UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques
i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

ESTUDI DE L'APRENENTATGE AUTOMÀTIC PER A LA DIAGNOSI DEL CÀNCER DE MAMA

Autora: Natàlia Puig i Casanovas

Director: Dr. Sergi Baena i Miret

Dr. Josep Vives i Santa-Eulàlia

Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 12 de juny de 2022

.

Abstract

In this project we will show and discuss the classification algorithms, specifically, for the breast cancer diagnosis. From a theoretical point of view, we will study and prove the basic results of multivariate analysis, such as: dimension theorem, properties of multivariate distributions and the necessary results of Principal Components Analysis (PCA) with their respectively proofs. Then, from a more practical point of view, we will present the observed data, understanding their meaning, studying their properties and the subsequent application of a PCA. Finally, using R programming language, we will apply the data to the classification algorithms Naive Bayes and Support Vector Machine, showing the results that they provide. As well as we will see a brief explanation of the K-NN algorithm.

Resum

En aquest treball veurem i discutirem els algoritmes de classificació, concretament, per el diagnòstic del càncer de mama. Des d'una vessant més teòrica, estudiarem i demostrarem els resultats bàsics de l'anàlisi multivariant, com ara: el teorema de la dimensió, les propietats de les distribucions multivariants i els resultats necessaris de l'Anàlisi de Components Principals, juntament amb les seves respectives demostracions. Posteriorment, des d'una vessant més pràctica, presentarem primer les dades observades, entenent el seu significat, estudiant les seves propietats i la posterior aplicació d'una PCA. Finalment, utilitzant el llenguatge R, aplicarem les dades als algoritmes de classificació Naive Bayes i Support Vector Machine, analitzant els resultats que ens proporcionen. Així com veurem una breu explicació de l'algoritme K-NN.

2020 Mathematics Subject Classification. Primary: 62H30, 62P99 Secundary: 62H10, 62H25, 68T99

Agraïments

Vull agrair als meus pares i al meu germà pel suport que m'han donat al llarg d'aquests quatre anys, per celebrar les meves victòries i ajudar-me a superar les dificultats. Per ensenyar-me a lluitar pel que realment desitjava, i no decaure quan les coses no anaven bé.

A la meva àvia i la meva besàvia, per ser les meves animadores incondicionals, perquè sense vosaltres tot això no hagués sigut possible, i sobretot al meu avi, que, tot i no haver pogut veure el final, sé que ha estat lluitant al meu costat fins a l'últim minut, donant-me l'energia necessària quan no la tenia.

A en Toni, per ser la persona que ha caminat al meu costat els últims quatre anys, per deixar-me aprendre d'ell i no deixar rendir-me mai, gràcies per celebrar els reptes assolits com si fossin teus i per tranquil·litzar-me en els moments més difícils.

Als meus tutors Sergi i Josep, per confiar en mi i les meves capacitats, especialment en Sergi, al qual vull agrair tot el que m'ha arribat a ensenyar, la paciència i constància que ha tingut al llarg de tot aquest camí, estic segura que sense ell aquest treball no hagués sigut el mateix.

I finalment, agrair a tots els companys i companyes que han fet aquest llarg camí amb mi. Al vostre costat tot ha sigut més fàcil.

Índex

1 Introducció **1**

1.1 Paraules clau 1

1.2 Context i justificació del treball 1

1.3 Problemàtica a resoldre 2

1.4 Objectius 3

1.5 Resultats esperats 4

1.6 Estructura de la memòria 4

2 Preliminars d'anàlisi multivariant **5**

2.1 Matriu de dades 5

2.1.1 Matriu de dades centrades 6

2.1.2 Teorema de la dimensió 7

2.2 Distribucions multivariants 9

2.2.1 Distribució normal multivariant 9

2.3 Anàlisis de components principals 11

2.3.1 Variabilitat explicada per les components 14

2.3.2 Nombre de components principals 14

3 Introducció a les dades **15**

3.1 Introducció 15

3.2 Les nostres dades 17

3.3 Variables simulades amb distribució normal 26

4 Diagnòstic del càncer de mama mitjançant algoritmes de classificació **27**

4.1 Support Vector Machine 29

4.1.1 Casos linealment separables 30

4.1.2 Casos linealment inseparables . . . 33
4.1.3 Funcions no lineals via kernels . . . 34
4.1.4 Diagnòstic de càncer de mama mitjançant l'algoritme de classificació Support Vector Machine . . . 38
4.1.5 Aplicació de les variables simulades amb distribució normal . . . 41
4.2 Naive Bayes . . . 42
4.2.1 Diagnòstic de càncer de mama mitjançant l'algoritme de classificació Naive Bayes . . . 43
4.2.2 Aplicació de les variables simulades amb distribució normal . . . 46
4.3 K-veí més proper . . . 46

5 Discussions **48**
5.1 Comparació amb l'article . . . 48

6 Conclusions **50**

A Annexos **52**
A.1 Preliminars en l'anàlisi univariant . . . 52
A.2 *Boxplot* . . . 55
A.3 Característiques estadístiques bàsiques de les dades . . . 56
A.4 P-valors obtinguts de l'aplicació del test de normalitat de Shapiro-Wilk . . . 57
A.5 Gràfic d'un anàlisi de components principals considerant tres components . . . 59
A.6 Creació de variables amb distribució normal molt correlacionades i poc correlacionades . . . 60
A.7 Altres algoritmes . . . 63
A.7.1 Xarxes neuronals artificials . . . 63
A.7.2 Arbres de decisió . . . 64
A.7.3 K-veí més proper (K-NN) . . . 65
A.8 Codi per la millora del rendiment de l'algoritme Support Vector Machine . . . 66

A.9 Codi per la millora del rendiment de l'algoritme Naive Bayes 67

A.10 Codi per l'algoritme K-NN 68

1 Introducció

El projecte

1.1 Paraules clau

Diagnòstic del càncer de mama, mètodes de classificació, aprenentatge automàtic, anàlisi multivariant, Support Vector Machine, Naive Bayes.

1.2 Context i justificació del treball

L'elecció del tema del Treball Final de Grau (TFG) és, segurament, el més important, ja que serà el que determinarà tot el que faràs als següents mesos. És per aquest motiu que vàrem estar-hi donant voltes durant un temps. Des d'un inici estàvem segurs d'enfocar-ho cap al món de l'estadística, per una banda, és el que més m'agrada i per aquest motiu tinc una gran motivació en aprendre'n moltes coses. Per altra banda, és al que l'hi he dedicat més temps al llarg de la carrera i, per tant, considero que hi puc aportar molt més coneixement. Un cop la branca estava escollida, teníem clar que volíem fer una aportació a la societat. Una primera idea va ser fer l'estudi a una població d'un país en via de desenvolupament, idea que es va veure eclipsada per l'actual: estudiar el diagnòstic del càncer de mama.

Actualment, el càncer de mama és un dels més freqüents amb una alta taxa de mortalitat entre les dones [30]. És sabut que si es fa un diagnòstic precoç d'aquest, la supervivència pot augmentar del 56% al 86% [36]. És per això que, als darrers anys, l'estudi del càncer, en particular, el de mama, ha sigut un dels grans temes a investigar en la Bioestadística i Bioinformàtica. En destaquem dos estudis interessants [8] i [34]. Això impulsa una quantitat de dades més elevada que en altres estudis. Aquestes dades poden ser útils a l'hora de realitzar noves investigacions com podria ser la nostra. En les últimes dècades s'han desenvolupat tècniques d'imatge per a la detecció precoç i el tractament del càncer de mama, així com també s'han desenvolupat diverses tècniques d'aprenentatge automàtic [38], tot i això, estem tot just a l'inici del llarg camí contra la lluita del càncer.

Així doncs, en aquest treball, considerarem l'article [28], el qual, només ens mostra els mètodes aplicats per a l'estudi del càncer de mama i els resultats obtinguts. L'ampliarem estudiant els diferents mètodes de classificació per a la detecció del càncer de mama i seleccionant-ne el millor, explicant en tot moment quina és la teoria que hi ha darrere i què fa que les coses

funcionin, així com explicant en detall el funcionament dels diferents algoritmes. Observem, doncs, que tenim dues grans fortaleses. Per una banda, fem un estudi d'una de les malalties que causa més mortalitat entre les dones, i, per altra banda, nodrim de fonament matemàtic tot allò que a simple vista només en veiem la pràctica. Esperem, doncs, aportar el nostre granet de sorra en aquesta batalla.

1.3 Problemàtica a resoldre

Com s'ha mencionat anteriorment la lluita contra el càncer està tot just a l'inici del camí. Actualment hi ha milers de persones investigant per a poder aconseguir petits avenços. Aquest treball agafarà com a base l'article publicat per la revista d'enginyeria sanitària [28], i a partir d'aquí trobarem el model de classificació òptim a l'hora de verificar si un tumor és benigne o maligne. Per això, recopilarem les dades adients, llavors les prepararem per a poder llegir-les adequadament, i, amb aquestes, aplicarem els diferents mètodes de classificació, entrenant els models amb una part de les dades i llavors observant amb les dades de test la seva eficiència. Un cop obtinguem els resultats de cada un d'ells, valorarem, tenint en compte que volem una millor sensibilitat, quin és el millor dels mètodes per a la nostra predicció. No oblidant mai que estem dins del món de les matemàtiques, és per aquest motiu, doncs, que explicarem també tota la matemàtica que hi ha darrere dels diferents mètodes, ampliant l'article amb un gran fonament matemàtic.

Com és sabut, el que s'amaga rere els models estadístics és l'anàlisi multivariant, així doncs, a partir del llibre [6], en farem un estudi exhaustiu i n'explicarem allò que pugui aportar una major informació.

Un dels primers problemes que hem detectat al llarg de la cerca del treball ha sigut la manera que tenen els articles d'expressar-se matemàticament, ja que, dista força de com hem après i ens hem desenvolupat en el grau. Si és cert que tenen una gran base pràctica, però a vegades no en fem prou amb només mostrar els resultats. Una altra gran dificultat ha sigut la recerca de dades, realment, per a qualsevol estudi suposa un inconvenient, es troben poques bases de dades accessibles i, un cop n'has trobat alguna, aquesta ha de tenir dades suficients i les característiques t'han d'encaixar, així doncs, no és tasca fàcil.

Amb aquest treball no pretenem descobrir la cura contra el càncer de mama, que seria la gran problemàtica a resoldre, no obstant hi aportarem tot el nostre coneixement.

1.4 Objectius

Tenim dos objectius principals, un de teòric, que consisteix en formalitzar matemàticament els algoritmes de classificació. I un de pràctic, amb el que volem proposar un mètode de classificació basat en l'aprenentatge automàtic per al diagnòstic del càncer de mama. És per això que estudiarem diferents tipus de mètodes de classificació, creant el seu corresponent codi, del qual, n'avaluarem la seva precisió. També estudiarem quins seran els valors o índexs que ens puguin aportar informació sobre l'eficiència de cada un dels models en particular, amb la finalitat d'aconseguir un estudi òptim. Donat que el nostre treball estarà basat en l'article [28], al final compararem els resultats obtinguts amb els mencionats en aquest, i així poder trobar petites errades, millores o victòries. Per a l'estudi considerarem les dades de la universitat de Wisconsin del repositori UCI machine learning, posteriorment en parlarem amb més detall. L'objectiu en aquest cas, serà tractar correctament les dades perquè siguin més accessibles, tant per a nosaltres, i així poder fer un millor estudi, com per a possibles investigadors que els pugui interessar. Objectius:

- Extracció, modificació, entrenament i estudi d'una base de dades
  - Selecció de les dades que es descarregaran, tenint en compte: nombre de dades, dispersió, tipus de característiques, etc.
  - Preparació correcta de les dades per al seu futur estudi
  - Entrenament de les dades per a cada un dels diferents mètodes
  - Millorar els diferents mètodes per a obtenir un millor resultat
  - Anàlisi de resultats
- Obtenir raonadament quin és el millor mètode de classificació pel càncer de mama
  - Elecció correcte del millor mètode en funció de l'anàlisi elaborat anteriorment
  - Millorar amb detall el mètode escollit
  - Estudi de les fortaleses i febleses
  - Comparació dels resultats amb l'article
  - Creació d'un model propi tenint en compte les fortaleses i febleses dels anteriors

- Dotar de fonament matemàtic les diferents nocions estadístiques bàsiques
  - Afegir definicions i formulacions de les nocions a treballar
  - Estudiar els teoremes i resultats que fonamenten (o justifiquen) el camí triat
  - Escriure amb rigorositat i claredat els passos a seguir

1.5 Resultats esperats

Per un costat, amb la bibliografia observada fins al moment, [28], s'espera que l'algoritme dels arbres de decisió sigui el que tingui una millor precisió, no obstant això, en farem un estudi exhaustiu per confirmar o refutar aquesta idea. També s'espera poder documentar matemàticament tot el que es va detallant al llarg del treball.

1.6 Estructura de la memòria

En el primer capítol, donat que és molt elemental, el podem trobar als annexos (vegeu secció A.1). Farem una breu introducció a l'anàlisi univariant. És cert que en aquest no parlarem de contingut curricular nou, ja que tot el que mencionarem ha estat explicat en les assignatures de probabilitats o estadística del grau, no obstant, serà útil per a la posterior ampliació al següent capítol, l'anàlisi multivariant. En aquest, introduirem noves terminologies que seran importants a l'hora de realitzar el nostre estudi. Un cop tinguem la base teòrica feta, en el següent capítol, veurem els aspectes més pràctics, començarem amb la introducció de les dades. Donat que hem cregut oportú que aquestes s'entenguessin bé, hem dedicat un breu capítol a la seva explícita definició. A la penúltima secció podrem trobar una detallada explicació sobre els tres mètodes de classificació, Support Vector Machine, Naive Bayes i K-veí més proper, conjuntament amb el seu respectiu codi, corresponent a l'aplicació d'aquests mètodes. Finalment, un últim capítol on extraiem les conclusions finals respecte a les diferències i semblances entre l'article i el treball.

2 Preliminars d'anàlisi multivariant

Segons Cuadras [6] l'anàlisi multivariant és la part de l'estadística i de l'anàlisi de dades que estudia, analitza, representa i interpreta les dades que resulten d'observar més d'una variable estadística sobre una mostra d'individus. Les variables observables són homogènies i correlacionades, sense que cap predomini entre elles. La informació estadística en l'anàlisi multivariant és de caràcter multidimensional, per tant, la geometria, el càlcul matricial i les distribucions multivariants juguen un paper fonamental.

Així doncs, per aquest motiu, el primer que farem és contextualitzar dels diferents conceptes geomètrics, matricials i estadístics que utilitzarem al llarg del treball. Si no s'especifica el contrari, les dades d'aquesta secció han estat extretes del llibre [6].

2.1 Matriu de dades

A continuació definirem què és una matriu de dades així com algunes propietats d'aquesta que ens serviran, posteriorment, tant per ordenar adequadament les dades com per poder operar amb elles.

Definició 2.1. *Una matriu de dades és una eina que ens permet ordenar un conjunt d'observacions dins d'un esquema de files i columnes. Els elements que la formen poden ser de diferents orígens, no obstant, acostumen a ser números [45].*

Suposem doncs que tenim unes observacions $(X_1, ..., X_p)$ les quals volem estudiar sobre els individus $(w_1, ..., w_n)$. La matriu de dades seria una matriu $n \times p$ de la següent forma:

$$X = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix},$$

on cada un dels elements de la matriu, x_{ij} correspon a l'observació X_j sobre l'individu w_i.
Amb les definicions anteriors notem que les files s'identifiquen amb els individus i les columnes amb les observacions.
A partir de la matriu de dades en podem deduir la matriu de covariàncies i la matriu de correlació, les quals ens ajudaran amb l'estudi. Les definim doncs de la següent manera:

Definició 2.2. *Matriu de covariàncies mostrals:*

$$S = \begin{pmatrix} s_{11} & \dots & s_{1p} \\ \vdots & \ddots & \vdots \\ s_{p1} & \dots & s_{pp} \end{pmatrix},$$

on $s_{jj'} = \frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x_j})(x_{ij'} - \bar{x_{j'}})$ i $\bar{x_j} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$.
Observem que es tracta d'una matriu simètrica i quadrada.

Definició 2.3. *Matriu de correlacions mostrals:*

$$R = \begin{pmatrix} 1 & r_{12} & \dots & r_{1p} \\ r_{21} & 1 & \dots & r_{2p} \\ \vdots & \ddots & \ddots & \vdots \\ r_{p1} & r_{p2} & \dots & 1 \end{pmatrix},$$

on $r_{jj'}$ és el coeficient de correlació mostral entre les columnes j i j' de la matriu X. Aquest ve donat per l'expressió $r_{jj'} = \frac{s_{jj'}}{s_j s_{j'}}$ on s_j i $s_{j'}$ són les desviacions típiques.

Definició 2.4. *Matriu de centrat*

$$H = I - \frac{1}{n}J,$$

on I és la matriu identitat i J una matriu formada per tot uns.

2.1.1 Matriu de dades centrades

Considerant, com abans, la matriu X com la matriu de dades. Tenim que la matriu de dades centrades serà $\bar{X} = (x_{ij} - \bar{x_j})$ on $\bar{x_j}$ està descrita com a la definició 2.2.
A continuació veurem les diferents relacions que hi ha entre les matrius anunciades anteriorment, juntament amb les seves respectives demostracions. Utilitzarem x' com a notació per indicar el transposat de x.

Proposició 2.5.

$$\bar{X} = HX.$$

Demostració:

$$\bar{X} = X - 1\bar{x}' = X - 1\frac{1}{n}1'X = (I - 1\frac{1}{n}1')X = (I - \frac{1}{n}J)X = HX.$$

□

Proposició 2.6.

$$S = \frac{1}{n} X'HX.$$

Demostració:

$$S = \frac{1}{n}\bar{X}'\bar{X} = \frac{1}{n}(HX)'(HX) = \frac{1}{n}X'H'HX = \frac{1}{n}X'HX.$$

Observem que efectivament $H'H = H$

$$H = \begin{pmatrix} 1-\frac{1}{n} & -\frac{1}{n} & \dots & -\frac{1}{n} \\ -\frac{1}{n} & 1-\frac{1}{n} & \vdots & -\frac{1}{n} \\ \vdots & \ddots & \ddots & \vdots \\ -\frac{1}{n} & -\frac{1}{n} & \dots & 1-\frac{1}{n} \end{pmatrix}$$

Així doncs, la matriu resultant de fer el producte $H'H = HH$, tindrà els elements de la diagonal de la forma;

$$(1-\frac{1}{n})^2 + (n-1)\frac{1}{n^2} = 1 - \frac{1}{n}.$$

I la resta d'elements seran:

$$2(1-\frac{1}{n})(-\frac{1}{n}) + (n-2)\frac{1}{n^2} = -\frac{1}{n}.$$

Que coincideix amb H □

Proposició 2.7.

$$R = D^{-1}SD^{-1},$$

on D és la matriu diagonal amb les desviacions típiques de les variables.

Demostració:

Fent el producte de matrius i tenint en compte que $r_{jj'} = \frac{s_{jj'}}{s_j s_{j'}}$ s'obté el resultat. □

2.1.2 Teorema de la dimensió

En la majoria dels estudis, el nombre de característiques és molt elevat per poder representar-lo, per això, és molt important fer una anàlisi de components principals (vegeu secció 2.3), i així, poder eliminar aquelles característiques que tenen una menor variabilitat. No obstant això, és rellevant veure que hi ha variables prescindibles, i que, per tant, és correcta la realització d'una anàlisi de components principals. Aquí entra en joc el teorema de la dimensió, el qual ens ensenya que, efectivament, hi ha variables que són combinació lineal de la resta i, en

conseqüència, són prescindibles.

El teorema de la dimensió per a la matriu de covariàncies diu que el rang $r = rang(S)$ determina la dimensió de l'espai vectorial generat per les variables observables, és a dir, el nombre de variables linealment independents és igual al rang de S. Abans de veure el teorema, necessitem els següents resultats que ens seran útils a l'hora d'entendre'l i demostrar-lo:

Proposició 2.8. *La matriu de covariàncies S és (semi) definida positiva.*

Demostració:

$$a'Sa = \frac{1}{n}a'X'HXa = \frac{1}{n}a'X'HHXa = b'b \geq 0,$$

on $b = n^{-\frac{1}{2}}HXa$. □

Definició 2.9. *Una variable composta Y és una combinació lineal de les variables observables amb coeficients* $A = (a_1, ..., a_p)'$,

$$Y = a_1X_1 + ... + a_pX_p.$$

Proposició 2.10. *Si* $Z = b_1X_1 + ... + b_pX_p$ *és una altre variable composta, es verifica que* $cov(Y, Z) = a'Sb$.

Ara sí, veiem el teorema:

Teorema 2.11. *Si r = rang(S)* $\leq p$*, on p és el nombre total de variables, hi ha r variables linealment independents i les altres* $p - r$ *són combinació lineal d'aquestes r variables.*

Demostració:

Considerem les p variables de manera que la matriu de covariàncies S_r de $X_1, ..., X_r$, sigui no singular,

$$S_r = \begin{pmatrix} s_{11} & \dots & s_{1r} \\ \vdots & \ddots & \vdots \\ s_{r1} & \dots & s_{rr} \end{pmatrix}.$$

Sigui X_j, $j > r$. La fila $(s_{j1}, ..., s_{jr})$ serà combinació lineal de les files S_r. Llavors les covariàncies entre X_j i $X_1, ..., X_r$ verifiquen:

$$s_{jj} = \sum_{i=1}^{r} a_i s_{ji}, \;\; s_{ji} = \sum_{i'=1}^{r} a_{i'} s_{ii'}.$$

Aleshores

$$\begin{aligned} var(X_j - \sum_{i=1}^{r} a_i X_i)^2 &= s_{jj} + \sum_{i,i'=1}^{r} a_i a_{i'} s_{ii'} - 2\sum_{i=1}^{r} a_i s_{ji} \\ &= \sum_{i=1}^{r} a_i s_{ji} + \sum_{i=1}^{r} a_i (\sum_{i'=1}^{r} a_{i'} s_{ii'}) - 2\sum_{i=1}^{r} a_i s_{ji} \\ &= \sum_{i=1}^{r} a_i s_{ji} + \sum_{i=1}^{r} a_i s_{ji} - 2\sum_{i=1}^{r} a_i s_{ji} \\ &= 0. \end{aligned}$$

Per tant

$$X_j - \sum_{i=1}^{r} a_i X_i = c \Rightarrow X_j = c + \sum_{i=1}^{r} a_i X_i,$$

on c és una constant. Per tant hem demostrat que X_j és combinació lineal de les X_i per $i \in \{1, ..., r\}$. □

2.2 Distribucions multivariants

A l'hora de començar l'estudi de les dades, observem que la majoria d'elles solen provenir d'una població caracteritzada per una distribució absolutament contínua amb funció de densitat $f(x_1, ..., x_p)$. De la mateixa manera que per a distribucions univariants, la funció de densitat ha de satisfer:

- $f(x_1, ...x_p) \geq 0$ per a tot $(x_1, ..., x_p) \in \mathbb{R}^p$,
- $\int_{\mathbb{R}^p} f(x_1, ..., x_p) dx_1 \cdots dx_p = 1$.

Entre les distribucions multivariants, destaquem la normal, ja que, les variables que segueixen aquesta distribució presenten unes bones característiques. No obstant això, hi ha altres distribucions que poden ser interessants com ara, la distribució de Wishart [9], [27], la de Hotelling [5] o la de Wilks [37]. Tot i que les nostres dades, com veurem posteriorment, no segueixen una distribució normal, sí que en crearem unes.

2.2.1 Distribució normal multivariant

Com ja s'ha estudiat al llarg del grau, tenim que, donada una variable aleatòria X, segueix una distribució normal univariant, $\mathrm{N}(\mu, \sigma^2)$, si la seva funció de distribució està definida de

la següent manera:

$$\Phi_{\mu,\sigma^2}(x) = \int_{-\infty}^{x} \varphi_{\mu,\sigma^2}(u)du = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left\{-\frac{(u-\mu)^2}{2\sigma^2}\right\} du, x \in \mathbb{R},$$

on μ és la mitjana, σ la desviació típica, i φ representa la funció de densitat (veure funció de densitat A.9).

Hem vist primer el cas univariant, ja que el cas multivariant és, senzillament, una generalització d'aquest, vegem-ho.

En comptes de considerar una única variable aleatòria, considerarem un vector d'aquestes $X = (X_1, ..., X_p)$, no només això, sinó que també generalitzarem μ a un vector de mitjanes, on la component i-èssima correspondrà a la mitjana de la i-èssima variable aleatòria. Finalment, transformarem la variància en una matriu de covariàncies definida positivament, Σ, així doncs, la funció de densitat quedarà de la següent forma.

$$f(x;\mu,\Sigma) = \frac{|\Sigma|^{-\frac{1}{2}}}{(\sqrt{2\pi})^p} \exp\left\{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right\}.$$

Per altra banda, com s'ha vist anteriorment (vegeu proposició A.10), X es pot escriure de la següent forma: $X = \mu + \sigma Y$ on $Y \sim N(0,1)$. Així doncs, tenint en compte això, per passar al cas multivariant tindrem:

$$X_1 = \mu_1 + a_{11}Y_1 + ... + a_{1p}Y_p,$$

$$\vdots$$

$$X_p = \mu_p + a_{p1}Y_1 + ... + a_{pp}Y_p$$

que podem escriure com $X = \mu + AY$, on $(Y_1, ..., Y_p)$ són v.a independents amb distribució N(0,1).

Propietats:

- De la segona observació veiem que $E(X) = \mu$ i la matriu de covariàncies $E[(X-\mu)(X-\mu)'] = \Sigma$.
- La distribució de cada variable marginal X_i és normal univariant.
- Tota combinació lineal de les variables $X_1, ..., X_p$ és també normal univariant.
- Si $\Sigma = diag(\sigma_{11}, ..., \sigma_{pp})$ és matriu diagonal, aleshores les variables $X_1, ..., X_p$ són estocàsticament independents.

Un cas particular de la distribució normal multivariant és la distribució normal bivariant, la qual consta, tal com diu el nom, de dues variables. Considerem doncs, X_1, X_2. La funció de probabilitat en aquest cas contindrà cinc paràmetres: dues mitjanes μ_1 i μ_2, dues desviacions estàndards σ_1 i σ_2 i la correlació entre les dues variables, ρ. La funció de densitat de la distribució normal bivariant es defineix per:

$$f(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)\right\}.$$

Podem veure que aquesta funció de probabilitat mostra un aspecte general en forma de campana. La superfície està centrada en el punt (μ_1, μ_2), és a dir, el baricentre. Per cada punt del pla inferior X_1, X_2, tenim un punt $f(X_1, X_2)$ situat a la superfície de la muntanya en forma de campana [41]. A continuació veiem la representació d'una distribució normal bivariant (vegeu Figura 2.1).

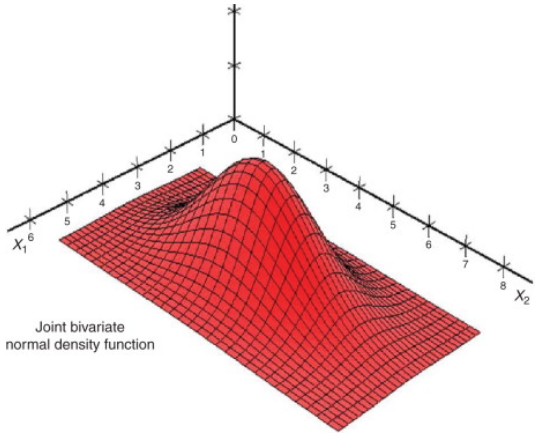

Figura 2.1: Gràfica d'una distribució normal bivariant [42].

2.3 Anàlisis de components principals

Com ja hem mencionat anteriorment en el teorema de la dimensió, un dels nostres objectius per poder entendre i representar correctament les dades serà reduir el nombre de característiques, si pot ser a, com a màxim, tres dimensions. Així doncs, per poder reduir-les, és necessari l'anàlisi de components principals, que ens donarà les eines que haurem d'utilitzar.

Les components principals són variables compostes no correlacionades tals que unes poques expliquen la major part de la variabilitat de X. D'aquesta manera, podem extreure informació a partir d'usar un nombre menor de variables sabent que aquestes contindran la major part de la variabilitat explicada per les dades. Veiem amb detall la seva definició i propietats:

Definició 2.12. *Siguin $t_1, \ldots, t_p$, p vectors normalitzats qualssevol. Les components principals són les variables compostes*

$$Y_1 = Xt_1, Y_2 = Xt_2, Y_3 = Xt_3, \ldots, Y_p = Xt_p$$

tals que:

- *$var(Y_1)$ és màxima condicionat a $t_1' t_1 = 1$.*
- *Entre totes les variables compostes Y tals que $cov(Y_1, Y) = 0$, la variable Y_2 és tal que $var(Y_2)$ és màxima condicionat a $t_2' t_2 = 1$.*
- *Si $p = 3$, la component Y_3 és una variable no correlacionada amb Y_1, Y_2 amb variància màxima.*
- *Anàlogament es defineixen les altres components principals si $p > 3$.*

Definició 2.13. *Diem que la transformació per components principals és una transformació lineal $X \to Y$ tal que $Y = XT$, on $T = [t_1, \ldots, t_p]$ és la matriu $p \times p$, tal que les seves respectives columnes defineixen les components principals.*

Teorema 2.14. *Si considerem ara $t_1, t_2, ..., t_p$ els p vectors propis normalitzats de la matriu de covariàncies S,*

$$St_i = \lambda_i t_i, \ t_i' t_i = 1, \qquad i = 1, \ldots, p.$$

Aleshores:

- *Les variables compostes $Y_i = Xt_i, i = 1, \ldots, p$, són les components principals.*
- *Les variàncies són els valors propis de S*

$$var(Y_i) = \lambda_i, \qquad i = 1, \ldots, p.$$

- *Les components principals són variables no correlacionades:*

$$cov(Y_i, Y_j) = 0, \qquad i \neq j = 1, \ldots, p.$$

Demostració:

Veiem primer que les variables $Y_i = Xt_i$, $i = 1, \ldots, p$ no estan correlacionades. Siguin $\lambda_1, \ldots, \lambda_p$ els valors propis de la matriu de covariàncies. Si existeixen $i, j \in \{1, \ldots, p\}$ tals que $\lambda_i = \lambda_j$, és a dir, tenim un valor propi amb multiplicitat dos, aleshores tindrem dos vectors

propis amb el mateix valor propi, diguem-los-hi, t_i i t_j. Respectivament si la multiplicitat es major que dos. Com hem vist a l'assignatura d'àlgebra, a partir del mètode de Gram-Schmidt [40], podem considerar la base ortonormal generada a partir d'aquests vectors propis, la qual estarà formada per vectors propis de la matriu de covariàncies i ortonormals entre sí, és a dir amb covariància 0 i variància 1. Per tant, tindrem $cov(Y_i, Y_j) = 0$. Veiem ara què passaria si són tots diferents. Siguin $\lambda_1 > \cdots > \lambda_p > 0$, podem assegurar que cap serà zero ja que prenem la matriu de covariàncies que té rang màxim. En aquest cas tenim:

$$cov(Y_i, Y_j) = t_i' S t_j = t_i' \lambda_j t_j = \lambda_j t_i' t_j$$

$$cov(Y_j, Y_i) = t_j' S t_i = t_j' \lambda_i t_i = \lambda_i t_j' t_i$$

$\Rightarrow (\lambda_j - \lambda_i) t_i' t_j = 0 \Rightarrow t_i' t_j = 0 \Rightarrow cov(Y_i, Y_j) = \lambda_j t_i' t_j = 0$, si $i \neq j$.
A més a més, per $i = j$, la variància de Y_i és:

$$var(Y_i) = \lambda_i t_i' t_i = \lambda_i.$$

Finalment veiem que les variables compostes $Y_i = X t_i$ són components principals, per això el primer que s'ha de satisfer és que la $var(Y_1)$ sigui màxima condicionat a $t_1' t_1 = 1$. Sigui doncs, $Y = \sum_{i=1}^{p} a_i X_i$ una variable composta, observem que

$$\begin{aligned}
Y &= \sum_{i=1}^{p} a_i X_i = \sum_{i=1}^{p} X_i \sum_{j=1}^{p} \alpha_j t_{ji} \\
&= X_1(\alpha_1 t_{11} + \cdots + \alpha_p t_{p1}) + \cdots + X_p(\alpha_1 t_{1p} + \cdots + \alpha_p t_{pp}) \\
&= (X_1, \ldots, X_p)(\alpha_1 t_{11} + \cdots + \alpha_p t_{p1}, \ldots, \alpha_1 t_{1p} + \cdots + \alpha_p t_{pp}) \\
&= X(\alpha_1(t_{11}, \ldots, t_{1p}) + \cdots + \alpha_p(t_{p1}, \ldots, t_{pp})) \\
&= X(\sum_{i=1}^{p} \alpha_i t_i) = \sum_{i=1}^{p} \alpha_i X t_i = \sum_{i=1}^{p} \alpha_i Y_i
\end{aligned}$$

on $a_i = \sum_{j=1}^{p} \alpha_j t_{ji}$. Així doncs $\sum_{i=1}^{p} a_i X_i = \sum_{i=1}^{p} \alpha_i Y_i$ condicionat a que $\sum_{i=1}^{p} \alpha_i^2 = 1$. Aleshores,

$$var(Y) = var(\sum_{i=1}^{p} \alpha_i Y_i) = \sum_{i=1}^{p} \alpha_i^2 var(Y_i) = \sum_{i=1}^{p} \alpha_i^2 \lambda_i \leq (\sum_{i=1}^{p} \alpha_i^2)\lambda_1 = var(Y_1),$$

que prova que Y_1 té variància màxima. Considerem ara les variables Y que no estan correlacionades amb Y_1. Les podem expressar com: $Y = \sum_{i=1}^{p} b_i X_i = \sum_{i=2}^{p} \beta_i Y_i$ condicionat a que $\sum_{i=2}^{p} \beta_i^2 = 1$. La igualtat la podem deduir de la mateixa manera que el cas anterior, a més, en aquest cas, observem que el sumatori comença a partir de $i = 2$, ja que hem

considerat les variables Y que no estan correlacionades amb Y_1 per tant $cov(Y, Y_1) = 0$ i $cov(Y, Y_1) = cov(\sum_{i=1}^{p} \beta_i Y_i, Y_1) = \sum_{i=1}^{p} \beta_i cov(Y_i, Y_1) = \beta_1 \lambda_1 = 0 \Leftrightarrow \beta_1 = 0$.
Aleshores:

$$var(Y) = var(\sum_{i=2}^{p} \beta_i Y_i) = \sum_{i=2}^{p} \beta_i^2 var(Y_i) = \sum_{i=2}^{p} \beta_i^2 \lambda_i \leq (\sum_{i=2}^{p} \beta_i^2)\lambda_2 = var(Y_2),$$

i, per tant Y_2 no està correlacionat amb Y_1 i té variància màxima. Si $p \geq 3$ la demostració que $Y_3, \ldots, Y_p$ són també components principals és anàloga. □

2.3.1 Variabilitat explicada per les components

La variància de la component principal Y_i és $var(Y_i) = \lambda_i$ i la variació total és $tr(S) = \sum_{i=1}^{p} \lambda_i$.
Per tant:

- Y_i contribueix amb la quantitat λ_i a la variació total $tr(S)$.
- Si $m < p$, $Y_1, \ldots, Y_m$ contribueixen amb la quantitat $\sum_{i=1}^{m} \lambda_i$ a la variació total $tr(S)$.
- El percentatge de variabilitat explicada per les m primeres components principals és:

$$P_m = 100 \frac{\lambda_1 + \cdots + \lambda_m}{\lambda_1 + \cdots + \lambda_p}.$$

Així doncs en les aplicacions s'espera que les primeres components expliquin un elevat percentatge de variabilitat.
A més, cal destacar que la transformació lineal T que maximitza la variabilitat geomètrica en dimensió m és la transformació per components principals $Y = XT$, és a dir, $T = [t_1, \ldots, t_m]$ conté els m primers vectors propis normalitzats de S. La demostració es pot trobar al llibre [6] al teorema 5.3.2.

2.3.2 Nombre de components principals

Hi ha diferents mètodes per trobar el nombre de components principals, com ara el criteri de Kaiser, el test d'esfericitat, entre d'altres que podem trobar al llibre [6], nosaltres utilitzarem el criteri del percentatge. Aquest consisteix en prendre un nombre m de components principals, tal que P_m sigui pròxim a un valor especificat per l'usuari, per exemple 80%.

3 Introducció a les dades

3.1 Introducció

Abans de veure els termes que utilitzarem al llarg del treball farem una breu introducció de què és el càncer. És necessari fer aquesta introducció sobre patologia per poder entendre, posteriorment, els diferents termes que anem fent servir. També per tenir un coneixement sobre la malaltia a la qual hi dedicarem tantes hores. Totes les dades d'aquesta secció, excepte aquelles que s'indica específicament, han estat extretes del llibre [35].

Definim el càncer com una acumulació d'alteracions cel·lulars en diverses etapes que afecten l'estructura (genètica) i expressió (epigenètica) dels gens que proporcionen a la cèl·lula la capacitat d'independitzar-se del control homeostàtic[3] de la resta de l'organisme, inclosa la seva proliferació indefinida[4], l'evasió de l'apoptosi[5] i la capacitat de propagar-se per l'organisme, primer localment i llavors a distància, arribant a l'extrem de causar-li la mort.

El càncer és la segona causa de mort per malaltia als Estats Units, només la supera els accidents cardiovasculars, i provoca més del 20% de totes les morts. En particular el càncer de mama és el més freqüent en dones, representa el 30% de les neoplàsies en aquestes. La seva mortalitat (14%) és més baixa que la del càncer de pulmó (25%), però més alta que la del càncer de còlon i recte (10%). La incidència del càncer de mama es manté estable (en dones és molt més elevada que en la resta de càncers) i la mortalitat ha disminuït gràcies al diagnòstic precoç (disposició de tècniques de diagnòstic més avançades) i la millora de les teràpies sistèmiques.

A la distribució poblacional del càncer influeixen múltiples factors: personals, com l'edat, el sexe, la raça o la predisposició genètica familiar; socials, fonamentalment els hàbits de vida, alimentació, addiccions; accés a la sanitat; ambientals o geogràfics, exposició a carcinògens químics, físics o biològics.

Un cop hem vist la informació bàsica del càncer i, donat que, l'objectiu del nostre treball és classificar la benignitat i malignitat del tumor, descriurem quin és el significat d'aquests termes. El concepte de benignitat i malignitat fa referència al comportament biològic general d'una neoplàsia, és a dir, la capacitat d'un tumor de causar malaltia greu i mort quan s'ha deixat a la seva lliure evolució. En general, els tumors benignes tenen un creixement lent,

[3] Tendència al manteniment de l'equilibri i de l'estabilitat interns en els diferents sistemes biològics.

[4] Augment indefinit del nombre de cèl·lules com a resultat del creixement i la multiplicació cel·lular.

[5] L'apoptosi és una forma de mort cel·lular programada (MCP) en els organismes pluricel·lulars.

expansiu i no metastatitzen. Mentre que els tumors malignes solen créixer ràpidament, destrueixen els teixits adjacents i poden metastatitzar. Aquestes característiques de cada tumor no tenen per què passar juntes, de fet, existeixen freqüents excepcions que relativitzen els conceptes de benignitat i malignitat tumoral.

El diagnòstic anatomopatològic de benignitat i malignitat d'un tumor consisteix en la predicció del comportament del tumor atenent a les seves característiques macroscòpiques i, especialment, les seves característiques microscòpiques. Aquest diagnòstic es basa en la correlació entre un patró anatomopatològic microscòpic i un comportament clínic. Vegem doncs quines són aquestes característiques:

Criteris macroscòpics:

Tot i que el diagnòstic es basa en una sèrie de criteris fonamentalment microscòpics, també en podem trobar alguns de macroscòpics, que són menys importants, però estan a l'abast de tots els metges i metgesses. Són els que utilitzem al nostre estudi. Aquests criteris són: *la mida del tumor*, en general, els tumors malignes acostumen a ser més grans que els benignes. Tot i això, la utilitat de la mida és molt limitada, ja que és poc útil en molts tumors; *característiques dels marges*, els tumors poden ser expansius, és a dir, ben delimitats i clarament definits, que és el cas dels benignes. Aquests, són tumors que no estan adherits als teixits i són fàcils d'expirar. Els tumors també poden ser infiltratius, cas dels malignes, que estan mal definits, destrueixen el teixit que els envolta i mostren marges mal definits amb parènquima adjacent. També trobem altres criteris com *el ritme de creixement*, els tumors benignes creixen habitualment de forma lenta, tot i això, hi ha alguns casos que presenten un creixement ràpid, en aquests, però, arriba un moment on s'estabilitza. Per altra banda, els tumors malignes creixen, en general, de forma més ràpida que els benignes, no obstant, aquest creixement pot ser molt variable depenent del càncer. Tenim altres criteris com: *la presència de necrosis*, *l'hemorràgia o la multiplicitat* (quantitat de tumors).

Criteris microscòpics:

Són molt més rellevants que els macroscòpics i són el fonament del diagnòstic de malignitat i benignitat. Dins el mateix repositori que hem utilitzat per a les nostres dades, podem trobar una altra base de dades tal que les seves variables corresponen a criteris microscòpics, no obstant, són variables discretes i, per tant, hem cregut oportú no utilitzar-les. Entre ells trobem *el límit del tumor*, els tumors benignes estan, en general, ben delimitats del teixit adjacent, mentre que els malignes s'infiltren i destrueixen el teixit que els rodeja. La diferenciació cel·lular, en l'anatomia patològica existeixen diferents termes que indiquen un ventall

d'alteracions de la diferenciació cel·lular, com ara: *el pleomorfisme*, en el cas maligne la mida i la morfologia entre cèl·lules presenten grans variacions mentre que, els tumors benignes, no; *l'alteració de la relació núcleo-citoplasma*, en els tumors malignes el nucli és gran en relació amb la mida del citoplasma. Entre d'altres com *l'hipercromàsia nuclear*, *els nuclèols predominants*, *les cèl·lules tumorals gegants* o *l'activitat mitòtica*.

Tot i que hi ha un seguit d'estàndards per a descriure la fisonomia del pit, aquest, és extremadament variable en mida, forma i pes. La mida ve determinada per diferents factors, com ara, el pes, l'exercici diari, la lactància materna, l'embaràs o la menstruació. Així més enllà d'una base, una variació de mida del teu pit podria ser el motiu d'un càncer.

3.2 Les nostres dades

Hem utilitzat el conjunt de dades del *Wisconsin Breast Cancer* de la UCI Machine Learning Repository, el qual ha estat creat per la universitat de Wisconsin. Les característiques estan calculades a partir d'una imatge digitalitzada d'aspirat amb agulla fina d'una massa mamària [44]. Els valors representen les característiques dels nuclis cel·lulars presentats en la imatge digital.

La base de dades anterior, està formada per 569 instàncies i 30 atributs, no obstant, per cada una de les característiques, la base de dades ens proporciona, la mitjana, l'error estàndard i el valor més elevat per totes les imatges, tot i això, nosaltres només ens quedarem amb la mitjana, i, per tant, considerarem 10 atributs. Veiem doncs aquestes característiques [44]:

- Número de codi. Senzillament, és un identificador per diferenciar les mostres, a l'estudi no ens aportarà cap informació.
- Diagnòstic. Ens indica si el tumor és benigne (B) o maligne (M).
- Radi. Mitjana de les distàncies del centre als punts del perímetre.
- Textura. Es mesura a partir de la variància de les intensitats de l'escala de grisos en els components de píxels.
- Perímetre.
- Àrea. Es compta a partir del nombre de píxels de l'interior del tumor i afegint mig píxel al perímetre.

- Uniformitat. Variació local de la longitud del radi. Es mesura a partir de la diferència entre la llargària de la línia radial i la longitud mitjana de les línies que l'envolten.

- Compacitat.

$$perimetre^2/area - 1,0.$$

  Aquest valor augmenta amb la irregularitat del límit.

- Concavitat. Severitat de les parts còncaves del contorn.

- Punts còncaus. Nombre de parts còncaves del contorn. És similar a la característica anterior, però en aquest cas mesura només el nombre i no la magnitud.

- Simetria.

- Dimensió fractal. Es calcula utilitzant l'aproximació a la frontera descrita per Mandelbrot [17].

A continuació realitzarem un estudi estadístic detallat de les diferents variables. Primer de tot, i per tenir una idea general de què ens trobarem, farem un *boxplot* i així poder veure de manera resumida la gran quantitat de dades que tenim. Per això separarem les variables en diferents grups segons els seus valors, ja que n'hi ha que prenen valors molt alts i d'altres molt petits, d'aquesta manera podrem apreciar bé els resultats (vegeu annexos A.2). Amb aquest podrem identificar valors atípics i comparar distribucions, així com estudiar la seva morfologia i simetria. Observem que en la majoria dels casos la mediana talla la caixa en dues parts iguals i, per tant, en aquest cas, ens trobem davant d'una distribució simètrica, on la mediana i la mitjana coincideixen. Aquestes propietats són les que segueixen les distribucions normals. A la resta podem considerar dos casos, el primer quan la part més llarga de la caixa és la part superior de la mediana, que, en aquest cas, les dades es concentren a la part inferior de la distribució i la mitjana és major que la mediana. El segon cas es produeix quan la part més llarga és la part inferior, aquí les dades es concentren a la part superior de la distribució i la mitjana és menor que la mediana. Finalment, veiem que podem observar valors atípics, aquests es representen amb els punts que sobresurten dels bigotis de la caixa.
Un cop tenim una idea general de les dades, anem a veure els petits detalls.
Primer de tot veurem un resum complet de les variables. Aquest ens serveix per poder detectar anomalies en els valors de les dades. A la secció 4.1.4 podem veure com hem inicialitzat la variable *dades*.

```
str(dades)
'data.frame': 569 obs. of  11 variables:
 $ Diagnostic     : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...
 $ Radi           : num  18 20.6 19.7 11.4 20.3 ...
 $ Textura        : num  10.4 17.8 21.2 20.4 14.3 ...
 $ Perimetre      : num  122.8 132.9 130 77.6 135.1 ...
 $ Area           : num  1001 1326 1203 386 1297 ...
 $ Uniformitat    : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
 $ Compacitat     : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
 $ Concavitat     : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
 $ PuntsConcaus   : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
 $ Simetria       : num  0.242 0.181 0.207 0.26 0.181 ...
 $ DimensioFractal: num  0.0787 0.0567 0.06 0.0974 0.0588 ...
```

Observem que les dades no estan equilibrades, ja que gairebé tenim el doble de casos benignes que malignes.

```
> length(dades$Diagnostic[dades$Diagnostic == "B"])
[1] 357
> length(dades$Diagnostic[dades$Diagnostic == "M"])
[1] 212
```

Als annexos A.3 podem trobar les característiques bàsiques de les dades. Una primera aproximació d'aquestes, és que podem trobar variables centrades, ja que, per exemple, la mediana està més propera al primer quadrant que al tercer.

A continuació estudiarem la seva normalitat. Ho veurem per a cada una de les classes, pel fet que si existeix normalitat, aquesta hauria de ser per classes. Podem veure els resultats obtinguts a l'hora d'aplicar el test de normalitat de Shapiro-Wilk als annexos (vegeu secció A.4).

Primer dividirem les dades segons la classe on es troben:

```
dades.BENIGNE <- dades[dades$Diagnostic == "B", -1]
dades.MALIGNE <- dades[dades$Diagnostic == "M", -1]
```

A continuació apliquem el test de normalitat **Shapiro-Wilk** per a cada classe, aquest considera com a hipòtesi nul·la que les dades son normals. Considerarem un nivell de significació α[6] del 0.05.

[6]Probabilitat de refutar la hipòtesi nul·la quan és verdadera.

- Classe 1, Benigne

```
shapiro.wilk.test.1 <- apply(dades.BENIGNE [,], 2, shapiro.test)
```

El p-valor[7] és molt significatiu, és a dir és menor que 0.05, per a totes les variables exceptuant el radi i el perímetre, així doncs, podem concloure que per aquesta classe com a mínim una de les variables no segueix una distribució normal univariant.

- Classe 2, Maligne

```
shapiro.wilk.test.2 <- apply(dades.MALIGNE [,], 2, shapiro.test)
```

En aquest cas, el p-valor per a totes les variables continua sent molt significatiu i, per tant, cap variable prové d'una distribució normal univariant.

Concloem que les variables de cada classe no segueixen una distribució normal multivariant, ja que, per contrari, a cada regió, totes les marginals haurien de seguir una distribució normal. Això és cert, ja que, hem vist anteriorment que si una variable composta segueix una distribució normal aleshores totes les seves marginals també.

Finalment, per acabar amb l'estudi de les dades realitzarem una PCA a aquestes. Tot i que hi ha diferents funcions a R com *prcomp*() per dur a terme una PCA, donat que a la secció 2.3 s'explica com reproduir-la, prendrem aquests passos per implementar-la.

El primer que farem és detallar el codi que hem seguit i llavors farem una explicació dels resultats obtinguts.

Passos que hem seguit per realitzar la PCA:

Traiem la variable "Diagnòstic"de les dades, ja que haurem de fer diferents operacions que necessiten dades numèriques, aquesta es troba a la posició 1.

```
dades.PCA<- dades[,-1]
```

En l'anàlisi de components principals, les variables sovint s'escalen. Això es recomana especialment quan les variables es mesuren en diferents escales, en cas contrari, els resultats aconseguits de la PCA es veuran greument afectats. L'objectiu és fer comparables les variables. Així doncs, les escalarem amb la funció *scale*().

```
dades.PCA <- scale(dades.PCA)
```

[7] El p-valor és la probabilitat d'obtenir un efecte com a mínim tan extrem com el de les dades de mostra, assumint que la hipòtesi nul·la és verdadera [10].

A continuació busquem la matriu de covariàncies de la matriu de dades, amb la funció *cov*().

```
sigma <-cov(as.matrix(dades.PCA))
```

I, finalment, busquem els vectors i valors propis d'aquesta matriu. Els vectors propis ens indicaran la direcció de les components principals i els valors propis la variabilitat de cada una d'elles.

```
eigen_sigma <- eigen(sigma)
```

Donat que el primer vector propi té tots els valors negatius, per treballar amb dades més corrents i comprensibles, canviarem el signe de tots els vectors propis. Com que aquests indiquen la direcció de les components, no afectarà si els hi canviem a tots el signe. Per definició, el primer que ha de satisfer una component principal és que, donat $Y_1 = Xt_1, \ldots, Y_p = Xt_p$, la $var(Y_1)$ sigui màxima condicionada a $t_1't_1 = 1$. Ens adonem però que els vectors ja estan normalitzats, ja que si realitzem el producte escalar mencionat anteriorment ens dóna 1 en tots els casos. Calculem doncs les components principals utilitzant que estan descrites de la següent forma $Y_1 = Xt_1, \ldots, Y_p = Xt_p$ on X és la matriu de dades.

```
#Realitzant el codi seguent obtindrem una matriu on cada columna correspondra a una
    component principal
components_principals <- matrix()
components_principals <- (as.matrix(dades.PCA) %*% as.vector(eigen_sigma$vectors[,1]))
for(vector in 2:dim(eigen_sigma$vectors)[1]){
 components_principals <- cbind(components_principals, (as.matrix(dades.PCA)) %*% as.vector(
    eigen_sigma$vectors[,vector]))
}
```

A continuació calcularem el percentatge de variabilitat explicada per cada una de les components principals. Recordem que aquest es calcula a partir dels valors propis de la matriu de covariàncies.

```
(cumsum(eigen_sigma$values)/sum(eigen_sigma$values))*100

54.78588  79.97302  88.77917  93.76926  97.49465  98.73607  99.53692  99.88582  99.99718
    100.00000
```

Donat que volem un percentatge proper al 80% (vegeu secció 2.3.2) ens quedarem amb dues components principals, tot i això, als annexos (vegeu secció A.5) podem trobar la representació

conseqüent de l'elecció de tres components. Per entendre-ho millor ho representarem, que de fet, era el nostre objectiu. Per fer-ho utilitzarem la funció *qplot*() del paquet *ggplot*2.

```
library(ggplot2)
library(e1071)
qplot(components_principals[,1],components_principals[,2],xlab = "Primera component principal
    ", ylab = "Segona component principal", colour=dades$Diagnostic, shape=dades$Diagnostic)
```

I obtenim la següent gràfica (vegeu Figura 3.2):

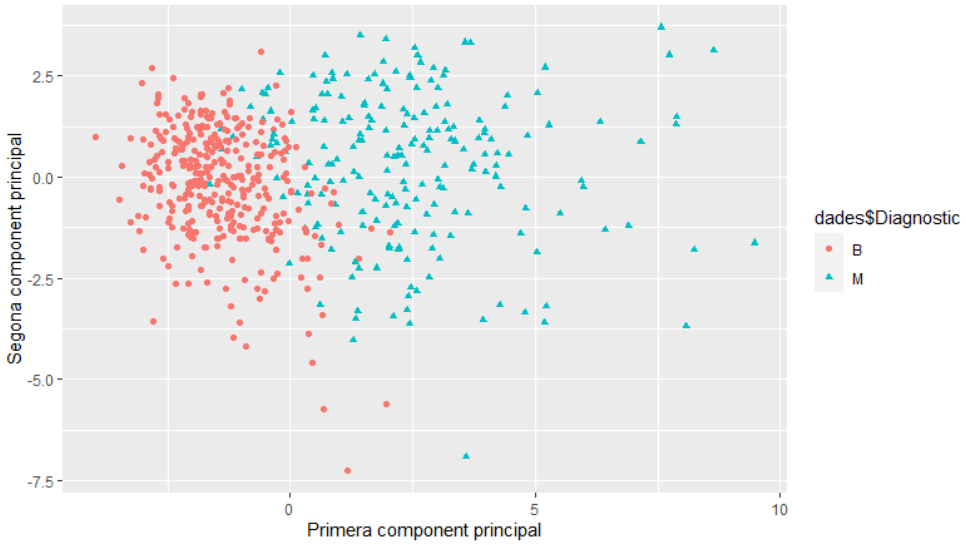

Figura 3.2: Gràfica de les dades un cop aplicada la PCA. Gràfica realitzada amb el programa RStudio.

Per poder entendre-la primer realitzarem una breu explicació intuïtiva de les components principals.

Suposem que tenim un cos amb unes certes característiques, n'hi poden haver tantes com siguin necessàries, però moltes d'elles mesuraran propietats relacionades i, per tant, seran redundants. Així doncs, podríem resumir el cos amb menys característiques. D'això és el que s'encarrega la PCA. Amb aquesta idea, no hem de pensar que es queda amb unes característiques i elimina completament les altres, sinó que en construeix unes de noves que resulten de resumir bé tota la llista de característiques. Evidentment, aquestes noves característiques es construeixen utilitzant les antigues. Bàsicament per trobar una nova característica, buscarà aquella on els punts projectats tinguin una màxima variació i un menor error de reconstrucció, és a dir una menor distància entre el punt original i el projectat. Ho podem entendre millor en la següent imatge (vegeu Figura 3.3), on la recta diagonal seria la nova component principal,

l'eix x i y dues característiques que vulguem comparar i que estiguin correlacionades, els punts blaus, les dades amb la base inicial i els vermells serien els punts un cop calculades les components principals. Observem que, la component de la imatge, té una variabilitat màxima, ja que, si anéssim canviant la direcció de la nova component, la distància entre el primer punt vermell i l'últim mai seria major que la de la imatge. Finalment, les línies vermelles, ens indiquen l'error de reconstrucció, de la mateixa manera que la variabilitat podem veure que aquest error és mínim a la imatge. De fet, el pla que asseguri una màxima dispersió serà també el pla que estigui més proper als punts originals, tal com expressa la següent equivalència:

$$\max \sum_i d_H^2(i, G) \Leftrightarrow \min \sum_i d_{\bar{H}}^2(i, G),$$

on H representa el subespai de projecció, $\bar{H}$ el subespai ortogonal d'H i G és el centre de gravetat del conjunt de punts.

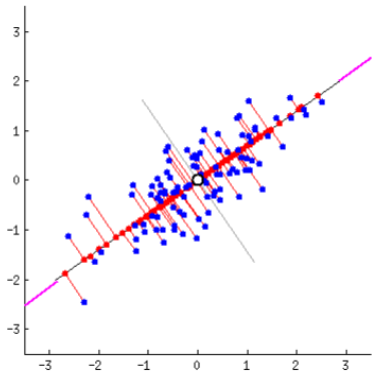

Figura 3.3: Gràfica de les dades un cop aplicada la PCA [22].

Anem doncs al nostre cas, tenim els següents vectors propis:

```
         [,1]        [,2]        [,3]         [,4]        [,5]        [,6]         [,7]
[1,] 0.36393793  0.313929073  0.12442759 -0.029558858  0.031067022  0.264180150 -0.04418839
[2,] 0.15445113  0.147180909 -0.95105659 -0.008916084  0.219922761  0.032206572  0.02055748
[3,] 0.37604434  0.284657885  0.11408360 -0.013458069  0.005945081  0.237819464 -0.08336923
[4,] 0.36408585  0.304841714  0.12337786 -0.013442682  0.019341222  0.331707454  0.26118796
[5,] 0.23248053 -0.401962324  0.16653247  0.107802033  0.843745292 -0.062225368  0.01129197
[6,] 0.36444206 -0.266013147 -0.05827786  0.185700413 -0.240182967 -0.005271104 -0.80380484
[7,] 0.39574849 -0.104285968 -0.04114649  0.166653523 -0.312533244 -0.601467155  0.36713629
[8,] 0.41803840 -0.007183605  0.06855383  0.072983951  0.009180198 -0.265613395  0.14131308
[9,] 0.21523797 -0.368300910 -0.03672364 -0.892998475 -0.112888068  0.061957003  0.04790201
[10,] 0.07183744 -0.571767700 -0.11358395  0.349331790 -0.264878077  0.567918997  0.34521359
```

```
             [,8]         [,9]         [,10]
 [1,] -0.084834062  0.474425305 -0.6690714888
 [2,]  0.007126797  0.004212629  0.0002497826
 [3,] -0.089258879  0.380167210  0.7404905337
 [4,] -0.144609749 -0.747347357 -0.0323589585
 [5,] -0.170503128  0.005847386  0.0036904058
 [6,] -0.063980134 -0.218732407 -0.0527527802
 [7,] -0.449573315  0.081170670 -0.0103668020
 [8,]  0.850918762 -0.022024652 -0.0037475480
 [9,] -0.016455606  0.009067850  0.0014669472
[10,]  0.065259461  0.129667491  0.0070573477
```

Per trobar els nous punts i així poder representar la gràfica de la PCA, hem fet el producte entre aquests vectors propis i la matriu de dades. Així doncs, la primera component de cada vector propi es multiplica per la variable Radi, la segona per Textura, la tercera per Perimetre, de la mateixa manera, Area, Uniformitat, Compacitat, Concavitat, PuntsConcaus, Simetria, DimensioFractal, per la quarta, cinquena, sisena, setena, vuitena, novena i desena component respectivament. Les coordenades dels punts nous seran cada una de les files de la matriu components_principals. En el cas que ens quedem amb $n < 10$ components, considerarem només les primeres n columnes d'aquesta matriu. Donat que el primer vector propi té tots els valors del mateix signe, estem sumant una ponderació de cada una de les característiques, podem veure que a la que es dóna més valor és a la característica PuntsConcaus, per tant, petits canvis en aquest valor poden canviar el resultat. A partir de la gràfica podem veure doncs que, com més alts siguin els valors de les dades més possibilitats tenim que un tumor sigui maligne. Per altre costat, la segona component principal té les característiques Radi, Textura, Perimetre i Area positives, mentre que la resta són negatives, tot i això, la segona component no ens separa les dades en dos grups ben definits. Per acabar amb la PCA representarem un *biplot*, per així poder estudiar el comportament de les variables. Per realitzar-lo utilitzarem la funció $fviz\_pca\_biplot()$ del paquet $factoextra$, la qual requereix la resposta de la funció *prcomp*() que ens retorna un objecte amb les desviacions típiques, és a dir l'arrel quadrada dels valors propis, i els vectors propis. Tot i això, li assignem els valors que hem obtingut anteriorment.

```
library(factoextra)
p <- prcomp(dades.PCA, retx=FALSE, scale=FALSE)
p$x <- components_principals
p$rotation <- eigen_sigma$vectors
```

```
rownames(p$rotation )<- c("Radi", "Textura", "Perimetre", "Area", "Uniformitat", "Compacitat"
    , "Concavitat", "PuntsConcaus", "Simetria", "DimensioFractal")
fviz_pca_biplot(p,
                axes = c(1,2),
                xlab='First Component',
                ylab='Second Component',
                geom= c("point"),
                habillage = dades$Diagnostic)
```

A continuació veiem el resultat (vegeu Figura 3.4):

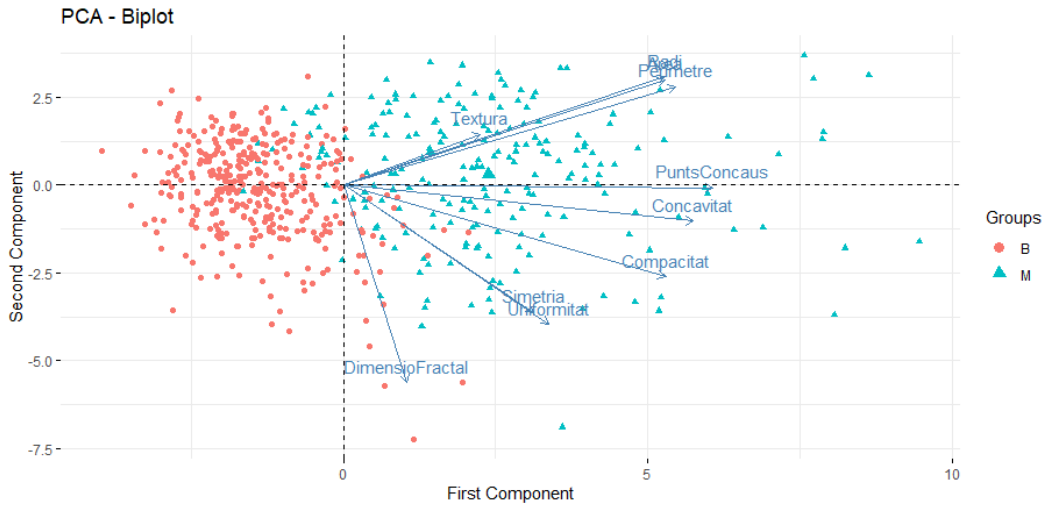

Figura 3.4: *Biplot* de les dades un cop aplicada la PCA. Gràfica realitzada amb el programa RStudio.

A la imatge podem observar el grup destacat anteriorment de Perimetre, Area, Radi i Textura, els quals segueixen una mateixa direcció. També podem destacar la variable DimensioFractal, la qual aporta una gran variabilitat a la segona component, mentre que a la primera gairebé no n'aporta. Així com, PuntsConcaus la qual te el comportament contrari. En aquest cas podríem calcular el rang de la matriu de variables i, donat que veiem un grup que aparentment sembla correlacionat, podríem descartar alguna de les variables, Radi, Perímetre o Àrea. No obstant, fer l'estudi podria ser interessant si tinguéssim moltes variables, en el nostre cas no és necessari reduir el nombre d'aquestes.

3.3 Variables simulades amb distribució normal

Com ja s'ha mencionat a la secció 2.2, les variables que segueixen una distribució normal tenen molt bones propietats, a més, en un estudi estadístic, el primer pas és considerar unes variables aleatòries, que s'acostumen a agafar normals, i veure que l'algoritme funciona bé per aquestes dades, en aquest cas, es procedeix amb l'estudi. És per això que s'ha considerat important veure com es poden crear de manera aleatòria, utilitzarem el paquet *mvtnorm*. Posteriorment, a les seccions, 4.1.5 i 4.2.2, veurem com apliquem aquestes dades als algoritmes SVM i Naive Bayes. Veiem doncs el codi:

```
 library(mvtnorm)
#Posem una llavor perque sempre doni els mateixos resultats
 set.seed(1)
#Nombre d'observacions, en generarem tantes com te el repositori que hem utilitzat al treball
n <- 569
# Nombre de variables, que tambe les farem coincidir amb el nombre de variables del
    repositori utilitzat.
p <- 10
sd <- 0.5
meanVec <- numeric(length = p)
X <- list()
sigma <- diag(rep(sd * sd, p))
X <- rmvnorm(n = n, mean = meanVec, sigma = sigma)
colnames(X)<- c("Radi", "Textura", "Perimetre", "Area", "Uniformitat", "Compacitat", "
    Concavitat", "PuntsConcaus", "Simetria", "DimensioFractal")
#Separacio entre dades de test i dades d'entrenament
indexes_partition <- createDataPartition(y = 1:dim(X)[1], p = 0.66,list = FALSE)
dades_train_X<- X[indexes_partition,]
dades_test_X <- X[-indexes_partition,]
#Generem aleatoriament tambe el diagnostic de les dades d'entrenament
dades_train_labels_X <- sample.int(2, dim(dades_train_X)[1], replace = TRUE) - 1
dades_train_labels_X <- ifelse(dades_train_labels_X == 0, "M", "B")
```

Amb aquest codi hem creat una matriu X de dades tals que les seves variables segueixen una distribució normal. Hem considerat variables sense correlació, ja que, el codi per a l'aplicació dels diferents mètodes i la manera d'interpretar els resultats és equivalent per a tots els casos. Tot i això, als annexos, (vegeu secció A.6), hem creat dos tipus de dades més, una on les seves variables estan poc correlacionades i l'altre molt correlacionades.

4 Diagnòstic del càncer de mama mitjançant algoritmes de classificació

Si no s'indica específicament, la informació d'aquesta secció ha estat extreta del llibre [26].

A continuació podrem veure l'aplicació de la teoria introduïda, per això, hem realitzat l'estudi de la predicció del càncer de mama amb els algoritmes k-veí més proper, Naive Bayes, Support Vector Machine (SVM), arbres de decisió i xarxes neuronals artificials. No obstant, a causa de l'extensió que suposaria l'explicació detallada de cada un d'ells, només en destacarem tres. Als annexos (vegeu secció A.7) podem veure una breu definició de tots els algoritmes juntament amb els seus avantatges i inconvenients.

Per una banda, pel seu gran fonament matemàtic i la relació estreta amb la geometria, que fa que veiem un altre enfocament de les matemàtiques en aquest treball, explicarem l'algoritme SVM. Per altra banda, pel seu interès matemàtic també explicarem l'algoritme de Naive Bayes. A més, donat que ha sigut l'algoritme que ens ha proporcionat una major sensibilitat, veurem una breu explicació del model K-NN.

L'objectiu d'aquest estudi és trobar un classificador que redueixi al màxim el nombre d'errors. Com és sabut, podem trobar dos tipus d'errors, els falsos negatius, és a dir que el model el classifiqui com a negatiu quan és positiu, o, per contrari els falsos positius. En aquest cas, s'ha donat més importància als falsos negatius, ja que, clínicament, és millor detectar de manera preventiva un càncer.

Per estudiar aquests valors entren en joc l'especificitat i la sensibilitat. La sensibilitat d'un model mesura la proporció de mostres positives que han sigut correctament classificades. I es calcula de la següent manera:

$$sensibilitat = \frac{TP}{TP + FN},$$

on TP = veritables positius i FN = falsos negatius. Observem doncs que, com més alta sigui la sensibilitat, menys falsos negatius hi haurà. Per altra banda, l'especificitat mesura la proporció de mostres negatives que s'han classificat correctament, es calcula de la següent forma:

$$especificitat = \frac{TN}{TN + FP},$$

on TN = veritables negatius i FP= són falsos positius. En aquest cas doncs, com més augmentem aquest valor menys falsos positius hi haurà. Per entendre millor tots els valors mencionats anteriorment vegem la següent taula (veure Taula 1).

		Real	
		P	N
Predit	P	TP	FP
	N	FN	TN

Taula 1: Relació entre els valors predits i els valors reals.

Finalment tenim la corba característica operativa del receptor (ROC), la qual s'utilitza habitualment per examinar la compensació entre la detecció de veritables positius i la reducció dels falsos positius. Les corbes es defineixen en una gràfica amb la proporció de vertaders positius a l'eix vertical i la proporció de falsos positius a l'eix horitzontal, aquests valors són equivalents a la sensibilitat i a l'especificitat. Veiem a continuació un exemple representatiu d'aquestes per a entendre-ho de millor manera (vegeu figura 4.5).

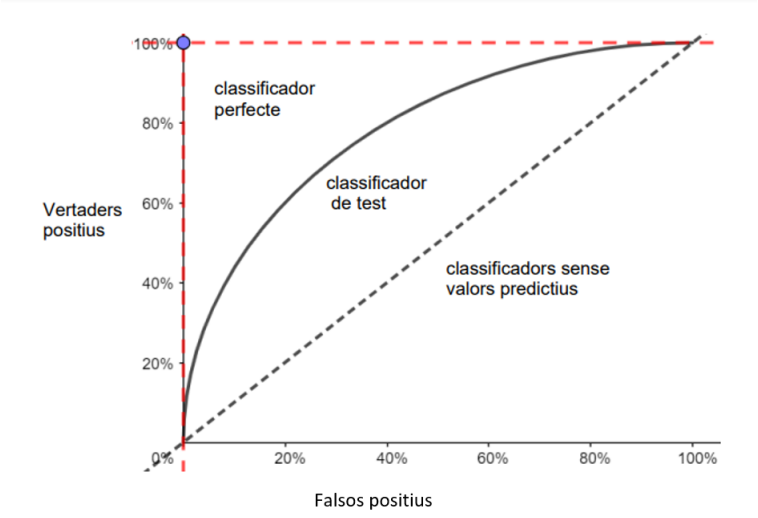

Figura 4.5: Descripció de la corba de ROC. Gràfica d'elaboració pròpia.

A la imatge podem observar tres tipus de classificadors, el primer d'ells, la línia diagonal, que representa un classificador sense valor predictiu, és a dir, detecta els vertaders positius i els falsos positius exactament a la mateixa velocitat, la qual cosa implica que el classificador no pot discriminar entre els dos. Aquesta és la línia base per la qual es poden jutjar altres classificadors. Les corbes ROC que cauen prop d'aquesta línia indiquen models poc útils. Per altra banda, el classificador perfecte té una corba que passa pel 100% i una taxa de fals positiu de 0%. És capaç d'identificar correctament tots els positius abans de classificar incorrectament qualsevol resultat negatiu. I, finalment, el classificador de test, que és on es troben la majoria

dels classificadors, com més a prop estiguin de la corba del classificador perfecte, millor serà per identificar valors positius. Intentarem doncs trobar el model amb la millor corba de ROC. Juntament amb la corba ROC tenim el valor AUC (Area Under the ROC Curve), que representa l'àrea que es troba a sota de la corba. Aquest índex es pot interpretar com la capacitat d'un classificador per distingir entre classes, com més alt sigui aquest valor, millor serà el rendiment del model a l'hora de distingir entre les classes positives i negatives.

4.1 Support Vector Machine

Consisteix en un conjunt d'algoritmes d'aprenentatge supervisat. L'objectiu del Support Vector Machine (SVM) és, mitjançant un hiperplà, dividir l'espai en dues parts per crear una partició de les dades en dos grups prou homogenis, i així poder classificar les noves dades en funció de la seva posició a l'espai.
Destaquem quatre aplicacions del SVM [15]:

- Detecció de cares.
- Categorització del text i hipertext, com per exemple, identificar l'idioma utilitzat en un document.
- Bioinformàtica: és útil tant per a la classificació de proteïnes com per a la classificació del càncer.
- Control predictiu generalitzat: s'utilitza per controlar dinàmiques que poden ser caòtiques a partir de paràmetres útils.

Veiem ara les seves fortaleses i mancances. Les afirmacions de la taula la podem trobar a [25]:

Fortaleses	**Mancances**
Tracta de manera eficient les dades no lineals. Simple, ràpid i efectiu. És eficaç per resoldre problemes de classificació i regressió. Té una gran estabilitat, un petit canvi a les dades no afecta enormement a l'hiperplà.	Escollir una funció de kernel adequada per gestionar les dades no lineals, és difícil. És difícil d'entendre i implementar. Requereix d'un temps d'entrenament llarg. La complexitat algorítmica i els requisits de memòria són molt alts, ja que ha d'emmagatzemar tots els vectors de suport a la memòria i aquests creixen bruscament amb la mida de les dades d'entrenament.

A continuació motivarem els conceptes principals darrere del SVM el que, com ja s'ha mencionat anteriorment, té un enfocament cap a la geometria amb el qual es produeix una matemàtica elegant amb models geomètricament intuïtius i teòricament ben fundats.
Donat que el nostre estudi està centrat en la identificació del càncer de mama ens focalitzarem en els classificadors SVM a partir d'hiperplans. Tota la informació d'aquesta secció la podem trobar a l'article [1].
Existeixen dos casos de SVM, quan les dades són linealment separables i quan no. Comencem doncs explicant el primer cas:

4.1.1 Casos linealment separables

En aquest cas existeix un hiperplà que classifica correctament tots els punts en els dos conjunts. És a dir, podem trobar l'hiperplà que separa les dades en dos conjunts homogenis. Suposem que tenim una classificació binària dels punts $(x_1, ..., x_m)$ on les seves etiquetes corresponents són $y_i = \pm 1$. En aquest cas la funció de classificació és $f(x) = sign(wx - b)$, on w determina l'orientació del pla discriminant i l'escalar b el desplaçament del pla des de l'origen. Aleshores existeixen infinits hiperplans que classifiquen correctament les dades d'entrenament com podem veure a la figura (vegeu Figura 4.6).

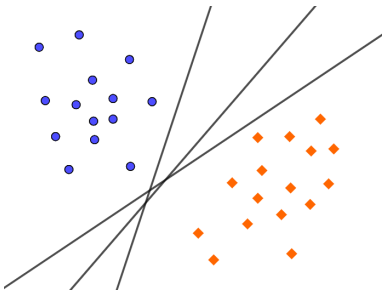

Figura 4.6: Dades linealment separables. Gràfica d'elaboració pròpia.

Però el que més ens interessarà serà aquell que estigui més allunyat de les dues classes, bàsicament perquè així podrem garantir que, si afegim noves dades, aquestes segueixin estant ben classificades. I com el podem construir? Doncs bé, tenim dues maneres de fer-ho.

Manera 1: Considerem l'envolupant convexa[8], llavors hem d'identificar quins són els dos punts de cada conjunt que estan més propers. Finalment, hem de construir el pla que divideixi aquests dos punts. Per fer-ho considerem la mínima recta que els uneix i construïm la mediatriu perpendicular a la unió dels dos vèrtexs que equidisti a cada un d'ells.

Com trobar els punts més propers en els dos conjunts?

Siguin I i J dos conjunts formats pels elements de cada classe, és a dir, I conté els elements de classe 1 i J els de classe -1. Considerem també els pesos $(\alpha_1, ...\alpha_m)$ i $(\beta_1, ..., \beta_m)$. De manera similar a com trobem el baricentre d'un objecte, buscarem els α_i i β_i que facin mínima la distància entre els dos punts c i d de les envolupants convexes, tenint en compte que no tenen per què coincidir amb cap punt $(x_1, ..., x_m)$. Per trobar el valor mínim, doncs, haurem de resoldre el següent problema quadràtic:

$$\min_{\alpha,\beta} \frac{1}{2}||c - d||^2,$$

on $c = \sum_{y_i \in I} \alpha_i x_i$, $d = \sum_{y_i \in J} \beta_i x_i$ tals que $\sum_{y_i \in I} \alpha_i = 1, \sum_{y_i \in J} \beta_i = 1, \alpha_i \geq 0$ i $\beta_i \geq 0$ per $i = 1, ..., m$.

Existeixen diferents algoritmes quadràtics com ara el mètode de quasi-Newton [20], els quals hem vist al llarg del grau, que poden resoldre problemes de mida petita (milers de punts). Però aquests algoritmes no serveixen quan la matriu original de dades, per a mètodes lineals, o la matriu de kernels, per a mètodes no lineals, ocupen una memòria de grans dimensions. I, per tant, per a conjunts de dades més grans s'han d'utilitzar tècniques alternatives, les

[8] Es defineix l'envolupant convexa d'un conjunt de punts S de dimensió n com la intersecció de tots els conjunts convexos que contenen S. Per N punts $p_1, ..., p_N$ l'envolupant convexa C ve donada per l'expressió $C = \sum_{j=1}^{N} \lambda_j p_j : \lambda_j \geq 0$ per tot j i $\sum_{j=1}^{N} \lambda_j = 1$, [33].

quals podem dividir en tres categories: tècniques en què els components del nucli s'avaluen i descarten durant l'aprenentatge, mètodes de descomposició en què s'utilitza un subconjunt de dades en evolució i, finalment, mètodes d'optimització que exploten específicament l'estructura del problema SVM.

Per a la primera categoria, l'enfocament més obvi és actualitzar seqüencialment els α_i. Aquest mètode s'utilitza a l'algoritme kernel Adatron [16]. Per a algunes variants pot ser molt fàcil d'implementar i pot donar una impressió ràpida del rendiment dels SVM en tasques de classificació. És equivalent al mètode de Hildreth [23] en teoria de l'optimització. Tot i això, no és tan ràpid com la majoria dels programes quadràtics, especialment en conjunts de dades petits.

En segon lloc, tenim els mètodes de descomposició. La idea és que, en lloc d'actualitzar seqüenciadament α_i, l'alternativa és actualitzar les α_i en paral·lel però només utilitzant un subconjunt de dades a cada etapa. En aquest cas es resol un programa quadràtic o lineal molt més petit per a cada conjunt de dades. Així, es resolen molts subproblemes petits en lloc d'un de gran. Hi ha molts codis basats en aquesta estratègia de descomposició, per exemple els codis SVM disponibles en línia com SVMTorch[4] i SVMLight [24] utilitzen aquesta estratègia de treball.

Finalment, l'últim mètode és directament atacar al problema de SVM des d'una perspectiva d'optimització i crear algoritmes explícitament per a l'estructura del problema. Ho podem veure al mètode Lagrangià de SVM (LSVM), que resol problemes de classificació lineal per milions de punts en minuts al Pentium III [29].

Manera 2: L'alternativa equivalent és intentar trobar un conjunt de dos plans paral·lels que divideixin els punts en grups homogenis i que maximitzin el marge entre els dos plans de suport. Diem que un pla és de suport si tots els punts de cada classe estan a un costat del pla. Així doncs, considerem un hiperplà de l'espai n-dimensional $wx - b = k$ on w és un vector de pesos. En el nostre cas considerarem $wx - b = 1$ i $wx - b = -1$. La distància entre aquests dos plans és $\gamma = \frac{2}{||w||}$[9]. Intentar maximitzar la distància entre els dos plans és equivalent a minimitzar els pesos $\frac{||w||}{2}$ al següent problema quadràtic:

$$\min_{w,b} \frac{1}{2}||w||^2,$$

tal que $wx_i \geq b + 1$, $y_i \in Class1$ i $wx_i \leq b - 1$, $y_i \in Class - 1$. Observem que podem simplificar les condicions per $y_i(wx_i - b) \geq 1$.

[9] Donats dos plans $\pi_1 = Ax + By + Cz + D_1 = 0$ i $\pi_2 = Ax + By + Cz + D_2 = 0$ la seva distància es calcula de la següent forma: $d(\pi_1, \pi_2) = \frac{|D_1 - D_2|}{\sqrt{A^2+B^2+C^2}}$

Com podem veure les dues maneres són equivalents. De fet, en el primer cas els vectors de suport determinen els punts més propers dels conjunts i en el segon són aquests mateixos que determinen la posició dels plans de suport.

4.1.2 Casos linealment inseparables

Suposem que tenim un cas com el de la següent imatge (veure imatge 4.7):

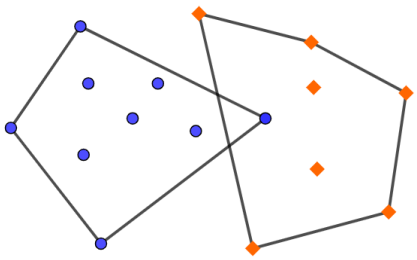

Figura 4.7: Dades linealment inseparables. Gràfica d'elaboració pròpia.

Observem que, en aquest cas, si intentem construir un pla de suport ens serà impossible, per tant, els mètodes mencionats anteriorment no ens serveixen. A continuació proposarem dues millores perquè aquests no fallin.

Al primer mètode, en el cas que hi hagi un conjunt finit de punts que ens eviten la separació lineal entre les dues classes, el que farem és restringir la influència de cada punt introduint una cota superior $D < 1$ de la següent manera:

$$d = \sum_{y_i \in J} \beta_i x_i \text{ tal que } \sum_{y_i \in J} \beta_i = 1 \text{ i } 0 \leq \beta_i \leq D,$$

similarment pel punt c.

Així doncs per una D suficientment petita els dos envolupants convexos reduïts no intersequen. En aquest cas per a trobar els dos punts més propers haurem de resoldre el següent problema quadràtic, tenint en compte la cota superior D:

$$\min_{\alpha,\beta} \frac{1}{2}||c - d||^2,$$

on $c = \sum_{y_i \in I} \alpha_i x_i$, $d = \sum_{y_i \in J} \beta_i x_i$ tal que $\sum_{y_i \in I} \alpha_i = 1, \sum_{y_i \in J} \beta_i = 1, 0 \leq \alpha_i \leq D$ i $0 \leq \beta_i \leq D$ per $i = 1, ..., m$.

Pel que fa al segon mètode, el que volem és que no hi hagi punts que es classifiquin malament,

ni punts que estiguin sobre de l'hiperplà. Així cada punt que caigui en el costat equivocat es considerarà un error. El que volem és maximitzar la distància entre els plans i minimitzar l'error de manera simultània. Per això, el que farem és utilitzar una variable anomenada de folgança, z_i, que serà una penalització ponderada. Així doncs, el problema a minimitzar serà el següent:

$$\min_w \frac{1}{2}||w||^2 + C\sum_{i=1}^{l} z_i \text{ tal que } y_i(wx_i - b) + z_i \geq 1 \text{ i } z_i \geq 0 \ i = 1, ..., m.$$

Hem d'anar amb compte amb el valor C, ja que al modificar-lo s'ajustarà la penalització. Per tant, com major sigui aquest paràmetre, més difícil serà l'optimització per aconseguir que una separació sigui cent per cent correcta. Per altra banda, si posem un valor molt petit posarà èmfasis en un marge general més ampli. Així doncs, és molt important trobar un equilibri.
Per acabar tenim un últim cas:

4.1.3 Funcions no lineals via kernels

Considerem ara un problema com el de la figura (veure Figura 4.8):

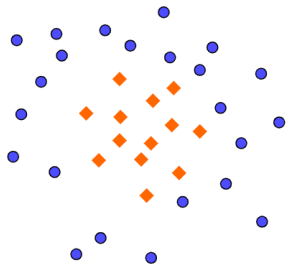

Figura 4.8: Dades disperses a l'espai. Gràfica d'elaboració pròpia.

En aquest cas observem que necessitaríem una circumferència per poder separar les dues classes. Un mètode clàssic que s'utilitza per convertir un algoritme de classificació lineal a un que no ho és, consisteix, simplement, en anar afegint característiques addicionals a les dades. Llavors, els algorismes de classificació lineal ja existents es poden aplicar al conjunt de dades ampliat a l'espai de característiques, produint funcions no lineals a l'espai d'entrada original.
Per entendre-ho millor veiem un exemple:
Suposem que tenim un espai de dimensió dos amb les característiques [r,s], i considerem una

aplicació que envia aquest espai a un de dimensió cinc amb característiques [r,s,rs,r^2,s^2]. A continuació construïm el discriminant lineal en aquell espai. Sigui $\theta(x) : \mathbb{R}^2 \to \mathbb{R}^5$ aleshores

$$\begin{aligned} x = [r, s] \quad &\longrightarrow \quad \theta(x) = [r, s, rs, r^2, s^2] \\ w_1 r + w_2 s \quad &\longmapsto \quad w_1 r + w_2 s + w_3 rs + w_4 r^2 + w_5 s^2. \end{aligned}$$

Així doncs la funció de classificació serà:

$$f(x) = sign(w\theta(x) - b) = sign(w_1 r + w_2 s + w_3 rs + w_4 r^2 + w_5 s^2 - b)$$

que és lineal a l'espai de característiques de dimensió cinc, però quadràtic en l'espai d'entrada de dimensió dos. Si volem afegir més característiques és quan apareix el problema, ja que, la dimensió de l'espai de sortida creix exponencialment degut a totes les permutacions que sorgeixen. Ara bé, no és pràctic calcular $\theta(x)$, és aquí on apareixen els kernels. Observem el següent problema no lineal quan utilitzem programes quadràtics:
Definim $\theta(x) : \mathbb{R}^n \to \mathbb{R}^{n'}$ $n' \gg n$. Necessitem optimitzar:

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j \theta(x_i)\theta(x_j) - \sum_{i=1}^{l} \alpha_i \text{ tal que } \sum_{i=1}^{l} y_i \alpha_i = 0 \text{ i } C \geq \alpha_i \geq 0 \; i = 1, ..., m.$$

Observem que la variable θ només apareix com un producte escalar en el nostre objectiu. Com veurem a continuació, aplicant els kernels de Hilbert-Schmidt, per a certes aplicacions θ i dos punts u i v, el producte escalar de les imatges dels punts, es pot avaluar utilitzant les funcions kernel, sense conèixer explícitament la funció θ. Tenim que, $\theta(u) \cdot \theta(v) \equiv K(u, v)$.
Abans de continuar farem un petit incís per a contextualitzar l'espai de Hilbert i els kernels en ell.

4.1.3.1 Els kernels a l'espai de Hilbert

Primer de tot definirem què és un espai de Hilbert [12]:

Definició 4.1. *Un espai de Hilbert $\mathbb{H}$ és:*

- *Un espai vectorial sobre els nombres reals $\mathbb{R}$.*
- *Té una norma $||\cdot||_{\mathbb{H}}$ (i.e. una mètrica per calcular distàncies entre vectors a l'espai).*
- *És complet amb la norma mencionada anteriorment (i.e. cada seqüència de Cauchy convergeix).*
- *La norma prové d'un producte escalar $\langle \cdot, \cdot \rangle_{\mathbb{H}}$ (i.e. $||f||_{\mathbb{H}} = \sqrt{\langle f, f \rangle_{\mathbb{H}}}$).*

Definició 4.2. *Sigui X un conjunt no buit. Considerem $K : X \times X \to \mathbb{R}$, diem que K és un kernel, si existeix un espai de Hilbert $\mathbb{H}$ sobre $\mathbb{R}$ i una aplicació $\phi : X \to \mathbb{H}$ tal que per a tot $x, y \in X$, $K(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathbb{H}}$.*

Per un espai de Hilbert $\mathbb{H}$ de funcions reals en X, i per un punt qualsevol $x \in X$, la funció d'avaluació a x està definida com la funció $L_x : H \to \mathbb{R}$ tal que per a tota funció $f \in \mathbb{H}$, $L_x(f) = f(x)$. En aquest cas, $\mathbb{H}$ s'anomena espai de Hilbert del nucli de reproducció si per qualsevol $x \in X$, L_x està acotada, és a dir, existeix una constant finita M tal que,

$$|L_x(f)| = |f(x)| \leq M||f||_{\mathbb{H}}.$$

Veiem doncs on els kernels entren en joc. Segons el teorema de representació de Riesz [2] per cada $x \in X$, existeix una funció $K_x \in \mathbb{H}$ tal que $f(x) = L_x(f) = \langle f, K_x \rangle_{\mathbb{H}}$ per tot $f \in \mathbb{H}$. Donat que K_x és ella mateixa una funció a $\mathbb{H}$, per qualsevol $y \in X$, tindrem $K_x(y) = L_y(K_x) = \langle K_x, K_y \rangle_{\mathbb{H}}$
Així doncs, definim:

$$K : X \times X \to \mathbb{R},$$

tal que $K(x, y) = \langle K_x, K_y \rangle_{\mathbb{H}}$. Per la definició que hem donat anteriorment podem dir que K és un kernel, el qual s'anomena kernel de reproducció de $\mathbb{H}$.
Veiem els exemples més comuns de kernels, en particular el que utilitzarem després per a millorar l'algoritme de SVM [3] .

- Kernel lineal
  $K(x, x') = x \cdot x'$.
- Kernel gaussià
  $K(x, x') = \exp\left\{-\frac{||x-x'||^2}{2\sigma^2}\right\}$, $\sigma > 0$, on, donat $x = (x_1, \ldots, x_d)$, $||x|| = \sqrt{\sum_{i=1}^{d} x_i^2}$.
- Kernel polinomial
  $K(x, x') = (x \cdot x' + 1)^d$, $d \in \mathbb{N}$.

El kernel gaussià s'està fent cada cop més popular i potent per al reconeixement de patrons. Les seves propietats estadístiques proporcionen enfocaments potencials per al seu ajustament i bones direccions per a futures investigacions, observem doncs la seva definició i diferents propietats.

Definició 4.3. *Sigui $X \in \mathbb{R}^{N\times p}$ un conjunt de dades on X conté N observacions, $X_1, ..., X_N$, $X_i \in \mathbb{R}^{1\times p}$ i cada una d'elles està etiquetada per un valor $y_i \in \{-1, 1\}$. Aleshores, com hem vist anteriorment,*

$$K(i,j) = \exp\left\{\frac{-||X_i - X_j||^2}{2\sigma^2}\right\}, \sigma > 0.$$

Observem que aquest kernel requereix l'ajustament del valor σ, el qual podríem trobar utilitzant el mètode de k-grups de validació creuada, [43], veient quin és el σ òptim amb les dades d'entrenament.

El comportament del kernel és evident quan s'examina amb detall, pren valors dins l'interval (0,1), exceptuant els $K(i,i)$ que prenen valor 1. Així doncs, la diagonal de la seva matriu estarà formada tota per uns. A més a més, observem que si prenem un valor molt petit de σ forçarem que els valors de la matriu siguin propers a 0. Mentre que si considerem un valor massa gran de σ els valors de la matriu seran molt propers a 1.

Per altra banda tenim una altra propietat comuna en tots els kernels, aquesta simplement indica que, per bons models, la següent relació és sempre certa:

$$(K(i,j)|(y_i = y_j)) > (K(i,j)|(y_i \neq y_j)).$$

Un primer pensament que tindríem és que per ajustar una matriu de kernels pel SVM, aquesta hauria de prendre valors grans, i, tot i que sembla coherent per la premissa mencionada anteriorment, ens estaríem equivocant, ja que la magnitud dels valors de la matriu no és un atribut important. El valor que sí que és rellevant és la dispersió de la variància dels valors de la matriu [13]. Així doncs, segons Shawe-Taylor i Cristianini si considerem valors petits de σ els classificadors s'ajustaran a qualsevol conjunt d'etiquetes i, per tant, es produirà un sobre ajustament. Per a valors grans impedeixen la capacitat dels classificadors de detectar patrons no trivials perquè el kernel es redueix gradualment a una funció constant. I, per tant, com bé s'ha dit a l'inici, el kernel gaussià requereix d'un correcte ajustament del valor σ per a poder fer un bon estudi de les dades.

Així doncs, tonant on ho havíem deixat, substituint el kernel obtindrem el problema:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{l}\alpha_i \text{ tal que } \sum_{i=1}^{l} y_i\alpha_i = 0 \text{ i } C \geq \alpha_i \geq 0 \ i = 1, ..., m.$$

Per tant substituint el kernel podem passar d'un algoritme lineal a un algoritme no lineal general.

4.1.4 Diagnòstic de càncer de mama mitjançant l'algoritme de classificació Support Vector Machine

A continuació explicarem amb detall els passos que hem seguit per a implementar l'algoritme de classificació SVM. Així com també avaluarem la seva sensibilitat a l'hora de detectar si un tumor és benigne o no.

Pas 1: Explorar i preparar les dades

El primer que farem és importar les dades utilitzant la funció *read.table*().

```
setwd("C:/Users/natap/OneDrive/Escritorio/TFG")
dades<-read.table("wdbc.data", header = FALSE, sep = ",", dec = ".")
```

A continuació donat que l'ID no ens aporta informació per a l'estudi l'eliminarem i només ens quedem amb els primers 10 atributs com hem mencionat anteriorment. Posem també l'etiqueta de cada una de les variables.

```
dades<-dades[2:12]
colnames(dades)<- c("Diagnostic", "Radi", "Textura", "Perimetre", "Area", "Uniformitat", "
    Compacitat", "Concavitat", "PuntsConcaus", "Simetria", "DimensioFractal")
```

Finalment, abans de separar les dades inicials en dades d'entrenament i dades de test, convertirem la característica Diagnostic en format factor, ja que per a la majoria de mètodes la necessitem en aquest format.

```
dades$Diagnostic<- factor(dades$Diagnostic)
```

Ara hem de preparar les dades per treballar amb elles. Amb aquest objectiu, les dividirem, com hem dit abans, amb les que utilitzarem per entrenar, dos terços de les dades, i les que utilitzarem per realitzar tests, un terç. Així doncs, de manera aleatòria dividirem les dades en dues parts.

```
#Utilitzarem la funcio createDataPartition del paquet caret
library(caret)
#Llavor perque indexes_partition ens doni sempre el mateix valor
set.seed(8)
#Seleccio aleatoria dels index per a les dades d'entrenament i de test
indexes_partition <- createDataPartition(y = 1:dim(dades)[1], p = 0.66,list = FALSE)
dades_train<- dades[indexes_partition, -1]
dades_test <- dades[-indexes_partition, -1]
```

```
#Guardem tambe les etiquetes de cada fila de dades
dades_train_labels <- dades[indexes_partition, 1]
dades_test_labels <- dades[-indexes_partition, 1]
```

Per acabar amb la preparació de les dades confirmarem que els subconjunts creats són representatius del conjunt total d'observacions, per això compararem els percentatges de tumors benignes i malignes que hi ha en les dades de sortida d'entrenament i de test amb la funció *prop.table*().

```
#Per a les dades de sortida d'entrenament
> tab_train <-prop.table(table(dades_train_labels))
> round(tab_train,4)*100
dades_train_labels
   B     M
63.93 36.07
#Per les dades de sortida de test
> tab_test<-prop.table(table(dades_test_labels))
> round(tab_test,4)*100
dades_test_labels
    B     M
60.42 39.58
```

Observem doncs que els percentatges dels dos casos es mouen entre els valors següents:

```
> clases_name=c("B","M")
> paste0((clases_name), " = ", round((tab_train + tab_test)/2, 4)*100, "%")
"B = 62.17%" "M = 37.83%"
```

I, per tant, les dades s'han dividit equitativament entre els dos conjunts de dades.

Pas 2: Entrenament del model amb les dades

Per modelar la relació entre les variables biològiques i si el càncer és benigne o no, utilitzarem l'algoritme de classificació SVM. Per això, utilitzarem la funció *ksvm*() del paquet *kernlab*. Començarem entrenant en el cas més simple: quan el kernel és lineal i amb el valor de cost per defecte (C=1).

```
library(kernlab)
#Cas simple, quan posem kernel= "vanilladot" significa que utilitzem el kernel lineal
SVM_model_S <- ksvm(as.factor(dades_train_labels) ~ ., data = dades_train , kernel = "
    vanilladot")
```

Podem observar a partir de la resposta de la funció $ksvm()$, que el model creat té 51 vectors de suport, i que té un error d'entrenament de 0.029178.

Pas 3: Predicció i avaluació del model

Ara per avaluar el rendiment de l'algoritme utilitzarem la funció $predict()$, el qual ens retorna les prediccions fetes pel model a partir de les dades de test.

```
SVM_prediction_S <- predict(SVM_model_S, dades_test)
```

Finalment, per observar les variables classificades correctament i les que no, utilitzarem la funció $confusionMatrix()$ del paquet *caret*.

```
library(caret)
mat_conf<- confusionMatrix(SVM_prediction_S, as.factor(dades_test_labels), positive = "M")
> mat_conf
          Reference
Prediction   B   M
         B 104   8
         M  12  68
```

A partir de la matriu podem observar que 20 de les 192 dades s'han classificat de manera incorrecta que això equival a un 10.41%. Donat que nosaltres buscàvem un model que tingués una millor sensibilitat, a partir de la mateixa matriu resultant d'aplicar la funció de confusionMatrix, podem veure que a sensibilitat és de 0.8958, el que implica que el model està fent un treball molt bo, no obstant, no és el millor.

Pas 4: Millora del rendiment del model

El model es pot millorar de dues maneres, per un costat podem variar els diferents valors de costos C, i buscar quin és el que ens dóna un millor resultat. Per altre costat, també podem utilitzar una funció kernel més complexa. De la forma que s'ha explicat anteriorment, hauríem de crear una funció que enviés les nostres característiques a un espai de dimensió més elevada. No obstant, hem de tenir en compte que tenim moltes característiques a l'espai de sortida, fet que pot comportar una gran dificultat computacional, tot i això, ho provarem. A la secció dels annexos A.8, hi podem veure primer l'aplicació de la funció kernel i llavors la prova dels diferents valors dels costos. En el primer cas, per fer-ho, utilitzarem el kernel Gaussià RBF, que s'ha demostrat que funciona bé per a molts tipus de dades. Observem que la sensibilitat millora, que era el nostre objectiu, tot i això, augmenta 0.3, així doncs hem de valorar en cada cas si ens surt a compte assumir el cost computacional d'aquesta segona

versió. En el segon cas, realitzem els mateixos passos que en el cas base, amb la diferència que recorrerem un vector de costos. Com podem veure, per a diferents valors de costos, no varia la sensibilitat, només millora l'especificitat en el cas $C > 1$ on es manté constant per a tots els valors.

Finalment, per aquest model, observarem la corba ROC (vegeu corba ROC 4.9) explicada a l'inici de la secció per poder realitzar un últim estudi. La programarem amb el paquet *pROC*.

```
roc_svm_test <- roc(response = as.factor(dades_test_labels), predictor = as.numeric(SVM_
    prediction_S))
plot(roc_svm_test,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)
legend(0.3, 0.2, legend = c("test-svm"), lty = c(1), col = c("blue"))
```

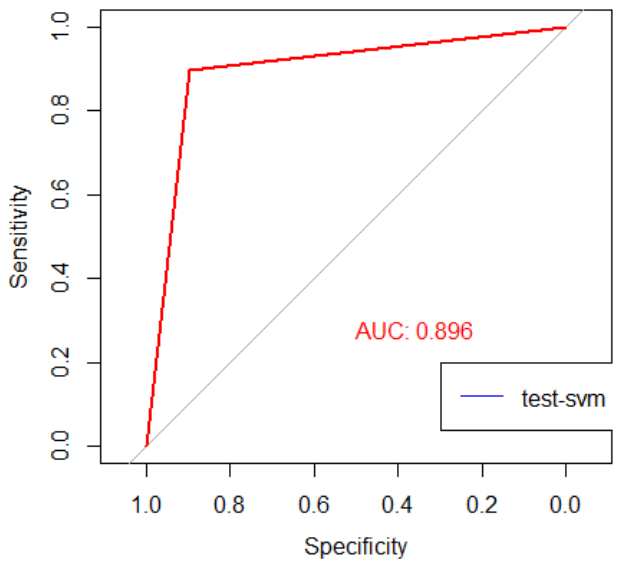

Figura 4.9: Corba de ROC de l'algoritme SVM. Gràfica realitzada amb RStudio.

Podem observar doncs que la corba està propera a la corba perfecta, el qual indica un bon funcionament del model.

4.1.5 Aplicació de les variables simulades amb distribució normal

Utilitzant el mateix codi que hem fet servir per les dades del repositori, veurem com aplicar-ho a les dades simulades de la secció 3.3, tals que les variables segueixen una distribució normal.

```
library(kernlab)
```

```
SVM_model <- ksvm(as.factor(dades_train_labels_X) ~ ., data = dades_train_X , kernel = "
    vanilladot")
SVM_prediction <- predict(SVM_model, dades_test_X)
```

Així doncs a *SVM_prediction* trobarem els resultats predits per l'algoritme.

4.2 Naive Bayes

Tot i que existeixen diferents algoritmes de classificació, els classificadors que utilitzen la probabilitat, com Bayes, són els mètodes que s'utilitzen més, per la seva simplicitat i criteri. L'algoritme de Naive Bayes (NB) és un dels més populars, simples i pràctics, i, per això s'utilitza en molts estudis. Està basat en el teorema de Bayes[10] amb la condició que les característiques han de ser independents. Simplement, utilitza les dades d'entrenament per estimar els valors de probabilitat necessaris per a la classificació. Veiem exactament com funciona en el cas discret:

Per a l'entrenament, tenim que cada conjunt està representat com un vector de característiques $< a_1, ..., a_n >$. Si considerem un conjunt com l'anterior, no observat encara, NB s'encarrega d'assignar-li la classe

$$c_{predicted} = \text{argmax}_{c \in C} \frac{p(a_1, ..., a_n|c)p(c)}{p(a_1, ..., a_n)}\,[11],$$

on C és el conjunt de totes les classes, $p(c)$ la probabilitat de la classe $c \in C$ i $p(a_1, \ldots, a_n)$ i $p(a_1, \ldots, a_n|c)$ són les probabilitats que les característiques $1, \ldots, n$ prenguin els valors $a_1, \ldots, a_n$, en el segon cas, amb l'afegit d'estar condicionada a què és de la classe c. A més a més, com hem dit abans, l'algoritme de Naive Bayes assumeix que les característiques són condicionalment independents donat un valor de la classe. Per tant, la fórmula es transforma en:

$$c_{predicted} = \text{argmax}_{c \in C} p(c) \prod_{i=1}^{n} p(a_i|c).$$

Així doncs, estem buscant el valor $c \in C$ que fa que el producte sigui màxim. Observem que si el valor d'una de les probabilitats és zero, llavors la probabilitat condicionada és zero, i en conseqüència el productori és nul, tot i que els altres valors de les probabilitats siguin molt

[10] Donat un conjunt finit o numerable d'esdeveniments, $A_1, ..., An, ...$, mútuament excloents i tals que la probabilitat de cada un d'ells és diferent de zero. I sigui B un altre esdeveniment qualsevol amb $P(B) > 0$ tal que en coneixem $P(B|A_i)$. Llavors la probabilitat de $P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)}$ [31].

[11] L'argmax és una operació que troba l'argument que dóna el valor màxim d'una funció de destí.

grans. Per això utilitzem el valor de Laplace, així considerem la següent formula;

$$p(a_i|c) = \frac{count(a_i, c) + L}{count(c)},$$

on $count(a_i, c)$ és el nombre d'aparicions de la característica i amb el valor a_i en els casos de la classe c al conjunt de dades d'entrenament, count(c) és el nombre d'instàncies que tenen la classe c i L el nombre de l'estimador de Laplace, que per defecte posarem a 1.
En el cas continu, o bé discretitza el problema utilitzant els quantils, o bé utilitza les funcions de densitat de probabilitat. Naive Bayes genera una distribució gaussiana per a cada variable predictora. La distribució es caracteritza per dos paràmetres, la seva mitjana i la desviació típica. Aleshores, basant-se en aquests valors de cada variable, la probabilitat que un valor sigui "x" es calcula mitjançant la funció de densitat. Llavors el procediment és similar al cas discret. En el nostre cas hem vist que les variables no segueixen una distribució normal, no obstant, veurem que no obtenim mals resultats, això és degut al fet que l'algoritme sabem que funciona bé per a variables normals, però això no significa que no funcioni per a variables que no ho són. A continuació veurem avantatges i inconvenients de l'algoritme [7].

Fortaleses	**Mancances**
És fàcil obtenir una estimació de probabilitat per a una predicció.	Suposa que totes les característiques són igualment importants i independents, fet que no és sempre cert.
Funciona bé tot i que faltin dades, i les dades siguin disperses.	No és ideal per bases de dades amb un gran nombre de variables numèriques.
Necessita relativament poques mostres per l'entrenament, però també funciona bé quan n'hi ha moltes.	Les probabilitats estimades són menys fiables que les classes que s'han predit.
Simple, ràpid i efectiu.	

4.2.1 Diagnòstic de càncer de mama mitjançant l'algoritme de classificació Naive Bayes

Pas 1: Explorar i preparar les dades

Aquest apartat és idèntic per a tots els algoritmes que hem utilitzat, ja que considerem les mateixes dades en tot moment, així doncs, suposem que ja hem realitzat aquest pas.

Pas 2: Entrenament del model amb les dades

De la mateixa manera que el mètode de SVM separarem les dades de manera aleatòria en dades d'entrenament i dades de test.

També, ja hem demostrat que cada una de les particions era una bona representació de les dades totals.

Comencem doncs entrenant el model en el cas que el valor de Laplace és nul, en aquest algoritme, igual que en l'anterior, s'entrena i es fan les prediccions en etapes separades. El primer pas és doncs, crear el model a partir de les dades d'entrenament, on utilitzarem la funció *naiveBayes*() de la llibreria *e*1071. Tot i que les dades són contínues i, per tant, podríem utilitzar l'algoritme de gaussian naive bayes, prenen valors numèrics i, en aquest cas, els dos algoritmes són equivalents [39].

```
#BAYES
library(e1071)
NB_model_S <- naiveBayes(dades_train, as.factor(dades_train_labels), laplace = 0)
```

Pas 3: Predicció i avaluació del model

El segon pas consisteix a fer prediccions sobre les dades de test amb els models de Naive Bayes anteriors. Per això utilitzarem la funció *predict*().

```
NB_prediction_S<- predict(NB_model_S, dades_test)
```

Ara que ja hem calculat les prediccions el següent pas serà avaluar el rendiment dels models. Com que estem davant d'un problema de classificació utilitzarem la matriu de confusió. Per això utilitzarem la funció $confusionMatrix()$ del paquet *caret*.

```
 #Matrius de confusio
mat_conf <- confusionMatrix(NB_prediction_S, as.factor(dades_test_labels))
          Reference
Prediction   B   M
         B 108  10
         M   8  66
Sensitivity : 0.9310
```

Observem doncs que 18 de 231 estan mal classificats, és a dir un $9,38\%$. Veiem també que la sensibilitat és de 0.9310. Podem dir doncs que en aquest cas, és un molt bon mètode de classificació.

Pas 4: Millora del rendiment del model

El primer que cal mencionar és que l'algoritme de Naive Bayes funciona millor per a variables discretes, així doncs, podríem discretitzar les dades, per exemple utilitzant els quartils perquè l'algoritme millorés, no obstant això, seria necessari valorar si la millora del model compensa la pèrdua de les dades.

Donat que el valor de Laplace pot variar, provarem diferents valors amb l'objectiu de trobar aquell que ens doni un millor resultat. Per això considerarem un vector d'estimadors de Laplace. Observem que, permetre que l'estimador de Laplace sigui igual a 1 garanteix, com ja s'ha explicat anteriorment, que el producte de probabilitats no sigui zero.

Seguirem doncs els mateixos passos que anteriorment, ara bé, canviant el valor de Laplace a cada iteració. Podem trobar el codi als annexos (vegeu secció A.9). Com podem veure, per als valors indicats, l'algoritme es queda igual. Així doncs, considerarem la millor predicció pel cas L=0, ja que com tots tenen les mateixes mesures de rendiment, s'escull el model més simple.

Finalment, per aquest model, observarem la corba ROC (vegeu corba ROC 4.10). Com abans, la realitzarem amb el paquet *pROC*.

```
roc_NB_test <- roc(response = as.factor(dades_test_labels), predictor = as.numeric(NB_
    prediction_S))
plot(roc_NB_test, col = "pink", print.auc=TRUE, print.auc.x = 0.7, print.auc.y = 0.3)
legend(0.3, 0.2, legend = c("test-NaiveBayes"), lty = c(1), col = c("blue"))
```

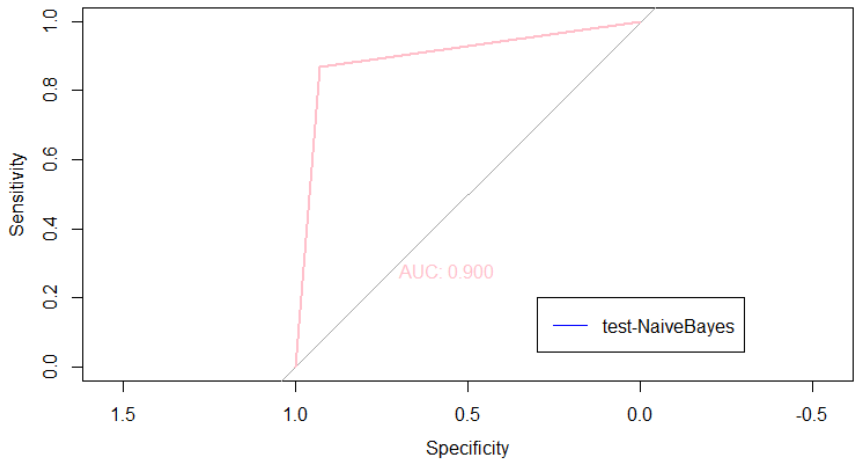

Figura 4.10: Corba de ROC de l'algoritme Bayes. Gràfica realitzada amb RStudio.

Podem observar doncs que la corba està molt propera a la corba perfecta, el qual indica un bon funcionament del model.

4.2.2 Aplicació de les variables simulades amb distribució normal

Igual que a la secció anterior, utilitzem el mateix codi que hem fet servir per les dades del repositori per veure com aplicar-ho a les dades simulades de la secció 3.3, tals que les variables segueixen una distribució normal.

```
library(e1071)
NB_model <- naiveBayes(dades_train_X, as.factor(dades_train_labels_X))
NB_prediction<- predict(NB_model, dades_test_X)
```

Donat que, com a la vida real, no podem saber si els algoritmes han classificat bé els tumors o no, farem una comparació entre els dos algoritmes. Utilitzant la funció *all.equal*() tenim que un $82,3\%$ de les observacions estan classificades de la mateixa manera en els dos algoritmes, SVM i Naive Bayes. Mentre que si comparem els algoritmes amb les dades reals coincideixen en un 90.63%. Considerant que tant les dades com la variable *Diagnostic* son valors aleatoris, podem considerar que els algoritmes funcionen correctament.

4.3 K-veí més proper

En aquest apartat veurem una breu explicació de com funciona l'algoritme K-veí més proper (K-NN) i en els annexos podrem trobar el codi utilitzat i les seves fortaleses i febleses (vegeu annexos A.10).

Per tenir una idea general, l'algoritme K-NN utilitza la informació dels exemples ja classificats per a classificar aquells que encara no ho estan. A més, és un tipus d'algoritme d'aprenentatge anomenat "gandul", ja que no aprèn un model a partir de les dades d'entrenament, sinó que memoritza tot el conjunt de dades d'entrenament i l'usa cada vegada que vol fer una predicció. Així doncs, considera que les característiques són les coordenades en un espai multidimensional, en ell hi representa les diferents dades. Suposem ara que la majoria d'elles ja estan separades en els dos grups, benigne i maligne, i a continuació volem veure si un tumor és benigne o no. Per això el situem a l'espai anterior i considerem els K punts que estiguin a una menor distància euclídia [32]. Llavors s'assignarà el nou punt a la classe que predomini entre els K veïns més propers trobats anteriorment.

La lletra K és el nombre de "veïns" més propers que s'utilitzaran per a la classificació, aquest

valor s'acostuma a agafar com l'arrel quadrada del nombre total de dades d'entrenament. A més a més, en els casos binaris, es pren el nombre K senar i així trenquem possibles empats de vot. Per entendre-ho millor veiem una imatge d'unes dades amb dues característiques i dues classes, considerem també K = 4 (vegeu imatge 4.11).

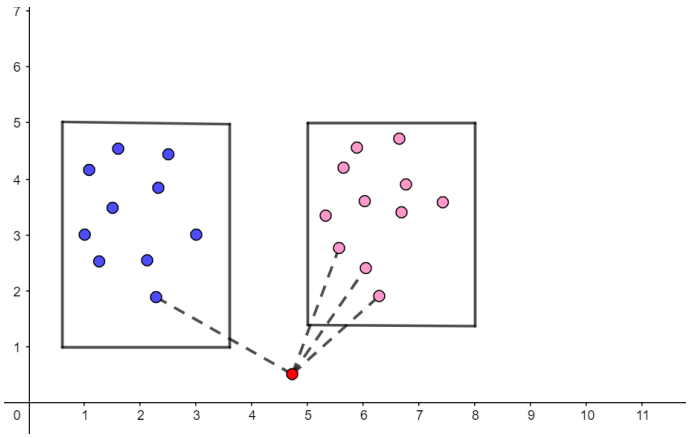

Figura 4.11: Representació pròpia del funcionament de l'algoritme KNN. Figura d'elaboració pròpia.

Observem que en aquest cas classificaríem el nou punt com els roses, ja que predominen entre els quatre veïns més propers.

En particular, si apliquem l'algoritme K-NN a les nostres dades obtenim que s'equivoca 22 de 192 vegades, és a dir un 11.45%. Tot i tenir un error major, que, per exemple, Naive Bayes, tenim que la seva sensibilitat és major, 0.948. Això és degut a que classifica correctament aquells tumors que son benignes, però ho fa pitjor per aquells que són malignes.

5 Discussions

5.1 Comparació amb l'article

L'article té quatre objectius principals, el primer d'ells és realitzar un preprocessament de les dades, el segon, la selecció de les característiques, el tercer, la selecció de l'algorisme d'aprenentatge automàtic que classifiqui millor les dades, i per últim, l'optimització dels paràmetres. Com podem veure, aquests objectius són els quatre passos que hem seguit a l'hora de realitzar la part pràctica del nostre treball.

En el cas de l'article, el qual utilitza les mateixes dades que en el nostre treball, es van realitzar tres experiments per a l'entrenament de les dades d'entrada, en cada un d'ells varen posar l'enfocament a un dels objectius. El primer es va enfocar en la selecció de les característiques, és a dir la realització de la PCA. A l'article no es menciona la quantitat de variables triades, ni quines van ser les escollides, però sí que observen que l'extracció de característiques amb mètodes híbrids, és a dir utilitzant més d'un mètode, millorava el rendiment del model escollit. En el nostre cas hem usat el criteri del percentatge, que ja ens proporciona la qualitat dels resultats desitjada.

El segon experiment va comparar els populars algoritmes d'aprenentatge supervisat. En aquests, les mètriques utilitzades van ser l'AUC i la matriu de confusió, entre d'altres. Els models aplicats varen ser k-veí més proper (K-NN), arbres de decisió (DT), Support Vector Machine (SVM), classificador gradient boosting (GB), random forest (RF), AdaBoost i Naive Bayes (GNB). A l'experiment de l'article, van veure que GNB va obtenir un ROC mitjà més alt del 77%. Tot i això, a partir dels resultats obtinguts, han trobat tres guanyadors: el classificador GB, el RF i el classificador d'arbre de decisió. En el nostre cas, com ja s'ha mencionat a la secció 4 hem programat els algoritmes k-veí més proper, Naive Bayes, SVM, arbres de decisió i xarxes neuronals artificials, i, tot i que pel resultat de l'article esperàvem que l'algoritme d'arbres de decisió fos el millor, hem trobat que el classificador K-NN ha sigut el que ens ha proporcionat una millor sensibilitat, 0.9483, que és el que nosaltres estàvem mirant, no obstant això, la gran majoria dels classificadors aporten uns bons resultats. Aquesta diferència pot ser deguda, a un sobreajustament del problema, molt típic en aquest algoritme. Aquesta diferència es pot haver donat per múltiples factors, però, donat que l'article només mostra els resultats no hem pogut estudiar a fons d'on provenen les diferències. Finalment, per entendre l'últim experiment necessitem definir els hiperparàmetres, aquests són aquells paràmetres

del model que no es poden estimar directament a partir de les dades. Normalment, alguns paràmetres s'han d'ajustar per aconseguir el rendiment desitjat d'un algorisme, aquests, en la majoria dels models, s'especifiquen manualment perquè no hi ha una fórmula analítica per calcular el valor adequat. L'article, mitjançant la programació genètica optimitza les dades i els paràmetres de control del model proposat. En el nostre cas s'avaluen els diferents paràmetres manualment, tot i això, no s'observen millores de rendiment quan els paràmetres varien. Volem destacar, però que, segons el teorema de "*No free lunch* "de Wolpert i Macready [46], cap algorisme funciona per a tots els problemes.

6 Conclusions

Finalment, en aquest últim capítol, observarem l'assoliment dels objectius plantejats a l'inici del treball (vegeu secció 1.4). Podem observar que n'hi havia tres de principals, dos de pràctics i un de teòric. El primer d'ells era: extracció, modificació, entrenament i estudi d'una base de dades. Tot i que hem realitzat assignatures com Estadística o Probabilitats al llarg del grau, es van focalitzar més en la part teòrica, i a la part pràctica, aquest punt ja se'ns donava fet, així doncs, en l'assoliment d'aquest objectiu vam trobar diferents dificultats, les principals, la falta de coneixement de les fonts d'informació i el tractament de les dades, ja que, no només és important saber-les llegir, sinó entendre-les i realitzar diferents aplicacions amb elles. Vull destacar l'anàlisi de components principals, que, tot i no ocupar una gran extensió en el treball, ha sigut un concepte totalment nou del qual n'he après molt. També afegir el punt biològic, com ja s'ha mencionat al llarg del treball, vam considerar que abans de poder treballar amb les dades, havíem de tenir una base consolidada d'allò que estàvem parlant, no obstant això, no em va suposar una dificultat, al contrari, vaig observar un nou camí que també m'agradava molt. Tot i el mencionat anteriorment considerem que s'ha assolit satisfactòriament, ja que hem dut a terme i hem obtingut bons resultats de cada un dels subapartats d'aquest objectiu.

El segon objectiu era: obtenir raonadament quin és el millor mètode de classificació pel càncer de mama. Quan vam començar a pensar el que seria el treball, la meva idea era posar especial èmfasi en aquest punt, ja que, al llarg del grau no havíem tocat ni els mètodes de classificació, ni R, amb aquesta complexitat, i m'interessava molt aprendre'n més, tot i això, finalment, tot i haver-hi treballat, va quedar eclipsat pel fonament matemàtic del qual hem volgut dotar el treball. Així doncs, per a una futura ampliació, considero que podria ser interessant programar personalment la gran majoria dels algoritmes, per així finalment acabar construint un model propi.

L'últim objectiu era: dotar de fonament matemàtic les diferents nocions estadístiques bàsiques. En un inici vaig pensar que seria la part més fàcil del treball, ja que és el que he estat realitzant dia rere dia els últims quatre anys, però no va ser així. Es van presentar tres dificultats, la primera, l'extensió de l'anàlisi univariant al multivariant, la segona, la comprensió dels diferents teoremes i proposicions que m'he anat trobant al llarg del camí, i finalment, la selecció del que realment era útil i interessant per a la posterior aplicació. No obstant, aquest objectiu s'ha assolit en la seva plenitud.

Tot i el que he mencionat anteriorment, estic molt contenta del que s'ha aconseguit amb el treball, ja que s'han assolit els objectius que ens havíem plantejat, així com també he ampliat el meu coneixement, en aspectes com l'anàlisi multivariant, els mètodes de classificació, l'anàlisi de les dades i l'ús del R, entre moltes altres coses.

Per acabar dir que he gaudit molt de la realització d'aquest treball i sobretot tot el que he arribat a aprendre, tant intel·lectualment com personalment.

A Annexos

A.1 Preliminars en l'anàlisi univariant

Abans de començar a endinsar-nos en els annexos, veurem un breu recull de les idees principals sobre l'anàlisi univariant apreses al llarg del grau de matemàtiques, en assignatures com probabilitats o estadística, amb l'objectiu de facilitar la posterior extensió a l'anàlisi multivariant. Tot el que explicarem en aquest capítol ha sigut extret dels apunts de les assignatures mencionades anteriorment.

Definició A.1. *Espai mostral Ω: Conjunt de tots els possibles resultats d'un experiment aleatori.*

Definició A.2. *Un esdeveniment és qualsevol subconjunt de l'espai mostral Ω, quan aquest és finit o infinit numerable. En aquests casos, el conjunt d'esdeveniments és el conjunt $\mathcal{P}(\Omega)$ de les parts de Ω.*

Definició A.3. *Donat un conjunt Ω (espai mostral), una col·lecció $\mathcal{F} \subset \mathcal{P}(\Omega)$ de subconjunts de Ω es diu que és una σ-àlgebra d'esdeveniments si:*

- $\Omega \in \mathcal{F}$,
- *Si* $A \in \mathcal{F}$ *aleshores* $A^c \in \mathcal{F}$,
- *Si* $A = \{A_n, n \in \mathbb{N}\} \subset \mathcal{F}$ *aleshores* $\cup_{n\in\mathbb{N}} A_n \in \mathcal{F}$.

Definició A.4. *La σ-àlgebra de Borel a $\mathbb{R}$ és la σ-àlgebra $\mathscr{B} \subset P(\mathbb{R})$ generada pels conjunts oberts de $\mathbb{R}$.*

Definició A.5. *Una variable aleatòria en un espai de probabilitat $(\Omega, \mathcal{F}, P)$ és una aplicació $X : \Omega \to \mathbb{R}$, que compleix que per tot $B \in \mathscr{B}$, l'antiimatge $X^{-1}(B) \in \mathcal{F}$.*

Definició A.6. *Donada X, v.a. en un espai de probabilitat $(\Omega, \mathcal{F}, P)$, la funció de distribució de probabilitat (cdf) de X és la funció $F_X : \mathbb{R} \to [0, 1]$ definida per:*

$$F_x(x) = P\{X \leq x\} = P(X \in (-\infty, x]) = P\left(X^{-1}(-\infty, x]\right).$$

Definició A.7. *Sigui F, cdf d'una v.a. X, diem que és absolutament contínua si és la integral d'una altra funció f,*

$$F(x) = \int_{-\infty}^{x} f(t)dt, \qquad x \in \mathbb{R},$$

on $f = F'$ és la densitat de probabilitat (pdf) de la v.a. X. Es diu que la v.a. X és absolutament contínua.

Proposició A.8. *Si F és una cdf absolutament continua, i $f : \mathbb{R} \to \mathbb{R}_+$ és la seva pdf, f satisfà:*

- *f és no negativa,*
- *f és integrable Riemann en $\mathbb{R}$,*
- *$\int_{-\infty}^{\infty} f(x)dx = 1$.*

Definició A.9. *Una v.a. X té distribució normal o gaussiana, amb paràmetres $\mu \in \mathbb{R}$ (mitjana) i $\sigma^2 \in \mathbb{R}_+$(variància, o desviació estàndard σ), si és absolutament contínua, amb pdf:*

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, x \in \mathbb{R}.$$

Notació: $X \sim N(\mu, \sigma^2)$.

Proposició A.10. *Si $X \sim N(\mu, \sigma^2)$, la variable*

$$Z \equiv \frac{X-\mu}{\sigma} \sim N(0,1)$$

Demostració: Tenim $X \sim N(\mu, \sigma^2)$. Sigui $Z \equiv \frac{X-\mu}{\sigma}$, donats $a, b \in \mathbb{R}$:

$$P(a < Z \leq b) = P(a < \frac{X-\mu}{\sigma} \leq b) = P(\mu + a\sigma < X \leq \mu + b\sigma)$$

$$= \int_{\mu+a\sigma}^{\mu+b\sigma} \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} dx = \int_a^b \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{u^2}{2}\right\} du.$$

□

De manera similar, tenim el següent resultat:

Proposició A.11. *Si $Z \sim N(0,1)$, la variable*

$$X \equiv \mu + \sigma Z \sim N(\mu, \sigma^2).$$

Definició A.12. *Si X és una v.a. definida a un espai de probabilitat $(\Omega, \mathcal{F}, P)$ i integrable respecte P, aleshores l'esperança matemàtica de X denotada com $E(X)$ es defineix com:*

$$E(X) = \int_\Omega X dP,$$

on la integral és una integral de Lebesgue respecte a la mesura de probabilitat P.

La condició perquè l'esperança existeixi és $E(|X|) < \infty$

Definició A.13. *La variància d'una v.a. X és una mesura de dispersió de X respecte a la seva mitjana $E(X)$. Es defineix com:*

$$Var(X) = E\left[(X - E(X))^2\right],$$

on suposem que $E(X^2) < \infty$.

Definició A.14. *Sigui X una v.a. amb esperança matemàtica $E(X) = \mu$. La desviació típica de X ve definida com:*

$$\sigma = \sqrt{E\left[(X-\mu)^2\right]} = \sqrt{E(X^2) - \mu^2},$$

on $E(X^2) < \infty$.

Així doncs, la desviació típica mostra quanta dispersió hi ha respecte a la mitjana.

Definició A.15. *La covariància de dues v.a. X i Y es defineix com:*

$$Cov(X,Y) = E([X - E(X)]\,[Y - E(Y)]).$$

Per tant, la covariància mesura la dispersió conjunta entre dues variables aleatòries.

Definició A.16. *El coeficient de correlació entre dues variables aleatòries X i Y ve definit per:*

$$R = \frac{\sigma_{XY}}{\sigma_X \cdot \sigma_Y},$$

on σ_{XY} és la covariància de (X,Y) i σ_X i σ_Y les desviacions típiques de les distribucions marginals.

Propietats del coeficient de correlació [14]:

- El valor del coeficient de correlació varia a l'interval [-1,1].
- Si R = 1, hi ha una correlació positiva perfecta. El coeficient indica una dependència total entre les dues variables, quan una d'elles augmenta, l'altre també ho fa en proporció constant.
- Si $0 < R < 1$ hi ha una correlació positiva.
- Si R = 0, no existeix relació lineal. Això no implica, però, que les dues variables siguin independents.

- Si $-1 < R < 0$, hi ha una correlació negativa.
- Si R = -1, hi ha una correlació negativa perfecta. El coeficient indica una dependència total entre les dues variables, quan una d'elles augmenta, l'altre disminueix en proporció constant.

A.2 *Boxplot*

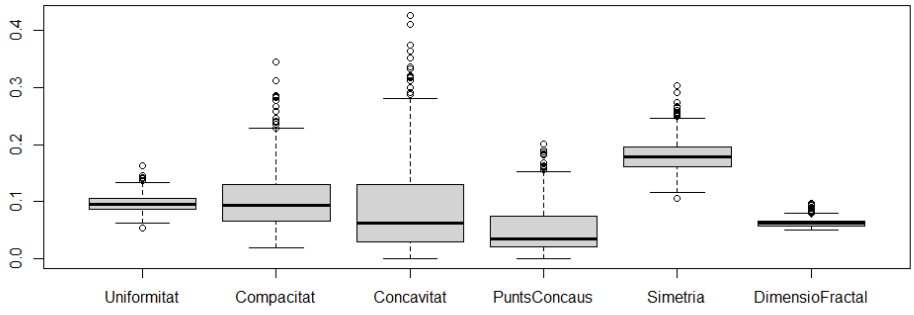

Figura 1.12: *Boxplot* realitzat mitjançant el programa R de les dades estudiades.

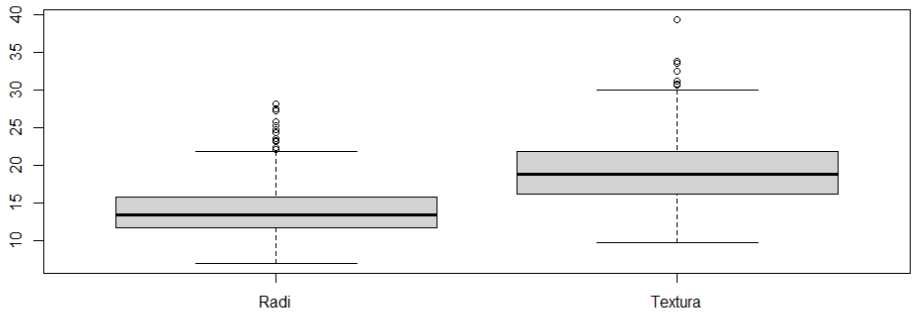

Figura 1.13: *Boxplot* realitzat mitjançant el programa R de les dades estudiades.

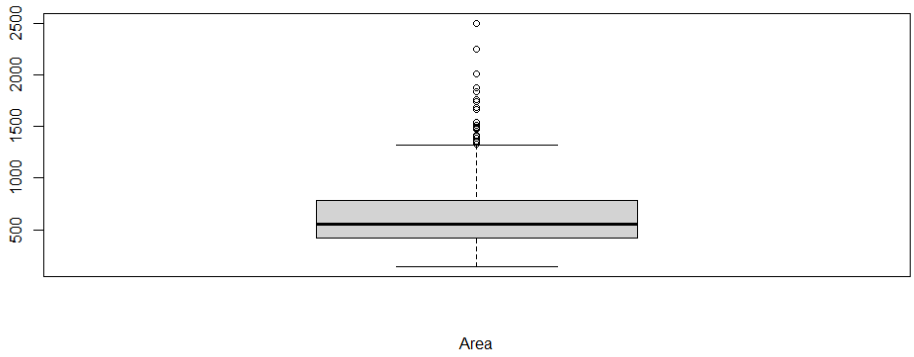

Figura 1.14: *Boxplot* realitzat mitjançant el programa R de les dades estudiades.

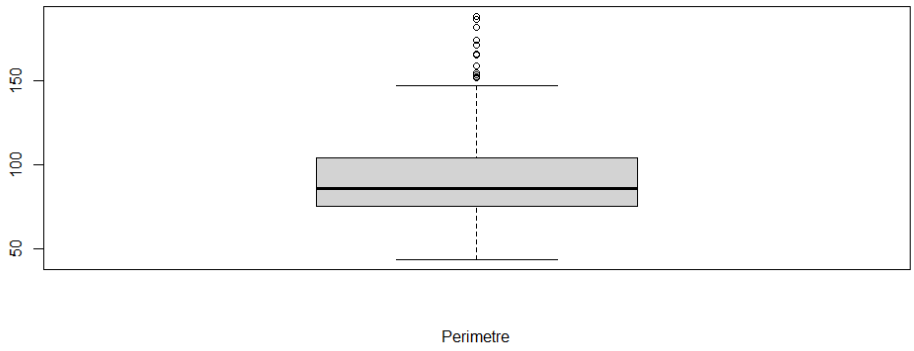

Figura 1.15: *Boxplot* realitzat mitjançant el programa R de les dades estudiades.

A.3 Característiques estadístiques bàsiques de les dades

```
> summary(dades[-1])
      Radi            Textura        Perimetre           Area         Uniformitat
    Compacitat
 Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5   Min.   :0.05263   Min.
    :0.01938
 1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3   1st Qu.:0.08637   1st Qu
    .:0.06492
 Median :13.370   Median :18.84   Median : 86.24   Median : 551.1   Median :0.09587   Median
    :0.09263
 Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9   Mean   :0.09636   Mean
    :0.10434
```

```
 3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Qu
    .:0.13040
 Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0   Max.   :0.16340   Max.
    :0.34540
   Concavitat       PuntsConcaus        Simetria      DimensioFractal
 Min.   :0.00000   Min.   :0.00000   Min.   :0.1060   Min.   :0.04996
 1st Qu.:0.02956   1st Qu.:0.02031   1st Qu.:0.1619   1st Qu.:0.05770
 Median :0.06154   Median :0.03350   Median :0.1792   Median :0.06154
 Mean   :0.08880   Mean   :0.04892   Mean   :0.1812   Mean   :0.06280
 3rd Qu.:0.13070   3rd Qu.:0.07400   3rd Qu.:0.1957   3rd Qu.:0.06612
 Max.   :0.42680   Max.   :0.20120   Max.   :0.3040   Max.   :0.09744
```

A.4 P-valors obtinguts de l'aplicació del test de normalitat de Shapiro-Wilk

```
> shapiro.wilk.test.1
$Radi
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.99665, p-value = 0.668
$Textura
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.94417, p-value = 2.385e-10
$Perimetre
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.9971, p-value = 0.7795
$Area
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.99064, p-value = 0.02278
$Uniformitat
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.97551, p-value = 9.507e-06
$Compacitat
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.92587, p-value = 2.644e-12
```

```
$Concavitat
    Shapiro-Wilk normality test
data:  newX[, i]
W = 0.73728, p-value < 2.2e-16
$PuntsConcaus
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.94672, p-value = 4.804e-10
$Simetria
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.97409, p-value = 5.182e-06
$DimensioFractal
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.887, p-value = 1.446e-15


> shapiro.wilk.test.2
$Radi
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.97766, p-value = 0.001895
$Textura
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.96909, p-value = 0.0001342
$Perimetre
    Shapiro-Wilk normality test
data:  newX[, i]
W = 0.97302, p-value = 0.0004326
$Area
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.93326, p-value = 2.97e-08
$Uniformitat
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.98469, p-value = 0.0215
$Compacitat
  Shapiro-Wilk normality test
```

```
data:  newX[, i]
W = 0.95743, p-value = 5.858e-06
$Concavitat
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.95293, p-value = 1.967e-06
$PuntsConcaus
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.96133, p-value = 1.583e-05
$Simetria
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.96488, p-value = 4.084e-05
$DimensioFractal
  Shapiro-Wilk normality test
data:  newX[, i]
W = 0.95076, p-value = 1.185e-06
```

A.5 Gràfic d'un anàlisi de components principals considerant tres components

A continuació veurem una ampliació de la secció 3.2 on hi podem trobar la realització d'una PCA. En la secció anterior, mitjançant el criteri del percentatge, s'escullen dues components principals, no obstant això, la quantitat òptima de components que hem d'escollir és un problema el qual, de moment, no té una resposta única. És per això que, a continuació veurem el gràfic que obtindríem si haguéssim escollit tres components principals, per a la representació hem fet servir la funció *scatterplot3d*. Primer de tot veiem el codi utilitzat

```
library("scatterplot3d")
shapes = c(16, 17)
#El valor dades esta inicialitzat de la mateixa manera que al llarg de tot el treball
shapes <- shapes[as.numeric(dades$Diagnostic)]
colors <- c("#999999", "#E69F00")
colors <- colors[as.numeric(dades$Diagnostic)]
scatterplot3d(components_principals[,1:3], xlab = "1a component", ylab = "2a component ",
    zlab = "3a component ", pch = shapes, color=colors,angle = 10)
```

El resultat obtingut és el següent (veure Figura 1.16).

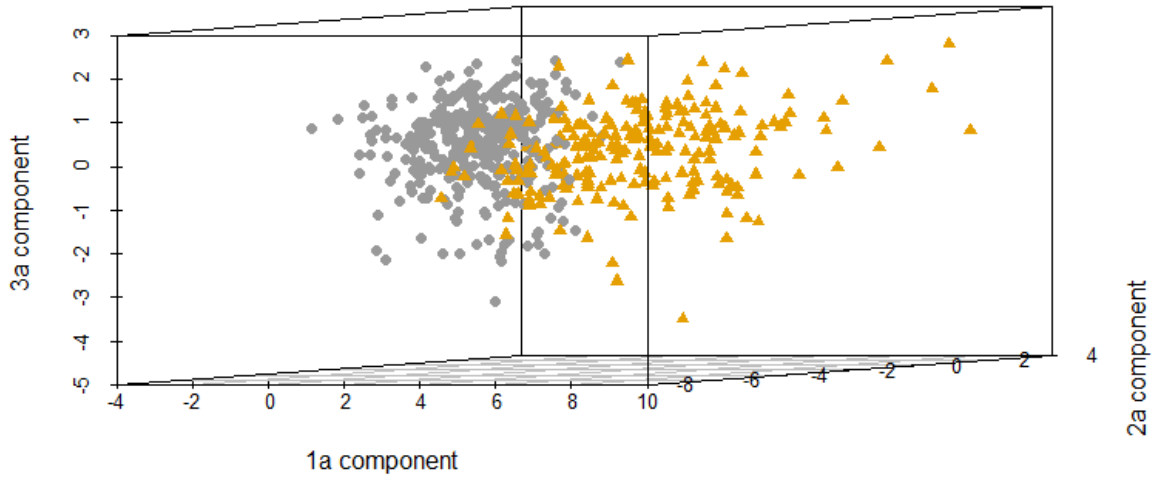

Figura 1.16: Gràfica 3D després d'haver realitzat una PCA a les dades. Gràfica realitzada amb el programa RStudio.

Com ja hem vist anteriorment, la tercera component principal aporta, aproximadament, un 6% més de variabilitat, tot i això no veiem una separació clara entre benignes i malignes en aquesta component.

A.6 Creació de variables amb distribució normal molt correlacionades i poc correlacionades

A la secció 3.3 hem creat dades tals que les seves variables segueixin una distribució normal i a més a més que no estiguin correlacionades. A continuació mostrarem el codi necessari per a que les dades, amb la propietat anterior, estiguin poc correlacionades o molt correlacionades, així com la seva aplicació als algoritmes SVM i Naive Bayes.

```
X_lc <- list() # Matriu amb variables poc correlacionades
X_hc <- list() # Matriu amb variables molt correlacionades
n <- 569
# Desviacio estandard de les variables
sd <- 0.5
meanVec <- numeric(length = p)
p <- 10
```

```r
#Posem una llavor perque sempre doni els mateixos resultats
 set.seed(1)
sdDiag <- diag(rep(sd, p))
#Dades amb variables poc correlacionades
#Considerem una matriu tota de 0
corMat_lc <- diag(0, p)
#Prenem la matriu triangular inferior exceptuant la diagonal i li posem valors ente 0 i 0.5 (
    sera la correlacio entre les variables)
corMat_lc[lower.tri(corMat_lc, diag = FALSE)] <-
  runif(p[1]*(p[1] - 1)/2, min = 0, max = 0.5)
#Fem la matriu simetrica
corMat_lc[upper.tri(corMat_lc)] <- t(corMat_lc)[upper.tri(corMat_lc)]
#Multipliquem per la transposada perque sigui simetrica i definida positiva
corMat_lc <- corMat_lc%*%t(corMat_lc)
#Dividim perque els valors estiguin entre 0 i 1
corMat_lc <- corMat_lc/(2*max(corMat_lc))
corMat_lc
#Les matrius de correlacio tenen la diagonal tota de 1 per tant l'hi assignem
diag(corMat_lc) <- 1
#Busquem la matriu de covariancies a partir de la matriu de correlacio
Sigma_lc <- sdDiag%*%corMat_lc%*%sdDiag
#Dades amb variables molt correlacionades
#Farem els mateixos passos que anteriorment, pero ara prendrem els valors de la matriu
    triangular inferior entre 0.5 i 1 perque
#tinguin una major correlacio
corMat_hc <- diag(0, p)
corMat_hc[lower.tri(corMat_hc, diag = FALSE)] <-
  runif(p*(p - 1)/2, min = 0.5, max = 1)
corMat_hc[upper.tri(corMat_hc)] <- t(corMat_hc)[upper.tri(corMat_hc)]
corMat_hc <- corMat_hc%*%t(corMat_hc)
corMat_hc <- corMat_hc/max(corMat_hc)
diag(corMat_hc) <- 1
Sigma_hc <- sdDiag%*%corMat_hc%*%sdDiag
#Dades amb variables poc correlacionades
X_lc <- rmvnorm(n = n, mean = meanVec, sigma = Sigma_lc)
colnames(X_lc)<- c("Radi", "Textura", "Perimetre", "Area", "Uniformitat", "Compacitat", "
    Concavitat", "PuntsConcaus", "Simetria", "DimensioFractal")
#Dades amb variables molt correlacionades
X_hc <- rmvnorm(n = n, mean = meanVec, sigma = Sigma_hc)
colnames(X_hc)<- c("Radi", "Textura", "Perimetre", "Area", "Uniformitat", "Compacitat", "
    Concavitat", "PuntsConcaus", "Simetria", "DimensioFractal")
```

```r
#Aplicacio dels algoritmes per a dades poc correlacionades
#Fixem una llavor
set.seed(8)
indexes_partition <- createDataPartition(y = 1:dim(X_lc)[1], p = 0.66,list = FALSE)
#Dades d'entrenament
dades_train_X_lc<- X_lc[indexes_partition,]
#Dades de test
dades_test_X_lc <- X_lc[-indexes_partition,]
#Generem tambe les etiquetes de cada fila de dades
dades_train_labels_X_lc <- sample.int(2, dim(dades_train_X_lc)[1], replace = TRUE) - 1
dades_train_labels_X_lc <- ifelse(dades_train_labels_X_lc == 0, "M", "B")
library(e1071)
NB_model_X_lc <- naiveBayes(dades_train_X_lc, as.factor(dades_train_labels_X_lc))
NB_prediction_X_lc<- predict(NB_model_X_lc, dades_test_X_lc)
library(kernlab)
library(caret)
SVM_model_X_lc <- ksvm(as.factor(dades_train_labels_X_lc) ~ ., data = dades_train_X_lc ,
    kernel = "vanilladot")
SVM_prediction_X_lc<- predict(SVM_model_X_lc, dades_test_X_lc)
all.equal(NB_prediction_X_lc, SVM_prediction_X_lc)

#Aplicacio dels algoritmes per a dades molt correlacionades
set.seed(8)
indexes_partition <- createDataPartition(y = 1:dim(X_hc)[1], p = 0.66,list = FALSE)
#Dades d'entrenament
dades_train_X_hc<- X_hc[indexes_partition,]
#Dades de test
dades_test_X_hc <- X_hc[-indexes_partition,]
#Generem tambe les etiquetes de cada fila de dades
dades_train_labels_X_hc <- sample.int(2, dim(dades_train_X_hc)[1], replace = TRUE) - 1
dades_train_labels_X_hc <- ifelse(dades_train_labels_X_hc == 0, "M", "B")
library(e1071)
NB_model_X_hc <- naiveBayes(dades_train_X_hc, as.factor(dades_train_labels_X_hc))
NB_prediction_X_hc<- predict(NB_model_X_hc, dades_test_X_hc)
library(kernlab)
library(caret)
SVM_model_X_hc  <- ksvm(as.factor(dades_train_labels_X_hc) ~ ., data = dades_train_X_hc ,
    kernel = "vanilladot")
SVM_prediction_X_hc  <- predict(SVM_model_X_hc , dades_test_X_hc)
all.equal(NB_prediction_X_hc, SVM_prediction_X_hc)
```

Si apliquem els algoritmes de SVM i Naive Bayes per aquests dos tipus de dades obtenim que, en el cas de variables poc correlacionades, un 76.56% de les observacions es classifiquen de la mateixa manera entre els dos algoritmes, mentre que en el cas de variables molt correlacionades ho fan en un 62.5%. Així doncs, podem dir que els algoritmes treballen millor per a variables no correlacionades. Les raons específiques d'aquest comportament se'ns escapen de les mans, tot i això, s'indica per a possibles ampliacions del treball.

A.7 Altres algoritmes

A.7.1 Xarxes neuronals artificials

Les xarxes neuronals artificials s'utilitzen per a trobar patrons que són massa complexos perquè un humà ho pugui resoldre. Tal com diu el nom utilitzen una xarxa de neurones o nodes artificials per resoldre els problemes. En general consten de capes d'entrada i de sortida, així com una capa oculta que s'encarreguen de transformar l'entrada en alguna cosa que la capa de sortida pugui utilitzar. El procés que porta a terme cada node consisteix a ponderar el senyal de cada entrada d'acord amb la seva importància i sumar-les en el cos cel·lular. Finalment, el senyal es transmet d'acord amb una funció d'activació [19].

Fortaleses	**Mancances**
Són flexibles i es poden utilitzar tant per a problemes de regressió com de classificació. Qualsevol dada que es pugui convertir en numèrica es pot utilitzar per al model.	A nivell computacional és extremadament lent d'entrenar, en particular si la topologia de la xarxa és complexa.
Són bones per modelar amb dades no lineals amb un gran nombre d'entrades, per exemple imatges.	Depenen molt de les dades d'entrenament i per tant són molt propensos a sobreajustar les dades.
Pot modelar patrons molt complexos millor que quasi qualsevol altre algoritme.	Dona com a resultat un model complex de caixa negre que és difícil d'interpretar.
Fa poques suposicions sobre les relacions subjacents a les dades.	
Un cop entrenat, les prediccions són ràpides.	

A.7.2 Arbres de decisió

Els arbres de decisió són classificadors potents, que utilitzen una estructura d'arbre per modelar les relacions entre les característiques i els resultats potencials. El seu nom prové de la manera a partir del qual pot ser il·lustrat (comença amb un tronc ample que, si es segueix cap amunt, es divideix en branques cada vegada més estretes). Per tenir una idea ràpida, tot comença amb un node d'arrel, on després passa a través dels nodes de decisió que requereixen la presa de decisions basades en atributs donats. Aquests espais són els encarregats de dividir les branques que indiquen els possibles resultats. En cas que es pugui prendre una decisió final l'arbre s'acaba amb un node fulla, que denota l'acció a prendre [11].

Fortaleses	**Mancances**
Procés d'aprenentatge altament automàtic, el qual pot tractar tant amb característiques numèriques com nominals, així com amb dades que falten.	Pot tenir problemes per modelar algunes relacions degut a les dependències de les divisions d'eixos paral·lels.
Funciona bé en una gran majoria dels problemes.	És fàcil sobre-ajustar o desajustar el model.
Exclou característiques que no tenen importància.	Els arbres grans poden ser difícils d'interpretar.
Dona com a resultat un model que es pot interpretar sense una base matemàtica.	No hi ha cap garantia de retornar un arbre de decisió 100 % eficient.
Es pot utilitzar en conjunts de dades grans i petites.	Petits canvis en les dades d'entrenament poden resultar grans canvis en la lògica de decisió.
Més eficient que altres models complexos.	
No requereix de cap transformació de les característiques si es tracta de dades no lineals perquè els arbres de decisió no tenen en compte múltiples combinacions ponderades simultàniament.	
La normalització no és necessària a l'arbre de decisions.	

A.7.3 K-veí més proper (K-NN)

Per a més informació sobre aquest algoritme podeu consultar a [21] [18].

Fortaleses	**Mancances**
Simple i efectiu.	És un algoritme molt lent.
Té una fase d'entrenament ràpida.	Fase de classificació bastant lenta.
No fa suposicions sobre la distribució de les dades subjacents. Crucial en el cas de dades no lineals.	Les dades desequilibrades causen problemes i molt sensible a valors atípics.
Molt fàcil d'implementar per a problemes de múltiples classes.	No produeix cap model, senzillament entén com funcionen les característiques a cada moment.
Permet que l'algoritme respongui ràpidament als canvis d'entrada durant l'ús.	Funciona bé amb un nombre reduït de variables d'entrada, però no a mesura que creix el nombre de variables.
Alta precisió, no és necessari que el comparem amb altres models d'aprenentatge automàtic.	S'ha de triar de manera òptima el valor de K.
Evoluciona constantment ja que és un aprenentatge que bàsicament compara els nous problemes amb les instàncies que s'han vist durant l'entrenament i que han estat emmagatzemades a la memòria.	Requereix de molta memòria, ja que necessita emmagatzemar totes les dades d'entrenament

A.8 Codi per la millora del rendiment de l'algoritme Support Vector Machine

```
#Construccio del model
SVM_model_RBF <- ksvm(as.factor(dades_train_labels) ~ ., data = dades_train , kernel = "
    rbfdot")
#Prediccio del conjunt de dades
SVM_prediction_RBF<- predict(SVM_model_RBF, dades_test)
#Calculem la matriu de confusio
mat_conf_RBF<- confusionMatrix(SVM_prediction_RBF, as.factor(dades_test_labels), positive = "
    M")
          Reference
Prediction   B   M
         B 107   6
         M   9  7
Sensitivity : 0.9211
```

```
#Escollim quins seran els valors dels costs
costs <- c(1,2,3,4,5,6)
#Prediccions de l'algoritme
SVM_model <- list()
SVM_prediction <- list()
j <- 1
for(C in costs){
  #Per cada valor de cost, guardem les diferents prediccions, tant pel cas normalitzat com el
     no normalitzat
  SVM_model[[j]] <- ksvm(as.factor(dades_train_labels) ~ ., data = dades_train , kernel = "
    vanilladot", C = C)
  j <- j + 1
}
SVM_evaluation <- data.frame(costs, accuracy = NA, accuracyL = NA,
                             accuracyU = NA, kappa = NA, Sensitivity= NA, Specificity= NA )

for(j in 1:length(costs)){
  #Prediccions
  SVM_prediction[[j]] <- predict(SVM_model[[j]], dades_test)
  #Matriu de confusio
  mat_conf <- confusionMatrix(SVM_prediction[[j]], as.factor(dades_test_labels))
  overall_measures <- round(mat_conf$overall[c("Accuracy", "AccuracyLower",
                                               "AccuracyUpper", "Kappa")], 3)
```

```
  classes_measures <- round(mat_conf$byClass[c("Sensitivity", "Specificity")], 3)
  SVM_evaluation[j, 2:5] <- overall_measures
  SVM_evaluation[j, 6:7] <- classes_measures

}
#Observem resultats
  costs accuracy accuracyL accuracyU kappa Sensitivity Specificity
1     1    0.896     0.844     0.935 0.784       0.897       0.895
2     2    0.901     0.850     0.939 0.795       0.897       0.908
3     3    0.901     0.850     0.939 0.795       0.897       0.908
4     4    0.901     0.850     0.939 0.795       0.897       0.908
5     5    0.901     0.850     0.939 0.795       0.897       0.908
6     6    0.901     0.850     0.939 0.795       0.897       0.908
7     7    0.901     0.850     0.939 0.795       0.897       0.908
8     8    0.901     0.850     0.939 0.795       0.897       0.908
9     9    0.901     0.850     0.939 0.795       0.897       0.908
```

A.9 Codi per la millora del rendiment de l'algoritme Naive Bayes

```
 #Escollim quins seran els valors de laplace
laplace_values <-c(0,0.25,0.5,0.75,1)
#Prediccions de l'algoritme
NB_model<- list()
NB_prediction <- list()
j<- 1
for(l in laplace_values){
  #Per cada valor de laplace, guardem les diferents prediccions, tant pel cas normalitzat com
     el no normalitzat
  NB_model[[j]] <- naiveBayes(dades_train, as.factor(dades_train_labels), laplace = l)

  j <- j + 1
}
#Avaluacio de l'algoritme
bayes_evaluation <- data.frame(laplace_values, accuracy = NA, accuracyL = NA,accuracyU = NA,
    kappa = NA, Sensitivity= NA, Specificity= NA )

for(j in 1:length(laplace_values)){
  #Prediccions
  NB_prediction[[j]] <- predict(NB_model[[j]], dades_test)
  #Matrius de confusio
```

```
  mat_conf <- confusionMatrix(NB_prediction[[j]], as.factor(dades_test_labels))
  overall_measures <- round(mat_conf$overall[c("Accuracy", "AccuracyLower","AccuracyUpper", "
    Kappa")], 3)
  classes_measures <- round(mat_conf$byClass[c("Sensitivity", "Specificity")], 3)
  bayes_evaluation[j, 2:5] <- overall_measures
  bayes_evaluation[j, 6:7] <- classes_measures

}
#Observem resultats
bayes_evaluation
  laplace_values accuracy accuracyL accuracyU kappa Sensitivity Specificity
1           0.00    0.906     0.856     0.943 0.803       0.931       0.868
2           0.25    0.906     0.856     0.943 0.803       0.931       0.868
3           0.50    0.906     0.856     0.943 0.803       0.931       0.868
4           0.75    0.906     0.856     0.943 0.803       0.931       0.868
5           1.00    0.906     0.856     0.943 0.803       0.931       0.868
```

A.10 Codi per l'algoritme K-NN

```
library(class)
KNN_prediction_S <- knn(train = dades_train, test = dades_test, cl= as.factor(dades_train_
    labels), k = sqrt(length(dades_train[,1])))
#Matriu de confusio
mat_conf <- confusionMatrix(KNN_prediction_S, as.factor(dades_test_labels))
mat_conf
          Reference
Prediction   B   M
         B 110  16
         M   6  60
Sensitivity : 0.9483
```

Referències

[1] Kristin P. Bennett i Colin Campbell. "Support Vector Machines: Hype or Hallelujah?" A: *SIGKDD Explor. Newsl.* 2.2 (2000). DOI: 10.1145/380995.380999. URL: https://doi.org/10.1145/380995.380999.

[2] Kurt Bryan. *The Riesz Representation Theorem*. URL: https://math.jhu.edu/~lindblad/632/riesz.pdf.

[3] Andrea Caponnetto. *Reproducing Kernel Hilbert Spaces*. 2006. URL: http://www.mit.edu/~9.520/spring06/Classes/class03.pdf.

[4] Ronan Collobert, Samy Bengio i C. Williamson. "SVMTorch: Support Vector Machines for Large-Scale Regression Problems". A: *Journal of Machine Learning Research* 1 (març de 2001). URL: https://www.jmlr.org/papers/volume1/collobert01a/collobert01a.pdf.

[5] A. G. Constantine. "The Distribution of Hotelling's Generalised T0 2". A: *The Annals of Mathematical Statistics* 37.1 (1966), pàg. 215 - 225. URL: http://www.jstor.org/stable/2238701.

[6] C.M. Cuadras. *Nuevos métodos de análisis multivariante*. Barcelona: CMC Editions, 2018.

[7] Khalil M. El Hindi Diab M. Diab. "Using differential evolution for fine tuning naïve Bayesian classifiers and its application for text classification". A: *ELSEVIER* 54 (2017), pàg. 183 - 199. DOI: 10.1016/j.asoc.2016.12.043. URL: https://doi.org/10.1016/j.asoc.2016.12.043.

[8] Cortina C; Turon G; Stork D; Hernando-Momblona X; Sevillano M; Aguilera M; Tosi S; Merlos-Suárez A; Stephan-Otto Attolini C; Sancho E; Batlle E. "A genome editing approach to study cancer stem cells in human tumors". A: *Embo Molecular Medicine* 9.7 (2017), pàg. 869 - 879. DOI: https://doi.org/10.15252/emmm.201707550.

[9] ML Eaton. *The Wishart Distribution*. 2007. URL: https://www.stat.pitt.edu/sungkyu/course/2221Fall13/lec2.pdf.

[10] Minitab Blog Editor. *Entendiendo las Pruebas de Hipótesis: niveles de Significancia (Alfa) y Valores P en Estadística*. 2019. URL: https://blog.minitab.com/es/entendiendo-las-pruebas-de-hipotesis-niveles-de-significancia-alfa-y-valores-p-en-estadistica.

[11] EDUCBA. *Decision Tree Advantages and Disadvantages*. 2020. URL: https://www.educba.com/decision-tree-advantages-and-disadvantages/.

[12] Statistical Odds Ends. *What is a reproducing kernel Hilbert space (RKHS)?* 2019. URL: https://statisticaloddsandends.wordpress.com/2019/09/13/what-is-a-reproducing-kernel-hilbert-space-rkhs/.

[13] Paul Evangelista, M. Embrechts i Boleslaw Szymanski. "Some Properties of the Gaussian Kernel for One Class Learning". A: set. de 2007, pàg. 269 - 278. ISBN: 978-3-540-74689-8. DOI: 10.1007/978-3-540-74690-4_28.

[14] Jason Fernando. *Correlation Coefficient*. 2021. URL: https://www.investopedia.com/terms/c/correlationcoefficient.asp.

[15] Data Flair. *Real-Life Applications of SVM (Support Vector Machines)*. 2022. URL: https://data-flair.training/blogs/applications-of-svm/.

[16] Cristianini N. Friess T.-T. i Campbell C. "The kernel adatron algorithm: a fast and simple learning procedure for support vector machines." A: *Morgan Kaufman Publishers* (1998), pàg. 188 - 196. URL: https://www.researchgate.net/publication/2610012_The_Kernel-Adatron_Algorithm_a_Fast_and_Simple_Learning_Procedure_for_Support_Vector_Machines.

[17] Fractal Fundation. *Fractal Dimension*. URL: https://fractalfoundation.org/OFC/OFC-10-4.html.

[18] Genesis. *Pros and Cons of K-Nearest Neighbors*. 2018. URL: https://www.fromthegenesis.com/pros-and-cons-of-k-nearest-neighbors/.

[19] Balaji Venkateswaran Giuseppe Ciaburro. *Neural Networks with R*. Packt, 2017.

[20] Philipp Hennig i Martin Kiefel. "Quasi-Newton methods: a new direction." A: *The Journal of Machine Learning Research* 14 (2013), pàg. 843 - 865. URL: https://arxiv.org/ftp/arxiv/papers/1206/1206.4602.pdf.

[21] M. R. Gray J. M. Keller i J. A. Givens. "A fuzzy K-nearest neighbor algorithm". A: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-15.4 (1985), pàg. 580 - 585.

[22] Zakaria Jaadi. *A Step-by-Step Explanation of Principal Component Analysis (PCA).* 2021. URL: https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

[23] Noreen Jamil, Xuemei Chen i Alexander Cloninger. "Hildreth's algorithm with applications to soft constraints for user interface layout". A: *Journal of Computational and Applied Mathematics* 288 (2015), pàg. 193 - 202. DOI: https://doi.org/10.1016/j.cam.2015.04.014. URL: https://www.sciencedirect.com/science/article/pii/S0377042715002320.

[24] T. Joachims. *SVMlight Support Vector Machine.* 2008. URL: https://www.researchgate.net/profile/Thorsten-Joachims/publication/243763293_SVMLight_Support_Vector_Machine/links/5b0eb5c2a6fdcc80995ac3d5/SVMLight-Support-Vector-Machine.pdf.

[25] Naresh Kumar. *Advantages and Disadvantages of SVM (Support Vector Machine) in Machine Learning.* 2019. URL: http://theprofessionalspoint.blogspot.com/2019/03/advantages-and-disadvantages-of-svm.html.

[26] Brett Lantz. *Machine Learning with R.* Birmingham: PACKT, 2013.

[27] S. Lauritzen. *Wishart and inverse Wishart distributions. [LATEX Slides].* 2009. URL: https://www.stats.ox.ac.uk/~steffen/teaching/bs2HT9/inverse.pdf.

[28] Dhahri H. Al Maghayreh E. Mahmood A. Elkilani W. Faisal Nagi M. "Automated Breast Cancer Diagnosis Based on Machine Learning Algorithms." A: *Journal of healthcare engineering* (2019). DOI: https://doi.org/10.1155/2019/4253641.

[29] O.L. Mangasarian i D. R. Musicant. *LSVM Software: Active Set Support Vector Machine Classification Software.* 2000.

[30] L. Marklund i L. Hammarstedt. "Impact of HPV in oropharyngeal cancer". A: *Journal of oncology* (2010).

[31] David Nualart i Marta Sanz. *Curs de probabilitats.* Facultat de Matemàtiques. Universitat de Barcelona, 1990, pàg. 7 - 9.

[32] Crue Math. *Euclidean Distance Formula*. URL: https://www.cuemath.com/euclidean-distance-formula/.

[33] Wolfram MathWoeld. *Convex Hull*. 2020. URL: https://mathworld.wolfram.com/ConvexHull.html.

[34] Pich O; Muiños F; Lolkema M; Steeghs N; Gonzalez-Perez A; Lopez-Bigas N. "The mutational footprints of cancer therapies". A: *Nature Genetics* 51.12 (2017), pàg. 1732 - 1740. DOI: https://doi.org/10.1038/s41588-019-0525-5.

[35] Jaume Ordi. *Anatomía patológica general*. Universitat de Barcelona, 2012, pàg. 369 - 393. ISBN: 978-84-475-3561-3.

[36] F. Paulin i A. Santhakumaran. "Extracting rules from feed forward neural networks for diagnosing breast cancer". A: *Artificial Intelligent Systems and Machine Learning* 1.4 (2009), pàg. 143 - 146.

[37] T. Pham-Gia. "Exact distribution of the generalized Wilks's statistic and applications". A: *Journal of Multivariate Analysis* 99.8 (2008), pàg. 1698 - 1716. DOI: https://doi.org/10.1016/j.jmva.2008.01.021. URL: https://www.sciencedirect.com/science/article/pii/S0047259X08000249.

[38] Elmore J. G. Wells C. K. Lee C. H. Howard D. H. Feinstein A. R. "Variability in radiologists' interpretations of mammograms". A: *New England Journal of Medicine* 331.22 (1994), pàg. 1493 - 1499.

[39] RDocumentation. *Gaussian Naive Bayes Classifier*. URL: https://www.rdocumentation.org/packages/naivebayes/versions/0.9.7/topics/gaussian_naive_bayes.

[40] Seb. *Gram Schmidt Process: A Brief Explanation*. URL: https://programmathically.com/gram-schmidt-process/.

[41] J. Tacq. *International Encyclopedia of Education (Third Edition)*. Elsevier, 2010, pàg. 332 - 338. ISBN: 978-0-08-044894-7. DOI: https://doi.org/10.1016/B978-0-08-044894-7.01351-8. URL: https://www.sciencedirect.com/science/article/pii/B9780080448947013518.

[42] J. Tacq. "Multivariate Normal Distribution". A: *International Encyclopedia of Education (Third Edition)*. Ed. de Penelope Peterson, Eva Baker i Barry McGaw. Third

Edition. Oxford: Elsevier, 2010, pàg. 332 - 338. ISBN: 978-0-08-044894-7. DOI: https://doi.org/10.1016/B978-0-08-044894-7.01351-8. URL: https://www.sciencedirect.com/science/article/pii/B9780080448947013518.

[43] Abhishek Jha Vladimir Lyashenko. *Cross-Validation in Machine Learning: How to Do It Right*. 2022. URL: https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right.

[44] William H. Wolberg W. Nick Street i O.L.Mangasarian. *Nuclear feature extraction for breast tumor diagnosis*. URL: https://minds.wisconsin.edu/bitstream/handle/1793/59692/TR1131.pdf;jsessionid=8B01049207C1CE16C3A1AACF04DFC848?sequence=1.

[45] Guillermo Westreicher. *Matriz de datos*. 2021. URL: https://economipedia.com/definiciones/matriz-de-datos.html.

[46] D.H. Wolpert i W.G. Macready. "No free lunch theorems for optimization". A: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pàg. 67 - 82. DOI: 10.1109/4235.585893.