir(dataDir):
        if os.path.isdir(os.path.join(dataDir, path)):
            count += 1

    for i in range(1, count + 1):
imagePath = os.path.join(dataDir, str(i).zfill(3) + "/" +
"Image.nii")
labelPath = os.path.join(dataDir, str(i).zfill(3) + "/" +
"Mask.nii")

        print(imagePath)
        print(labelPath)

        extractor =
      featureextractor.RadiomicsFeatureExtractor(params)
        extractor.enableAllFeatures()
```

```
            try:
                result = extractor.execute(imagePath, labelPath)
```

```
            # Add the folder number to the result - ultima columna
            result['folder_number'] = i

            df = df._append(result, ignore_index=True)
        except Exception as e:
            print(f"Error processing case {i}: {e}")

    for i in df.columns.values:
        if "diagnostics" in i:
            df.drop([i], axis=1, inplace=True)

    return df
    from radiomics import featureextractor

    dataDir =
"C:/Users/Paula/Desktop/RADIOMICA/Pacients_nii/Nifti_Data/Manos"
    params = os.path.join(os.getcwd(),
    "C:/Users/Paula/Desktop/RADIOMICA/params.yaml")
    df = radiomic_features_extraction(dataDir, params)
    df.head()
    # Ruta donde quieres guardar el archivo Excel
    excel_file_path = "C:/Users/Paula/Desktop/RADIOMICA/caract_MANOS.xlsx"

    # Guardar el DataFrame en un archivo Excel
    df.to_excel(excel_file_path, index=False)
```

Codi 2. Codi per extreure les característiques radiòmiques de les imatges en format NIfTI.

Annex IV

En aquest annex es presenta el codi utilitzat per a la selecció de variables i creació de models predictius.

1) Importar llibreries

```
# Importar les llibreries necessàries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import mutual_info_regression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.datasets import make_classification
from sklearn.feature_selection import f_classif, SelectKBest,
mutual_info_regression, VarianceThreshold
from sklearn.feature_selection import SelectFromModel
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import Conv1D, MaxPooling1D, Flatten, Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import confusion_matrix, recall_score,
balanced_accuracy_score
from keras.optimizers import Adam
from sklearn.model_selection import train_test_split, KFold
from sklearn.metrics import accuracy_score, confusion_matrix,
ConfusionMatrixDisplay
from imblearn.over_sampling import SMOTE
from keras import backend as K
from keras.models import Sequential, Model
from keras.layers import Conv1D, MaxPooling1D, Dense, Flatten
from sklearn.model_selection import KFold
from sklearn.model_selection import learning_curve
from imblearn.over_sampling import SMOTE
from collections import Counter
from sklearn.linear_model import LassoCV
from sklearn.model_selection import RandomizedSearchCV, StratifiedKFold,
cross_val_score, train_test_split
from sklearn.metrics import roc_auc_score, precision_score,
```

```
accuracy_score, recall_score, f1_score
from sklearn.metrics import roc_curve, auc, confusion_matrix
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, Flatten, Dense,
Input, Dropout
from tensorflow.keras.regularizers import l2
from tensorflow.keras import backend as K
from sklearn.model_selection import GridSearchCV, RepeatedStratifiedKFold
import tensorflow as tf
from tensorflow.keras import layers, models
```

Codi 3. Codi per importar les llibreries.

2) Definir funcions

```
# Funció per calcular i visualitzar la matriu de confusió
def plotCM(ytrue, ypred_prob, optimal_threshold, clases=None,
normalize=False, ax=None):

   # Convertir probabilitats a prediccions binàries utilitzant el llindar
òptim
    ypred = (ypred_prob >= optimal_threshold).astype(int)

    # Calcular la matriu de confusió
    CM = confusion_matrix(ytrue, ypred)

    # Normalitzar la matriu de confusió
    if normalize:
       CM = 100 * CM / CM.sum(axis=1).reshape(-1, 1) # Normalització per
files

    # Generar les classes a partir de les etiquetes reals
    if clases is None:
        clases = list(set(ytrue))

    # Crear el dataframe per la matriu de confusió
    df = pd.DataFrame(CM, index=clases, columns=clases)
    df.index.name = 'True'; df.columns.name = 'Predicted'
```

```
    # Crear la figura i els eixos
    if ax is None:
        plt.figure(figsize=(8, 6))
        ax = plt.gca()

    # Visualitzar la matriu de confusió amb el colormap 'Blues'
    sns.heatmap(df, annot=True, fmt='d', cmap='Blues', cbar=False,
square=True, annot_kws={'fontsize':16}, ax=ax)

    # Configurar etiquetes i títol
    ax.set_xlabel('Predicted', fontsize=12)
    ax.set_ylabel('Actual', fontsize=12)
    ax.set_title('Confusion Matrix', fontsize=15)

    # Ajustar l'eix de les etiquetes de les classes per a que surtin
rectes
    ax.set_xticklabels(clases, rotation=0, ha="center")
    ax.set_yticklabels(clases, rotation=0)

    plt.show()

    return CM
```

Codi 4. Codi per definir la funció que ens permet visualitzar la matriu de confusió.

```
# Funció per calcular i visualitzar la corba ROC i trobar el llindar òptim
per a models CNN i sklearn
def curveROC(X, y, model):
    # Obtenint les probabilitats predicades
    if hasattr(model, "predict_proba"): # Si és un model sklearn
        y_prob = model.predict_proba(X)
        y_prob = y_prob[:, 1] # Probabilitats de la classe positiva
    else: # Si és un model Keras
        y_prob = model.predict(X).ravel()

    # Calcular fpr, tpr, thresholds
    fpr, tpr, thresholds = roc_curve(y, y_prob, pos_label=1)
```

```
    roc_auc = auc(fpr, tpr)

    # Calcular el millor llindar
    optimal_idx = np.argmax(tpr * (1-fpr))
    optimal_threshold = thresholds[optimal_idx]

    # Gràfic ROC curve
    sns.set(font_scale=1)
    plt.plot(fpr, tpr, label='ROC curve (area = %0.2f)' % roc_auc)
    plt.plot(fpr[optimal_idx], tpr[optimal_idx], 'o', label='Optimal
threshold = %0.2f' % optimal_threshold)
    plt.plot([0, 1], [0, 1], 'k--') # Corba de prediccions aleatòries
    plt.xlim([-0.05, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate or (1 - Specificity)')
    plt.ylabel('True Positive Rate or (Sensitivity)')
    plt.title('Receiver Operating Characteristic')
    plt.legend(loc="lower right")
    plt.show()

    return optimal_threshold, roc_auc
```

Codi 5. Codi per definir la funció que ens permet visualitzar la corba ROC.

```
# Funció per calcular i visualitzar la corba d'aprenentatge
def plotLearningCurve(estimator, X, y, cv, scoring='accuracy', n_jobs=-
1,
train_sizes=np.linspace(0.1, 1.0, 5)):

    # Generar de la corba d'aprenentatge
    train_sizes, train_scores, test_scores = learning_curve(estimator, X,
y, cv=cv, n_jobs=n_jobs,

train_sizes=train_sizes, scoring=scoring)

    # Visualitzar la corba d'aprenentatge
    plt.figure()
    plt.fill_between(train_sizes, np.mean(train_scores, axis=1) -
```

```
np.std(train_scores, axis=1),
                     np.mean(train_scores, axis=1) + np.std(train_scores,
axis=1), alpha=0.1, color="red")
    plt.fill_between(train_sizes, np.mean(test_scores, axis=1) -
np.std(test_scores, axis=1),
                     np.mean(test_scores, axis=1) + np.std(test_scores,
axis=1), alpha=0.1, color="green")
    plt.plot(train_sizes, np.mean(train_scores, axis=1), 'o-',
color="red", label="Training score")
    plt.plot(train_sizes, np.mean(test_scores, axis=1), 'o-',
color="green", label="Cross-validation score")


    plt.title("Corba d'Aprenentatge")
plt.xlabel("Exemples d'entrenament")
plt.ylabel("Precisió")
plt.ylim(0, 1)
plt.legend(loc="best")
plt.grid()
plt.show()
```

Codi 6. Codi per definir la funció que ens permet visualitzar la corba d'aprenentatge.

3) Importar dades

```
#Importar dades
file_path =
"C:/Users/Paula/Desktop/RADIOMICA/excels/excelsRADIOMICA/RadiomicaManos/Ra
diomica_Manos.xlsx"
radiomic_features = pd.read_excel(file_path)
num_tac = radiomic_features['NumTAC']
radiomic_features = radiomic_features.set_index('NumTAC')
radiomic_features = radiomic_features.iloc[:,:-1]


#Eliminar NA
radiomic_features = radiomic_features.dropna()
radiomic_features =
radiomic_features.drop(radiomic_features[radiomic_features['Respuesta']==4
].index)
```

```
# La columna de la variable objectiu és 'respuesta'
X = radiomic_features.drop('Respuesta', axis=1) # Característiques
y = radiomic_features.Respuesta.values              # Variable objectiu
y = y.astype(int)

#Estandaritzar les característiques utilitzant StandardScaler de
scikit-learn
X_scaled = StandardScaler().fit_transform(X)

#Convertir la variable objectiu a una sèrie de pandas
serie = pd.Series(y)

#Recomptar els valors de la variable objectiu
recomp = serie.value_counts()
print(recomp)
```

```
#Mostrar primeres files del DataFrame
radiomic_features.head()
# Combinar les categories 2 i 3 en una sola categoria
y = np.where(y == 3, 2, y)
y = np.where(y == 1, 0, 1)
#Convertir la variable objectiu a una sèrie de pandas
serie = pd.Series(y)

#Recomptar els valors de la variable objectiu
recomp = serie.value_counts()
print(recomp)
# Aplicar SMOTE
smote = SMOTE(sampling_strategy='auto', random_state=42, k_neighbors=2)
X_resampled, y_resampled = smote.fit_resample(X, y)

# Convertir les dades reamostrades en DataFrames
X_resampled = pd.DataFrame(X_resampled, columns=X.columns)
y_resampled = pd.Series(y_resampled, name='Respuesta')

# Replicar la columna 'NumTAC' per a les dades reamostrades
num_tac_resampled = smote.fit_resample(num_tac.values.reshape(-1, 1),
y)[0].flatten()
```

```
# Afegir la variable objectiu reamostrada i la columna 'NumTAC'
radiomic_features_resampled = X_resampled.copy()
radiomic_features_resampled['Respuesta'] = y_resampled.values
radiomic_features_resampled['NumTAC'] = num_tac_resampled

# Comprovar el nombre de mostres per a cada classe després de SMOTE
recomp_resampled = Counter(y_resampled)
print(recomp_resampled)

# Mostrar les primeres files del DataFrame resultant
radiomic_features_resampled.head()

#Recolocar les variables de 'NumTAC' i 'Respuesta' al principi del df
columnes = list(radiomic_features_resampled.columns)
columnes.remove('NumTAC')
columnes.insert(0, 'NumTAC')
columnes.remove('Respuesta')
columnes.insert(1, 'Respuesta')

# Reordenar les columnes del DataFrame
radiomic_features_resampled = radiomic_features_resampled[columnes]

# Mostrar les primeres files per confirmar la nova posició de la columna
'Respuesta'
radiomic_features_resampled.head()
# Guardar el DataFrame actualitzat en un nou fitxer Excel
output_file_path =
"C:/Users/Paula/Desktop/RADIOMICA/excels/excelsRADIOMICA/RadiomicaManos/Ra
diomica_Manos_Resampled.xlsx"
radiomic_features_resampled.to_excel(output_file_path, index=False)
output_file_path =
"C:/Users/Paula/Desktop/RADIOMICA/excels/excelsRADIOMICA/RadiomicaManos/Ra
diomica_Manos_Resampled.csv"
radiomic_features_resampled.to_csv(output_file_path, index=False)
radiomic_features_resampled.head()
```

Codi 7. Codi per importar i fer el processament de dades.

a) **Importar dades equilibrades**

```
#Importar dades equilibrades
file_path =
"C:/Users/Paula/Desktop/RADIOMICA/excels/excelsRADIOMICA/RadiomicaManos/Ra
diomica_Manos_Resampled.xlsx"
radiomic_features = pd.read_excel(file_path)
radiomic_features = radiomic_features.set_index('NumTAC')
radiomic_features.head()
# La columna de la variable objectiu és 'respuesta'
X = radiomic_features.drop('Respuesta', axis=1) # Característiques
y = radiomic_features.Respuesta.values              # Variable objectiu
y = y.astype(int)
```

```
#Estandaritzar les característiques utilitzant StandardScaler de
scikit-learn
X_scaled = StandardScaler().fit_transform(X)

#Convertir la variable objectiu a una sèrie de pandas
serie = pd.Series(y)

#Recomptar els valors de la variable objectiu
recomp = serie.value_counts()
print(recomp)

#Mostrar primeres files del DataFrame
radiomic_features.head()
```

Codi 8. Codi per importar les dades equilibrades.

4) Selecció de característiques

a) **Correlació**

```
# Convertir totes les columnes a valors numèrics, convertint els errors en
NaN
radiomic_features = radiomic_features.apply(pd.to_numeric,
errors='coerce')

# Calcular la correlació amb la variable objectiu 'Respuesta'
corr_with_target =
```

```
radiomic_features.corrwith(radiomic_features['Respuesta']).abs()

# Seleccionar característiques amb una correlació superior a 0.15
high_corr_features = corr_with_target[corr_with_target >
0.15].index.tolist()
print("Característiques amb alta correlació:", high_corr_features)

# Crear un DataFrame amb les característiques seleccionades
df_corr = radiomic_features[high_corr_features]
col_reduced_correlacio = df_corr.columns
print("Columnes reduïdes:", col_reduced_correlacio.to_list())
```

Codi 9. Codi per aplicar el mètode de correlació.

b) **Informació mútua (MI)**

```
# Calcular la informació mútua (MI) entre les característiques i la
variable objectiu
mi = mutual_info_regression(X, y)
mi_scores = pd.DataFrame(mi, index=X.columns, columns=['MI
Scores']).sort_values(by='MI Scores', ascending=False)

# Seleccionar característiques amb una puntuació de MI superior a 0.02
selected_features = mi_scores[mi_scores['MI Scores'] > 0.02]
no_selected_features = mi_scores[mi_scores['MI Scores'] < 0.02]
selected_features_IM = selected_features.transpose()
print("Característiques seleccionades basades en MI:",
selected_features_IM.columns.to_list())
```

Codi 10. Codi per aplicar el mètode d'informació mútua.

c) **Selecció supervisada amb RF**

```
# Crear un classificador Random Forest
rf = RandomForestClassifier(n_estimators=100, random_state=42)
# Utilitzar SelectFromModel per ajustar i seleccionar les característiques
selector = SelectFromModel(rf)
X_important = selector.fit_transform(X, y)

# Obtenir els noms de les característiques seleccionades
```

```
selected_feature_names = X.columns[selector.get_support()]

# Crear un DataFrame amb les característiques seleccionades
X_important_RF = pd.DataFrame(X_important, columns=selected_feature_names)

# Imprimir les columnes del conjunt de dades reduït
print("Forma del conjunt de dades reduït:",
X_important_RF.columns.to_list())
```

Codi 11. Codi per aplicar el mètode de selecció supervisada amb Random Forest.

d) **ANOVA**

```
# Eliminar característiques constants
constant_filter = VarianceThreshold(threshold=0)
X_filtered = constant_filter.fit_transform(X)

# Obtenir els noms de les característiques que no són constants
non_constant_features = X.columns[constant_filter.get_support()]

# Actualitzar X amb només les característiques no constants
X_no_constants = pd.DataFrame(X_filtered, columns=non_constant_features)

# Computar els valors F d'ANOVA per a les característiques
f_values, p_values = f_classif(X_no_constants, y)

# Seleccionar un nombre de característiques basat en els p-values més
petits
selector = SelectKBest(f_classif, k='all').fit(X_no_constants, y)

# Si es vol veure quines característiques han estat seleccionades:
selected_features_bool = selector.get_support()
selected_features_anova = [column for (column, selected) in
zip(X_no_constants.columns, selected_features_bool) if selected]

# Filtrar les característiques seleccionades amb p-values inferiors a 0.1
filtered_features_p_values = [(feature, p_value) for feature, p_value in
zip(selected_features_anova, p_values[selected_features_bool]) if p_value
```

```
< 0.1]
filtered_features_ANOVA = pd.DataFrame(filtered_features_p_values,
columns=['Feature', 'P-Value'])

# Per imprimir les característiques seleccionades i els seus p-values
corresponents
print("Característiques seleccionades i els seus p-values:")
for feature, p_value in zip(selected_features_anova,
p_values[selected_features_bool]):
    if p_value < 0.1: #per què 0.1? 0,05?
        print(f"{feature}: p-value = {p_value}")
```

Codi 12. Codi per aplicar el mètode d'ANOVA.

e) **LassoCV**

```
# Aplicar LassoCV amb augment del nombre d'iteracions i tolerància
lasso = LassoCV(cv=5, random_state=42, max_iter=100000,
tol=1e-4).fit(X_scaled, y)

# Obtenir les característiques seleccionades (coeficients diferents de
zero)
coef = pd.Series(lasso.coef_, index=X.columns)
caracteristiques_importants_LassoCV = coef[coef != 0].index.tolist()

# Crear un DataFrame amb aquestes característiques per a ús posterior
df_global = radiomic_features[caracteristiques_importants_LassoCV]

# Mostrar les característiques seleccionades
print("Característiques seleccionades:",
caracteristiques_importants_LassoCV)
```

Codi 13. Codi per aplicar el mètode de LassoCV.

f) **Taula de pressència-absència**

```
# Recopilar les característiques seleccionades per cada mètode
features_presence = {
    "Correlació":
radiomic_features.columns.isin(df_corr.columns).astype(int),
    "Informació mútua":
```

```
radiomic_features.columns.isin(selected_features_IM.columns).astype(int),
    "Random Forest":
radiomic_features.columns.isin(X_important_RF.columns).astype(int),
    "ANOVA":
radiomic_features.columns.isin(filtered_features_ANOVA['Feature']).astype(
int),
    "LassoCV":
radiomic_features.columns.isin(df_global.columns).astype(int)
}

# Crear la taula de presència-absència
presence_absence_table = pd.DataFrame(features_presence,
index=radiomic_features.columns)
print("Taula de presència-absència:")
print(presence_absence_table.head())

# Guardar la taula en un fitxer Excel
presence_absence_table.to_excel('C:/Users/Paula/Desktop/RADIOMICA/excels/r
esultats_Seleccio_Caract/Manos/presence_absence_Manos.xlsx')
# Funció per seleccionar variables basades en la presència en els cinc
mètodes
def seleccionar_variables(df):
    return df[df.sum(axis=1) >= 5].index.tolist()

# Utilitzar la funció per seleccionar les variables
variables_seleccionades = seleccionar_variables(presence_absence_table)
print("Variables seleccionades:", variables_seleccionades)
print("Nombre de variables seleccionades:", len(variables_seleccionades))
# Filtrar el DataFrame per mantenir només les variables seleccionades
df_seleccionades = radiomic_features[variables_seleccionades]

# Guardar les característiques seleccionades en un nou fitxer Excel
df_seleccionades.to_excel('C:/Users/Paula/Desktop/RADIOMICA/excels/resulta
ts_Seleccio_Caract/Manos/variables_seleccionades_MANOS.xlsx', index=False)
# Guardar el DataFrame com a CSV
df_seleccionades.to_csv('C:/Users/Paula/Desktop/RADIOMICA/excels/resultats
_Seleccio_Caract/Manos/variables_seleccionades_MANOS.csv', index=False)
```

Codi 14. Codi per crear la taula de presència-absència.

g) Correlació de les característiques seleccionades amb la resposta

```
# Calcular la matriu de correlació
correlation_matrix = dades_model.corr()

# Seleccionar la variable objectiu 'Respuesta'
correlation_with_response = correlation_matrix.loc['Respuesta']

# Suprimir la columna "Respuesta" de la sèrie de correlació
correlation_with_response = correlation_with_response.drop('Respuesta')

# Crear un gràfic de barres horitzontals de la correlació de cada variable
amb 'Respuesta'
plt.figure(figsize=(10, 6))
sns.barplot(x=correlation_with_response.values,
y=correlation_with_response.index, palette='coolwarm')

plt.title('Correlation of Variables with "Respuesta"')
plt.xlabel('Correlation')
plt.ylabel('Variable')
plt.xlim(-0.5, 0.5)
plt.show()
```

```
variables_seleccionades.append('Respuesta')
# Seleccionar les dades que volem conservar
dades_model = radiomic_features[variables_seleccionades]
```

Codi 15. Codi per estudiar la correlació entre les característiques seleccionades i la resposta.

5) Creació de models

```
#Importar les característiques seleccionades
df_seleccionades =
'C:/Users/Paula/Desktop/RADIOMICA/excels/resultats_Seleccio_Caract/Manos/v
ariables_seleccionades_MANOS.csv'
df_seleccionades = pd.read_csv(df_seleccionades)
```

a) Random Forest

```
# Dividir les dades en entrenament i prova
X_train, X_test, y_train, y_test_rf = train_test_split(df_seleccionades,
y, test_size=0.2, random_state=42)
```

```
print("Variables que utilitzem per l'entrenament:", X_train.shape)
print("Variables que utilitzem pel test:", X_test.shape)

# Escalar les dades per tal de millorar la convergència del model
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Definir els hiperparàmetres per evitar el sobreajustament
param_dist_rf = {
    'n_estimators': [50, 100, 150, 200, 250, 300, 350, 400],
    'max_depth': [3, 5, 7],
    'min_samples_split': [10, 15, 20],
    'min_samples_leaf': [5, 10],
    'max_features': ['sqrt'],
    'bootstrap': [True]
}

# Model Random Forest amb RandomizedSearchCV
rf = RandomForestClassifier(random_state=42, class_weight='balanced')
random_search_rf = RandomizedSearchCV(estimator=rf,
param_distributions=param_dist_rf,
                                      n_iter=50, cv=5, verbose=2,
random_state=0, n_jobs=-1)
random_search_rf.fit(X_train_scaled, y_train)
best_rf = random_search_rf.best_estimator_
```

```
# Validació creuada
skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)
n_scores = cross_val_score(best_rf, X_train_scaled, y_train,
scoring='accuracy', cv=skf, n_jobs=-1)
print('Mean accuracy with CV: %.3f +/- %.3f' % (np.mean(n_scores) * 100,
np.std(n_scores) * 100))

# Ajustar el model als millors paràmetres
best_rf.fit(X_train_scaled, y_train)
```

```
# Avaluar el conjunt de prova
y_prob_rf = best_rf.predict_proba(X_test_scaled)[:, 1]

# Visualitzar la corba ROC i obtenir el llindar òptim
plt.figure()
optimal_threshold, rf_auc = curveROC(X_test_scaled, y_test_rf, best_rf)

# Prediccions basades en el llindar òptim
y_pred_rf_optimal = (y_prob_rf >= optimal_threshold).astype(int)

# Visualitzar la matriu de confusió utilitzant el llindar òptim
plt.figure()
conf_matrix_optimal = plotCM(y_test_rf, y_prob_rf, optimal_threshold,
normalize=False)

# Calcular les mètriques basades en la matriu de confusió òptima
tn_opt, fp_opt, fn_opt, tp_opt = conf_matrix_optimal.ravel()
sensibilitat_optimal = tp_opt / (tp_opt + fn_opt)
especificitat_optimal = tn_opt / (tn_opt + fp_opt)
precisio_optimal = precision_score(y_test_rf, y_pred_rf_optimal)
f1_optimal = f1_score(y_test_rf, y_pred_rf_optimal)
acc_optimal = accuracy_score(y_test_rf, y_pred_rf_optimal)
# Resultats de les mètriques
print(f"Threshold òptim: {optimal_threshold:.3f}")
print(f"Sensibilitat (Recall) òptima: {sensibilitat_optimal:.3f}")
print(f"Especificitat òptima: {especificitat_optimal:.3f}")
print(f"Precisió òptima: {precisio_optimal:.3f}")
print(f"F1-score òptim: {f1_optimal:.3f}")
print(f"Exactitud (Accuracy) òptima: {acc_optimal:.3f}")
print(f'ROC AUC: {rf_auc:.3f}')
# Generar la corba d'aprenentatge
plt.figure()
plotLearningCurve(best_rf, X_train_scaled, y_train, cv=skf)
plt.show()
```

Codi 16. Codi crear el model de Random Forest.

b) SVM amb kernel RBF

```
# Escalar les dades per tal de millorar la convergència del model i
dividir les dades en entrenament i prova
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df_seleccionades)
X_train, X_test, y_train, y_test = train_test_split(df_seleccionades, y,
test_size=0.2, random_state=42, stratify=y)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
# Definir els hiperparàmetres
param_dist_rbf = {
    'C': [0.0001, 0.001, 0.01, 0.1, 1, 5, 10, 20, 30, 50, 100],
    'gamma': [0.0001, 0.0005, 0.001, 0.01],
    'kernel': ['rbf']
}

svm_rbf = SVC(kernel='rbf', probability=True)
random_search_rbf = RandomizedSearchCV(svm_rbf,
param_distributions=param_dist_rbf, n_iter=29, cv=5, scoring='roc_auc',
random_state=42, n_jobs=-1)
random_search_rbf.fit(X_train_scaled, y_train)

best_svm_rbf = random_search_rbf.best_estimator_
best_params_rbf = random_search_rbf.best_params_
print("Millors paràmetres per SVM RBF:", best_params_rbf)

# Validació creuada amb més splits per una millor estabilitat
skf = StratifiedKFold(n_splits=45, shuffle=True, random_state=0)
n_scores = cross_val_score(best_svm_rbf, X_train_scaled, y_train,
scoring='accuracy', cv=skf, n_jobs=-1)
print(f'Cross-validated accuracy: {np.mean(n_scores):.3f} ±
{np.std(n_scores):.3f}')
```

```
# Ajustar el model amb les dades d'entrenament
best_svm_rbf.fit(X_train_scaled, y_train)

# Avaluar el model sobre el conjunt de prova
```

```
y_prob_rbf = best_svm_rbf.predict_proba(X_test_scaled)[:, 1]

# Visualitzar la corba ROC i obtenir el llindar òptim
plt.figure()
optimal_threshold, roc_auc = curveROC(X_test_scaled, y_test, best_svm_rbf)

# Prediccions basades en el llindar òptim
y_pred_rbf_optimal = (y_prob_rbf >= optimal_threshold).astype(int)

# Visualitzar la matriu de confusió utilitzant el llindar òptim
plt.figure()
conf_matrix_optimal = plotCM(y_test, y_prob_rbf, optimal_threshold,
normalize=False)

# Calcular les mètriques basades en la matriu de confusió òptima
tn_opt, fp_opt, fn_opt, tp_opt = conf_matrix_optimal.ravel()
sensibilitat_optimal = tp_opt / (tp_opt + fn_opt)
especificitat_optimal = tn_opt / (tn_opt + fp_opt)
precisio_optimal = precision_score(y_test, y_pred_rbf_optimal)
f1_optimal = f1_score(y_test, y_pred_rbf_optimal)
acc_optimal = accuracy_score(y_test, y_pred_rbf_optimal)

# Resultats de les mètriques
print(f"Threshold òptim: {optimal_threshold:.3f}")
print(f"Sensibilitat (Recall) òptima: {sensibilitat_optimal:.3f}")
print(f"Especificitat òptima: {especificitat_optimal:.3f}")
print(f"Precisió òptima: {precisio_optimal:.3f}")
print(f"F1-score òptim: {f1_optimal:.3f}")
print(f"Exactitud (Accuracy) òptima: {acc_optimal:.3f}")
print(f'ROC AUC: {roc_auc:.3f}')

# Generar i visualitzar la corba d'aprenentatge
plt.figure()
plotLearningCurve(best_svm_rbf, X_train_scaled, y_train, cv=skf,
scoring='accuracy')
```

Codi 17. Codi per crear el model de SVM amb kernel RBF

c) Xarxa Neuronal (CNN)

```
# Dividir les dades en entrenament i prova
X_train, X_test, y_train, y_test = train_test_split(df_seleccionades, y,
test_size=0.2, random_state=42)

# Escalar les dades
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Remodelar les dades per ajustar-les a la CNN
X_train_reshaped = X_train_scaled.reshape(X_train_scaled.shape[0],
X_train_scaled.shape[1], 1)
X_test_reshaped = X_test_scaled.reshape(X_test_scaled.shape[0],
X_test_scaled.shape[1], 1)

# Definir el model CNN
model = models.Sequential([
    layers.Conv1D(filters=32, kernel_size=3, activation='relu'),
    layers.MaxPooling1D(pool_size=2),
    layers.Conv1D(filters=64, kernel_size=3, activation='relu'),
    layers.MaxPooling1D(pool_size=2),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

# Entrenament del model
history = model.fit(X_train_reshaped, y_train, epochs=20,
validation_data=(X_test_reshaped, y_test))

# Prediccions
y_pred = (model.predict(X_test_reshaped) > 0.5).astype("int32")

# Matriu de confusió
```

```
plotCM(y_test, y_pred, optimal_threshold)
```

```
# Calcular les mètriques basades en la matriu de confusió òptima
tn_opt, fp_opt, fn_opt, tp_opt = conf_matrix_optimal.ravel()
sensibilitat_optimal = tp_opt / (tp_opt + fn_opt)
especificitat_optimal = tn_opt / (tn_opt + fp_opt)
precisio_optimal = precision_score(y_test, y_pred)
f1_optimal = f1_score(y_test, y_pred)
acc_optimal = accuracy_score(y_test, y_pred)

# Resultats de les mètriques
print(f"Threshold òptim: {optimal_threshold:.3f}")
print(f"Sensibilitat (Recall) òptima: {sensibilitat_optimal:.3f}")
print(f"Especificitat òptima: {especificitat_optimal:.3f}")
print(f"Precisió òptima: {precisio_optimal:.3f}")
print(f"F1-score òptim: {f1_optimal:.3f}")
print(f"Exactitud (Accuracy) òptima: {acc_optimal:.3f}")
print(f'ROC AUC: {roc_auc:.3f}')

# Corba ROC amb punt òptim
optimal_threshold, roc_auc = curveROC(X_test_reshaped, y_test, model)

# Corba d'Aprenentatge
plt.figure(figsize=(8, 6))
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Learning Curve')
plt.legend()
plt.show()
```

Codi 18. Codi per crear el model de CNN.