Universitat de Girona
Escola Politècnica Superior

Projecte fi de grau

Estudi: Grau en Enginyeria Informàtica

Títol: Aprenentatge automàtic per a la generació de plans de tractament de tumors amb radioteràpia

Document: Memòria

Alumne: Joan Passarrius Pedrosa

Tutor: Beatriz López Ibáñez
Departament: Enginyeria Elèctrica, Electrònica i Automàtica
Àrea: Enginyeria de Sistemes i Automàtica

Tutor extern: Daniel Lambisto Castro
Departament: Institut Català d'Oncologia (ICO)
Àrea: Servei de Física Mèdica i Protecció Radiològica

Convocatòria (mes/any): Febrer 2022

Universitat de Girona
Escola Politècnica Superior

PROJECTE FI DE GRAU

Aprenentatge automàtic per a la generació de plans de tractament de tumors amb radioteràpia

Autor:
Joan PASSARRIUS PEDROSA

Febrer 2022

Grau en Enginyeria Informàtica

Tutors:
Beatriz LÓPEZ IBÁÑEZ
Daniel LAMBISTO CASTRO

Agraïments

Per començar voldria dedicar aquest treball a la meva família, ja que sense el seu suport i ànims durant tota aquesta etapa no hauria pogut arribar fins a on estic a dia d'avui.

A més, m'agradaria agraïr als meus tutors, per una banda a la Beatriz López Ibáñez per tota la seva ajuda i sobretot paciència a l'hora de realitzar el projecte i, per l'altra a en Daniel Lambisto Castro tant per haver confiat en mi per dur a terme la investigació que proposava, com per tots els moments que ha dedicat a explicar-me i proporcionar-me tant de forma presencial a l'hospital com telemàticament, totes les dades, informació, detalls i conceptes de l'àmbit físic i mèdic necessaris en tot moment per a poder entendre i desenvolupar aquest treball.

Índex

CAPÍTOL 1

Introducció

La radioteràpia consisteix en l'ús de radiacions ionitzants per al tractament de determinats tipus de tumors. És una teràpia oncològica, com pot ser la cirurgia o la quimioteràpia, que elimina, amb les radiacions, les cèl·lules canceroses. El problema d'aquesta tècnica és que aconseguir un tractament on la radiació només afecti la regió ocupada pel tumor i que, a més, aquest tractament sigui el més efectiu possible, és molt complicat. Això es deu a què el procés de generar un pla de tractament per cada pacient és bastant complex donat el nombre de factors que s'han de tenir en compte a l'hora de realitzar els càlculs per acabar generant cada model, i el temps que requereix la generació de cadascun d'ells.

En el present treball ens centrarem en analitzar els tractaments amb feixos fotons produïts en un accelerador lineal (AL) (veure 5.1.1) i modulats en una tècnica coneguda com a VMAT (Volumetric Modulated Arc Therapy) (veure 5.1.2). Tots els pacients d'aquest treball han estat tractats a l'Institut Català d'Oncologia (ICO) de Girona amb un accelerador Clinac iX de la marca Varian amb energia de 6MV.

1.1 Motivació

El motiu principal que m'ha portat a escollir aquest projecte és el fet de poder posar en pràctica els conceptes adquirits durant la carrera i poder aprofundir més en els coneixements relacionats amb la Intel·ligència Artificial.

Un altre aspecte molt rellevant que m'ha fet decidir per aquest estudi ha estat el fet que el projecte està relacionat amb l'àmbit de la medicina. Això otorga un extra de motivació, ja que a part d'ampliar els coneixements propis, també serveix per ajudar en l'avenç i millora de procediments mèdics, que al final repercutiran en augmentar tant l'eficàcia com la velocitat de generació de tractaments podent oferir els millors serveis a persones que pateixen qualsevol tipus de tumor.

Així doncs he considerat que el projecte d'en Daniel (tutor extern de l'ICO)

és perfecte, ja que la idea que proposa és molt atractiva en quant a l'aspecte de recerca i innovació, i s'ajusta molt als objectius personals que m'agradaria assolir.

1.2 Propòsit

El propòsit principal d'aquest projecte consisteix en estudiar la relació entre les diferents mètriques que s'utilitzen per generar els tractaments de radioteràpia utilitzant tècniques de Machine Learning per tal d'aconseguir automatitzar el procés de generació de plans per al tractament de tumors amb radioteràpia, a més d'aconseguir desenvolupar un model predictiu capaç de predir tractaments òptims per a cada pacient reduint al mínim l'afectació de la dosi en els teixits sans i, per tant, reduir-ne la toxicitat i els afectes adversos.

L'abast que s'ha establert comprèn l'estudi, anàlisi i disseny d'un sistema predictiu programat en Python capaç de predir plans de tractament on es minimitzi l'error utilitzant tècniques de Machine Learning.

1.3 Objectius

Els objectius d'aquest P/TFG són els següents:

- Estudiar el comportament de les diferents mètriques utilitzades en un pla de tractament obtingut a través d'un *planificador* i trobar quines són les que aporten més informació, i per tant tenen una major relació a l'hora de decidir si un tractament passa els índexs de validesa per tal de poder-lo utilitzar per a tractar un pacient. Totes les mètriques que s'utilitzen per elaborar un pla de tractament són les recomanades per la Societat Catalana de Física Mèdica, però no totes aquestes mètriques tenen per què ser rellevants i donar-nos la mateixa informació. En concret s'estudiarà la utilització de diferents mètodes de regressió per a seleccionar les millors mètriques.

- Fer l'anàlisi, el disseny i la implementació d'un sistema intel·ligent que permeti obtenir prediccions sobre quins són els tractaments que passen l'índex d'acceptació per poder ser utilitzats en un pacient rebent d'entrada les mètriques seleccionades.

CAPÍTOL 2

Viabilitat

Tenint en compte que el desenvolupament d'aquest projecte no depèn únicament de la recerca en l'àmbit informàtic, sinó que la part principal d'aquest recau en l'àmbit físic-mèdic contingut en el treball que es desenvolupa a l'ICO cal fer una separació dels recursos necessaris per al desenvolupament d'aquesta investigació.

2.1 Recursos

- Recursos mèdics:
  - Accelerador lineal Varian Clinac® iX LinAc: 1.500.000 € .
- Recursos informàtics:
  - Ordinador: 1450 € .
  - Software i llibreries: 0 € .

2.2 Recursos humans

Per a desenvolupar aquest projecte són necessaris especialistes en diferents àmbits, un en l'àmbit físic-mèdic i l'altre en l'àmbit informàtic.

- Físic mèdic 18 € /h.
- Enginyer informàtic: en aquest cas les tasques excercides són diverses per la qual cosa podem fer distincions entre cadascuna d'elles:
  - Analista: 18 € /h.
  - Programador: 12 € /h.

2.3 Viabilitat econòmica

Tenint en compte els costs tant dels recursos materials com dels recursos humans podem fer un estudi econòmic aproximat del cost total que suposa el desenvolupament d'aquest projecte. Val a dir, que els costos provinents dels recursos emprats per part del personal de l'ICO són independents del desenvolupament d'aquest projecte donat que és una col·laboració. Per tant el seu valor consta únicament de forma informativa i no es tindran en compte en el còmput total.

Feina	**Especialista**	**Temps estimat (h)**	**Cost (€)**
Preprocessament de les dades	Analista	20	360
Tractament de dades	Analista	20	360
Codificació	Programador	130	1560
Selecció millors mètriques	Físic mèdic	30	540
Proves	Programador	40	480
Anàlisi de resultats	Analista	50	900
Interpretació de resultats	Físic mèdic	40	720
Documentació	Analista	96	1728
Total recursos humans		426	6648

Material	**Cost (€)**
Ordinador	1450
Software (*Sklearn, Yellowbrick, etc...*)	0
Total recursos materials	1450

El cost total aproximat del projecte és de: $1450 + 6648 = \mathbf{8098}\,€$.

CAPÍTOL 3

Metodologia

La metodologia implementada pel desenvolupament d'aquest projecte ha estat *Agile*, específicament utilitzant el conjunt de bones pràctiques que s'engloben dins *Scrum*.

3.1 *Scrum*

Scrum [Scrum 021] és un marc de treball on s'apliquen de forma regular un cojunt de bones pràctiques per treballar de forma col·laborativa, en equip, i obtenir el millor resultat d'un projecte.

A *Scrum* es realitzen entregues parcials i regulars del producte final, prioritzades segons el benefici que aporten al receptor del projecte o client. És altament recomanable per projectes en entorns complexos, on es necessita obtenir resultats aviat, on els requisits són cambiants o poc definits, on la innovació, la competitivitat, la flexibilitat i la productivitat són fonamentals.

3.1.1 Fonaments

Scrum es fonamenta en:

- El desenvolupament incremental dels requisits d'un projecte en blocs temporalment curts i fixes. Solen ser iteracions de dues setmanes tot i que es poden allargar si és necessari.

- La priorització dels requisits segons el valor que tenen pel client i el cost de desenvolupament en cada iteració.

- El control del projecte. Per una banda, al final de cada iteració es mostra al client els resultats obtinguts, de forma que es puguin prendre les decisions necessàries en funció del què s'observa i del context del projecte en cadascun d'aquests moments. Per l'altra, l'equip es sincronitza diàriament i es realitzen les adaptacions necessaries.

- La sistematització de la col·laboració i la comunicació tant entre l'equip com amb el client.
- La temporalització de les activitats del projecte, per ajudar a la presa de decisions i aconseguir resultats.

3.1.2 Beneficis

Els principals beneficis que proporciona *Scrum* són:

- Gestió regular de les expectatives del client i basada en resultats tangibles.
- Resultats anticipats.
- Flexibilitat i adaptació respecte les necessitats del client, canvis en el mercat, etc.
- Mitigació de riscos en el projecte.
- Productivitat i qualitat.
- Alineament entre el client i l'equip de desenvolupament.
- Motivació de l'equip de treball.

3.2 Aplicació de la metodologia

Per a l'aplicació d'aquesta metodologia es necessita una entitat equivalent a la representació del client marcant els requisits de l'estudi i fixant una ruta de treball en cadascuna de les entregues depenent dels resultats obtinguts en cadascuna d'elles. En aquest cas, donat que la idea del treball i els requisits han estat marcats des d'un principi pel tutor extern, en Daniel Labisto ha estat qui ha realitzat aquest paper.

Donat que l'organització temporal del treball s'ha vist afectada per diversos motius, principalment la manca de temps degut a la feina i en gran mesura a causa de la pandèmia del SARS-CoV-2, al final la metodologia adoptada s'ha hagut d'adaptar a les necessitats i circumstàncies de cada moment i, per tant, tot i mantenir el sistema de tasques i entregues, la metodologia no ha estat idèntica a *Scrum* ja que en aquesta s'estableix un espai de temps que aproxima a ser de dues setmanes entre cadascuna de les etapes d'entrega, reestructuració i distribució de tasques i, en aquest cas, aquest temps ha variat molt depenent de la disponibilitat.

3.3 Gestió de tasques

Per tal d'organitzar el flux de treball i simplificar la gestió del projecte i fer un seguiment de forma més efectiva de l'estat i prioritat de les diferents tasques, s'ha utilitzat una eina anomenada *Notion*. *Notion* és una eina de treball tot en un, amb una interfície d'usuari molt simple que permet tant crear notes, documents, *wikis* o bases de dades, com planificar, gestionar i organitzar projectes i informació. A més permet treballar de forma col·laborativa en el cas de formar part d'un grup de treball.

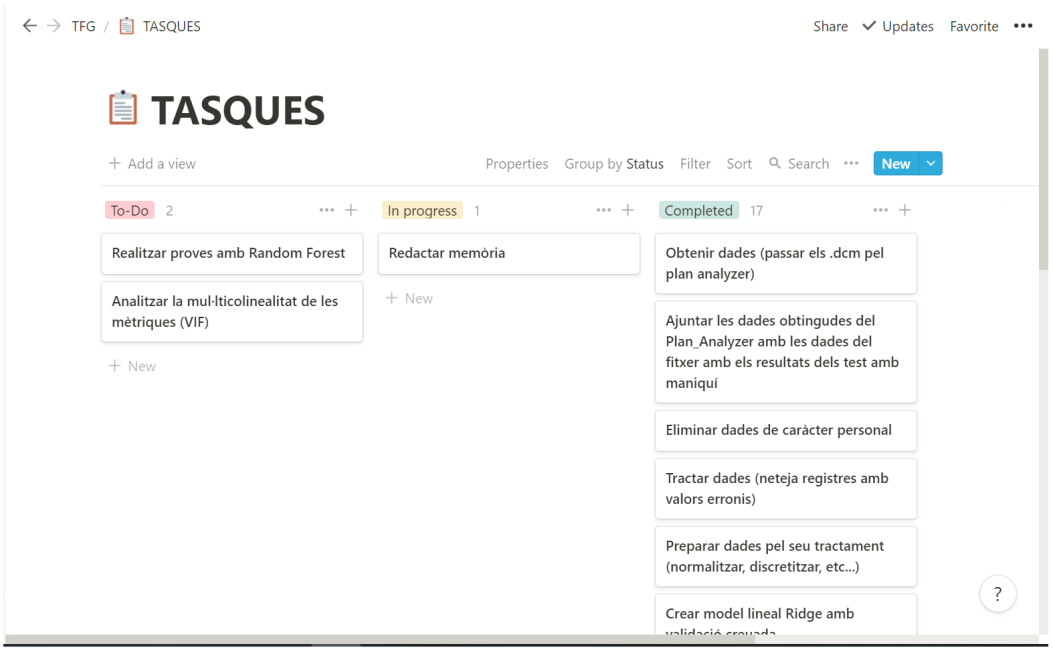


Figura 3.1: Captura de la gestió de tasques utilitzant Notion.

CAPÍTOL 4

Planificació

En aquest capítol es descriuran els detalls de la planificació del projecte per tal d'assolir els objectius establerts en la introducció.

La planificació del projecte es divideix en dues etapes:

La primera etapa inclou tota la preparació: l'estudi de les tècniques d'intel·ligència artificial adoptades en el projecte i, l'estudi dels tractaments de radioteràpia i les dades que els composen. Les dades són les que s'obtenen a l'ICO mitjançant primerament un planificador que genera els tractaments a partir del TAC d'un pacient i posteriorment fent proves amb un maniquí per saber-ne els índexs de viabilitat.

La segona etapa és on es desenvolupa la resta del projecte i se'n va fent el seguiment. En aquesta part es duen a terme tant les tasques de tractament de dades per poder utilitzar-les en els diferents algoritmes, com la part de generar models de regressió per a analitzar-ne el comportament. A més, en aquesta part és on també es generen els models de classificació per intentar predir la viabilitat d'un tractament.

- **Etapa 1**
  - **Planificació del projecte:**
    - Reunió amb tutors.
    - Anàlisi de requeriments i establiment d'objectius.
    - Temps: 10h.
  - **Estudis previs:**
    - Estudi de les regressions: Lasso, Ridge, LassoLARS, ElasticNet.
    - Estudi dels algoritmes de classificació: boscos aleatoris, màquines de supor vectorial i KNN.
    - Estudi de les llibreries de Pyhton.
    - Reunions ICO per a la introducció als tractaments de radioteràpia.

    - ◇ Estudi de les mètriques d'un tractament i lectura d'estudis relacionats.
    - ◇ Lectura de projectes relacionats.
    - ◇ Temps: 60h.

- **Etapa 2**
  - **Desenvolupament:**
    - ◇ Generació de les dades.
    - ◇ Neteja i disseny de la base de dades.
    - ◇ Anàlisi de les mètriques.
    - ◇ Anàlisi dels resultats i selecció mètriques rellevants.
    - ◇ Experimentació amb models de regressió.
    - ◇ Experimentació amb boscos aleatoris.
    - ◇ Experimentació amb màquines de suport vectorial.
    - ◇ Temps: 330h.
  - **Documentació:** documentació del projecte en general
    - ◇ Temps: 96h.

La planificació presenta les consideracions següents:

- El P/TFC es va començar el Febrer de 2020.
- Hi ha hagut períodes en els quals per motius laborals no s'ha pogut avançar el treball.
- El desenvolupament del treball s'ha vist afectat per la pandèmia del SARS-CoV-2 (Covid-19).

4.1 Estructura de descomposició del treball

L'estructura de descomposició del treball o *WBS* [Institute 2019] pel seu nom en anglès (*Work Breakdown Structure*) és una eina que permet representar de forma esquemàtica la descomposició jeràrquica de l'abast del treball que s'ha de dur a terme en un projecte en diverses activitats, assolint un grau de detall suficient per planejar-lo i controlar-lo de forma adequada.

L'estructura de descomposició d'aquest treball s'adjunta a continuació:

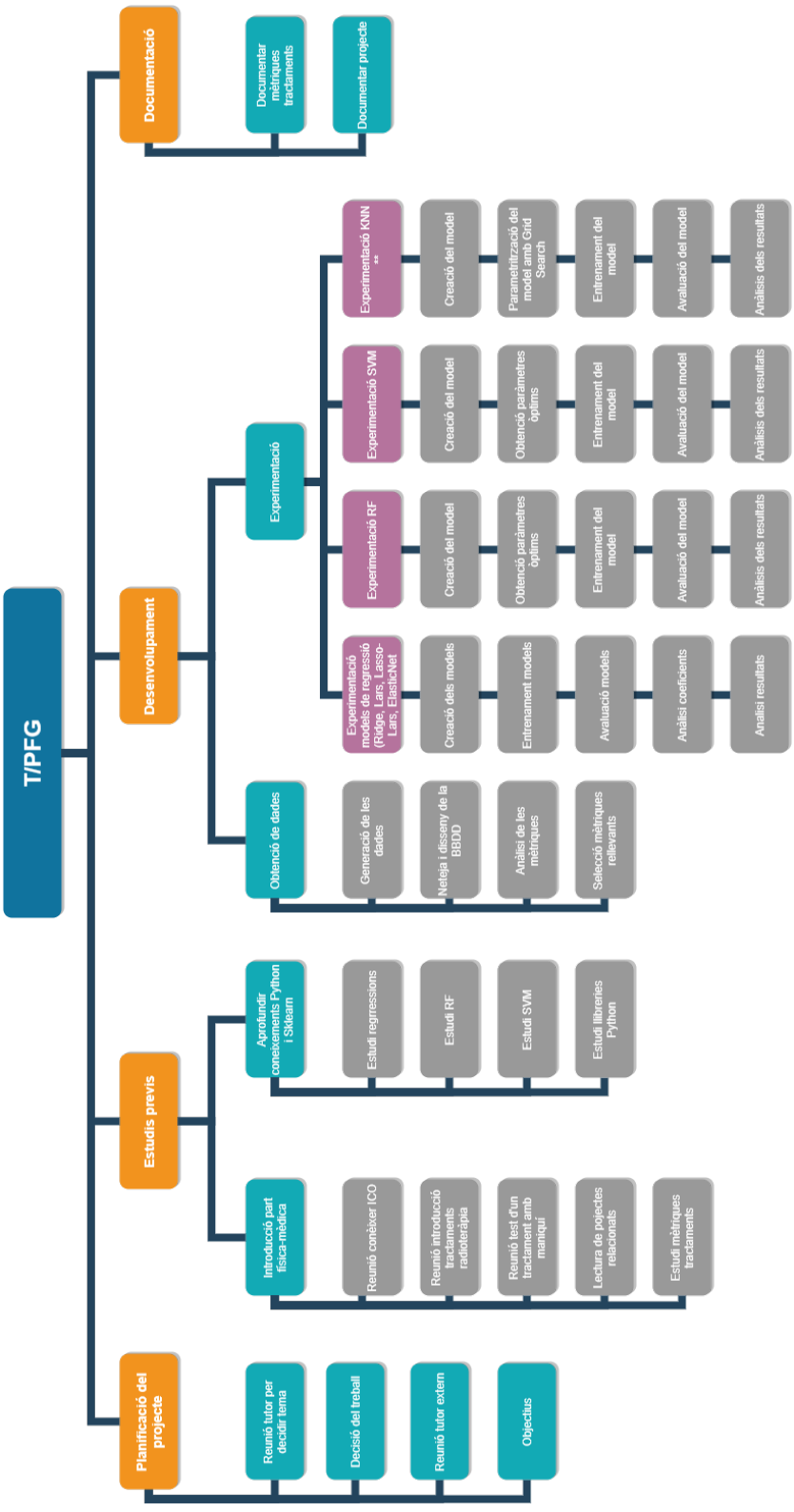


Figura 4.1: Descomposició de la planificació del treball en activitats.

4.2 Diagrama de Gantt

El diagrama de Gantt és una eina que permet a través d'un diagrama de barres, exposar de forma cronològica totes les activitats o tasques d'un projecte.

El diagrama de Gantt s'adjunta a continuació:

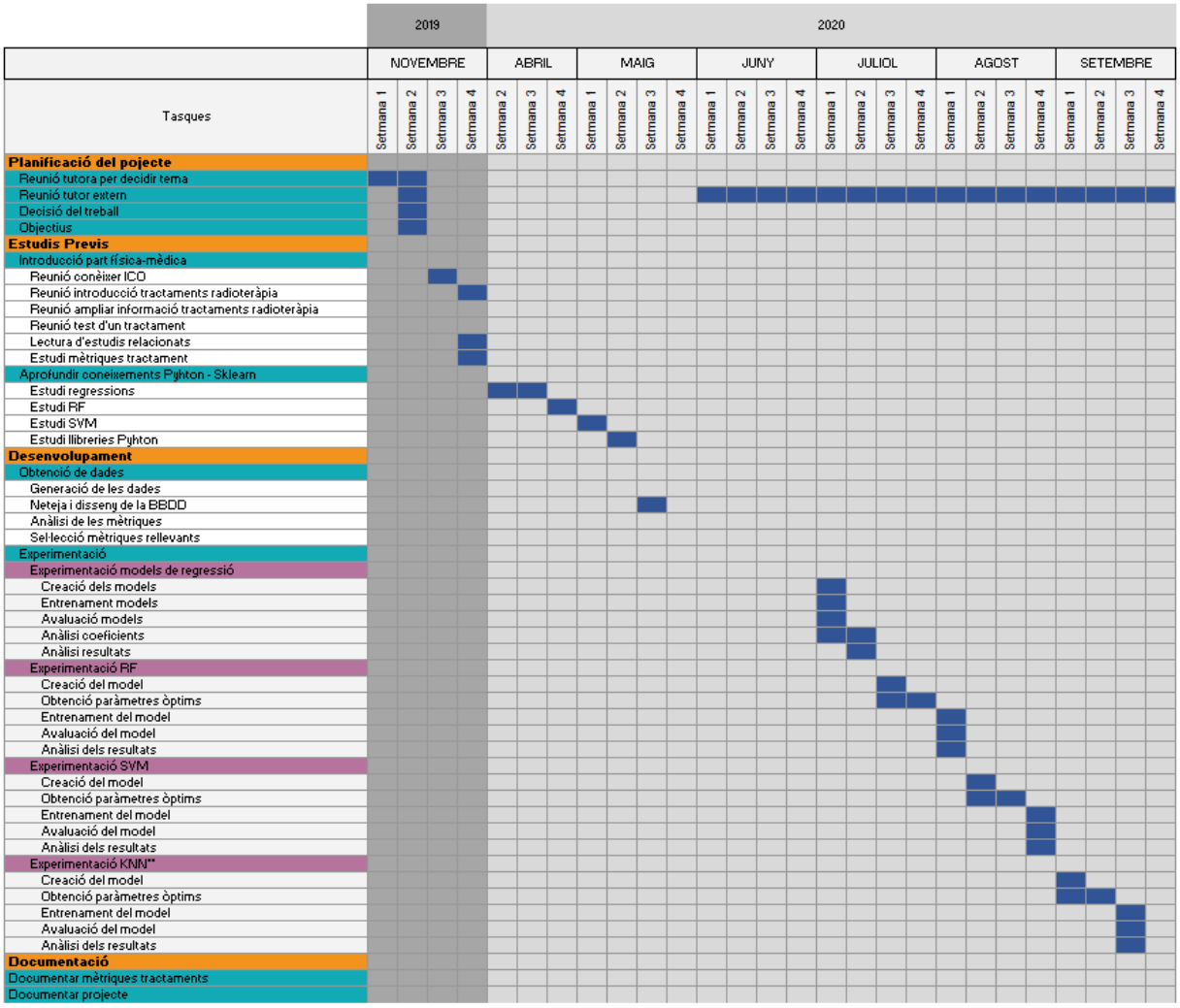


Figura 4.2: Diagrama de Gantt de 2019 i 2020.

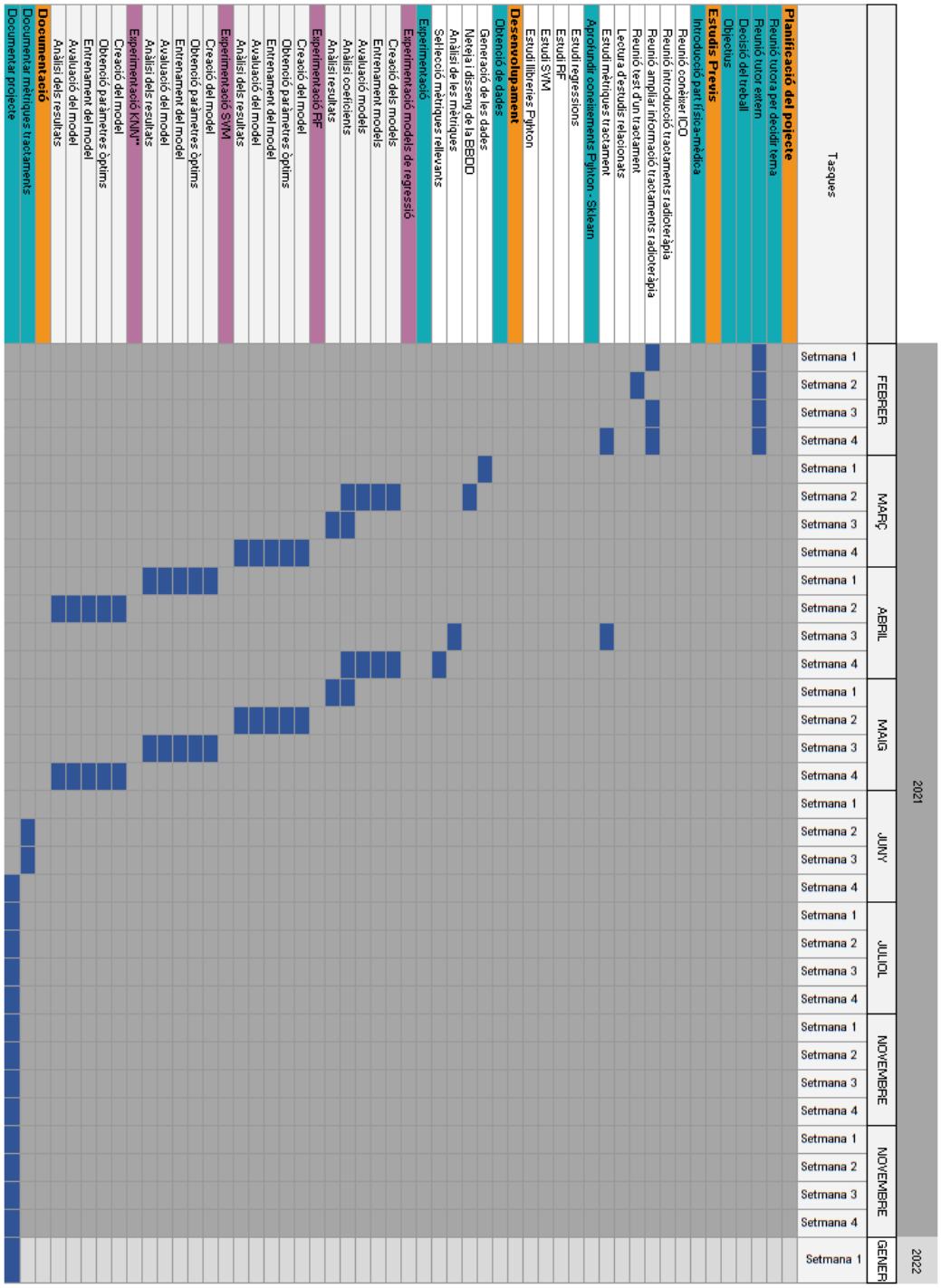


Figura 4.3: Diagrama de Gantt de 2021 i 2022.

CAPÍTOL 5

Marc de treball i conceptes previs

En l'apartat actual s'introduiran tots els conceptes necessaris per entendre millor aquest projecte.

5.1 Radioteràpia

5.1.1 Accelerador Linial

Un accelerador linial (AL) és un aparell capaç d'accelerar electrons fins a energies superiors als 6 MeV. Aquests electrons es fan xocar contra un blanc de tungstè per crear fotons amb una energia màxima igual a la dels electrons incidents, que són les partícules que utilitzarem per tractar els teixits desitjats.

Mitjançant un braç mòbil que pot girar al voltant del pacient, farem que els fotons incideixin en el pacient des de diferents angles per centrar el màxim de dosi de radiació en la zona a tractar mentre preservem els teixits sans (figura 5.1).

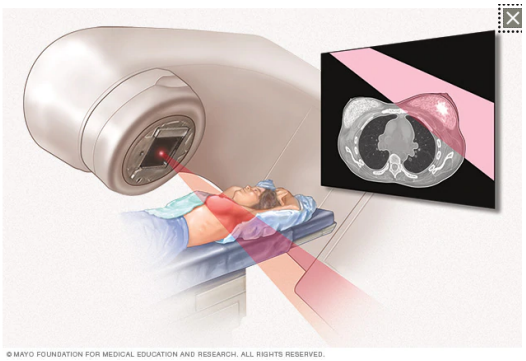


Figura 5.1: Esquema on es veu el braç de l'AL girat en una posició per tractar una mama. Imatge original Mayo Clinic.

A la sortida del feix, existeix un dispositiu anomenat *Multi Leaf Collimator* (MLC) que aconsegueix donar forma al feix de radiació per ajustar-lo a la zona a tractar (veure figura 5.2).

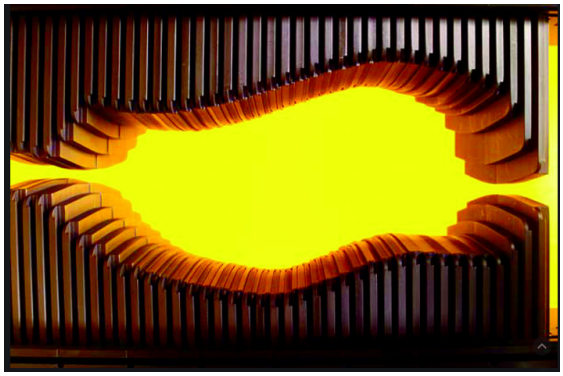


Figura 5.2: Fotografia retocada artísticament d'un MLC Millenium™ de 120 làmines de la marca Varian

5.1.2 VMAT

Per disminuir la dosi de radiació als teixits sans augmentant la dosi subministrada al volum a tractar, una de les estratègies més eficaces és que el feix de radiació incideixi sobre el pacient pel màxim d'angulacions possibles del braç. La tècnica VMAT fa girar el braç 360º dues vegades entorn el cos del pacient, ajustant l'MLC i modulant la radiació que surt en cada punt per subminstrar la dosi adequada al volum a tractar, protegint el teixit sa. En el següent vídeo es pot veure una simulació d'una sessió de tractament de radioteràpia, on es veu com es posiciona al pacient, se li fa un TAC previ al tractament per verificar que la posició és la correcta i, finalment es tracta amb una tècnica de VMAT. https://www.youtube.com/watch?v=lVUS6ZcBKaQ.

La configuració de les làmines i de la quantitat de radiació que surt del capçal en cada posició és el que s'anomena "pla de tractament". En els tractaments de VMAT aquest pla de tractament és molt complex i s'obté amb un *planificador*, un programa informàtic que optimitza les posicions de l'MLC i la quantitat de radiació que surt del capçal per tal que la dosi dins del pacient sigui la desitjada.

5.1.3 Correlació entre la dosi calculada i la rebuda pel pacient

El *planificador* opera en dos nivells: primer de tot, en la fase d'*optimització* calcula de manera ràpida sobre una imatge TAC del pacient quina serà la dosi que rebran els diferents òrgans en funció de tres paràmetres: l'angle del braç, la configuració de l'MLC i la quantitat de radiació donada per aquesta posició de braç (en endavant, *Taxa de Dosi*). Modificant la configuració de l'MLC i la taxa de dosi per cadascun dels 720º que el braç gira entorn el pacient (360º en sentit

horari, 360º en sentit antihorari) obtenim una distribució de dosi dins del pacient vàlida per tractar. En una segona fase de *càlcul*, es refina aquesta distribució de dosi utilitzant algoritmes de càlcul del transport de la radiació i diposició de dosi més acurats, a partir de la configuració obtinguda en la fase d'*optimització*.

El sistema d'optimització és un software propietari sotmès a patents que els venedors no solen voler compartir; el tractarem, per tant, com una caixa negra pel que fa el nostre treball: li donem uns inputs (el TAC del pacient i certs paràmetres de dosi desitjada) i ens retorna un output (configuració de l'MLC i taxa de dosi per cada angle). Aquest output l'utilitzem com a input d'un algorisme de càlcul i obtenim la distribució de dosi en el TAC del pacient. Els algorismes de càlcul sí que són subjectes a controls per part dels físics mèdics i podem assegurar que les mesures de dosi coincideixen amb les prediccions del sistema de càlcul dins d'un marge inferior al 3% *per camps coneguts*.

El primer que cal preguntar-se ara és si la dosi calculada és igual a la que rebrà el pacient i, en cas de que no siguin exactament iguals, quin marge ens permetem per acceptar el tractament com a vàlid. Analitzem primer de tot les causes que podrien fer que la dosi calculada i la rebuda per el pacient no fossin iguals:

1. Causes provinents del software de càlcul: els programes de càlcul de dosi utilitzen models matemàtics de transport de radiació i deposició de dosi basats en les dades subministrades per l'usuari. Aquests models funcionen molt bé en condicions controlades (camps amb configuració de l'MLC estàndard, dosi dipositada en maniquí d'aigua, etc) però, com qualsevol model, poden donar diferències quan ens allunyem d'aquestes condicions [Smilowitz 2015]. També sabem ([Muñoz-Montplet 2018, Bueno 2017]) que l'elecció de l'algorisme de càlcul de dosi pot donar diferents distribucions de dosi degut al tractament que s'utilitza per calcular el transport de radiació, la deposició de dosi, o com modelitza el nostre AL. En aquest treball, tots els plans de tractaments estan calculats amb el planificador Acuros (AXB) amb l'opció de calcular el transport de radiació en medi i la diposició de dosi en aigua.

2. Causes provinents de l'AL: en un pla de tractament li demanem al nostre AL que variï la configuració de l'MLC, la taxa de dosi i l'angle de braç de manera contínua. La fase d'optimització ja té en compte que aquests moviments mecànics no es poden fer a velocitat infinita i que la variació de la taxa de dosi no és instantània i, per tant, el nostre AL hauria de ser capaç de reproduir el pla de tractament sense cap problema. Però a la realitat l'AL pot no ser capaç de subministrar el pla que li demanem: pot tenir problemes mecànics que hagin passat desapercebuts en els controls

diaris i, per exemple, ser incapaç de moure el braç a la velocitat normal; per certs angles de braç (i.e: 90º i 270º), el moviment de l'MLC és bastant normal que es faci en contra la força de la gravetat i per tant la velocitat màxima de les làmines sigui més lenta que la teòrica.

Les implicacions clíniques d'un error en el càlcul de dosi en un pacient poden ser greus: infradosificacions en el tumors que poden repercutir en el control de la malaltia, o sobredosificacions en òrgans propers que comprometin l'estat de salut futur del pacient [Ash 1994].

Per saber si la distribució de dosi que la màquina subministrarà al pacient correspon amb la calculada procedim de la següent manera:

1. Quan considerem que la distribució de dosi calculada *sobre el TAC del pacient* és la correcta per tractar la seva malaltia, calculem quina seria la distribució de dosi *sobre un maniquí conegut* **mantenint els mateixos paràmetres a l'AL** (figura 5.3).

2. Posicionem el nostre maniquí a la taula de tractament. Subministrem el tractament amb l'AL com si fos el pacient real. El nostre maniquí conté detectors de radiació distribuïts dins seu. Mesurem la dosi que arriba a cada detector.

3. Comparem la dosi mesurada *sobre el maniquí* amb la calculada *sobre el maniquí*. La comparació es fa amb el que es coneix com a test gamma (veure 5.1.3.1). Si la comparació és favorable, s'accepta la planificació com a vàlida. Si no es passa el test, el procés de planificació comença de nou.

En el nostre cas, el maniquí de mesura és el ArcCheck de l'empresa Sun Nuclear i l'anàlisi gamma és fa amb el software de SNC-Patient de la mateixa empresa (figura 5.4).

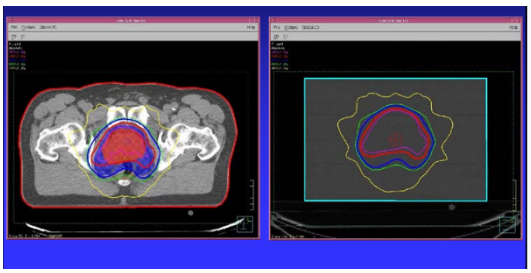


Figura 5.3: Distribució de dosi al pacient vs distribució en un maniquí [Mijnheer].

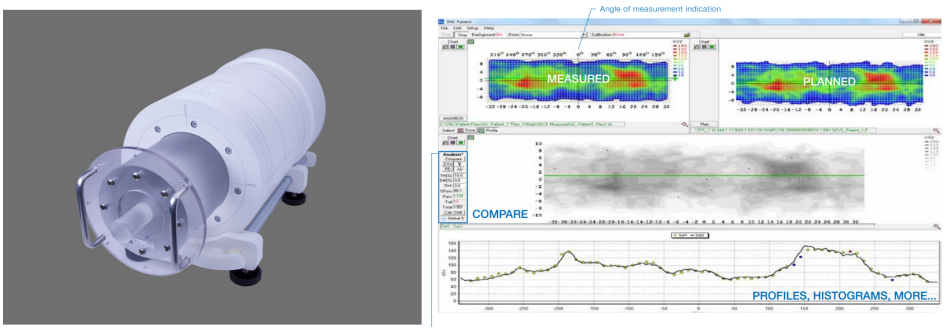


Figura 5.4: Imatges del maniquí ArcCheck i del software d'anàlisi. [SunNuclear Corporation 021]

A part de les possibles crítiques al test γ com a prova de predicció, [2nd ESTRO Physics Workshop-Malaga 2018 , Hussein 2017] el mètode descrit a dalt té dos desavantatges importants:

i) Consumeix temps de màquina (aprox 90 minuts al dia) que es podria dedicar a tractar pacients –un tractament estàndard dura 12 min a l'ICO de Girona, per tant, entre 7 i 8 pacients més al dia.

ii) Durant la fase de planificació no es pot saber *a priori* si un pla passarà el test gamma. Això pot fer que l'usuari decideixi no forçar les capacitats de la màquina per aconseguir una dosimetria millor pel pacient, per por a haver de començar tot el procés una altra vegada un cop fet el test.

5.1.3.1 El test gamma (γ)

Si volem comparar dues distribucions espacials de dosi, el primer que en ve al cap és calcular punt a punt la diferència entre les dues; en el nostre cas, per cada punt on tenim la dosi calculada, comparar aquesta dosi amb la dosi mesurada en el mateix punt. Aquest mètode es coneix com a diferència de dosi (*Dose Diference* en anglès. En endavant DD). Direm que les dues distribucions de dosi son comparables si la diferència entre les dues és menor a una valor δd que definim *a priori*. És un mètode fiable quan la nostra distribució de dosi no varia massa abruptament en l'espai. Ara bé, podem tenir distribucions de dosi amb molt gradient de dosi (la dosi varia molt entre un punt de càlcul i els punts propers). Per tant, utilitzar un mètode com el DD pot conduir a errors en zones amb molt de gradient, un error de pocs milímetres posicionant el detector pot donar inconsistències. A més, la dosi es calcula en un punt, però el detector de radiació té cert volum; això fa que la dosi no sigui constant en tot el volum de detecció, i podem tenir discrepàncies. Per solucionar-ho s'utilitza el mètode DTA (*Distance To Agreement*): per cada punt de mesura es busca el punt més proper

que tingui la mateixa dosi calculada. Direm que les distribucions són comparables si aquesta distància es menor a un δr.

Ara bé, en un tractament de radioteràpia, volem distribucions de dosi amb zones amb poc gradient de dosi (dins de la zona a tractar, per assegurar una dosi homogènia) i zones amb gradients de dosi alts (just en la frontera de la zona a tractar, on volem que la dosi caigui ràpidament per no irradiar el teixit sa.) Com podem unificar els dos criteris?

El test més utilitzat internacionalment [Miften 2018] és el test gamma (γ) proposat per Low al 1998 [Low 1998] i reproduït a l'equació 5.1

$$\Gamma(r_c, r_m) = \sqrt{\frac{(r_m - r_c)^2}{\delta r^2} + \frac{(d_m - d_c)^2}{\delta d^2}}$$
$$\gamma(r_m) = \min\{\Gamma(r_c, r_m)\}\ \forall r_c \tag{5.1}$$

on els subíndex c i m corresponen a *calculat* i *mesurat*, r_i és la posició espacial del punt, d_i la dosi en el punt i, δr i δd les toleràncies de dosi i distància anteriors.

S'observa que la fórmula primer compara cada punt mesurat amb tots els punts calculats i després es queda amb el valor mínim. Quan $\gamma(r_c) > 1$ diem que el punt no passa el test (això és fàcil de veure si pensem que el valor mínim de gamma es troba en $r_m = r_c$, llavors, si $d_m - d_c > \delta d$, $\gamma > 1$).

Visualment, en una dimensió espacial, cada punt de mesura es converteix en el centre d'un el·lipsoide de semieixos δr i δd. Si existeix un punt de dosi calculada dins d'aquest el·lipsoide, aquest punt té un índex $\gamma < 1$.

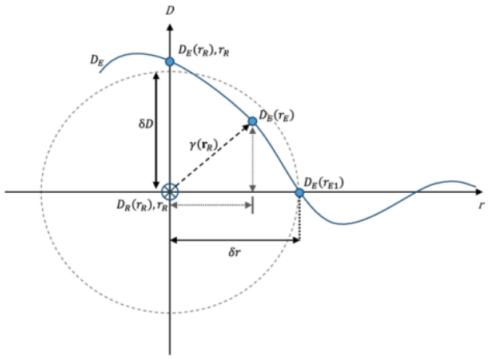


Figura 5.5: Representació gràfica de l'índex γ on δd i δr tenen el mateix valor absolut. De [Hussein 2017]

Es pot veure també que, en regions amb poc gradient de dosi, el terme $r_m - r_c$ tendeix a minimitzar-se i predomina el terme $d_m - d_c$, per tant γ es comporta com a DD. En canvi, en regions amb molt gradient, és fàcil trobar punts molt propers al punt de mesura on $d_m = d_c$ i, per tant, el terme que predomina és $r_m - r_c$ i γ es comporta com a DTA.

Un altre avantatge que té l'índex γ és que el podem calcular encara que els punts de càlcul i els punts de mesura no estiguin situats a les mateixes coordenades espacials exactes. Això no passa, per exemple, amb DD.

Les recomanacions internacionals [Miften 2018] ens diuen que hem d'utilitzar $\delta d = 3\%$ i $\delta r = 2mm$[1] ($\gamma(3\%, 2mm)$, per abreviar) i donar el pla per bo sempre que el 95% dels punts tinguin $\gamma < 1$, mentre que rejectem tots els plans on menys del 90% dels punts tinguin $\gamma < 1$. Els plans on superin el test entre el 90-95% dels punts s'han d'analitzar més detalladament.

Tot i això, els plans utilitzats en aquest treball tenen un biaix gran: són plans que s'han utilitzat en pacients, així que no n'hi ha cap amb menys del 90% de punts que no sobrepassin el test γ; per obtenir més representació de plans "dolents", hem utilitzat una prova γ més restrictiva, i hem calculat el $\gamma(2\%, 2mm)$.

[1] S'exclouen de l'anàlisi tots els punts on la dosi mesurada es inferior al 10% de la dosi prescrita. Es fa així perquè donarien problemes de càlcul (com tractem dosis =0 en la matriu de càlcul?) i perquè se sap que certs detectors responen de manera diferent a les zones de baixa dosi.

5.2 Conceptes i algoritmes en intel·ligència artificial i estadística

En aquest apartat es descriuran els conceptes i algoritmes en intel·ligència artificial i estadística que s'han utilitzat en aquest treball [James 2013].

5.2.1 Algoritmes de Machine Learning

Els models de *Machine Learning* es basen bàsicament en tres tipus d'aprenentatge:

- **Aprenentatge supervisat**: Aquests algoritmes s'utilitzen quan es té un conjunt de dades totalment classificades, és a dir, per cada entrada es sap la seva solució. En aquests casos es crea una funció on se li passa un conjunt de les dades, les dades d'entrenament. Quan la funció s'ha entrenat i ja s'ha adaptat a les dades, posteriorment s'utilitza en un nou conjunt de dades de prova per fer una predicció i veure com d'exacte és el model. La finalitat és crear una funció que s'ajusti el millor possible a les noves dades.

- **Aprenentatge no supervisat**: Aquests algoritmes s'utilitzen quan les dades no estan classificades, per la qual cosa no es té cap indicació prèvia sobre quines són les respostes esperades. Per tant, en aquests algoritmes es tracten les dades d'entrada com a variables aleatories, i el sistema haurà de ser capaç d'identificar els patrons per classificar les noves entrades.

- **Aprenentatge per reforç**: Aquests algoritmes funcionen a través de la prova i error. S'obtenen les dades d'entrada a través de la informació que es va obtenint del món exterior com a resposta a les seves accions.

Tots els models utilitzats en aquest projecte són models d'aprenentatge supervisat.

5.2.2 Regressió Lineal

La regressió lineal és una tècnica estadística utilitzada en Machine Learning que intenta modelar la relació lineal entre les variables predictores independents X i una variable dependent Y.
Hi ha diferents tipus de models de regressió lineal, però nosaltres ens centrarem en la regressió lineal múltiple ja que en el nostre conjunt de dades hi trobem vàries propietats o variables independents. Un model de regressió lineal múltiple es pot representar de forma matemàtica com:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \varepsilon$$

on: β_i són els coeficients, X_i les variables independents o predictors, Y la variable dependent i ε l'error associat.

Durant el procés d'ajustar el model s'utilitza una funció de pèrdua o funció de costos coneguda com la suma de quadrats de les diferències. El valor dels coeficients s'escullen de forma que es minimitzi el resultat d'aquesta funció

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

on: β_0 és l'intercepció amb y (constant), y_i és la variable dependent, n és el nombre de dades d'aprenentatge, p és el nombre de variables independents, β_j és el coeficient i, x_{ij} és la variable independent.

Ara això ajustarà els coeficients en funció de les dades d'entrenament. Si hi ha soroll en aquestes dades llavors els coeficients estimats no seran capaços de generalitzar el càlcul per a predir les noves dades. És en aquest punt on passa a tenir importància la regularització, per tal de disminuir les estimacions d'aquells coeficients que perjudiquen en la capacitat de generalitzar del model cap a 0.

5.2.3 Regularització

Un dels aspectes més importants a tenir en compte a l'hora d'entrenar un model és evitar el què es coneix com *underfitting* i *overfitting*.

L'*overfitting* és l'efecte que es produeix quan un model estadístic s'adapta massa bé a les dades d'entrenament, és a dir, intenta aprendre dels sorolls de les dades d'entrenament. Entenem com a soroll tota aquella informació que no representa una propietat real de les nostres dades, sinó que són factors aleatoris. El problema és que aquests sorolls no es troben a les noves dades i acaba impactant negativament en la capacitat del model per generalitzar, fent que l'algorisme no pugui funcionar amb precisió contra noves dades. Un model pateix d'*overfitting* quan la variància és molt alta i el biaix és molt baix.

L'*underfitting*, per altra, banda es produeix quan el nostre model té un biaix molt alt, és a dir, el nostre model no és capaç de captar amb precisió la relació real entre els predictors i els resultats. Es produeix quan un model és massa simple, cosa que pot ser el resultat d'un model que necessita més temps d'entrenament, més funcions d'entrada o menys regularització. Igual que per l'*overfitting*, quan un model pateix d'*underfitting*, no pot establir la tendència dominant dins les

dades, cosa que provoca errors d'entrneament i un rendiment baix del model. Això pot passar, per exemple, quan no tenim prou dades d'entrenament i el model no és capaç de trobar cap forma de generalitzar.

Si un model no pot generalitzar bé amb les noves dades, no es pot aprofitar per a tasques de classificació o predicció. La generalització d'un model a noves dades és, en última instància, el que ens permet utilitzar algoritmes d'aprenentatge automàtic per fer prediccions i classificar dades. És per això que cal anar amb compte i intentar evitar les situacions comentades.

Hi ha diverses tècniques per afrontar aquestes situacions, com ara la validació creuada (veure 9.4.1) i la regularització .

La regularització és una forma de regressió que restringeix o redueix les estimacions dels coeficients cap a zero aconseguint així reduir tant el biaix com la variància del model. En altres paraules, aquesta tècnica dissuadeix l'aprenentatge d'un model més complex o flexible per evitar el risc de sobreajustament. Això s'aconsegueix mitjançant la introducció d'un terme de penalització en la funció de costos que comporta una penalització més alta a les corbes complexes.

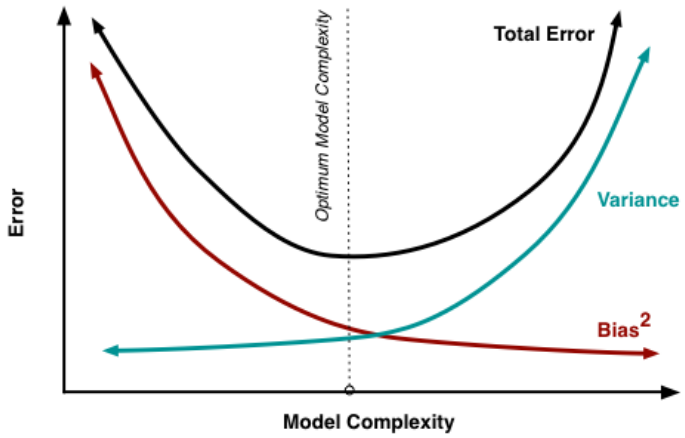


Figura 5.6: Representació gràfica de l'evolució del biax i la variància segons la complexitat del model. De [Wikipedia contributors 2021a]

Les regressions que s'han utilitzat per l'anàlisi de les dades en aquest projecte utilitzen tècniques de regularització. Així doncs, gràcies a aquesta particularitat, s'ha pogut realitzar l'estudi matemàtic sobre quines són les propietats que realment aporten informació a l'hora de determinar la validesa d'un tractament. A

més de poder determinar si existeix algun tipus de correlació entre les variables independents.

5.2.4 Regressió Ridge

La regressió Ridge és un tipus de regressió lineal on s'inclou una penalització equivalent al quadrat de la magnitud dels coeficients. En altres paraules, podríem dir que s'afegeix una restricció per tal de minimitzar la funció de pèrdua. Podem representar aquesta restricció de la següent manera:

$$c > 0, \sum_{j=0}^{p} \beta_j^2 < c$$

on: c representa el valor del coeficient, p el nombre total de variables independents i β el coeficient.

A més, per tal de poder decidir com d'influent és aquesta restricció, també s'afegeix a la funció un terme de penalització que es representa amb el símbol (λ). Aquest nou paràmetre decideix fins a quin punt volem penalitzar la flexibilitat del nostre model. Per tal d'augmentar la flexibilitat d'un model el què és fa és augmentar el valor dels coeficients, per tant, si el què pretenem és minimitzar la funció de pèrdua llavors estem obligant als coeficients a fer-se petits i per tant estem reduint la flexibilitat. Fent això, el què estem aconseguint és reduir la influència de cada variable en la predicció. Com més alt és el valor de λ més restrictiva és la restricció. Si el valor de λ fos zero la restricció s'anul·laria i ens quedaríem amb una regressió lineal bàsica.
Així doncs la funció de costos per a la regressió Ridge quedaria de la següent manera:

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=0}^{p} \beta_j^2$$

on: β_j són els coeficients, X_i les variables independents o predictors i, y_i la variable dependent.

La regressió Ridge és útil quan tenim un conjunt de predictors on tots ells aporten informació al model. El què s'aconsegueix és donar més rellevància a les variables que contribueixen més en disminuir l'error del model i reduir la importància d'aquelles que aporten menys informació sense haver de prescindir-ne, ja que continuen aportant informació i ignorar-les ens conduiria a la disminució de la precisió per pèrdua d'informació.

5.2.4.1 Sklearn RidgeCV

Aquest algorisme es troba implementat dins la llibreria d'sklearn de python i es pot utilitzar a través de la funció *RidgeCV()*. A continuació s'exposa un exemple parametritzat de la funció:

*RidgeCV(alphas=(0.1, 1.0, 10.0), *, fit_intercept=True, normalize=False, scoring=None, cv=None, gcv_mode=None, store_cv_values=False)* [scikit-learn developers 2021e].

La part més important a l'hora d'utilitzar aquestes funcions és la bona definició dels paràmetres. Algun dels paràmetres més importants són:

- **alphas**: Llista de valors d'α que es provaran durant l'entrenament de l'algorisme. S'escollirà la millor utilitzant validació creuada. α és el terme de penalització de la funció de costos, també es pot expressar com λ.
- **normalize**: És un booleà que indica si les dades s'han de normalitzar abans de tractar.
- **cv**: Nombre enter que indica el nombre de particions utilitzades en la validació creuada.
- **gcv mode**: Indica l'estratègia utilitzada per fer la validació creuada.

5.2.5 Regressió Lasso

La regressió Lasso és un altre tipus de regressió lineal que, conceptualment, és molt similar a la regressió Ridge, ja que només en varia la restricció que s'afegeix a la funció de costos. En aquest cas, la restricció, en comptes de penalitzar la suma dels quadrat dels coeficients, penalitza la suma del seu valor absolut fent que per valors elevats del terme λ varis coeficients passin a ser exactament zero, fet que no arriba a passar mai amb la regressió Ridge.

La penalització de la regressió Lasso (L1) es pot representar de la següent manera:

$$c \geq 0, \sum_{j=0}^{p} |\beta_j| < c$$

on: c representa el valor del coeficient, p el nombre total de variables independents i β el coeficient.

Tenint en compte la penalització anterior, la funció d'una regressió Lasso queda de la següent forma:

$$RSS = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=0}^{p}|\beta_j|$$

on: β_j són els coeficients, X_i les variables independents o predictors i y_i la variable dependent.

Per tal de determinar la influència de la restricció, igual que s'ha vist en la regressió Ridge, aquí també s'inclou el terme de penalització (λ).

La regressió Lasso pot ser útil per exloure variables insignificants de l'equació del model. En altres paraules, la regressió Lasso pot ajudar en la selecció de característiques.

5.2.5.1 Selecció de variables en la regressió Lasso

El fet que permet que la regressió Lasso resulti en estimacions dels coeficients iguals a zero, a diferència de la regressió Ridge, es pot explicar utilitzant la imatge que es mostra a continuació:

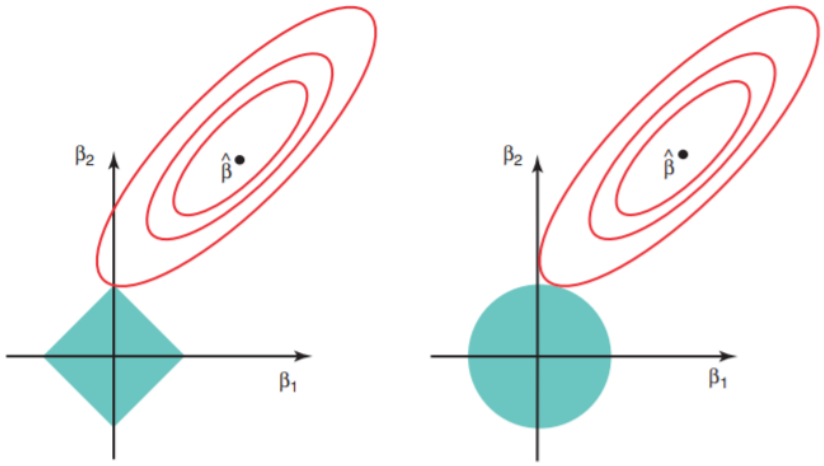


Figura 5.7: Contorns de les funcions d'error i restricció per a les regressions Lasso (esquerra) i regressions Ridge (dreta). Les zones blaves són les regions de restricció, mentre que les el·lipses vermelles són els contorns de la *RSS*. $\hat{\beta}$ és la solució de la funció de costos per un coeficient [James 2013].

Les el·lipses centrades al voltant de $\hat{\beta}$ representen regions de la constant de la

funció de costos. En altres paraules, tots els punts d'una el·lipse comparteixen el mateix valor de la funció de costos. A mesura que les el·lipses s'allunyen de les estimacions dels coeficients de la funció de mínims quadrats, el valor de la funció augmenta. Les estimacions dels coeficients de les regressions Lasso i Ridge venen donades pel primer punt en què una el·lipse entra en contacte amb la regió de restricció. Com que la regressió Ridge té una restricció circular, aquesta intersecció no es produirà gairebé mai en un eix i, per tant, les seves estimacions dels coeficients seran diferents de zero. No obstant això, la restricció Lasso, al tenir una forma similar a un diamant, compta amb cantonades en cadascun dels eixos, de manera que l'el·lipse sovint tallarà la regió de restricció en un eix, fet que comportarà que un dels coeficients sigui igual a zero.

5.2.5.2 Sklearn LassoCV

Aquest algorisme es troba implementat dins la llibreria d'sklearn de python i es pot utilitzar a través de la funció *LassoCV()*. A continuació s'exposa un exemple parametritzat de la funció:

LassoCV(, eps=0.001, n_alphas=100, alphas=None, fit_intercept=True, normalize=False, precompute='auto', max_iter=1000, tol=0.0001, copy_X=True, cv=None, verbose=False, n_jobs=None, positive=False, random_state=None, selection='cyclic)* [scikit-learn developers 2021c]

La part més important a l'hora d'utilitzar aquestes funcions és la bona definició dels paràmetres. Alguns dels paràmetres més importants són:

- **alphas**: Llista de valors d'α que es provaran durant l'entrenament de l'algorisme. S'escollirà la millor utilitzant validació creuada. α és el terme de penalització de la funció de costos, també es pot expressar com λ.

- **normalize**: És un booleà que indica si les dades s'han de normalitzar abans de tractar.

- **cv**: Nombre enter que indica el nombre de particions utilitzades en la validació creuada.

- **max_iter**: Nombre màxim d'iteracions de l'algorisme.

- **selection**: Paràmetre que indica el criteri seguir a l'hora d'escollir el coeficient a actualitzar en cada iteració.

5.2.6 Comparativa regressió Ridge i Lasso

En general, es pot esperar que Lasso tingui un millor rendiment en un entorn on un nombre relativament petit de predictors tenen coeficients substancials i els predictors restants tenen coeficients molt petits o iguals a zero. La regressió Ridge tindrà un millor rendiment quan la resposta sigui una funció de molts predictors, tots amb coeficients de mida similar. Tanmateix, el nombre de predictors relacionats amb la resposta no es coneix mai a propri per a conjunts de dades reals. Es pot utilitzar una tècnica com la validació creuada per determinar quin enfocament és millor en un conjunt de dades concret.
Igual que amb la regressió Ridge, quan les estimacions de mínims quadrats (*RSS*) tenen una variància excessivament alta, la solució Lasso pot produir una reducció de la variància a costa d'un petit augment del biaix i, en conseqüència, pot generar prediccions més precises.
A diferència de la regressió Ridge, Lasso realitza una selecció de variables i, per tant, resulta en models més fàcils d'interpretar.

5.2.7 LARS algorithm

LARS o regressió de l'angle mínim és un algorisme de regressió molt útil quan es treballa amb conjunts de dades de grans dimensions, és a dir, dades amb un gran nombre d'atributs. Això és així gràcies a l'eficiència i l'eficàcia que proporciona a l'hora de selecionar les variables més significatives d'un model.

Per entendre bé el funcionament i els avantatges d'aquest algorisme es consideren a continuació altres mètodes de selecció de models que el precedeixen:

- **Selecció cap endavant**: És un mètode de selecció de variables ràpid i senzill, però que comporta alguns desavantatges.

  En aquest algorisme, el model comença sense variables i, a cada pas, s'afegeix la variable amb més potència explicativa. El procés s'atura quan es compleix alguna regla pre-establerta, o bé quan totes les variables sota consideració s'han afegit al model.

  El problema d'aquest mètode és que a cada pas que es realitza, s'afegeix de forma completa una de les variables en el model. Si les dades contenen variables que aporten informació al model però estan fortament correlacionades entre elles, en el moment en què en un dels passos s'afegeix una d'aquestes variables perquè es detecta que és la que aporta més informació al model, les altres variables ja no aportaran gaire poder explicatiu i,

per tant, serà poc probable que s'incloguin en el model.

- **Regressió progressiva cap endavant:** La regressió progressiva cap endavant intenta solucionar l'avarícia de la selecció directa afegint les variables al model de forma parcial en comptes de fer-ho de forma completa.

  En aquest algorisme, el model comença amb totes les variables al conjunt actiu, però amb els coeficients a 0. Quan es troba la variable amb més potència explicativa s'actualitza el seu pes fins que n'apareix una amb una potència explicativa superior. Quan la correlació de dos variables és igual, es continua actualitzant el pes de la primera fins que n'hi hagi una altra que la superi. El procés s'atura quan l'actualització de la variable més correlacionada ja no aporta un increment significatiu d'informació al model.

  El problema d'aquest mètode és el gran nombre d'actualitzacions que s'han de realitzar, provocant que sigui un algorisme molt poc eficient.

Amb LARS el què es busca és millorar l'eficiència de la qual careix la regressió progressiva per passos. Això s'aconsegueix fent que, en el moment en què hi ha diverses variables amb la mateixa correlació respecte la variable independent, en lloc de continuar augmentant una de les variables, s'actualitzen totes en una direcció equiangular entre elles.

Els passos d'aquest mètode s'exposen a continuació:

1. S'inicialitzen tots els coeficients β_x a 0.
2. S'identifica el predictor, X_j, que està més correlacionat amb la variable indepedent Y.
3. S'augmenta el coeficient β_j en la direcció que està més correlacionada amb Y i es para quan es troba algun altre predictor X_k que tingui una correlació igual o major que X_j.
4. Si la correlació és la mateixa, s'extenen els coeficients (β_j, β_k) en una direcció equiangular tant a X_j com a X_k.
5. Es repeteix fins que tots els predictors estan en el model o s'han complert els requeriments del model.

Els avantatges principals d'utilitzar LARS són:

- És molt eficient quan el nombre de característiques és significativament superior al nombre de mostres.

- Computacionalment és tan ràpid com la regressió progressiva cap endavant i té el mateix ordre de complexitat que els mínims quadrats.
- Intuïtivament, si dues característiques es relacionen de forma similar amb la variable independent (tenen una correlació aproximadament igual), els seus coeficients han d'augmentar a la mateixa velocitat. LARS compleix aquesta idea.
- **És fàcilment modificable per utilitzar-se en altres estimadors, com Lasso**.

Tot i els beneficis d'utilitzar LARS, cal destacar i tenir en compte que és especialment sensible als efectes del soroll que es troba a les dades, cosa que pot comportar resultats imprevisibles.

5.2.8 Regressió Lasso LARS

La regressió Lasso LARS és igual que un model Lasso implementat mitjançant l'algorisme LARS, és a dir, en comptes d'utilitzar l'algorisme del gradient descendent, per minimitzar la funció de la suma de quadrats de les diferències per trobar el valor dels coeficients, utilitza l'algorisme LARS.

5.2.8.1 Sklearn LassoLarsCV

Aquest algorisme es troba implementat dins la llibreria d'sklearn de python i es pot utilitzar a través de la funció *LassoLarsCV()*. A continuació s'exposa un exemple parametritzat de la funció:

LassoLarsCV(, fit_intercept=True, verbose=False, normalize=False, max_iter=500, normalize=True, precompute='auto', cv=None, max_n_alphas=1000, n_jobs=None, eps=*$2.220446049250313e^{-16}$*, copy_X=True, positive=False)* [scikit-learn developers 2021d]

La part més important a l'hora d'utilitzar aquestes funcions és la bona definició dels paràmetres. Alguns dels paràmetres més importants són:

- **normalize**: És un booleà que indica si les dades s'han de normalitzar abans de tractar.
- **cv**: Nombre enter que indica el nombre de particions utilitzades en la validació creuada.
- **max_iter**: Nombre màxim d'iteracions de l'algorisme.

5.2.9 Regressió ElasticNet

Elastic Net és un model de regressió lineal que combina les restriccions utilitzades en les regressió Lasso i Ridge. Això permet generar un model on només alguns dels coeficients no siguin nuls, mantenint les propietats de regularització de Ridge.

A l'ajuntar les dues restriccions o regularitzacions amb la suma dels quadrats de les diferències, s'obté la funció de costos de la regressió ElasticNet. La funció es descriu a continuació:

$$RSS_{ElasticNet} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\rho\sum_{j=0}^{p}|\beta_j| + \lambda(1-\rho)\sum_{j=0}^{p}\beta_j^2$$

on: β_j són els coeficients, X_{ij} les variables independents o predictors, y_i la variable dependent, λ el terme penalitzador i ρ el terme que controla la combinació de les restriccions.

Durant l'optimització de la funció de costos, el terme de penalització de la restricció Ridge redueix els pesos del model cap a valors propers a zero. Gràcies a aquesta penalització dels pesos, el model es fa més senzill, més genèric i menys propens al sobre-ajustament. El terme de penalització de la restricció Lasso causa que els pesos més propers a zero esdevinguin zero i, per tant, s'eliminin alguns dels predictors presents en la funció inicial, aconseguint un model més predictiu.

Segons el valor de ρ es tenen en compte les següents consideracions:

- Si ρ és 0, la regressió ElasticNet es comporta igual que una regressió Ridge.
- Si ρ és 1, la regressió ElasticNet es comporta igual que una regressió Lasso.

També s'ha de tenir en compte que si el valor de λ és 0, la regressió ElasticNet es comporta com una regressió lineal bàscia.

5.2.9.1 Sklearn ElasticNet

Aquest algorisme es troba implementat dins la llibreria d'sklearn de python i es pot utilitzar a través de la funció *ElasticNetCV()*. A continuació s'exposa un exemple parametritzat de la funció:

ElasticNetCV(, l1_ratio=0.5, eps=0.001, n_alphas=100, alphas=None, fit_intercept=True, normalize=False, precompute='auto', max_iter=1000, tol=0.0001, cv=None, copy_X=True, verbose=0, n_jobs=None, positive=False, random_state=None, selection='cyclic')* [scikit-learn developers 2021b]

La part més important a l'hora d'utilitzar aquestes funcions és la bona definició dels paràmetres. Alguns dels paràmetres més importants són:

- **l1_ratio**: Valor entre 0 i 1 que determina el pes en la combinació de restriccions del model.
- **alphas**: Llista de valors d'α que es provaran durant l'entrenament de l'algorisme. S'escollirà la millor utilitzant validació creuada. α és el terme de penalització de la funció de costos, també es pot expressar com λ.
- **normalize**: És un booleà que indica si les dades s'han de normalitzar abans de tractar.
- **cv**: Nombre enter que indica el nombre de particions utilitzades en la validació creuada.
- **max_iter**: Nombre màxim d'iteracions de l'algorisme.
- **selection**: Paràmetre que indica el criteri a seguir a l'hora d'escollir el coeficient a actualitzar en cada iteració.

5.2.10 Arbres de decisió

Els arbres de decisió són un mètode d'aprenentatge supervisat no paramètric que s'utilitzen per a la classificació i la regressió. L'objectiu és crear un model capaç de predir el valor d'una variable objectiu mitjançant l'aprenentatge de regles de decisió senzilles inferides a partir de les característiques de les dades.

Els arbres de decisió estan formats per tres elements:

- **Nodes**: Contenen els atributs i els criteris per a dividir les dades.
- **Arcs**: Indiquen el resultat de l'operació booleana.
- **Fulles**: Nodes finals on es classifica la mostra.

Un arbre de decisió es pot representar gràficament de la següent forma:

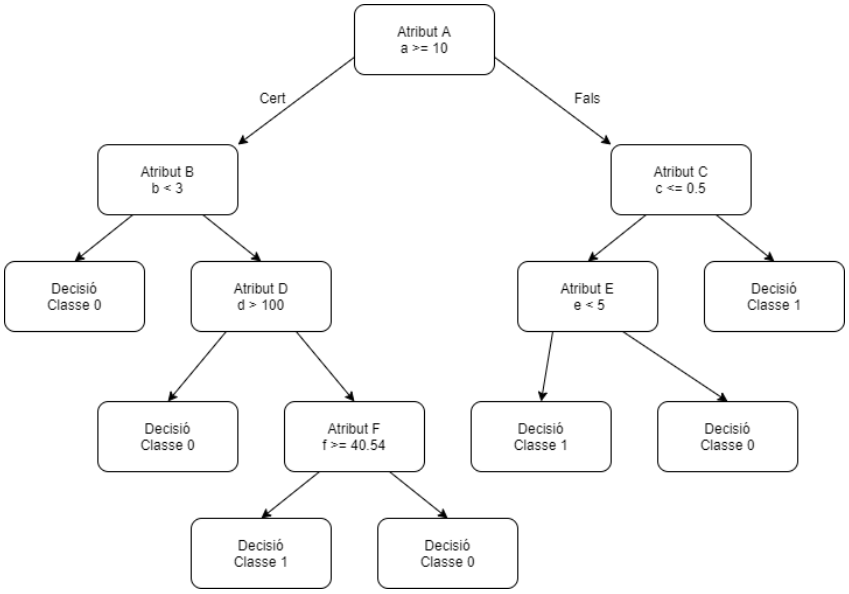


Figura 5.8: Exemple d'arbre de decisió binari.

Aquests arbres es creen segons un algorisme on, de forma recursiva, es van escollint els atributs segons el grau d'informació que aporten al model. Un cop s'escull l'atribut, es realitza de forma exhaustiva la cerca del millor punt per dur a terme la divisió de les dades. Aquest procés recursiu de tria d'atributs i escollir el millor valor per on dividir les dades s'atura quan es compleix un criteri preestablert. El criteri més comú consisteix en indicar un nombre mínim d'instàncies o mostres d'entrenament pels nodes fulla, d'aquesta forma si el nombre de mostres que arriben en un node és inferior a aquest mínim, la divisió no s'accepta i el node es pren com a node final. La construcció de l'arbre es dona per finalitzada quan tots els nodes més externs de totes les branques són fulles.

5.2.11 Boscos aleatoris

Els mètodes d'aprenentatge de conjunts estan formats per un conjunt de classificadors, per exemple, arbres de decisions i les seves prediccions s'agrupen per identificar el resultat més popular. Un dels mètodes de conjunts més conegut és el *Bagging*. En aquest mètode es selecciona una mostra aleatòria de dades d'un conjunt d'entrenament amb substitució, això significa que es pot agafar la mateixa mostra més d'una vegada. Després de generar els diversos subconjunts de dades els models es formen de forma independent.

L'algorisme de bosc aleatori és una extensió del mètode de *bagging*, ja que utilitza el *bagging* i la selecció aleatòria d'atributs per crear un bosc d'arbres de decisió no correlacionats. La selecció aleatòria d'atributs, genera subconjunts

aleatoris d'atributs, fet que garanteix una correlació baixa entre els arbres de decisió. Aquesta és una diferència clau entre els arbres de decisió i els boscos aleatoris. Tot i que els arbres de decisió consideren totes les possibles divisions de característiques, els boscos aleatoris només seleccionen un subconjunt d'aquestes.

Els algorites de boscos aleatoris tenen tres hiperparàmetres principals, que s'han d'establir abans de l'entrenament. Aquests inclouen la mida del node, el nombre d'arbres i, el nombre de característiques mostrejades. A partir d'aquí, el classificador de bosc aleatori es pot utilitzar per resoldre problemes de regressió o classificació.

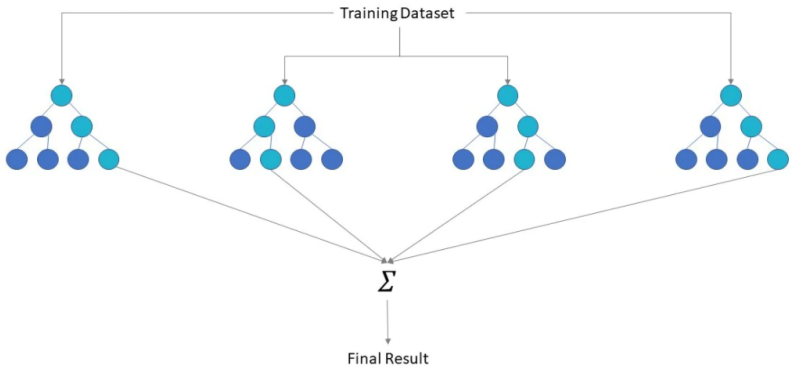


Figura 5.9: Representació gràfica d'un bosc aleatori. El resultat s'obté de la combinació de prediccions dels arbres de decisió que el componen [IBM 2020]

L'algorisme del bosc aleatori es compon d'una col·lecció d'arbres de decisió, on cadascun d'ells està format per una mostra de dades extreta d'un conjunt d'entrenament amb reemplaçament, anomenada mostra de *bootstrap*. D'aquesta mostra d'entrenament se'n reserva una part com a dades de prova, coneguda com a mostra fora de la bossa (oob). A continuació s'inclou més aleatorietat al model mitjançant la selecció aleatòria dels atributs, afegint més diversitat al conjunt de dades i, reduint la correlació entre els arbres de decisió. Depenent del tipus de problema a resoldre el resultat de la predicció variarà. Per a una tasca de regressió, el resultat s'obtindrà calculant la mitja de les prediccions fetes per tots els arbres de decisió i, per a una tasca de classificació, es farà un recompte de la variable categòrica i el resultat en serà la més freqüent. Finalment, s'utilitza la mostra reservada de dades (oob) per a la validació creuada, finalitzant la predicció i obtenint una precisió bastant ajustada a la realitat del nou model.

Els boscos aleatoris són algorismes molt fiables i redueixen bastant el risc de

sobreajustament gràcies al fet de d'utilitzar de forma conjunta varis models no correlacionats disminuïnt així la variància i l'error en la predicció del model final. En contrapartida, cal tenir en compte que els boscos aleatoris poden arribar a ser models molt complexos i computacionalment lents degut a la gran quantitat de dades que arriben a tractar.

5.2.11.1 Sklearn RandomForestClassifier

Aquest algorisme es troba implementat dins la llibreria d'sklearn de python i es pot utilitzar a través de la funció *RandomForestClassifier()*. A continuació s'exposa un exemple parametritzat de la funció:

*RandomForestClassifier(n_estimators=100, *, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None)* [scikit-learn developers 2021a]

La part més important a l'hora d'utilitzar aquestes funcions és la bona definició dels paràmetres. Alguns dels paràmetres més importants són:

- **n_estimators**: Nombre d'arbres de decisió del bosc.
- **min_samples_split**: Nombre mínim de mostres en els nodes dels arbres per a dividir les dades.
- **max_features**: Nombre de característiques a tenir en compte.
- **criterion**: Mesura que s'utilitza per avaluar la qualitat de la divisió.
- **bootstrap**: Indicador per determinar si s'utilitzen les mostres de *bootsrap* a l'hora de construir els arbres.
- **class_weights**: Pesos associats a cada classe.

5.2.12 Màquines de suport vectorial

Una màquina de suport vectorial és un model que representa els punts de mostra en un espai vectorial de tantes dimensions com característiques hi hagi en les dades, separant les classes el més àmpliament possible per un hiperpla de

separació que està definit segons uns vectors de suport. Els vectors de suport són els punts de les dades que es troben més a prop de l'hiperpla i influeixen en la seva posició i orientació.

La dimensió de l'hiperpla depèn del nombre de característiques de les dades. En casos ideals on les dades són linealment separables, el separador es representa com una línia quan el nombre de dimensions és 2, com un pla quan es tracta de 3 dimensions, i com un hiperpla per a dimensions superiors.

L'eficiència del model recau en trobar un hiperpla que separi les dades de forma que es minimitzi el nombre de casos on la mostra d'una de les classes cau dins de l'espai d'una classe que no és la seva.

Per separar les mostres segons les seves classes hi ha molts hiperplans possibles que es podrien triar. L'objectiu doncs, és trobar un pla que tingui el marge màxim, és a dir, la distància màxima entre punts de diferent classe. Maximitzar la distància del marge proporciona cert reforç per a què les mostres es puguin classificar amb més confiança.

A continuació es pot veure l'exemple d'una màquina de suport vectorial simple:

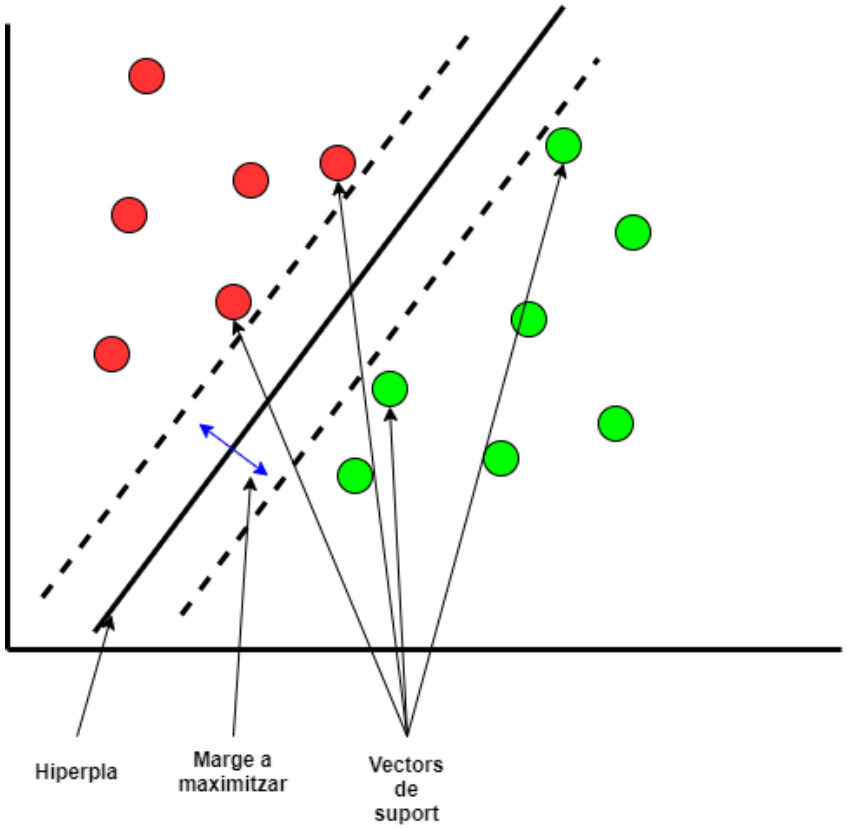


Figura 5.10: Representació gràfica d'una màquina de suport vectorial de dues dimensions i dues classes on les dades són linealment separables.

5.2.12.1 Sklearn SVC

Aquest algorisme es troba implementat dins la llibreria d'sklearn de python i es pot utilitzar a través de la funció *SVC()*. A continuació s'exposa un exemple parametritzat de la funció:

SVC(, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=- 1, decision_function_shape='ovr', break_ties=False, random_state=None)* [scikit-learn developers 2021g]

La part més important a l'hora d'utilitzar aquestes funcions és la bona definició dels paràmetres. Alguns dels paràmetres més importants són:

- **C**: Paràmetre de penalització que permet ajustar el marge de decisió. Per a valors molt alts de C, el marge és molt estricte, més restrictiu, fent que

els punts no puguin trobar-se dins dels marges establerts. Per valors petits de C, el marge serà molt més suau i es permetrà que els punts puguin situar-se sobre aquesta franja sense cap problema.

- **gamma**: Defineix la influència que té cada exemple de les dades d'entrenament. Com més gran és el valor de γ més a prop s'han de trobar la resta d'exemples per veure's afectats per aquest.

- **class_weight**: Pesos associats a cada classe.

5.2.13 K Nearest Neighbors

L'algorisme de *k-nearest-neighbors* o *knn* és un mètode d'aprenentatge automàtic supervisat no paramètric que es basa en assumir que les coses similars es troben a prop les unes de les altres i s'allunyen a mesura que deixen d'assemblar-se. A vegades també es defineix com un model no generalitzador, ja que simplement recorda les dades que ha vist durant l'entrenament.

Els algorismes de *knn* es poden utilitzar tant per a tasques de classificació quan les etiquetes de les dades siguin valors discrets, com per a tasques de regressió quan les etiquetes siguin valors continus.

El principi que es troba darrera aquest mètode consisteix en trobar un número predefinit de mostres d'entrenament K que es trobin més a prop del nou punt i predir-ne l'etiqueta a partir d'elles. El número de mostres pot ser una constant definida per l'usuari o variar segons la densitat local de punts.

És interessant destacar que, quan s'utilitza *knn* per a tasques de classificació, és favorable definir la constant K com un nombre senar ja que d'aquesta manera s'eviten situacions d'empat en el recompte de les etiquetes.

La distància entre els punts, en general, pot ser qualsevol mesura mètrica: la distància euclidiana és l'opció més utilitzada.

L'algorisme de *knn* per a predir la classe d'una nova mostra es descriu a continuació [Onel Harrison 2018]:

1. Introduir la nova mostra.

2. Inicialitzar K amb el nombre de veïns a tenir en compte.

3. Per cada exemple de les dades d'entrenament:

3.1 Calcular la distància entre l'exemple i la nova mostra.

3.2 Afegir en una llista la distància calculada i l'índex de l'exemple d'entrenament.

4. Ordenar la llista de distàncies de forma ascendent.
5. Agafar els K primers elements de la llista ordenada.
6. Obtenir les etiquetes dels K elements.
7. Si es fa regressió, retornar la mitja de les etiquetes dels K elements seleccionats.
8. Si es fa classificació, retornar la moda de les etiquetes dels K elements seleccionats.

A continucació s'exposa un exemple gràfic d'un *knn*:

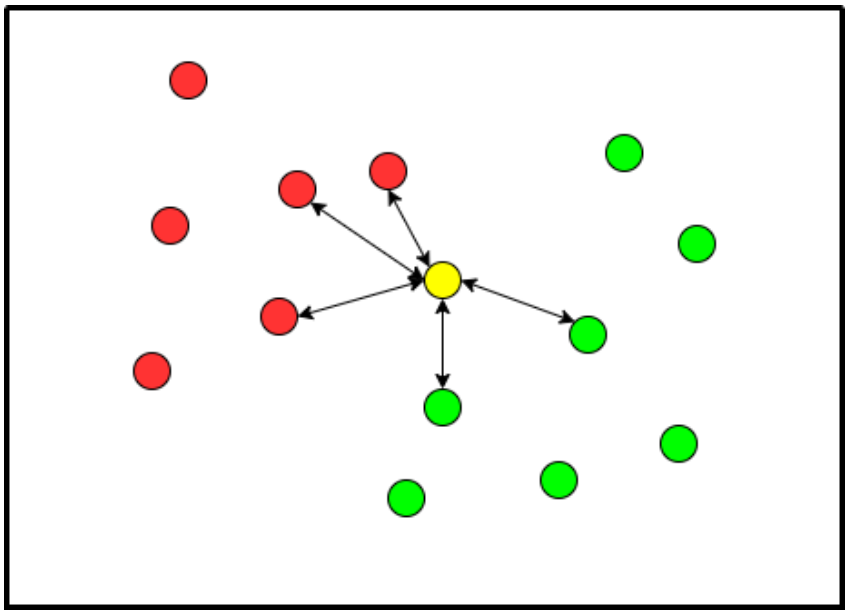


Figura 5.11: Representació gràfica d'un classificador *knn* amb K=5. El resultat de la predicció serà classificar el nou punt com a vermell, ja que és la moda de les etiquetes dels K elements més pròxims a aquest.

5.2.13.1 Sklearn KNeighbors Classifier

Aquest algorisme es troba implementat dins la llibreria d'sklearn de python i es pot utilitzar a través de la funció *KNeighborsClassifier()*. A continuació s'exposa un exemple parametritzat de la funció:

*KNeighborsClassifier(n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None, **kwargs)* [scikit-learn developers 2021f]

La part més important a l'hora d'utilitzar aquestes funcions és la bona definició dels paràmetres. Alguns dels paràmetres més importants són:

- **n_neighbors**: Nombre de veïns a tenir en compte, és la constant K.
- **algorithm**: Algoritme utilitzat per calcular els veïns més propers.
- **criterion**: Paràmetre utilitzat per calcular la distància de Minkowski (quan n'és la mètrica seleccionada). Quan p = 1, és equivalent a utilitzar la distància de Manhattan, per p = 2 és equivalent a utilitzar la distància Euclidiana. Per altres valors de p, es manté l'ús de la distància de Minkowski.
- **metric**: Mètrica que s'utilitzarà per calcular les distàncies.

5.2.14 Multicolinealitat

La multicolinealitat és una de les qüestions més importants en l'anàlisi de regressió, ja que produeix estimacions de coeficients inestables i fa que els errors creixin molt. Ens referim a ella quan hi ha una forta dependència lineal entre dos o més regressors. Normalment, sempre hi ha present en major o menor mesura un cert grau de multicolinealitat en les dades. La multicolinealitat perfecte és un cas extrem i poc comú, que es produeix quan dues o més variables independents en un model de regressió mostren una relació lineal perfecte. En aquest últim cas, els coeficients de regressió són indeterminats i els seus errors són infinits. La multicolinealitat pot donar lloc a diversos problemes:

- **Problemes d'estimacions**: tenint en compte el mètode dels mínims quadrats les estimacions dels coeficients de regressió estan donades per:

  $$\beta OLS = \left(X'X\right)^{-1} X'Y$$

  En presència d'un alt grau de multicolinealitat, la matriu X'Y és quasi singular; per tant, hi ha una desalineació del sistema d'equacions que hauria de proporcionar les estimacions dels paràmetres, que per tant són inexactes.

- **Problemes de predicció**: des d'un punt de vista pràctic, és inútil fer cap mena de predicció per a un valor de la variable dependent Y, a causa de la incertesa dels valors de β.

- **Problemes d'interpretació**: en presència de multicolinealitat, les estimacions dels coeficients són poc eficients i la seva interpretació esdevé complicada. Els regressors no expliquen la variació de la variable dependent com haurien de fer normalment.

En un model de regressió esperem que gran part de la varància pugui ser explicada (R^2). Com més gran sigui la variància explicada, millor serà el model. Tot i que, en presència de multicolinealitat, tant la variància, com els errors, i les estimacions dels coeficients poden estar inflats.

Un estimador molt utilitzat per evitar eliminar variables d'un model és el de mínims quadrats ordinaris.

5.2.14.1 Formes de detectar la multicolinealitat

No hi ha proves específiques per detectar la multicolinealitat, però algunes característiques dels models estimats poden proporcionar-ne pistes que n'indiquin la seva presència:

- Un R^2 alt amb coeficients de regressió no significatius (t-values).
- Una alta correlació entre les variables individuals.
- Un factor d'inflació de la variància (VIF) elevat. El VIF es calcula per a cada variable del model a partir de l'expressió:

$$VIF_j = \frac{1}{1-R_j^2}$$

on R_j^2 és el múltiple R^2 per a la regressió de l'element j del vector d'estimacions sobre les altres covariables. Un VIF elevat revela una dependència lineal entre la columna j i les columnes restants de la matriu X i, per tant, la presència de multicolinealitat.

5.2.14.2 Formes d'evitar la multicolinealitat

Quan es detecta la multicolinealitat en un model de regressió és important entendre quines variables explicatives estan causant el problema, i la solució més fàcil és eliminar les variables problemàtiques del model, de manera que no inclogui variables que estiguin correlacionades. Tanmateix, a vegades, a més de la dificultat que suposa triar quines variables excloure, fer-ho no sempre és una bona solució, ja que depèn de si l'objectiu de l'anàlisi és l'explicació o la predicció.

Quan l'objectiu és la predicció, no sorgeix cap problema si, entre les variables dependents, dos regressors tenen el mateix "significat": només és possible deixar de banda una d'elles. Quan, en canvi, l'objectiu és l'explicació, els regressors es seleccionen segons una regla teòrica, de manera que no és possible eliminar cap variable: el model s'ha d'explicar amb exactitud, inlcoent-hi totes les variables seleccionades. Una alternativa a l'eliminació de variables per tal de solventar el problema de la multicolinealitat és l'ànalisi de components principals que converteix les variables originals en un nou conjunt de variables linealment no correlacionades. Com que l'eliminació d'una variable comporta un problema de mala especificació del model, un remei viables és l'ús d'un estimador diferent del dels mínims quadrats. Això es deu al fet que el mètode *OLS* proporciona estimacions que podrien ser estadísticament no significatives en presència de multicolinealitat. Algunes alternatives habituals són els mínims quadrats parcials i la regressió Ridge. Aquest útim, en particular, té l'avantatge de reduir el terme de variància dels coeficients.

CAPÍTOL 6

Requisits del sistema

En aquest apartat es descriuen els requeriments que ha de complir el sistema resultant, tant funcionals com no funcionals.

6.1 Requisits funcionals

Els requeriments funcionals que s'han determinat són els següents:

- Donades les dades de múltiples tractaments de radioteràpia, obtenir la informació referent al seu comportament (lineal o no lineal) i detecar les mètriques més rellevants.
- Classificar un tractament de radioteràpia com vàlid o invàlid. Es tracta d'una classificació binària.

6.2 Requisits no funcionals

Gràcies a la utilització de les *Jupyter Notebook*, ja sigui a través de la versió web o utilitzant l'aplicació d'escriptori, i de *Python 3.0*, qualsevol ordinador serà capaç d'executar el codi.

CAPÍTOL 7

Estudi i decisions

En aquest capítol es descriuen els estudis realitzats i decisions que s'han pres al respecte durant el desenvolupament del projecte.

7.1 Python

7.2 Scikit-learn

Scikit-learn és la principal llibreria que existeix per treballar amb Machine Learning, inclou la implementació d'un gran nombre d'algoritmes d'aprenentatge. Es pot utilitzar per classificacions, extracció de característiques, regressions, agrupacions, reducció de dimensions, selecció de models, o preprocessament de dades.

És una llibreria molt simple d'utilitzar gràcies a la varietat de mètodes que es poden trobar predefinits, i també té una capacitat d'adaptació molt gran a les necessitats dels usuaris gràcies a la parametrització de tots els seus mètodes. A més a més, aquesta llibreria ofereix una interfície molt consistent per tots els models i s'integra molt bé amb altres llibreries científiques de Python.

En el desenvolupament d'aquest treball s'han utilitzat els següents paquets d'aquesta llibreria:

7.2.1 Sklearn.metrics

Aquest mòdul implementa diverses funcions per avaluar el rendiment d'un sistema de classificació. Algunes de les mètriques poden requerir estimacions de probabilitat, valors de confiança, o valors de decisions binàries. La majoria de les implementacions permeten que cada mostra proporcioni una contribució ponderada de la puntuació general.

D'aquest mòdul s'han utilitzat les funcions següents:

- **classification_report**: Genera i retorna en format de text un informe amb la precisió (fracció de casos veritablement positius entre els casos seleccionats com a positius), la sensibilitat (fracció de casos veritablement positius entre el total de casos positius) i, la puntuació F1 (es pot interpretar com la mitjana ponderada de la precisió i la sensibilitat) del model.

- **confusion_matrix**: Aquesta funció calcula la matriu de confusió per avaluar la precisió d'una classificació.

  Una matriu de confusió és aquella on, en una classificació binària, s'especifiquen el nombre de falsos negatius, falsos positius, positius certs i, negatius certs.

  - Fals negatiu: Mostra classificada com a negativa però és positiva.
  - Fals positiu: Mostra classificada com a positiva però és negativa.
  - Positiu cert: Mostra classificada com a positiva i és positiva.
  - Negatiu cert: Mostra classificada com a negativa i és negativa.

  La matriu té el següent format:

Negatius certs	Falsos positius
Falsos negatius	Positius certs

- **r2_score**: Aquesta funció retorna el valor del coeficient de determinació. Aquest coeficient determina la qualitat del model per replicar resultats, i la proporció de variació dels resultats que pot explicar-se coneixent els valors de les variables independents. L'R^2 varia des d'un mínim de 0.0 (no s'explica la variància), fins a un màxim de 1.0 (s'explica tota la variància). Existeixen casos dins de la definició computacional d'R^2 on aquest valor pot prendre valors negatius [Wikipedia contributors 2021b]

7.2.2 Sklearn.linear_model

En aquest mòdul s'implementen els algoritmes de regressió LassoCV, LassoLarsCV, RidgeCV i, ElasticNetCV (veure 5.2). S'han utilitzat per estudiar la relació lineal, quadràtica i, polinòmica de fins a grau 6 de les nostres dades respecte l'índex de validesa.

7.2.3 Sklearn.ensamble

Aquest mòdul inclou mètodes basats en conjunts per a la classificació, regressió i, detecció d'anomalies. D'aquest mòdul s'ha fet servir l'algoritme de *Random Forest Classifier* (veure 5.2.11).

7.2.4 Sklearn.neighbors

En aquest mòdul s'implementa l'algorisme de *K-Nearest-Neighbour Classifier* (veure 5.2.13).

7.2.5 Sklearn.svm

Aquest paquet implementa l'algorisme de *Support Vector Classifier* (veure 5.2.12).

7.2.6 Sklearn.preprocessing

Aquest mòdul inclou mètodes pel tractament de les dades que s'utilitzaran en els algoritmes de Machine Learning. En aquest projecte s'ha utilitzat l'Standard Scaler.

Aquest mètode s'utilitza per ajustar el rang de valors de les variables per tal que les variables amb valors molt grans no tinguin una influència més alta que la resta en aquells algoritmes on els càlculs depenen de les unitats de les variables independents. Això s'aconsegueix eliminant la mitja i escalant les dades de forma que la variància sigui igual a 1.

7.3 Pandas

Pandas és una llibreria de codi obert utilitzada per l'anàlisi i manipulació de dades. Disposa de diverses estructures de dades que permeten la manipulació d'aquestes d'una forma molt flexible i eficient.

L'estructura utilitzada en aquest projecte ha estat el *DataFrame* que comparteix moltes característiques amb les taules d'una base de dades com podria ser *Oracle*.

7.4 Matplotlib

Matplotlib és una llibreria que s'utilitza per crear visualitzacions estàtiques, animades i, interactives en Python.

7.4.1 Pyplot

Pyplot és una col·lecció de funcions que fan que *Matplotlib* funcioni com MATLAB. Les funcions contingudes dins aquest paquet s'utilitzen per realitzar algun canvi en una figura com pot ser: modificar el format de visualització, canviar etiquetes dels eixos, afegir elements a la visualització, modificar rangs, crear una visualització, etc.

7.5 Yellowbrick

És una llibreria que extén la API de Scikit-Learn per facilitar la selecció de models i l'ajust d'hiperparàmetres. També fa ús de la llibreria Matplotlib vista en l'apartat anterior per a les visualitzacions de gràfiques i altres estadístics.

D'aquesta llibreria s'han utilitzat els mètodes de Alpha Selection i Prediction Error Plot.

7.5.1 Alpha Selection

L'Alpha Visualizer és un mètode que s'utilitza per demostrar com els diferents valors d'α influeixen en la selecció del model durant la regularització de models lineals. En termes generals, α augmenta l'efecte de la regularització. Si α és zero no hi ha regularització i, quant major és α més influeix el paràmetre de regularització en el model final.

La regularització està dissenyada per penalitzar la complexitat del model, per tant, quant més alt sigui el valor d'α menys complex és el model. Les α amb valors massa grans, per altra banda, augmenten l'error degut al biaix (falta d'ajust). Per tant, és important seleccionar una α òptima de manera que l'error es minimitzi en ambdues direccions.

Per fer això l'Alpha Visualizer utilitza un dels models de regressió amb validació creuada continguts dins la llibreria Scikit-learn. En aquest model se li passa una llista d'α que es seleccionarà en funció de la puntuació de la validació creuada de cada α. El visualitzador el què fa és mostrar a través d'un gràfic la relació entre

cadascuna de les α de la llista amb l'error obtingut. A través d'aquesta representació es pot detectar si el model respon a la regularització, ja que, a mesura que augmenta o disminueix α, el model respon i l'error disminueix. Si la gràfica és irregular o aleatòria, llavors potencialment el model no és sensible a la regularització que estem aplicant i se n'hauria de provar una de diferent [scikit-yb developers].

7.5.2 Prediction Error Plot

Aquest mètode s'utilitza per generar una gràfica on es mostra la relació entre els objectes reals de les nostres dades i els valors predits pel nostre model. Això ens permet veure quanta variància hi ha en el nostre model.

CAPÍTOL 8

Anàlisi i disseny del sistema

8.1 Disseny

En aquest apartat es descriurà de forma gràfica el funcionament intern dels sistemes construïts mitjançant diagrames de blocs.

8.1.1 Diagrama de blocs pel sistema de regressors

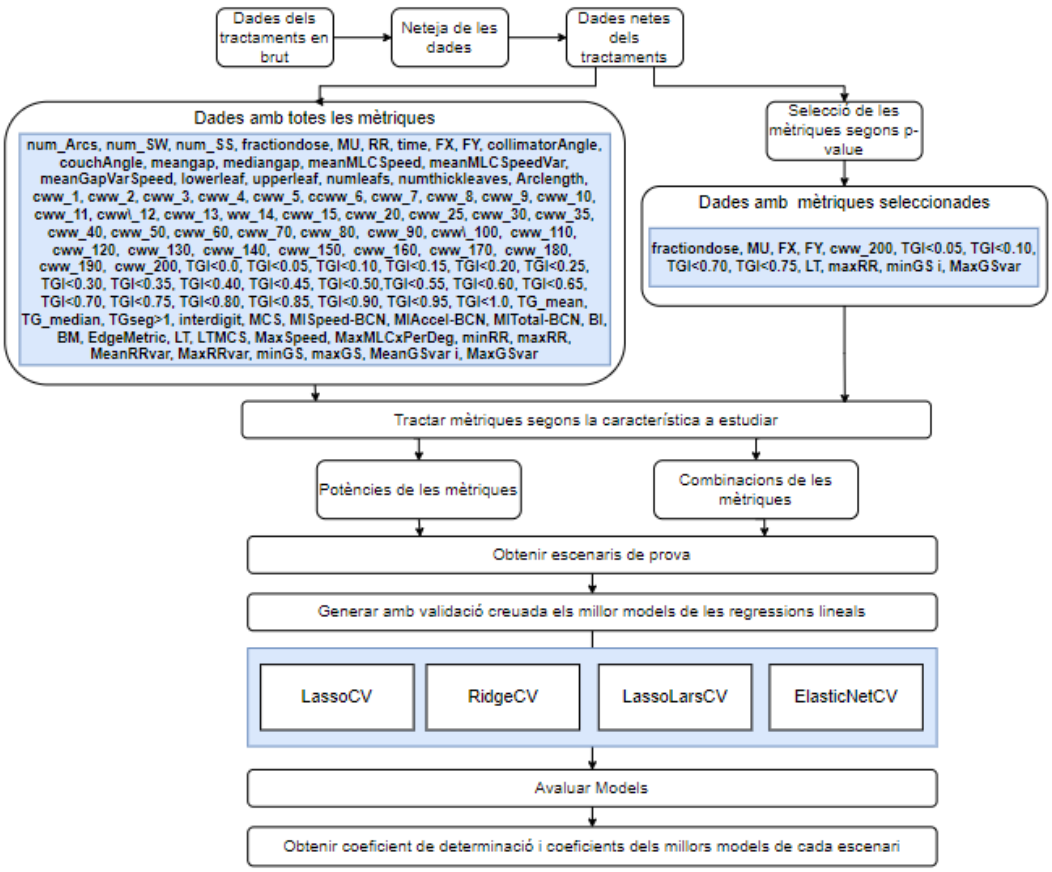


Figura 8.1: Diagrama de blocs on es descriu el funcionament del sistema dissenyat per a utilitzar els regressors.

8.1.2 Diagrama de blocs pel sistema de classificadors

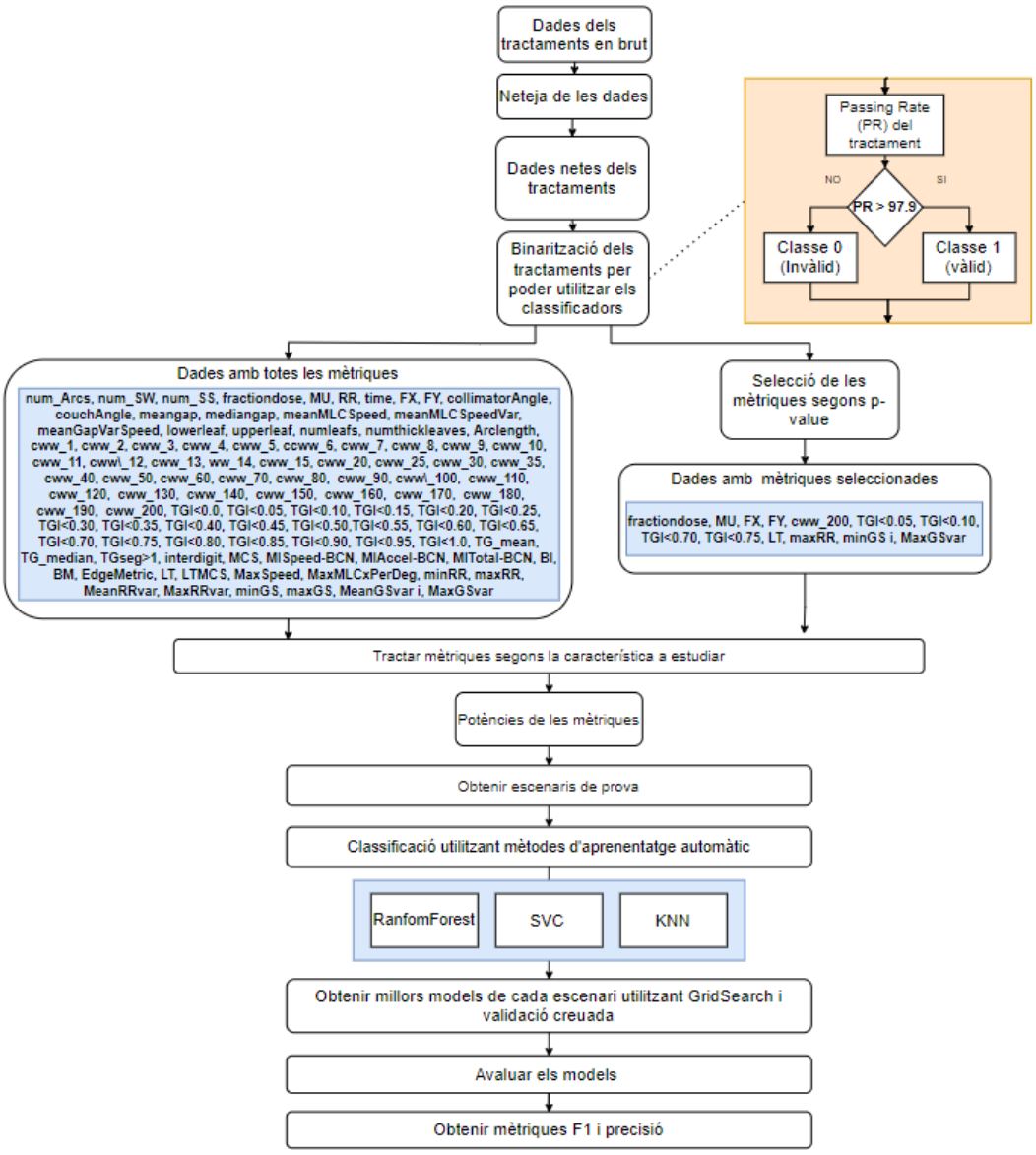


Figura 8.2: Diagrama de blocs on es descriu el funcionament del sistema dissenyat per a utilitzar els classificadors.

8.2 Dades

Aquest projecte no consta d'una base de dades que gestioni la informació dels tractaments. Les dades es generen i es guarden en fitxers *csv* que posteriorment es llegeixen utilitzant la llibreria *Pandas* de *Python*. Amb aquesta llibreria es

crea, a partir de les dades dels fitxers *csv*, una taula virtual que es podria entendre com una taula d'*SQL* d'Oracle sobre la qual es treballa.

S'han utilitzat dues versions d'aquesta taula, la primera consta de totes les mètriques dels tractaments i, la segona n'és una versió reduïda amb les mètriques seleccionades a partir d'un anàlisi estadístic.

L'estructura de les taules es mostra a continuació:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 899 entries, 0 to 898
Data columns (total 102 columns):
% J                 float64
 num_Arcs           int64
 num_SW             int64
 num_SS             int64
 fractiondose       float64
 MU                 float64
 RR                 float64
 time               float64
 FX                 float64
 FY                 float64
 collimatorAngle    float64
 couchAngle         float64
 meangap            float64
 mediangap          float64
 meanMLCSpeed       float64
 meanMLCSpeedVar    float64
 meanGapVarSpeed    float64
 lowerleaf          int64
 upperleaf          int64
 numleafs           int64
 numthickleaves     float64
 Arclength          float64
 cww_1              float64
 cww_2              float64
 cww_3              float64
 cww_4              float64
 cww_5              float64
 ccww_6             float64
 cww_7              float64
 cww_8              float64
 cww_9              float64
 cww_10             float64
 cww_11             float64
 cww_12             float64
 cww_13             float64
 ww_14              float64
 cww_15             float64
 cww_20             float64
 cww_25             float64
 cww_30             float64
 cww_35             float64
 cww_40             float64
 cww_50             float64
 cww_60             float64
 cww_70             float64
  cww_80            float64
  cww_90            float64
 cww_100            float64
  cww_110           float64
  cww_120           float64
  cww_130           float64
  cww_140           float64
  cww_150           float64
  cww_160           float64
  cww_170           float64
  cww_180           float64
  cww_190           float64
  cww_200           float64
 TGI<0.0            float64
 TGI<0.05           float64
 TGI<0.10           float64
 TGI<0.15           float64
 TGI<0.20           float64
 TGI<0.25           float64
 TGI<0.30           float64
 TGI<0.35           float64
 TGI<0.40           float64
 TGI<0.45           float64
 TGI<0.50           float64
 TGI<0.55           float64
 TGI<0.60           float64
 TGI<0.65           float64
 TGI<0.70           float64
 TGI<0.75           float64
 TGI<0.80           float64
 TGI<0.85           float64
 TGI<0.90           float64
 TGI<0.95           float64
 TGI<1.0            float64
 TG_mean            float64
 TG_median          float64
 TGseg>1            float64
 interdigit         float64
 MCS                float64
 MISpeed-BCN        float64
 MIAccel-BCN        float64
 MITotal-BCN        float64
 BI                 float64
 BM                 float64
 EdgeMetric         float64
 LT                 float64
 LTMCS              float64
 MaxSpeed           float64
 MaxMLCxPerDeg      float64
 minRR              float64
 maxRR              float64
 MeanRRvar          float64
 MaxRRvar           float64
 minGS              float64
 maxGS              float64
 MeanGSvar          float64
 MaxGSvar           float64
dtypes: float64(96), int64(6)
memory usage: 716.5 KB
```

Figura 8.3: Estructura taula completa.

```
Dataframe info:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 899 entries, 0 to 898
Data columns (total 30 columns):
% J                 899 non-null float64
 fractiondose       899 non-null float64
 MU                 899 non-null float64
 time               899 non-null float64
 FX                 899 non-null float64
 FY                 899 non-null float64
 mediangap          899 non-null float64
 meanMLCSpeed       899 non-null float64
 meanMLCSpeedVar    899 non-null float64
 meanGapVarSpeed    899 non-null float64
 lowerleaf          899 non-null int64
 upperleaf          899 non-null int64
  CWW_200           899 non-null float64
 TG_median          899 non-null float64
 interdigit         899 non-null float64
 MCS                899 non-null float64
 MISpeed-BCN        899 non-null float64
 BI                 899 non-null float64
 BM                 899 non-null float64
 EdgeMetric         899 non-null float64
 LT                 899 non-null float64
 LTMCS              899 non-null float64
 minRR              899 non-null float64
 maxRR              899 non-null float64
 MeanRRvar          899 non-null float64
 MaxRRvar           899 non-null float64
 minGS              899 non-null float64
 maxGS              899 non-null float64
 MeanGSvar          899 non-null float64
 MaxGSvar           899 non-null float64
dtypes: float64(28), int64(2)
memory usage: 210.8 KB
```

Figura 8.4: Estructura taula amb mètriques seleccionades.

8.3 Mètriques d'un tractament

Com s'ha pogut veure, els tractaments de radioteràpia es defineixen pels valors d'un conjunt de mètriques. Per tal d'etendre la informació que aporten aquestes mètriques a continuació se'n descriuen algunes de les més importants:

- **Num_arcs**: Nombre de voltes que efectua el capçal entorn el pacient.
- **Fractiondose**: Dosi diària prescrita pel metge.
- **Motor Units (MU)**: Mesura de la radiació abans d'entrar al pacient, és a dir, la radiació que surt de la màquina.
- **Repetition Rate (RR)**: Mesura que indica la quantitat de radiació per unitats de temps. Els valors oscil·len entre els 80 i els 600 UM/min.

- **Time**: Duració del tractament. Serà major o menor depenent del nombre d'arcs.
- **FX i FY**: Mesures estàtiques lligades a la mida del tumor.
- **Collimator_angle**: Angle del col·limador.
- **MeanGap**: Tamany mig dels forats que generen les làmines del col·limador.
- **MedianGap**: Mediana dels forats del col·limador.
- **MeanMLCSSpeed**: Velocitat mitjana de l'MLC, velocitat mitja de les làmines.
- **MeanMLCSpeedVar**: Variació mitjana de l'MLC, és a dir, l'acceleració mitjana de les làmines.
- **Lowerleaf**: Làmina inferior involucrada en el tractament.
- **Upperleaf**: Làmina superior involucrada en el tractament.
- **Numleafs**: Número total de làmines involucrades en el tractament.
- **Numthickleafs**: Nombre de làmines gruixudes involucrades en el tractament. Les làmines gruixudes es troben allunyades del centre del tractament, si el tumor és gros les làmines gruixudes entren en acció i implica una mala modelització de *Tongue and Groove* (TG).

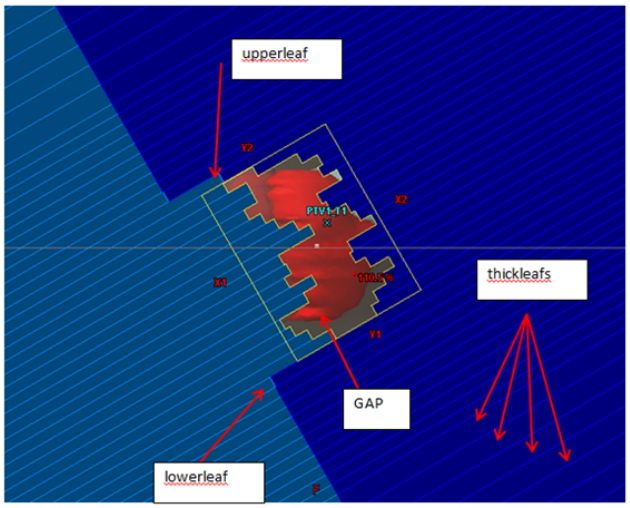


Figura 8.5: Representació gràfica d' *upperleaf, lowerleaf i, thickleaf*.

- **Arclength**: Angle total girat entorn al pacient. Si per anatomia no es gira 360º, s'han de forçar altres components del capçal.

- **Tongue and Groove (TG)**:

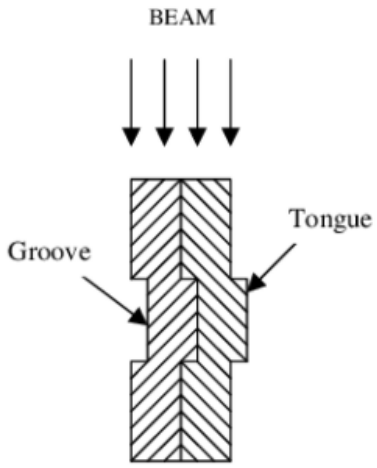


Figura 8.6: Representació de *Tongue and Groove*.

- **TGI<X**: Fracció del camp que té un TG menor que X.
- **TG_mean**: Mitja de TG en un arc.
- **TG_median**: Mediana de TG en un arc.
- **Interdigit**: Hi ha interdigitació quan les làmines d'un costat es creuen amb les altres.

Figura 8.7: *Interdigit*.

- **Modulation Complexity Score (MCS)**: Índex que permet una avaluació quantitativa de la complexitat del pla, a una escala fixa, que es pot aplicar a tots els llocs de tractament i pot proporcionar més informació relacionada amb el lliurament de la dosi que els paràmetres simples del feix.
- **Beam Inhomogenity (BI)**: Ratio entre el perímetre i l'àrea del GAP.

- **Beam Modulation (BM)**: Mètrica que indica el nivell de modulació del feix de radiació respecte les diferents àres formades per les làmines de l'MLC. Pot prendre valors compresos entre 0 (feixos poc modulats) i 1 (feixos molt modulats).
- **Edgemetric**: Mètrica que penalitza el procés d'optimització durant la planificació de la teràpia d'arc modulat (VMAT) en funció del BI per tal de disminuir les imprecisions dosimètriques degudes a la complexitat de les obertures per on ha de passar la radiació.
- **Leaf Travel (LT)**: Distància mitja que recorren les fulles.
- **LTMCS**: Combinació entre LT i MCS.
- **MinRR**: RR mínim.
- **MaxRR**: RR màxim.
- **Minimum Gantry Speed (MinGS)**: Velocitat mínima del capçal.
- **Maximum Gantry Speed (MaxGS)**: Velocitat màxima del capçal.
- **MeanGSvar**: Velocitat mitja del capçal.
- **MaxGSvar**: Velocitat màxima del capçal.

CAPÍTOL 9

Experimentació

En aquest capítol s'exposen els experiments realitzats. Com que els objectius d'aquest treball consisteixen en analitzar la relació entre el *Passing Rate* (PR) (índex de validesa o resultat del test γ) respecte les diferents mètriques que descriuen la complexitat d'un tractament, i generar un model capaç de classificar nous tractaments segons la seva viabilitat, aquest capítol constarà de diversos apartats on es detallarà el procediment seguit per assolir cadascun dels objectius.

9.1 Càrrega i estructuració de les dades

Per començar es disposa tant dels PR resultants dels anàlisis gamma (2%-2mm, 3%-2mm i, 3%-3mm) de múltiples pacients tractats a la ICO-Girona en un *Clinca iX* amb *VMAT* en diferents localitzacions i, analitzats amb el sistema de verificació *ArcCheck* de SunNuclear, com dels fitxers *DICOM-RP*, d'on s'obtenen les mètriques dels tractaments utilitzant un script per MatLab: *PlanAnalyzer_v7*.

Ja que els resultats dels anàlisis gamma i les mètriques es troben en fitxers separats, primerament s'han carregat les dades dels dos fitxers i s'han netejat eliminant els registres amb valors erronis o nuls. Posteriorment, s'han filtrat els registres dels anàlisis gamma per obtenir el subconjunt de les dades on els registres només pertanyen a anàlisis gamma 2%-2mm, els més restrictius.

Finalment utilitzant l'identificador del pacient i l'identificador del pla de tractament com a camps comuns entre els dos conjunts de dades, s'han ajuntat els PR resultants dels anàlisis amb les mètriques corresponents.

Com a resultat, el nou conjunt conté la informació de 899 tractaments i consta de 101 mètriques, a més del PR representat amb la variable J. Les mètriques del conjunt són: num_Arcs, num_SW, num_SS, fractiondose, MU, RR, time, FX, FY, collimatorAngle, couchAngle, meangap, mediangap, meanMLCSpeed, meanMLCSpeedVar, meanGapVarSpeed, lowerleaf, upperleaf, numleafs, numthickleaves, Arclength, cww_1, cww_2, cww_3, cww_4, cww_5, ccww_6, cww_7, cww_8, cww_9, cww_10, cww_11, cww_12, cww_13, ww_14, cww_15,

cww_20, cww_25, cww_30, cww_35, cww_40, cww_50, cww_60, cww_70, cww_80, cww_90, cww_100, cww_110, cww_120, cww_130, cww_140, cww_150, cww_160, cww_170, cww_180, cww_190, cww_200, TGI<0.0, TGI<0.05, TGI<0.10, TGI<0.15, TGI<0.20, TGI<0.25, TGI<0.30, TGI<0.35, TGI<0.40, TGI<0.45, TGI<0.50, TGI<0.55, TGI<0.60, TGI<0.65, TGI<0.70, TGI<0.75, TGI<0.80, TGI<0.85, TGI<0.90, TGI<0.95, TGI<1.0, TG_mean, TG_median, TGseg>1, interdigit, MCS, MISpeed-BCN, MIAccel-BCN, MITotal-BCN, BI, BM, EdgeMetric, LT, LTMCS, MaxSpeed, MaxMLCxPerDeg, minRR, maxRR, MeanRRvar, MaxRRvar, minGS, maxGS, MeanGSvar i, MaxGSvar.

	% J
count	899.000000
mean	97.119800
std	2.664024
min	79.900000
25%	96.300000
50%	97.900000
75%	98.900000
max	100.000000

Figura 9.1: Descripció del *Passing Rate*. El recompte total de mostres és de 899, amb una mitja del 97.12% i una desviació de 2.664. Cal destacar la importància del valor del percentil 50, ja que marca el punt on les dades es distribueixen de forma proporcional, és a dir, estan balancejades i, indica el punt a partir del qual categoritzar de forma binària la variable per realitzar tasques de classificació.

Com s'ha indicat en l'apartat de problemes (veure 9.5), a causa de la multicolinealitat, no s'han pogut utiltzar els coeficients dels models de regressió per seleccionar les millors mètriques. Per aquest motiu, s'ha optat per una de les solucions que consisteix en realitzar un anàlisi estadístic, en el qual s'obtenen els p-value i s'identifiquen les més rellevants, és a dir, les que tenen un p-value < 0.05 i que, per tant, estan més correlacionades amb l'índex de validesa. Amb les mètriques seleccionades s'ha generat un nou conjunt de dades format d'igual forma per 899 exemples però, en aquest cas, solament 13 mètriques.

El nou conjunt de mètriques està format per: fractiondose, MU, FX, FY, cww_200, TGI<0.05, TGI<0.10, TGI<0.70, TGI<0.75, LT, maxRR, minGS i, MaxGSvar.

S'han utilitzat els dos conjunts de dades per realitzar les proves.

9.2 Implementació dels models de regressió

En aquesta primera part d'experimentació es treballa amb models de regressió lineals múltiples per correlacionar el PR de cada tractament amb les diferents mètriques. Els models de regressió emprats són: Lasso, LassoLARS, Ridge i, ElasticNet.

9.2.1 Procediment

A part d'estudiar la correlació entre el PR i les mètriques, també s'ha realitzat l'estudi del comportament de les variables tenint en compte certes consideracions. Per tal d'obtenir aquesta informació s'han tractat les dades per ampliar-ne el seu abast. A continuació es detallen aquests canvis:

1. La primera ampliació de les dades consisteix en generar les potències de fins a grau p = 6 de les diferents mètriques. Aquesta ampliació s'ha realitzat per tal d'avaluar la dependència de les variables amb elles mateixes. D'aquesta forma es tenen 6 escenaris diferents a analitzar:

   - **p = 1**:
   $$Y = \beta_1 X_1 + ... + \beta_n X_n$$
   - **p = 2**:
   $$Y = \beta_1 X_1 + \beta_2 X_1^2 + ... + \beta_k X_n + \beta_{k+1} X_n^2$$
   - **p = 3**:
   $$Y = \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_1^3 + ... + \beta_k X_n + \beta_{k+1} X_n^2 + \beta_{k+2} X_n^3$$
   - **p = 4**:
   $$Y = \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_1^3 + \beta_4 X_1^4$$
   $$+...+$$
   $$\beta_k X_n + \beta_{k+1} X_n^2 + \beta_{k+2} X_n^3 + \beta_{k+3} X_n^4$$
   - **p = 5**:
   $$Y = \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_1^3 + \beta_4 X_1^4 + \beta_5 X_1^5$$
   $$+...+$$
   $$\beta_k X_n + \beta_{k+1} X_n^2 + \beta_{k+2} X_n^3 + \beta_{k+3} X_n^4 + \beta_{k+4} X_n^5$$

- **p = 6**:

$$Y = \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_1^3 + \beta_4 X_1^4 + \beta_5 X_1^5 + \beta_6 X_1^6$$

$$+...+$$

$$\beta_k X_n + \beta_{k+1} X_n^2 + \beta_{k+2} X_n^3 + \beta_{k+3} X_n^4 + \beta_{k+4} X_n^5 + \beta_{k+5} X_n^6$$

2. La segona ampliació es basa en generar combinacions amb repeticions de les variables per tal de determinar l'existència de dependències creuades. Aquest canvi s'aplica només per agrupacions de m = {1,2,3} variables, ja que la generació de les combinacions resulta en un nombre molt elevat de variables, cosa que implica un cost computacional molt elevat i, augmentant la complexitat del model d'aquesta forma tampoc s'espera obtenir millors correlacions. El nombre de variables resultant d'aquest procés ve determinat per la següent fórmula:

$$C_{n,p} = \frac{(n+p-1)!}{p!(n-1)!}$$

on: $C_{n,p}$ és el nombre de variables resultants de la combinació, n és el nombre de variables inicials, i p el nombre de variables a agrupar. Els casos a analitzar tindran la forma següent (exemple amb 3 variables):

- **p = 1**:

$$Y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

- **p = 2**:

$$Y = \beta_1 X_1^2 + \beta_2 X_1 X_2 + \beta_3 X_1 X_3 + \beta_4 X_2^2 + \beta_5 X_2 X_3 + \beta_6 X_3^2$$

- **p = 3**:

$$Y = \beta_1 X_1^3 + \beta_2 X_1^2 X_2 + \beta_3 X_1^2 X_3 + \beta_4 X_1 X_2^2 + \beta_5 X_1 X_3^2 +$$

$$\beta_6 X_1 X_2 X_3 + \beta_7 X_2^3 + \beta_8 X_2^2 X_3 + \beta_9 X_2 X_3^2 + \beta_1 0 X_3^3$$

Tenint en compte que s'han de generar els models de regressió amb les dades de la forma que s'acaben de descriure, s'ha implementat un bucle que s'executarà 6 vegades en el primer cas descrit, un per cada potència, i pel segon cas tindrem un bucle que s'executarà 3 vegades, un per cada combinació. Tot això es realitzarà

dues vegades, una on s'utilitzaran totes les mètriques i l'altra on s'utilitzaran només les mètriques seleccionades.

9.2.2 Generació dels models

Els models de regressió s'han generat utilitzant les funcions *LassoCV(), LassoLarsCV(), RidgeCV() i, ElasticNetCV* del paquet *sklearn.metrics*.

A les funcions se'ls passa per paràmetres el nombre de particions a utilitzar per a la validació creuada i, una llista amb 200 valors d'α (el terme de penalització de les regularitzacions) espaiats uniformament en una escala logarítmica compresos entre 1e-03 i 1e+01. Utilitzant la validació creuada i l'algorisme *AlphaSelection* de la llibreria *Yellowbrick* es selecciona la millor alpha, és a dir, la que minimitza l'error en la predicció i a partir de la qual s'escull el millor model de cada regressió.

Les dades que s'utilitzen durant l'entrenament s'escalen mitjançant el mètode *StandardScaler()* per tal d'evitar que les mètriques amb valors més grans tinguin una correlació més alta amb el PR de forma errònia.

9.3 Implementació dels classificadors

En aquesta segona part d'experimentació es treballa amb tres models de classificació binària diferents. Cadascun d'ells es basa en una metodologia concreta per intentar generar prediccions del PR de noves dades. Els models utilitzats són RandomForest (veure 5.2.11), SupportVectorMachine (5.2.12) i, K-NearestNeighbors (veure 5.2.13).

9.3.1 Procediment

El procediment adoptat en aquesta part ha estat el mateix que per l'anterior descartant la segona ampliació de les dades generant-ne les combinacions, ja que, tal i com es detallarà en el capítol de resultats, la segona ampliació implica augmentar el nivell de complexitat dels models sense obtenir millors resultats.

Llavors, s'ha implementat un bucle que s'executa 6 vegades, una per cada grau de les potències, i a dins del qual s'han generat els sis models diferents de cada classificador.

Per provar els classificadors primer s'han utilitzat totes les mètriques i posteriorment les mètriques seleccionades.

9.3.2 Generació dels models

Els models de classificació s'han generat utilitzant les funcions de *RandomForestClassifier()* del paquet *ensemble*, *SVC()* del paquet *svm* i, *KNeighborsClassifier()* del paquet *neighbors*, tots ells de la llibreria *sklearn*.

Per tal que els models de classificació resultants siguin òptims és molt important realitzar una molt bona elecció dels valors dels seus hiperparàmetres. Per aquest motiu, s'ha utilitzat un algorisme que en realitza la cerca exhaustiva generant totes les combinacions possibles per a uns valors donats. L'algorisme en qüestió està implementat en el paquet *model_selection* d'*sklearn* i s'ha utilitzat mitjançant la funció *GreedSearchCV()*. Per tal d'avaluar el model amb la combinació òptima dels d'hiperparàmetres aquesta funció utilitza la tècnica de la validació creuada que porta integrada.

Per a cada classificador s'han establert els següents valors dels hiperparàmetres que es combinaran mitjançant *greed search*:

- **RandomForestClassifier**:

  n_estimadors: [50, 100, 250, 500, 1000]
  criterion: ['gini', 'entropy']
  max_features: ["auto", "sqrt", "log2"]
  min_samples_split: [2,4,8,16,32]

- **SVC**:

  kernel: ['linear','rbf', 'poly', 'sigmoid']
  C: [1,0.1,0.001,100,10]
  gamma: ['scale', 'auto', 0.0001, 0.001, 0.01, 0.1, 0.2, 0.3]
  degree: [2, 3]

- **KNeighborsClassifier**:

  n_neighbors: [2,3,4,5,6,7,8,9,10,15,20,25,30,35,40,45,50,60,61,62,63,64,65,66,67,68,69,70,80,90,100]
  algorithm: ['auto', 'ball_tree', 'kd_tree', 'brute']
  p: [1,2]
  weights: ['uniform', 'distance']
  leaf_size: [5,10,15,20,25,30,35,40,45,50]

Les dades que s'utilitzen durant l'entrenament s'han escalat només per a la generació dels models de màquines de suport vectorial i pels k veïns més propers, ja que, tenen en compte les mètriques de forma conjunta i per tant és necessari situar-ne els valors dins del mateix rang per evitar que les mètriques amb valors més grans tinguin una correlació més alta amb el PR de forma errònia. Pel què fa als models de boscos aleatoris, no és necessari escalar les dades perquè les variables es seleccionen de forma aleatòria i s'avaluen independentment les unes de les altres.

9.4 Proves

Com ja s'ha explicat en la introducció d'aquest apartat, al ser aquest un projecte de recerca, les proves que s'han realitzat s'han enfocat a mesurar l'eficàcia dels diferents models de regressió i classificació.

Per tal de mesurar l'eficàcia dels models generats s'han aplicat diverses mètriques utilitzant la tècnica de la validació creuada.

En el cas concret dels classificadors, la tècnica de la validació creuada, s'ha utilitzat de forma conjunta amb el mètode de cerca exhaustiva o *greed search* per a l'obtenció de la millor combinació dels hiperparàmetres dels models.

A continuació s'explica en què consisteix la tècnica de la validació creuada, el mètode de *Grid Search*, i les mètriques utilitzades per avaluar els resultats.

9.4.1 Cross Validation

La *cross validation* o validació creuada és una tècnica que serveix per avaluar els resultats d'una sèrie d'anàlisis estadístics sobre unes dades i garantir que aquests resultats són independents a les característiques d'aquestes dades. Per fer-ho s'aplica la mètrica repetidament sobre diferents subconjunts de les dades d'entrenament.

Es tracta d'una tècnica molt utilitzada en sistemes com els que es desenvolupen en el present treball, on s'ha de realitzar una predicció i es vol mesurar la precisió d'aquesta.

L'estratègia utilitzada en aquest treball es coneix com *k-folds cross validation* i funciona segons l'algorisme següent:

1. Es divideixen les dades en K particions.

2. Per K vegades:

   2.1 Es reserva una de les particions per utilitzar-la com a dades de prova i la resta s'utilitzen per a entrenar el model.

   2.2 S'entrena el model amb les dades d'entrenament.

   2.3 Es fan les prediccions amb les dades de la partició de prova reservada.

   2.4 S'apliquen les mètriques corresponents i s'obtenen els resultats.

3. Es calcula la mitjana aritmètica de tots els resultats.

Els subconjunts de dades es formen mantenint el balanç de les classes del conjunt global. Gràcies a la generació d'aquests subconjunts de dades, on a cada iteració del procés s'utilitzen dades d'entrenament i test diferents, s'aconsegueixen models que no s'esbiaixen cap a unes dades concretes.

La validació creuada s'ha aplicat pels següents casos:

- Per trobar, en cadascuna de les regressions utilitzades, el millor valor d'α que minimitza l'error en la predicció del model.
- Per trobar, en cadascun dels classificadors, la millor combinació de paràmetres que maximitza la precisió del model.

9.4.2 Grid Search

Els models d'aprenentatge automàtic tenen hiperparàmetres que cal establir per adaptar el model al conjunt de dades.

Normalment, es coneixen els efectes generals dels hiperparàmetres en un model, però resulta difícil determinar la millor manera d'establir un hiperparàmetre i les combinacions d'hiperparàmetres que interactuen per a un conjunt de dades determinat. Sovint existeixen funcions heurístiques o regles empíriques per a configurar els hiperparàmetres.

Una alternativa millor consisteix en buscar de forma objectiva diferents valors per als hiperparàmetres del model i, escollir-ne un subconjunt que doni com a resultat un model que proporcioni el millor rendiment en un conjunt de dades determinat. Aquest procediment s'anomena optimització d'hiperparàmetres o ajust d'hiperparàmetres i està disponible al paquet *model_selection* de *sklearn*. El resultat d'una bona optimització dels hiperparàmetres és un únic conjunt d'hiperparàmetres que sutilitzarà per configurar un model amb un nivell de rendiment elevat [Jason Brownlee 2020].

Grid Search és un d'aquests mètodes descrits que s'utilitzen per a l'optimització d'hiperparàmetres.

El GridSearch s'ha utilitzat per determinar el millor conjunt d'hiperparàmetres a l'hora de generar els models de classificadors.

9.4.3 R^2 score, el coeficient de determinació

El coeficient de determinació és una mesura estadística que representa la proporció de la variància d'una variable dependent que s'explica per una variable independent o variables en un model de regressió.

R^2 es medeix en una escala de 0 a 1. Un valor de 1 indica un model que prediu perfectament la variable objectiu. Un valor de 0 indica que el model no té cap valor predictiu.

La fórmula matemàtica per calcular el coeficient de determinació és la següent:

$$R^2 = \frac{\sum_{i=1}^{N} \left(\hat{y}_i - \bar{Y}\right)^2}{\sum_{i=1}^{N} \left(y_i - \bar{Y}\right)^2}$$

on: N és el nombre d'observacions, $\hat{y}_i$ és l'estimació de la mostra segons el model, y_i és el resultat real de l'observació, i $\bar{Y}$ representa la mitjana real de les observacions.

Durant la generació dels models de regressió, el coeficient de determinació s'ha utilitzat com a mètrica en la validació creuada per determinar el valor d'α que genera el model amb una correlació més alta entre el PR i les variables independents.

9.4.4 Matriu de confusió

La matriu de confusió és una eina de visualització que s'utilitza en l'aprenentatge supervisat per representar les prediccions de cada classe en un sistema binari.

Les columnes de la matrius representen les prediccions del sistema, mentre que les files representen el valor real. La matriu ens queda dividida en quatre categories:

- Positius certs (TP): Instàncies classificades com a bones i són bones.

- Falsos positius (FP): Instàncies classificades com a bones però són dolentes.
- Negatius certs (TN): Instàncies classificades com a dolentes i són dolentes.
- Falsos negatius (FN): Instàncies classificades com a dolentes però són bones.

		VALOR PREDIT	
		Positiu	Negatiu
VALOR REAL	Positiu	Positiu Cert	Fals negatiu
	Negatiu	Fals positiu	Negatiu cert

Figura 9.2: Matriu de confusió.

A partir d'una matriu de confusió es poden extreure un gran nombre d'observacions sobre com es comporta un model, algunes de les més importants són:

- Sensibilitat: Indica la proporció d'instàncies classificades com a positives d'entre totes les instàncies que realment són positives.

$$TPR_{TruePositiveRate} = \frac{TP}{TP + FN}$$

- Especificitat: Indica la proporció d'instàncies classificades com a negatives d'entre totes les instàncies que realment són negatives.

$$TNR_{TrueNegativeRate} = \frac{TN}{TN + FP}$$

- Precisió: Resumeix la fracció d'exemples assignats a la classe positiva que pertanyen a la classe positiva.

$$PPV_{PositivePredictiveValue} = \frac{TP}{TP + FP}$$

- Valor Predictiu Negatiu (VPN): Resumeix la fracció d'exemples assignats a la classe negativa que pertanyen a la classe negativa.

$$VPN_{NegativePredictiveValue} = \frac{TN}{TN + FN}$$

- Exactitud: Mesura la proporció de prediccions que són correctes d'entre el total de prediccions fetes.

$$ACC_{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- F_1 Score (9.4.5):

$$F_1 = \frac{2TP}{2TP + FP + FN}$$

on: TP i TN són els positius i negatius classificats de forma correcta respectivament, mentre que FP i FN són els positius i negatius classificats incorrectament respectivament.

A partir de la informació que s'extreu de la matriu de confusió s'han avaluat els resultats obtinguts en els models de classificació.

9.4.5 F1 Score

La puntuació F, també anomenada puntuació F1, és una mesura de la precisió d'un model en un conjunt de dades. S'utilitza per avaluar sistemes de classificació binaris, que classifiquen els exemples en "positius"o "negatius".

La puntuació F és una manera de combinar la precisió i la sensibilitat del model, i es defineix com la mitjana harmònica de la precisió i la sensibilitat del model.

$$F1 = 2\frac{PS}{P + S}$$

on: P és la precisió i S és la sensibilitat.

9.4.6 Factor d'inflació de la variància (VIF)

El VIF és una estadística que mesura l'impacte de la multicolinealitat entre les variables predictores o explicatives en un model de regressió sobre la precisió de l'estimació. Expressa el grau en què la multicolinealitat entre els predictors degrada la precisió d'una estimació.

Es calcula com $(1/(1-R^2))$ per a cadascuna de les k-1 equacions de variables independents. Per exemple, donades 4 variables predictores independents, les equacions de regressió independents es formen utilitzant cada variable independent k-1 com a variable dependent:

$$X_1 = X_2 X_3 X_4$$

$$X_2 = X_1 X_3 X_4$$

$$X_3 = X_1 X_2 X_4$$

Cada model de variable independent retornarà un valor R^2 i un valor VIF. Aleshores, el terme a excloure del model es basa en el valor de VIF. Si X_j està altament correlacionat amb els predictors restants, el seu factor d'inflació serà molt gran.

Com a norma generalitzada, el valor del VIF no ha de ser mai superior a 10. En cas contrari s'indicaria la presència de multicolinealitat.

9.5 Problemes

Descripció dels problemes trobats durant el desenvolupament del projecte

9.5.1 Cost computacional

El conjunt de dades del qual es disposa en aquest projecte consta de 899 exemples de tractaments i cadascun d'ells està format per 102 variables, on una d'elles és la variable dependent i la resta són les variables predictores.

Tot i que a priori el nombre de variables predictores és assequible per a realitzar qualsevol càlcul, a mesura que s'augmenta la complexitat dels escenaris a costa de generar les potències de fins a grau sis o, generant les combinacions amb repeticions de fins a tres variables, el seu nombre augmenta en gran mesura.

En la següent taula es pot observar l'augment del nombre de variables en cada escenari:

Escenari	Variables predictores
p=1	101
p=2	202
p=3	303
p=4	404
p=5	505
p=6	606
$C_{101,1}$	101
$C_{101,2}$	5151
$C_{101,3}$	176851

Figura 9.3: Nombre de variables predictores de cada escenari.

Com s'observa a la taula anterior, el nombre de variables que han de ser tractades a l'hora de generar els models de regressió en els darrers casos, on s'utilitzen les combinacions de variables per estudiar com es relacionen entre elles, augmenta en gran mesura. Tal com es pot esperar, el temps que ha requerit generar els models per aquests darrers casos ha augmentat d'igual forma. L'últim cas, en concret, ha estat el més conflictiu, ja que després d'executar el codi durant varis dies no s'ha aconseguit obtenir el primer dels quatre models que s'havien previst. Per aquest motiu, quan s'han realitzat les proves no s'han inclòs els resultats d'aquest darrer cas.

9.5.2 Multicolinealitat

Durant l'etapa inicial d'experimentació es va observar com les diferents regressions no eren capaces de generar un model que representés de forma lineal la tendència real de les dades. Això es pot observar de forma clara a través del coeficient de determinació, ja que en cap dels models s'aconsegueix superar un 25% de la variància explicada.

Es va realitzar l'estudi de la correlació entre les diferents mètriques calculant-ne el VIF i es va detectar que diverses es troben altament correlacionades entre elles, aquest fet afecta negativament en la capacitat de modelitzar d'alguns dels algoritmes utilitzats.

S'han realitzat diferents tasques per minimitzar-ne l'impacte:

- Eliminar les mètriques amb un grau de correlació més elevat calculant-ne el VIF.
- Realitzar un anàlisi estadístic de la rellevància de les mètriques obtenint-ne el p-value per cadascuna d'elles i eliminar aquelles que segons aquest valor no aportaven informació sobre el passing rate dels tractaments.
- Crear noves mètriques resultants de la combinació de vàries d'elles.

En cap dels casos contemplats per intentar disminuir l'efecte de la multicolinealitat es va assolir una millora destacable.

De totes maneres, alguns dels algoritmes utilitzats realitzen una avaluació independent de les mètriques fent que la multicolinealitat no sigui un problema a l'hora d'interpretar-ne els resultats.

CAPÍTOL 10

Resultats

10.1 Resultats

S'han testat els escenaris mostrats en els apartats corresponents a l'anàlisi de les mètriques amb regressions i a la classificació dels tractaments segons els diferents algoritmes d'aprenentatge automàtic (veure 9.2 i 9.3).

10.1.1 Regressions

10.1.1.1 Resultats amb totes les mètriques

- **Primera ampliació**

	Valors d'R^2			
Escenari	**Lasso**	**LassoLars**	**Ridge**	**ElasticNet**
p1	0.168	0.188	0.209	0.194
p2	0.169	0.205	0.225	0.211
p3	0.166	0.218	0.158	0.213
p4	0.065	0.222	-0.138	0.212
p5	-0.259	0.193	-1.884	0.173
p6	-1.172	0.165	-8.395	0.112

Figura 10.1: Resum dels coeficients de determinació.

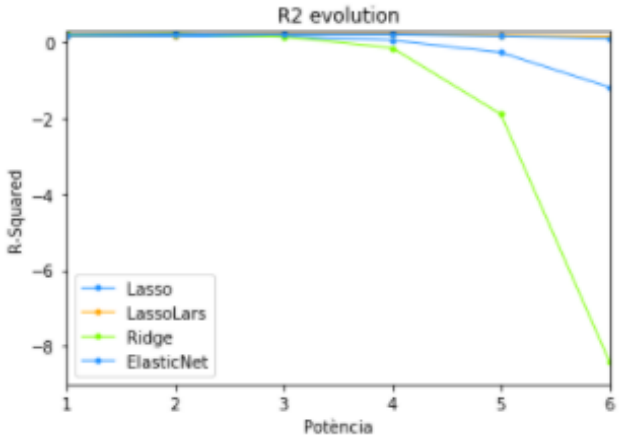


Figura 10.2: Gràfic amb l'evolució dels coeficients de determinació segons la potència i el model de regressió.

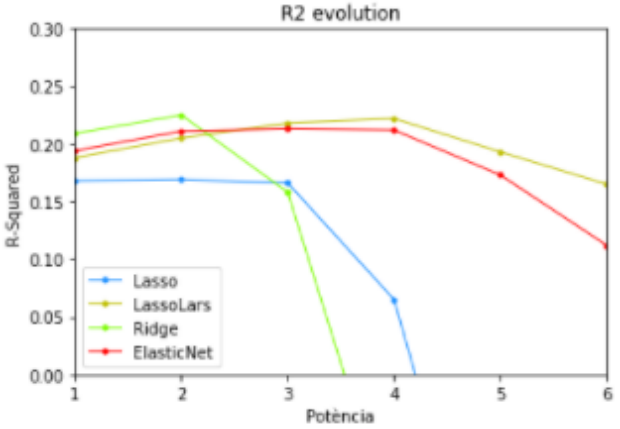


Figura 10.3: Gràfic ampliat de l'evolució dels coeficients de determinació segons la potència i el model de regressió.

Tal i com s'observa a través de les taules i els gràfics anteriors l'escenari en el qual s'obté una millor modelització de les dades és on s'utilitza una regressió Lasso Lars amb p=4, aconseguint un coeficient de determinació de 0.222.

Aquests resultats indiquen que, fent ús de models lineals, en el millor dels casos estudiats, no és possible relacionar la informació que proporcionen les diferents mètriques amb la viabilitat d'un tractament més d'un 22.2%. Això comporta que d'existir una relació entre les mètriques i la viabilitat d'un tractament no seria una relació lineal.

Els millors resultats, en qualsevol dels regressors, s'obtenen en els primers escenaris on els valors de p són més baixos. D'aquesta forma s'observa com augmentant el nivell de complexitat de les mètriques, més complicat és obtenir un model que les relacioni de forma lineal amb la validesa del tractament.

- **Segona ampliació**

	Valors d'R^2			
Escenari	Lasso	LassoLars	Ridge	ElasticNet
$C_{101,1}$	0.199	0.177	0.231	0.200
$C_{101,2}$	0.245	0.245	-0.102	0.247
$C_{101,3}$	-	-	-	-

Figura 10.4: Resum dels coeficients de determinació.

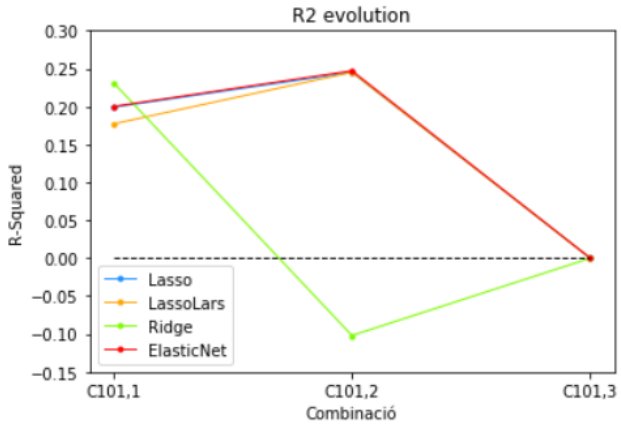


Figura 10.5: Gràfic amb l'evolució dels coeficients de determinació segons la combinació de variables i el model de regressió.

El millor resultat s'aconsegueix en l'escenari $C_{101,2}$ amb ElasticNet amb un $R^2 = 0.247$.

En general s'observa com a mesura que s'augmenta el nombre de combinacions, augmenta el valor de R^2 per tots els regressors menys per Ridge.

Això pot ser degut a què no és un model que seleccioni les variables sinó que n'augmenta o disminueix el pes, fent que a mesura que augmenta el nombre de variables sigui més complicat establir el valor dels coeficients i determinar-ne la relació amb la variable dependent.

A priori, amb els resultats obtinguts sembla que generar noves combinacions de variables aporta més informació, però també per la seva pròpia naturalesa, és a dir, pel fet d'estar relacionant vàries mètriques és molt possible que s'estigui augmentant el grau de multicolinealitat. En un model on hi és present, que augmenti l'R^2 pot indicar que la multicolinealitat també augmenta.

Per extreure conclusions s'haurien d'obtenir resultats per combinacions de més variables, però computacionalment és molt complicat (veure 9.5).

10.1.1.2 Resultats amb mètriques seleccionades

- **Primera ampliació**

	Valors d'R^2			
Escenari	Lasso	LassoLars	Ridge	ElasticNet
p1	0.051	0.143	0.143	0.142
p2	0.051	0.146	0.157	0.144
p3	0.051	0.149	0.156	0.169
p4	0.051	0.157	0.155	0.166
p5	0.051	0.168	0.140	0.168
p6	0.051	0.098	- 0.211	0.135

Figura 10.6: Resum dels coeficients de determinació.

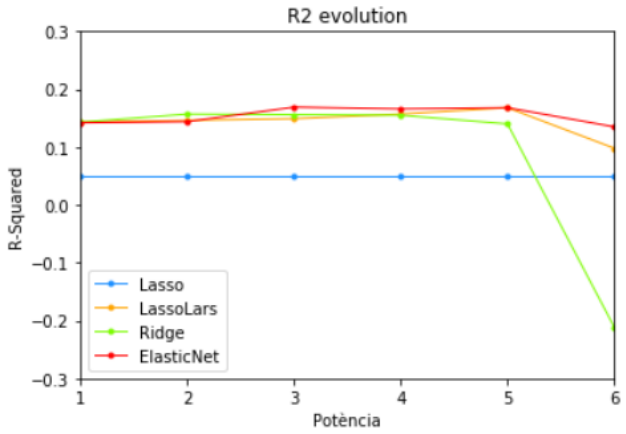


Figura 10.7: Gràfic amb l'evolució dels coeficients de determinació segons la potència i el model de regressió.

Es pot observar tant en la taula com en el gràfic anterior com l'escenari en el qual s'obté una millor modelització de les dades és on p=5 amb Lasso-LARS i ElasticNet amb R^2=0.168.

És interessant observar com a mesura que s'augmenta el nombre de variables mitjançant les potències, és a dir, relacionant les variables amb elles mateixes, quan el nombre de mètriques inicial és molt petit, la informació que aquestes aporten augmenta. En canvi, quan el nombre de mètriques resultant torna a ser alt, el rendiment baixa de cop.
En el cas de Lasso veiem com el fet de disposar de menys propietats, és a dir, al disminuir la dimensionalitat de les dades, disminueix molt la seva capacitat de generar models que s'ajustin a les dades.

- **Segona ampliació**

	Valors d' R^2			
Escenari	Lasso	LassoLars	Ridge	ElasticNet
$C_{13,1}$	0.179	0.179	0.179	0.178
$C_{13,2}$	0.172	0.159	0.198	0.168
$C_{13,3}$	0.163	0.161	-0.017	0.157

Figura 10.8: Resum dels coeficients de determinació.

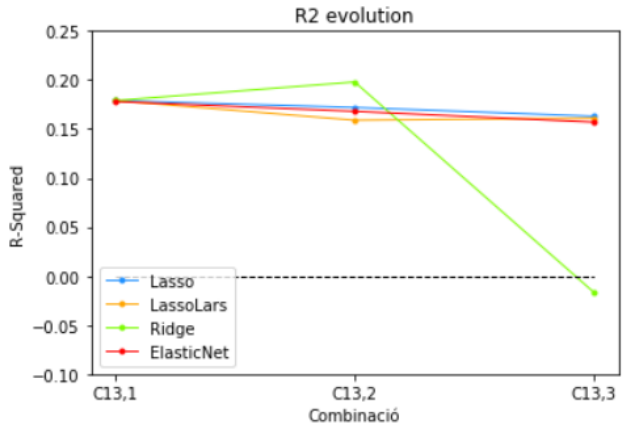


Figura 10.9: Gràfic amb l'evolució dels coeficients de determinació segons la combinació de variables i el model de regressió.

El millor resultat s'obté en l'escenari $C_{13,2}$ de Ridge amb un R^2 =0.198.

En aquest cas, es veu com els valors dels coeficients de determinació disminueixen a mesura que s'augmenta el nombre de variables agrupades en les combinacions.

Finalment, comparant els resultats d'R^2 obtinguts en les dues ampliacions s'observa clarament com en els casos on només s'han utilitzat les variables predictores seleccionades segons la seva rellevància estadística, els regressors estan generant models que són pitjors a l'hora de relacionar de forma lineal els predictors amb la variable objectiu. Amb això podem determinar que eliminar mètriques, ja sigui per intentar disminuir el grau de multicolinealitat, com per simplificar la complexitat dels models, per menys informació que aportin en principi, estem perdent la informació necessària. A més, estem reduint molt el nombre de mètriques, cosa que afecta negativament a l'hora de construir qualsevol model lineal.

10.1.2 Classificadors

En totes les proves realitzades, cada classificador s'ha configurat amb la millor combinació de paràmetres obtinguts al realitzar una *GridSearch* (veure 9.4.2) amb validació creuada utilitzant *stratified* k-folds amb k=5.

S'han utilitzat totes les mètriques del conjunt de dades. En primera instància es pretenia utilitzar els subconjunts de mètriques que aportessin informació segons els coeficients obtinguts amb les diferents tècniques de regressió, però es va observar un alt grau de correlació entre múltiples mètriques (veure 9.5.2).

Aquest fet ha comportat que els coeficients obtinguts per les regressions no siguin del tot fiables i, a més, s'ha observat que eliminar mètriques comporta la pèrdua d'informació rellevant.

Per tal de poder realitzar les tasques de classificació de forma adequada s'han binaritzat els *Passing Rate* dels tractaments. Per fer-ho se n'han obtingut els percentils i s'ha seleccionat el valor corresponent al percentil 50 per fer la divisió, ja que, d'aquesta manera s'aconsegueix una base de dades balancejada mantenint un llindar inferior per a l'índex de validesa que segueix sent acceptable. Així doncs els tractaments han quedat separats com a vàlids o invàlids a partir d'un "Passing Rate"major o menor a 97.90 respectivament.

Els classificadors s'han entrenat amb un 80% de les dades, mentre que s'han provat amb el 20% restant. Les dades que conformen aquests subconjunts s'han seleccionat aleatòriament de forma que la proporció del 50% per cada classe s'ha perdut en certa forma. Tot i que el desbalanceig provocat és molt baix pot comportar que mètriques, com per exemple l'*accuracy* o els propis *score* dels classificadors, no siguin les millors per determinar la qualitat del sistema, ja que, en un cas extrem on es té un conjunt de tractaments on el 80% d'aquests són vàlids i el 20% són invàlids, si un model classifica tots els tractaments com a vàlids acabarà tenint una precisió del 80% que a priori semblaria un resultat acceptable però, en realitat, les classificacions no s'estarien realitzant correctament. Per aquest motiu es va decidir utilitzar la F_1Score (9.4.5) per avaluar la fiabilitat global del classificador.

Tanmateix, com que l'objectiu final és seleccionar els tractaments que s'han d'utilitzar en pacients s'ha de prestar especial atenció a la capacitat dels models per classificar de forma correcta els tractaments vàlids. Determinar que un tractament és vàlid quan no ho és és un fet que no hauria de succeir en cap cas. Per aquest motiu, tot i avaluar el comportament global dels classificadors utilitzant la F_1Score, també s'ha prestat especial atenció a la seva precisió (9.4.4).

10.1.2.1 Resultats amb totes les mètriques

- **Classificador 1: Random Forest**

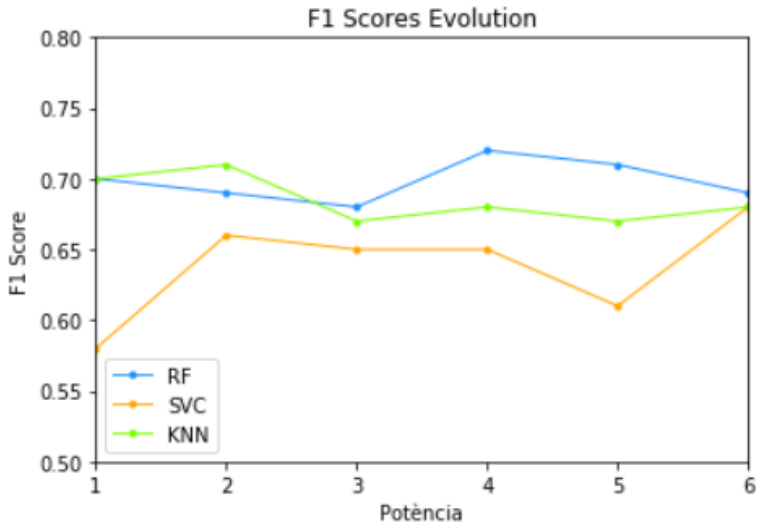
Escenari	Precisó	VPN	Sensibilitat	Especificitat	Accuracy	F1
p1	0.72	0.64	0.68	0.68	0.68	0.70
p2	0.69	0.63	0.68	0.63	0.66	0.69
p3	0.68	0.62	0.68	0.62	0.66	0.68
p4	0.73	0.67	0.71	0.68	0.70	0.72
p5	0.72	0.65	0.70	0.67	0.69	0.71
p6	0.68	0.63	0.70	0.61	0.66	0.69

- Classificador 2: Support Vector Classifier

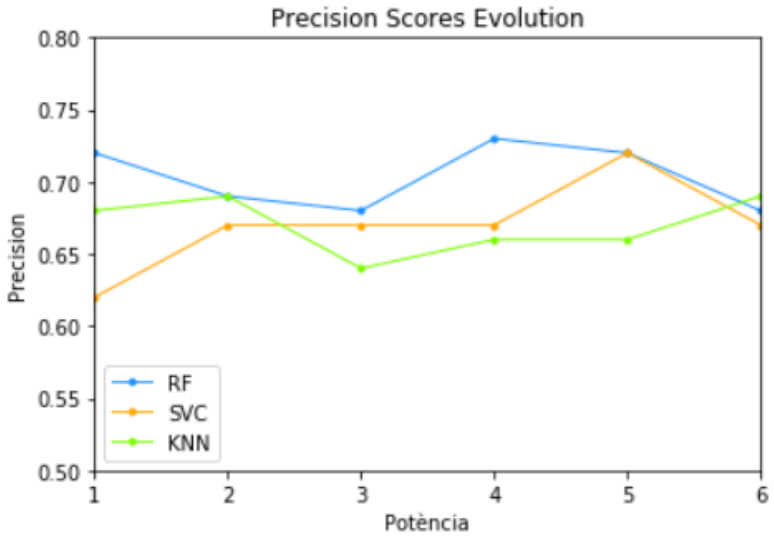
Escenari	Precisó	VPN	Sensibilitat	Especificitat	Accuracy	F1
p1	0.62	0.53	0.55	0.60	0.57	0.58
p2	0.67	0.60	0.65	0.62	0.64	0.66
p3	0.67	0.59	0.63	0.62	0.63	0.65
p4	0.67	0.59	0.63	0.63	0.63	0.65
p5	0.72	0.57	0.53	0.76	0.63	0.61
p6	0.67	0.62	0.69	0.59	0.64	0.68

- Classificador 3: K Nearest Neighbors

Escenari	Precisó	VPN	Sensibilitat	Especificitat	Accuracy	F1
p1	0.68	0.64	0.61	0.68	0.61	0.70
p2	0.69	0.66	0.73	0.61	0.68	0.71
p3	0.64	0.60	0.70	0.54	0.63	0.67
p4	0.66	0.62	0.70	0.57	0.64	0.68
p5	0.66	0.60	0.68	0.57	0.63	0.67
p6	0.69	0.62	0.67	0.65	0.66	0.68



(a) Representació gràfica de l'evolució de les $F_1 Score$ de cada classificador obtingudes en els diferents escenaris.



(b) Representació gràfica de l'evolució de les precisions de cada classificador obtingudes en els diferents escenaris

El primer que es pot observar en els resultats és que el millor classificador en qualsevol dels escenaris és el Random Forest. Aquesta superioritat pot venir atribuïda al fet d'analitzar les variables de forma indepentent, cosa que el faria salvar en certa mesura el problema de la multicolinealitat.

En específic, el millor resultat s'obté en l'escenari p=4 amb un F1 = 0.72. No

obstant això, tot i que a priori pot semblar un resultat prou prometedor, observant la precisió es pot veure que el grau d'encert del model a l'hora de classificar els tractaments positius és del 73%, això significa que un 27% dels tractaments classificats com vàlids són erronis.

De forma general, tant comparant els resultats obtinguts en els diferents escenaris d'un mateix model, com comparant els resultats dels diferents classificadors en cada escenari, s'observa que les diferències en les F_1 i les precisions són mínimes. A més, analitzant els gràfics es pot veure com, augmentar la complexitat del model no comporta una davallada de la seva efectivitat

Finalment, observant els valors de la precisió es veu com en cap cas s'aconsegueix superar el 73% d'encert a l'hora de determinar de forma correcta els tractaments vàlids.

10.1.2.2 Resultats amb mètriques seleccionades

- **Classificador 1: Random Forest**

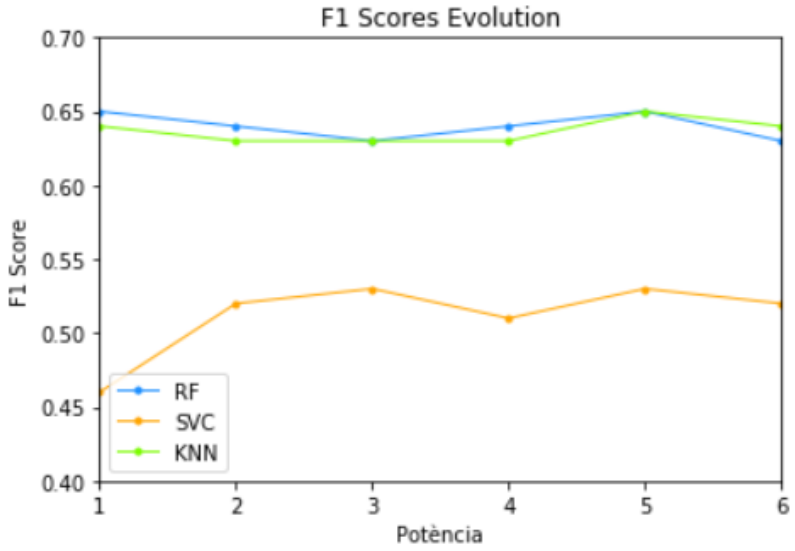
Escenari	Precisó	VPN	Sensibilitat	Especificitat	Accuracy	F1
p1	0.67	0.59	0.63	0.63	0.63	0.65
p2	0.66	0.58	0.62	0.62	0.62	0.64
p3	0.65	0.57	0.61	0.61	0.61	0.63
p4	0.65	0.57	0.63	0.59	0.61	0.64
p5	0.65	0.58	0.64	0.59	0.62	0.65
p6	0.66	0.57	0.60	0.63	0.62	0.63

- **Classificador 2: Support Vector Classifier**

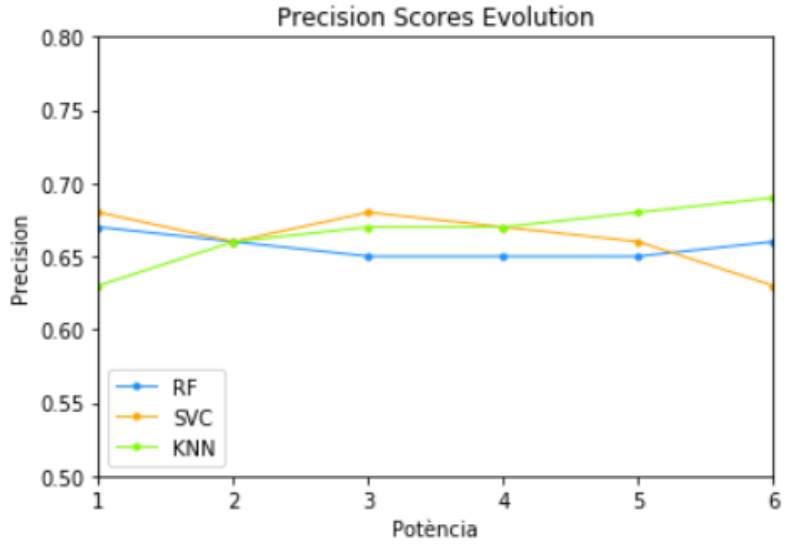
Escenari	Precisó	VPN	Sensibilitat	Especificitat	Accuracy	F1
p1	0.68	0.51	0.35	0.80	0.56	0.46
p2	0.66	0.52	0.43	0.73	0.57	0.52
p3	0.68	0.53	0.43	0.76	0.58	0.53
p4	0.67	0.52	0.41	0.76	0.57	0.51
p5	0.66	0.52	0.44	0.73	0.57	0.53
p6	0.63	0.51	0.44	0.70	0.56	0.52

- **Classificador 3: K Nearest Neighbors**

Escenari	Precisó	VPN	Sensibilitat	Especificitat	Accuracy	F1
p1	0.63	0.56	0.64	0.55	0.60	0.64
p2	0.66	0.57	0.60	0.63	0.62	0.63
p3	0.67	0.57	0.59	0.65	0.62	0.63
p4	0.67	0.57	0.59	0.65	0.62	0.63
p5	0.68	0.59	0.61	0.66	0.63	0.65
p6	0.69	0.59	0.60	0.68	0.64	0.64



(a) Representació gràfica de l'evolució de les $F_1 Score$ de cada classificador obtingudes en els diferents escenaris.



(b) Representació gràfica de l'evolució de les precisions de cada classificador obtingudes en els diferents escenaris

El millor resultat que s'ha obtingut d'F_1 ha estat 0.65. Aquest valor s'ha obtingut en els escenaris p=1 i p=5 pel RandomForest i KNN.

En aquest cas veiem com el fet d'analitzar les variables de forma independent ja no proporciona un avantatge al RandomForest. Al reduir el nombre de variables independents els arbres de decisió estaran formats per menys nodes que deci-

deixen la classe d'un exemple i, per tant, atorgaran més pes a algunes mètriques a l'hora de prendre decisions que potser no aporten suficient informació per a realitzar les distincions.

Amb els resultats que s'han obingut es pot observar com els classificadors RandomForest i KNN són capaços de determinar la classe d'un tractament de forma més acurada, mentre que els pitjors resultats sempre s'obtenen amb SVC. Aquests resultats són adequats si recuperem els resultats obtinguts en les regressions on s'ha pogut veure que d'existir relació entre les mètriques i la variable objectiu aquesta no seria lineal. Això confirma que fer ús d'altres algoritmes no lineals pot servir per obtenir millors resultats.

A més, comparant els resultats obtinguts ens els escenaris on s'utilitzen totes les mètriques amb els casos on s'utilitzen únicament les mètriques seleccionades, s'observa com en el darrer cas tant els valors de F1 com els valors de la precisió es degraden, de forma aproximada, un 10%. Amb aquests resultats es confirma el què s'ha pogut entreveure també amb els resultats de les regressions, i és que al descartar mètriques s'ha perdut informació relacionada amb el valor de la variable independent γ.

10.2 Normativa i legislació

El projecte realitzat compleix al complet la legislació vigent. Això és així donat que no es guarda ni es gestiona cap mena d'informació de caràcter personal. Per altra banda, les llibreries utilitzades són de codi obert, pel què es poden utilitzar sempre que s'indiqui el seu ús.

CAPÍTOL 11

Conclusions

El caràcter d'aquest treball està encarat de forma explícita a la recerca i té com objectiu determinar si és possible fer una predicció de la viabilitat dels tractaments de radioteràpia a través de la informació que s'obté de les diferents mètriques que defineixen aquests tractaments.

Amb aquest enfocament es van definir dues etapes del projecte, la primera consistia en l'estudi de les mètriques i, l'altra consistia en el desenvolupament d'un model predictiu que fos capaç de predir amb les mètriques seleccionades en l'estudi de l'etapa anterior quins tractaments són els que superen els índexs de viabilitat establerts.

A trets generals podem dir que els objectius marcats s'han assolit:

- S'han estudiat les mètriques i s'han extret conclusions interessants que marquen un nou full de ruta per seguir la investigació. La principal conclusió és que la correlació entre la viabilitat del tractament i les mètriques de complexitat analitzades indica que, d'existir relació entre elles, aquesta no és una relació lineal, per tant s'hauria de seguir la investigació utilitzant regressions no lineals o altres mètodes.

  A més, s'ha descobert una característica important de les dades que és la multicolinealitat i s'ha determinat que intentar-la solventar pot comportar en la majoria dels casos la pèrdua d'informació rellevant. Amb això s'ha establert que per futurs estudis, és imprescindible fer ús de totes les mètriques.

- Utilitzant els mètodes de Machine Learning inclosos dins la llibreria de Scikitlearn de Python s'ha aconseguit obtenir un model que és capaç de predir en un 72% d'efectivitat quins són els tractaments que passen els índexs de viabilitat per poder ser utilitzats en pacients sense necessitat de fer proves amb maniquí. Tot i no ser un bon resultat donada la finalitat del predictor, ja que l'existència de falsos positius n'impossibiliten el seu ús en casos reals, s'ha pogut demostrar que no és possible fer una predicció de la viabilitat dels tractaments de radioteràpia utilitzant els mètodes de boscos aleatoris, màquines de suport vectorial, o KNN.

CAPÍTOL 12

Treball futur

12.1 Utilitzar tècniques de Deep Learning

Una aposta per seguir treballant en el desenvolupament d'un model que sigui capaç d'identificar els tractaments òptims per a ser utilitzats en pacients seria ampliar les fronteres del marc en què s'engloba aquest treball i fer ús d'altres tècniques en intel·ligència artificial com poden ser les xarxes neurals.

Bibliografia

[2nd ESTRO Physics Workshop-Malaga 2018] 2nd ESTRO Physics Workshop-Malaga 2018. *ESTRO newsletter*. https://www.estro.org/ESTRO/media/ESTRO/About/Newsletters/Newsletter%202019/newsletter-jan-febr-2019_final.pdf. (Cited on page 19.)

[Ash 1994] D. Ash and T. Bates. *Report on the clinical effects of inadvertent radiation underdosage in 1045 patients*. Clinical Oncology, vol. 6, no. 4, pages 214 – 226, 1994. (Cited on page 18.)

[Bueno 2017] M. Bueno, M.A. Duch, D. Jurado-Bruggeman, S. Agramunt-Chaler and C. Muñoz-Montplet. *Experimental verification of Acuros XB in the presence of lung-equivalent heterogeneities*. Radiation Measurements, vol. 106, pages 357 – 360, 2017. Proceedings of the 18th International Conference on Solid State Dosimetry (SSD18), Munich, Germany, 3 – 8 July 2016. (Cited on page 17.)

[Hussein 2017] Mohammad Hussein, Catharine Clark and Andrew Nisbet. *Challenges in calculation of the gamma index in radiotherapy – Towards good practice*. Physica Medica, vol. 36, pages 1–11, 04 2017. (Cited on pages 19 and 21.)

[IBM 2020] IBM. *Random Forest*, December 2020. Disponible a https://www.ibm.com/cloud/learn/random-forest. (Cited on page 35.)

[Institute 2019] P.M. Institute. Practice standard for work breakdown structures - third edition. Project Management Institute, 2019. (Cited on page 10.)

[James 2013] Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani. An introduction to statistical learning, volume 112. Springer, 2013. Disponible a http://www.ime.unicamp.br/~dias/Intoduction%20to%20Statistical%20Learning.pdf. (Cited on pages 22 and 27.)

[Jason Brownlee 2020] Jason Brownlee. *Hyperparameter Optimization With Random Search and Grid Search*, 2020. Disponible a https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/. (Cited on page 68.)

[Low 1998] Daniel A. Low, William B. Harms, Sasa Mutic and James A. Purdy. *A technique for the quantitative evaluation of dose distributions*. Medical Physics, vol. 25, no. 5, pages 656–661, 1998. (Cited on page 20.)

[Miften 2018] Moyed Miften, Arthur Olch, Dimitris Mihailidis, Jean Moran, Todd Pawlicki, Andrea Molineu, Harold Li, Krishni Wijesooriya, Jie Shi, Ping Xia, Nikos Papanikolaou and Daniel A. Low. *Tolerance limits and methodologies for IMRT measurement-based verification QA: Recommendations of AAPM Task Group No. 218*. Medical Physics, vol. 45, no. 4, pages e53–e83, 2018. (Cited on pages 20 and 21.)

[Mijnheer] Ben Mijnheer. *IMRT plan validation*. https://inis.iaea.org/collection/NCLCollectionStore/_Public/40/003/40003885.pdf?r=1&r=1. (Cited on page 18.)

[Muñoz-Montplet 2018] Carles Muñoz-Montplet, Jordi Marruecos, Maria Buxó, Diego Jurado-Bruggeman, Ingrid Romera-Martínez, Marta Bueno and Joan C. Vilanova. *Dosimetric impact of Acuros XB dose-to-water and dose-to-medium reporting modes on VMAT planning for head and neck cancer*. Physica Medica, vol. 55, pages 107 – 115, 2018. (Cited on page 17.)

[Onel Harrison 2018] Onel Harrison. *Machine Learning Basics with the K-Nearest Neighbors Algorithm*, 2018. Disponible a https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761. (Cited on page 39.)

[scikit-learn developers 2021a] scikit-learn developers. *sklearn.ensemble.RandomForestClassifier*, 2021. Disponible a https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. (Cited on page 36.)

[scikit-learn developers 2021b] scikit-learn developers. *sklearn.linearmodel.ElasticNetCV*, 2021. Disponible a https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNetCV.html. (Cited on page 33.)

[scikit-learn developers 2021c] scikit-learn developers. *sklearn.linearmodel.LassoCV*, 2021. Disponible a https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoCV.html. (Cited on page 28.)

[scikit-learn developers 2021d] scikit-learn developers. *sklearn.linearmodel.LassoLarsCV*, 2021. Disponible a https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoLarsCV.html. (Cited on page 31.)

[scikit-learn developers 2021e] scikit-learn developers. *sklearn.linearmodel.RidgeCV*, 2021. Disponible a https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeCV.html. (Cited on page 26.)

[scikit-learn developers 2021f] scikit-learn developers. *sklearn.neighbors.KNeighborsClassifier*, 2021. Disponible a https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html. (Cited on page 41.)

[scikit-learn developers 2021g] scikit-learn developers. *sklearn.svm.SVC*, 2021. Disponible a https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. (Cited on page 38.)

[scikit-yb developers] scikit-yb developers. *Alpha Selection*. Disponible a https://www.scikit-yb.org/en/latest/api/regressor/alphas.html. (Cited on page 51.)

[Scrum 021] Scrum. *WHAT IS SCRUM?*, (Accedit: Juny 2021). Disponible a https://www.scrum.org/resources/what-is-scrum?gclid=CjwKCAiArOqOBhBmEiwAsgeLmbqXxlLnFB5rn3NfK9_rta-5fRKUb59pH7zJoHSNBENhDnjQ-8O1qxoCNcsQAvD_BwE. (Cited on page 5.)

[Smilowitz 2015] Jennifer B. Smilowitz, Indra J. Das, Vladimir Feygelman, Benedick A. Fraass, Stephen F. Kry, Ingrid R. Marshall, Dimitris N. Mihailidis, Zoubir Ouhib, Timothy Ritter, Michael G. Snyder, Lynne Fairobent and AAPM Medical Physics Practice Guideline Task Group. *AAPM Medical Physics Practice Guideline 5.a.: Commissioning and QA of Treatment Planning Dose Calculations - Megavoltage Photon and Electron Beams*. Journal of applied clinical medical physics, vol. 16, no. 5, pages 14–34–14–34, Sep 2015. 26699330[pmid]. (Cited on page 17.)

[SunNuclear Corporation 021] SunNuclear Corporation. *ArcCHECK® and 3DVH® The Ultimate 4D Patient QA Solution*, (Accedit: Juny 2021). Disponible a http://epsilonelektronik.com/wp-content/uploads/2015/05/ArcCHECK-3DVH.pdf. (Cited on page 19.)

[Wikipedia contributors 2021a] Wikipedia contributors. *Bias–variance tradeoff*, 2021. Disponible a https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff. (Cited on page 24.)

[Wikipedia contributors 2021b] Wikipedia contributors. *Coeficiente de determinación*, 2021. Disponible a https://es.wikipedia.org/wiki/Coeficiente_de_determinaci%C3%B3n#cite_note-2. (Cited on page 48.)