USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA
FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

Métodos de Clasificación con datos obtidos mediante LiDAR

Andrea Blanco Pérez

2021-2022

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

Métodos de Clasificación con datos obtidos mediante LiDAR

Andrea Blanco Pérez

Xullo, 2022

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Traballo proposto

Área de Coñecemento: Estatística e Investigación de Operacións
Título: Métodos de Clasificación con datos obtidos mediante LiDAR
Breve descrición do contido
Mediante procedementos de LiDAR aéreo pódese obter unha representación tridimensional dos obxectos dunha área concreta, ademais doutros parámetros relacionados con estes puntos. O obxectivo destes procedementos é determinar o tipo de combustible de cada píxel para zonas non determinadas a partires dun conxunto de entrenamento no que se poden aplicar os procedementos de clasificación usuais. Neste TFG planease a descrición e comparativa dos métodos de clasificación habituais aplicados a estes datos.
Recomendacións
Outras observacións

Índice

Resumo **VIII**

Introdución **XI**

0.1. Filtrado e preprocesado de datos XV

1. Primeira aproximación á aprendizaxe supervisada **1**

1.1. Introdución á aprendizaxe estatística 1

1.2. Aprendizaxe supervisada 2

1.2.1. Validación cruzada 4

1.2.2. Tipos de aprendizaxe supervisada 7

2. Métodos de clasificación clásicos **11**

2.1. Clasificador de Bayes 12

2.1.1. Estimación da función de densidade 14

2.1.2. Clasificador de Bayes inxenuo 15

2.2. K Puntos Próximos (KNN) 16

2.2.1. Determinación da distancia métrica 17

2.2.2. Determinación do valor de $\mathcal{K}$ 22

3. Modelos Aditivos Xeneralizados **25**

3.1. Introdución aos GAMs 25

3.2. Consideracións para a estimación de GAMs 27
3.2.1. Representación das funcións suaves 28
3.2.2. Estimación do modelo 28
3.3. Aplicación a problemas de clasificación 32
3.3.1. One Versus One 32
3.3.2. One Versus All 33

4. Aplicación dos métodos de clasificación 35
4.1. Particionado dos datos 35
4.2. Métricas de rendemento 36
4.2.1. Métricas para clasificación binaria 36
4.2.2. Adaptación á clasificación multiclase 37
4.3. Resultados e comparativa 40
4.4. Exemplos de aplicación 43

A. ANEXO I: Código R 45
A.1. Particionado de datos 45
A.2. Naive Bayes 46
A.3. K Veciños Próximos 48
A.4. GAM 49
A.5. Clasificador aleatorio 52

Bibliografía 53

Resumo

O considerable aumento da aparición e severidade de incendios forestais nas últimas decadas, supuxo o inicio de investigacións orientadas a evitar e reducir o seu impacto e aparecemento. Neste marco, xorde o modelo *Prometheus* como un capaz de resumir de forma representativa a distribución das masas forestais. Este proceso está respaldado pola tecnoloxía LiDAR, que é capaz de obter puntos tridimensionais cunha gran precisión e facilidade para percorrer áreas amplas.

Neste traballo proponse a revisión e comparación de algoritmos de clasificación aplicados ao etiquetado de masas forestais seguindo o modelo *Prometheus*. Deste xeito, realizarase unha introdución á aprendizaxe estatística e de forma máis concreta á clasificación. Tomando o anterior como base, presentaranse distintos métodos aplicables ao problema de estudo, indicando o seu marco teórico e características. Concretamente, estudarase o método de Bayes, KNN e Modelos Lineais Xeneralizados para a súa aplicación ao anterior problema.

Na última parte deste traballo, introduciranse as métricas para medir o seu rendemento e analizaranse os resultados obtidos tras a súa aplicación. Deste xeito será posible coñecer a súa eficacia sobre nubes de puntos reais e comparar o seu rendemento.

Abstract

In recent decades the increase of the appearance and severity of fires led to the outset of several researches for avoiding and reducing its impact. In this situation, the *Prometheus* model emerge as one capable of summarizing the distribution of the vegetation in forests in a representative way. This process is supported by the LiDAR technology, which is capable of obtaining three-dimensional points with high precision and easiness for covering vast areas.

The purpose of this project is the revision and comparison of vegetation classification algo-

rithms following the *Prometheus* model. Primarily, an introduction to statistical learning and, particularly, to classification will be made. Taking into account this, several methods will be presented, specifying its theoretical context and main characteristics. Specifically, Bayes method, KNN and Generalized Additive Models will be studied for its application to this problem.

In the last part of this project, some performance metrics will be introduced for measuring and analysing the results of the classifiers. All of this will make possible to evaluate its effectiveness in real cloud points and compare its performance.

Introdución

Na actualidade os incendios forestais constitúen un verdadeiro problema para diversos países europeos, entre os que se encontra España. O seu clima e as súas características medioambientais convírteno nun dos cinco países de Europa con máis incendios forestais, concentrándose a súa maioría no noroeste da península, Novo et al. (2020). Adicionalmente, nas últimas décadas a súa aparición e severidade aumentou considerablemente, o que produciu profundos danos en ecosistemas forestais.

Co obxectivo de desenvolver estratexias de prevención e extinción de incendios, nos últimos anos leváronse a cabo diversos estudos relacionados coa distribución e clasificación das masas forestais. Estes permiten coñecer con maior precisión o composición da vexetación e, deste xeito, anticipar riscos potenciais de incendios, a súa taxa de propagación ou a súa severidade, entre outros, García-Cimarras et al. (2021).

Ante esta situación xorde o modelo *Prometheus* (Arroyo et al. (2008)), un sistema desenvolvido por investigadores europeos que pretende resumir e representar toda esta información. Este está destinado orixinalmente a ecosistemas mediterráneos e propón sete tipos de combustible posibles cos que clasificar as masas forestais. Cada un destes tipos ou clases engloba un conxunto de características que deben cumprir as masas forestais. Aínda que o criterio fundamental desta clasificación é o tipo e altura dos elementos estudados, tamén se ten en conta a densidade e distribución da vexetación nalgún dos tipos de combustible (Figura 1).

Cada un dos tipos de combustible ou *fuel* defínense como segue:

- **Tipo 1**. Terreo no que predomina a vexetación herbácea, fundamentalmente son pradeiras ou zonas de cultivo (vexetación herbácea $> 50\,\%$).

- **Tipo 2**. Comprende terras con arbustos baixos (de entre 30 e 60 cm de altura, $>60\,\%$) e cunha alta porcentaxe de vexetación herbácea (30-40 %).

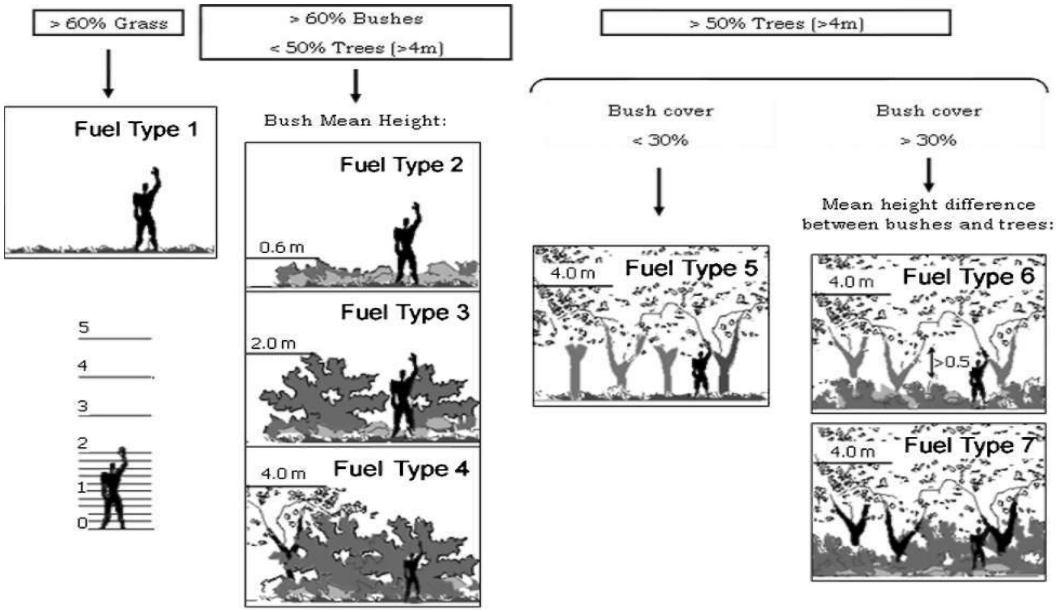

Figura 1: Descrición dos tipos de combustible do modelo Prometheus.

- **Tipo 3**. Terreos con arbustos de altura mediana-alta (>60 %, de entre 0,6 e 2 metros de altura) e árbores novas resultado da reforestación ou rexeneración natural.
- **Tipo 4**. Área con arbustos altos (>60 %, de entre 2 e 4 metros) e árbores novas.
- **Tipo 5**. Zona boscosa cuxo chan ten pouca vexetación e é baixa. As árbores son predominantes (>50 %) e máis altas ca nos casos anteriores (4 metros). Pode haber arbustos pero en pouca cantidade (< 30 %).
- **Tipo 6**. Zona boscosa na que a distancia entre a parte baixa das copas das árbores e a parte alta da capa que cobre a superficie do chan é de máis de 0,5 metros. Na parte inferior do terreo hai fundamentalmente arbustos baixos (> 30 %) e vexetación herbácea. As árbores teñen unha altura superior a 4 metros e cobren máis do 50 % do terreo.
- **Tipo 7**. Zona boscosa na que a distancia entre a parte baixa das copas das árbores e a parte alta da capa que cobre a superficie do chan é menor de 0,5 metros. De novo as árbores teñen unha altura superior a 4 metros e cobren máis do 50 % do terreo. Os arbustos cobren unha superficie maior ao 30 %.

Observamos, polo tanto, que os catro primeiros tipos de combustible fundaméntanse na altura e no tipo de vexetación, mentres que os tres últimos dependen tamén da densidade e distribución da vexetación baixo o dosel arbóreo.

Polo tanto, o modelo *Prometheus* permite condensar e comprender a composición das masas forestais. Para iso é necesario obter certa información relativa a estas como a súa posición espacial

ou distribución. Inicialmente, con este propósito, empregáronse tecnoloxías como a *fotogrametría*, concretamente, a aérea. Esta baséase na interpretación de fotografías aéreas baixo unha serie de condicionantes cos que se obteñen as propiedades espaciais e xeométricas dos obxectos (Quirós Rosado (2015)).

A pesar do seu uso estendido, esta tecnoloxía presenta unha serie de inconvintes, como a incapacidade de determinar a distribución da vexetación baixo o dósel arbóreo. Por este motivo, outras técnicas baseadas en sensores remotos comezaron a empregarse, como é o caso de *Light Detection And Ranging* ou LiDAR.

A tecnoloxía LiDAR é un sistema de medición masiva de posicións de forma remota. Baséase nun sensor de varrido láser que vai emitindo pulsos e recollendo os retornos obtidos contra a superficie. Desta forma, é posible obter a representación tridimensional dos obxectos reflictidos (Zamora-Martínez (2017)).

Existen dúas formas de obter esta información, a través de *LiDAR terrestre* e *LiDAR aéreo*, sendo esta última na que se centrará este traballo. A través deste, os datos recóllense por medio dunha aeronave que dispón dun escáner láser (ALS), que emite pulsos láser e mide a intensidade de retorno, un sistema de navegación inercial (INS), que calcula continuamente os xiros e traxectoria da aeronave, e un receptor GPS (Sistema de Posicionamento Global), que obtén a altura e posición do medio aéreo. Desta forma, a aeoronave vai sobrevoando o terreo que se quere estudar facendo distintos varridos co láser e, incluso, percorrendo as mesmas zonas en varias ocasións para obter a máxima información posible sobre o terreo.

O equipamento descrito anteriormente permite coñecer de forma precisa a altura ou elevación dos obxectos ou masas forestais, a súa posición e outros parámetros que se presentarán a posteriori. Adicionalmente, tamén é posible analizar a densidade e distribución da vexetación en función da intensidade do láser dos retornos, xa que, canto máis denso sexa o dosel arbóreo, menos poderá penetrar o pulso láser sobre esa masa forestal (Arroyo et al. (2008)).

Facendo uso da anterior tecnoloxía, obtense información dos puntos LiDAR asociados ao terreo estudado. Concretamente, os tipos de datos almacenados para cada punto son os seguintes (ASPRS (2008)):

- **Valores x, y e z**: son usados conxuntamente para determinar as coordenadas de cada punto, sendo a última a altitude ou elevación.
- **Intensidade**: número natural que representa a magnitude do pulso de retorno. Esta non está influenciada polas condicións atmosféricas ou por sombras ou luz, senón que está determinada polas propiedades da topografía e densidade do terreo e polos obxectos que hai neste.

- **Número de Retorno** (*Return Number*): natural que representa o número de retorno para un pulso de saída determinado. Cada pulso láser pode ter máis dun retorno e esta secuencia de retornos debe ser precisada. Por exemplo, para un pulso láser de saída, o seu primeiro retorno terá un *Return Number* de 1, o segundo será de 2, e así sucesivamente.

- **Número (total) de Retornos** (*Number of Returns*): natural que indica a cantidade total de retornos para un pulso concreto de saída. Por exemplo, un determinado punto pode ter *Return Number* de dous mentres que o *Number of Returns* é de cinco, noutras palabras, este correspóndese co segundo retorno dos cinco totais.

- **Indicador da Dirección de Escaneado** (*Scan Direction Flag*): denota a dirección na que se está a realizar o varrido do láser. Se o valor do bit é 1 este faise cara a dereita e se é 0 cara a esquerda.

- **Límite da Liña de Voo** (*Edge of Flight Line*): bit que indica se o punto está no borde da zona escaneada, é dicir, se este é o último antes de que o varrido láser cambie de dirección (valor 1) ou se, en caso contrario, non é un punto do borde do terreo analizado (valor 0).

- **Red, Green e Blue**: estes tres campos indican o valor asociado a cada cor primaria do modelo *RGB*[1]. Así, con esta especificación determínase a cor coa que representar cada punto LiDAR nun software de visualización.

A continuación, móstrase como exemplo un extracto dunha nube de puntos LiDAR, na que se poden observar os distintos campos presentados previamente.

```
example<-read.table("../puntos/exemploLiDAR.xyz",header = T,sep =' ')
head(example)
```

```
##          x       y      z     I RN NoR SD EoFL C     R     G     B
## 1 591408.0 4771992 550.70  6939  1   1  1    0 0  7710  7196  5140
## 2 591408.2 4771990 550.85  7453  1   1  1    0 0  8224  7710  5911
## 3 591408.4 4771992 550.12  7196  1   1  1    0 0  7453  7453  5397
## 4 591409.0 4771994 550.64 11308  1   1  1    0 0 11308 11308 11308
## 5 591409.6 4771989 550.00  4883  1   1  1    0 2  4626  5397  4369
## 6 591409.7 4771993 550.66  7710  1   1  1    0 0  8481  7710  6682
```

Adicionalmente, a través da ferramenta *Olivia* (véxase Martínez et al. (2018) e Blanco (2022)), desenvolvida polo Grupo de Arquitectura de Computadores (GAC) da Universidade de Santiago

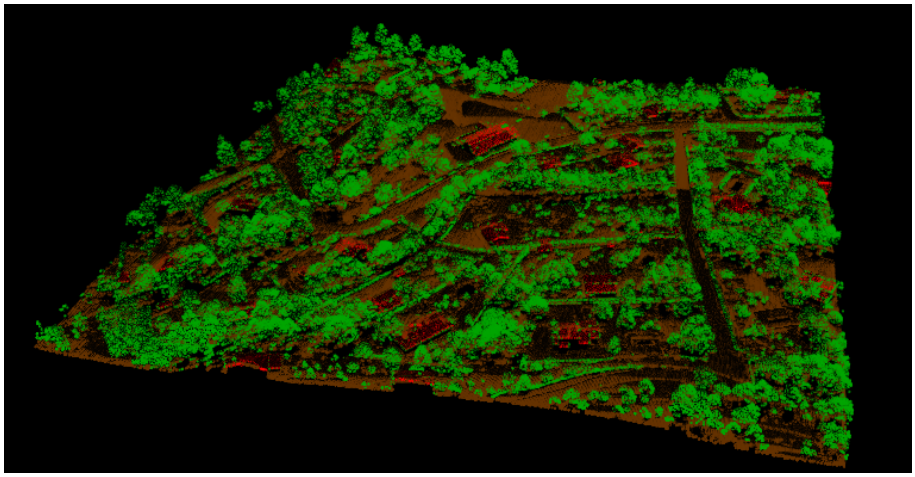

Figura 2: Representación tridimensional dunha nube de puntos LiDAR usando o software de visualización Olivia. A cor dos puntos representa a súa clasificación, sendo os verdes vexetación, os vermellos edificacións e os marróns puntos do chan.

de Compostela (USC), pódense visualizar as nubes de puntos LiDAR (Figura 2).

O obxectivo deste traballo é o estudo e revisión de distintos métodos de clasificación para o modelo *Prometheus* aplicables a puntos LiDAR de masas forestais. Desta maneira, a súa estrutura contará con tres bloques fundamentais. O primeiro estará orientado á introdución á aprendizaxe estatística e particularmente á clasificación, indicando conceptos fundamentais e resultados teóricos de interese. No segundo bloque presentaranse os métodos de estudo, indicando as súas características e marco teórico. Finalmente, na última parte, introduciranse as métricas de rendemento que se aplicarán aos clasificadores para avalialos e comparalos aplicándoos sobre nubes de puntos reais.

0.1. Filtrado e preprocesado de datos

Para poder clasificar as masas forestais en función dos anteriores tipos de combustible, é necesario determinar que datos obtidos mediante LiDAR son relevantes. Ao mesmo tempo, prescindiremos daqueles que non aportan información de interese.

Por un lado, as *Coordenadas x e y* indican a lonxitude e latitude, respectivamente, dun punto, as cales realmente non son relevantes á hora de determinar a súa clasificación. Por outro lado o *Número de Retorno* e o *Número de Retornos* non aportan información adicional xa que todos os puntos das mostras dispoñibles teñen un 1 nestos campos. En canto ao *Indicador da Dirección*

[1] O modelo de cor RGB baséase na suma ou síntese aditiva das cores lumínicas primarias vermello, verde e azul. A través deste é posible representar unha cor mediante a mezcla por adición das tres anteriores.

de Escaneado e *Límite da Liña de Voo*, tal e como están definidos, son datos que non inflúen de ningún xeito na determinación do tipo de combustible dun punto. Finalmente, os tres campos asociados a *Red*, *Green* e *Blue* soamente teñen sentido no marco da visualización de puntos LiDAR.

Desta maneira, tendo en conta como se definen os tipos de combustible do modelo *Prometheus*, os únicos datos que son relevantes son a *Intensidade* e a *Coordenada z*. Por un lado, a *Intensidade* permite coñecer a magnitude do retorno láser. Esta depende dos obxectos cos que o pulso láser se atopa. De feito, canto máis densa sexa a copa dos árbores dunha área, máis difícil será que o láser penetre, co que o valor da intensidade será menor.

A *Coordenada z*, por outro lado, indica a elevación dun punto LiDAR. Esta elevación non se corresponde coa altura real do punto, senón que é a elevación con respecto ao nivel do mar. No noso caso, a altura real dos puntos é fundamental, xa que é un dos discriminantes principais entre varios tipos de combustible do modelo. Por este motivo, será necesario obter, dalgunha forma, este parámetro para cada un dos puntos.

Con este obxectivo, farase uso dun clasificador automático de obxectos mediante LiDAR desenvolvido polo GAC da USC. Este software emprega unha serie de resultados obtidos en fases previas, entre os que se atopa o *MDT* ou *Modelo Dixital do Terreo*. Este modelo, obtido a través dos retornos láser dos puntos do chan, representa a distribución e posición dos puntos do solo, modelando este mesmo. Deste xeito, co *MDT*, poderemos obter a altura real de cada punto LiDAR, simplemente restando a *coordenada z* deste coa do punto do chan coa mesma latitude e lonxitude.

Adicionalmente, disponse dun conxunto de 51 puntos LiDAR con clase coñecida, pertencentes ás nubes de puntos existentes. Estes serán fundamentais para a determinación de posibles algoritmos de clasificación para o problema de estudo deste traballo.

Concretamente, foron recollidos de forma física nos lugares onde se fixo uso de tecnoloxía LiDAR. Destas medicións coñécense as *coordenadas x* e *y* e o tipo de combustible, como se amosa no seguinte exemplo.

```
pts<-read.table("../puntos/groundTruth.xyz", header=T, sep=' ')
head(pts)
```

```
##        X       Y Type
## 1 593216 4774473    2
## 2 593253 4774479    5
## 3 593201 4774526    1
## 4 593310 4774546    7
```

```
## 5 593270 4774562    5
## 6 593289 4774588    6
```

En xeral, estas medicións non teñen correspondencia cun punto concreto das nubes de puntos existentes. Ademais de ser un conxunto moi limitado, existe tamén un erro de precisión nas súas coordenadas asociado á forma de recoller esta información e ao seu posterior contraste coas nubes de puntos dispoñibles. Por este motivo, tendo en conta tamén a continuidade das masas forestais, decidiuse considerar os veciños de cada unha das medicións anteriores, contidos nun entorno de 1 metro, como puntos con clase coñecida.

Desta forma, partirase dun conxunto de puntos máis amplo con clase coñecida (*Type*). Ademais, tamén se disporá da súa altura real (z) e da intensidade (I) asociadas, tal e como se presenta na seguinte mostra:

```
pts<-read.csv("../groundTruth.csv", header=T)
head(pts)
```

```
##       z     I Type
## 1 12.16  8481    6
## 2  0.58  4883    3
## 3  3.06 20560    4
## 4  9.63 12850    7
## 5  9.70  6939    7
## 6  7.13  3341    6
```

Así, o anterior conxunto de referencia constituirá a base para a contrución dos clasificadores que se desenvolverán neste traballo.

Capítulo 1

Primeira aproximación á aprendizaxe supervisada

1.1. Introdución á aprendizaxe estatística

A aprendizaxe estatística é unha área de investigación clave para moitas áreas de ciencia, finanzas ou industria. Baséase no estudo de datos, de patróns e de como estimar determinados aspectos dos datos a partir destes. Deste xeito, o seu obxectivo final é a determinación de patróns comúns nos datos e o seu uso para a predición das características en novos datos.

Para a realización deste proceso é necesario analizar e estudar os datos coñecidos e determinar as variables de entrada e as de saída que se deben ter en conta. Os dous últimos, son conceptos fundamentais na aprendizaxe estatística. As variables de entrada, tamén coñecidas coma preditores ou variables independentes, son aquelas coas que se traballará para obter as variables de saída, xa que as primeiras afectan ás segundas. Estas últimas, tamén chamadas variables dependentes ou de resposta, son aquelas cuxo comportamento ou valor se ve afectado por algunha variable independente.

Notación 1.1. Denotarase $\boldsymbol{X}$ á matriz de dimensión $n \times p$ cuxas columnas se corresponden coas p distintas variables de entrada X_j e cada fila con cada unha das n observacións dispoñibles. Deste xeito, $\boldsymbol{x}_i$ é un vector p-dimensional que fai referencia á i-ésima observación, con $i = \{1, ..., n\}$. A súa j-ésima compoñente x_{ij}, denota ao elemento (i, j) da matriz $\boldsymbol{X}$. Deste xeito,

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}.$$

De igual maneira, $\boldsymbol{Y}$ fai referencia á variable de saída, que supoñemos que é unidimensional. Polo tanto,

$$\boldsymbol{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},$$

sendo y_i con $i = \{1, ..., n\}$ as variables resposta de cada observación.

Desta maneira, no marco da aprendizaxe estatística buscarase estimar as variables dependentes en función das independentes, recoñecendo, nestas últimas, patróns que poidan explicar e estimar os comportamentos das variables de saída.

Todo o anterior proceso toma como partida un conxunto de observacións coñecidas. Concretamente, a natureza do problema considerado determinará a forma de estudo e tratamento destas observacións. No marco da aprendizaxe estatística, distínguense dúas aproximacións: a *supervisada* e *non supervisada*.

Por un lado, a aprendizaxe *supervisada* parte dun conxunto de medicións cuxas variables resposta son coñecidas. Deste xeito, estúdase a relación entre as variables independentes e as dependentes, tratando de determinar como as primeiras afectan ás segundas.

Pola contra, a aprendizaxe *non supervisada* soamente parte dun conxunto de medicións das variables de entradas, sen coñecer as súas variables resposta. Esta aproximación baséase no estudo da relación entre as distintas observacións, determinando patróns ou características comúns entre elas. Desta maneira, aquelas que sexan máis similares entre si serán catalogadas dentro do mesmo grupo ou categoría. Concretamente, unha ferramenta empregada neste caso é o *clustering*, a través da cal as medicións se asocian en grupos relativamente distintivos.

Este traballo centrarase na *aprendizaxe supervisada* por ser a que mellor se adapta ao problema de estudo.

1.2. Aprendizaxe supervisada

A *aprendizaxe supervisada*, como xa se indicou previamente, é unha aproximación na que se parte dun conxunto de observacións cuxas variables dependentes son coñecidas. Esta fundaméntase na determinación da relación existente entre as variables de entrada e as de saída. De feito, cobra vital importancia a determinación de patróns e dependencias entre ambos tipos de variables do conxunto de observacións de partida, tamén coñecido como **conxunto de adestramento**.

Definición 1.2. Denomínase **conxunto de adestramento** a un conxunto da seguinte forma

$A = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$ formado polos n pares $(\boldsymbol{x}_i, y_i) = (x_{i1}, .., x_{ip}, y_i)$ e sendo $\boldsymbol{x}_i$ a i-ésima observación de $\boldsymbol{X}$ e y_i o valor da resposta asociada a cada observación, $i = \{1, .., n\}$.

Deste xeito, a aprendizaxe supervisada baséase na obtención dun modelo cunha función asociada h que permita predicir as variables resposta ante observacións de entrada non vistas previamente, tomando como base un conxunto de adestramento A. A anterior función fai referencia á ideal do problema, é dicir, aquela coa que se obterían as predicións coa maior exactitude. Tendo en conta o anterior, a efectos prácticos calcularase unha función $\hat{h}$ que estime h co menor erro posible.

Observación 1.3. Concretamente, o principal obxectivo da aprendizaxe supervisada é o cálculo dunha función $\hat{h} : \boldsymbol{X} \longrightarrow \boldsymbol{Y}$ que prediga a variable $\boldsymbol{Y}$ partindo dun conxunto de adestramento A. Esta función parte das variables de entrada e obtén $\hat{\boldsymbol{Y}} = \hat{h}(\boldsymbol{X})$, sendo $\hat{\boldsymbol{Y}}$ a predición de $\boldsymbol{Y}$.

De forma xeral, a función $\hat{h}$ condensa toda a información relevante do conxunto de adestramento e permite aplicala a novas observacións. De feito, a flexibilidade ou grao no que esta función depende dos datos do conxunto de adestramento pode levar a aparición dalgún dos seguintes fenómenos:

- **Sobreaxuste**. Consiste na modelación moi detallada dos datos de adestramento. Deste xeito, a función $\hat{h}$ adáptase demasiado ao conxunto de adestramento e non é capaz de xeneralizar esta información a novas observacións.
- **Subaxuste**. Fenómeno baseado nun axuste insuficiente ao conxunto de adestramento, é dicir, á escasa adaptación a estes datos.

Calquera dos anteriores fenómenos poden limitar a exactitude e calidade de predición da función, conducindo a predicións erróneas. Unha forma de detectar estos fenómenos é estudar o seu *erro de adestramento*. Aínda que a súa definición formal varía en función do tipo de aprendizaxe supervisada, en todos os casos este correspóndese co grao de desacerto entre as predicións das observacións de adestramento e os seus valores reais. Un erro de adestramento alto implica que o modelo non se adapta ben ás observacións de partida (subaxuste) mentres que un erro moi baixo indica que se adapta moi ben ao conxunto de adestramento, o que pode ser un indicativo de sobreaxuste do modelo.

De forma natural, introdúcese tamén o *erro de test* dun modelo. Este, de igual forma que o de adestramento, mide o desacerto nas predicións dun conxunto de observacións, denominado *conxunto de validación*, fronte aos seus valores reais. Concretamente, o anterior conxunto, estará formado por observacións cuxa variable resposta é coñecida que non foron empregadas para construír o modelo.

Un aspecto a ter en conta da estimación deste erro é o **compromiso entre nesgo e varianza**, concepto que se introducirá a continuación e se desenvolverá en máis detalle en apartados próximos. Concretamente, dada unha observación $\boldsymbol{x}$ non empregada para o adestramento do modelo, o seu erro asociado pódese descompoñer en tres cantidades fundamentais: a varianza de $\hat{h}(\boldsymbol{x})$ ou Var $\left(\hat{h}(\boldsymbol{x})\right)$, o cadrado do nesgo de $\hat{h}(\boldsymbol{x})$ e a varianza do erro irreducible sistemático ε asociado ao propio problema, de modo que $y = h(\boldsymbol{x}) + \varepsilon$.

O **nesgo** defínese como o erro asociado ás suposicións erróneas do propio algoritmo de aprendizaxe, é dicir, ao erro sistemático asociado á estimación de h. Concretamente, un nesgo alto supón mala xeneralización do algoritmo, o que pode resultar en subaxuste.

Por outro lado, a **varianza** fai referencia á sensibilidade ou variabilidade da estimación de h en $\boldsymbol{x}$ ante pequenas fluctuacións ou cambios no conxunto de adestramento. Unha varianza alta no modelo pode ser indicativo de sobreaxuste.

Deste xeito, é importante ter en conta estes dous primeiros aspectos na construción do modelo, tratando de acadar un equilibrio entre ambos.

A través do *erro de test* é posible coñecer con máis exactitude o rendemento dun modelo e como este xeneraliza a información do conxunto de adestramento. Á diferenza do erro de adestramento, cuxo cálculo e directo e sinxelo, a estimación do erro de test non resulta sempre tan directa. Ademais, este está moi condicionado ao conxunto de validación empregado. Nos casos nos que este non é moi extenso, empréganse diversas técnicas para estimalo, como é o caso da *validación cruzada*.

1.2.1. Validación cruzada

A *validación cruzada* comprende un conxunto de técnicas que teñen como obxectivo obter unha estimación dun determinado parámetro ou métrica asociada a un método de aprendizaxe supervisada, como por exemplo o *erro de test* (E_T). En moitos casos estas fan posible comparar distintos métodos de aprendizaxe estatística ou incluso analizar a flexibilidade adecuada para un método concreto.

Neste apartado presentaranse tres técnicas de validación cruzada: *leave-one-out cross-validation* ou *validación cruzada clásica*, *validación cruzada de k-iteracións* ou *validación cruzada aleatoria* e a *aproximación do conxunto de validación*. Todas elas baséanse na construción dun conxunto de validación a partir do conxunto de adestramento para despois seguir unha serie de procedementos para estimar un parámetro ou medida dun modelo. Neste caso, presentarase o exemplo da estimación do erro de test, mais, os procedementos e técnicas desenvolvidas son equivalentes para calquera outra métrica ou parámetro.

Validación cruzada clásica

A *validación cruzada clásica*, tamén coñecida como *leave-one-out cross-validation*, é unha técnica que se basea en extraer unha observación do conxunto de adestramento e tomala como conxunto de validación. Deste xeito, o primeiro é empregado para construír o modelo, mentres que a observación extraída utilízase para obter o erro de test. Este proceso repítese para as n observacións do conxunto de adestramento, de modo que cada unha delas é empregada para validar o modelo axustado coas restantes.

Finalmente obtéñense n erros de test distintos que se denotarán E_i, asociados a cada unha das observacións. Desta maneira, a estimación do erro de test do modelo calcúlase como a súa media:

$$E_T = \frac{1}{n}\sum_{i=1}^{n} E_i.$$

Esta estimación ten unha varianza asociada, xa que os conxuntos de adestramento son moi semellantes, diferindo nunha única observación dous a dous, o que implica que a predición de h pode cambiar moito dependendo do conxunto de adestramento considerado. Pola contra, o nesgo para este método é moi baixo, xa que este foi construído empregando todas as observacións dispoñibles menos unha.

Adicionalmente, existe un inconvinte de carácter computacional, especialmente se n é grande, xa que sería necesario construír o modelo n veces para obter a estimación do erro, o que pode ser moi custoso.

Validación cruzada de k-iteracións

A *validación cruzada de k-iteracións* ou *k-fold cross-validation* baséase en dividir aleatoriamente todas as observacións en k conxuntos ($k \leq n$), con aproximadamente o mesmo número de elementos. Así, cada un destes é empregado como conxunto de validación en cada unha das k iteracións que se teñen que realizar, mentres que os $k-1$ restantes conforman o conxunto de adestramento que axusta o modelo.

Deste xeito, para cada iteración $j = 1, .., k$ calcúlase o erro de test asociado ao conxunto de validación j, que denotaremos E_j. A través deste proceso obteranse k estimacións do erro de test: $E_1, .., E_k$. Partindo do anterior, o erro de test estímase como a media ponderada das anteriores estimacións,

$$E_T = \frac{1}{N}\sum_{i=1}^{k} n_i E_i,$$

sendo n_i o número de observacións de cada un dos conxuntos de validación asociados e $N = \sum_{i=1}^{k} n_i$. De feito, a *validación clásica* pode considerarse coma un caso concreto da *aleatoria* no que $k = n$ e consecuentemente $n_i = n - 1$ para $i = \{1, ..., n\}$.

Unha cuestión que se presenta a continuación é que valor de k tomar, debido ao **compromiso entre nesgo e varianza** asociado á estimación do erro de test. Como xa se adiantou, para valores de k próximos a n a situación sería semellante á da *validación clásica*, na que a varianza é moi elevada e nesgo baixo. Pola contra, para valores de k baixos, preséntase xusto a situación contraria: nesgo elevado e varianza baixa, debido a unha pobre xeneralización do algoritmo ao tomárense conxuntos de adestramento máis limitados.

Polo tanto, a anterior é unha cuestión a ter en conta á hora de desenvolver esta técnica de validación cruzada. Aínda que esta realmente dependendería do propio modelo e do conxunto de observacións dispoñibles, existen diversos estudos acerca do comportamento e rendemento desta aproximación para determinados valores de k. Concretamente, a validación cruzada de 5 ou 10-iteracións é considerada de forma xeral como un bo compromiso entre nesgo e varianza (Hastie et al. (2017)).

Aproximación do conxunto de validación

A técnica do *conxunto de validación* consiste en dividir de xeito aleatorio as observacións dispoñibles entre o conxunto de adestramento e o de validación. Aínda que depende do número de datos de referencia, unha práctica moi habitual é tomar un número de observacións de adestramento maior que o de validación. Un exemplo podería ser considerar o $75\,\% - 25\,\%$ das observacións en cada un dos conxuntos anteriores, respectivamente.

De igual maneira ca nas anteriores técnicas, o primeiro conxunto empregarase para axustar o modelo e o segundo validalo. Deste xeito, facendo uso das anteriores predicións, poderase obter unha estimación do erro de test.

A *aproximación do conxunto de validación* é unha técnica sinxela e fácil de implementar, pero presenta dous inconvintes. O primeiro é que o erro calculado a través desta aproximación depende moito de que observacións están nos conxuntos de adestramento e validación. Adicionalmente, como o modelo é adestrado cunha menor cantidade de observacións, o erro de test tenderá a ser maior do que realmente sería se se empregaran todas as observacións na súa construción. Polo anterior motivo, é habitual repetir este procedemento de división aleatorio de conxuntos un número considerable de veces, como por exemplo 100, que posteriormente se promedian.

Os anteriores procedementos de validación cruzada introducíronse sen indicar os detalles de como se calcula o erro de test e o de adestramento dos modelos construídos. Este cálculo depende

da propia natureza do problema e do **tipo de aprendizaxe supervisada** asociado. No seguinte apartado presentaranse os dous tipos, indicándose unha aproximación para calcular os erros asociados en cada caso.

1.2.2. Tipos de aprendizaxe supervisada

Dependendo da natureza do problema de aprendizaxe supervisada, a variable de resposta pode ser cuantitativa ou cualitativa. As primeiras son aquelas que se poden medir numericamente, mentres que as segundas, tamén coñecidas como categóricas, son aquelas que fan referencia a unha categoría, característica ou calidade.

No caso de que a variable de saída sexa cuantitativa, o problema será considerado coma un de **regresión**, mentres que se é categórica, será un de **clasificación** (Hastie et al. (2017)).

Regresión

A **regresión** é un tipo de aprendizaxe supervisada na que as variables resposta son cuantitativas. Existen distintas aproximacións para obter modelos de regresión cos que obter a función de regresión h, como pode ser a través da *regresión lineal*, *non lineal* ou *árbores de decisión*, entre outros.

Neste tipo de problemas, a aproximación máis empregada para calcular os erros asociados ao modelo é a través do **erro cuadrático medio**. Deste xeito, defínese o **erro cadrático medio de adestramento** ou ECM_A como se segue.

Definición 1.4. Defínese o *erro cadrático medio de adestramento* ou ECM_A dun problema de regresión como

$$ECM_A = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{h}(\boldsymbol{x}_i))^2,$$

con $\hat{h}$ a función de regresión e $(\boldsymbol{x}_i, y_i) \in A$, $i = 1, .., n$, sendo A o conxunto de adestramento do modelo.

De xeito análogo preséntase o **erro cuadrático medio de test** ou ECM_T.

Definición 1.5. Sexa $\hat{h}$ unha función de *regresión* calculada a partir dun conxunto de adestramento A, e un conxunto $B = \{\boldsymbol{x}_1, y_1), .., (\boldsymbol{x}_m, y_m)\}$ de m observacións verificando $A \cap B = \emptyset$. Defínese o *erro cuadrático medio de test* ou ECM_T como

$$ECM_T = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{f}(\boldsymbol{x}_i))^2$$

con $(\boldsymbol{x}_i, y_i) \in B$, $i = 1, .., m$.

Un aspecto a ter en conta deste último erro, como xa se indicou previamente, é o **compromiso entre nesgo e varianza**. Concretamente, dada unha observación $\boldsymbol{x}$ non empregada para o adestramento do modelo, o seu erro asociado pódese descompoñer na suma de tres cantidades fundamentais, a varianza de $\hat{h}(\boldsymbol{x})$, o cadrado do nesgo de $\hat{h}(\boldsymbol{x})$ e a varianza do erro irreducible asociado ao cálculo de h tal e como se presenta na seguinte proposición (Ramasubramanian and Singh, 2016, p. 489-490).

Proposición 1.6. *Sexa $\hat{h}$ unha función de regresión construída empregando un conxunto de adestramento A e $\boldsymbol{x}$ unha observación. O erro esperado para a observación pódese escribir da seguinte maneira:*

$$E(y - \hat{h}(\boldsymbol{x}))^2 = Var(\hat{h}(\boldsymbol{x})) + [Bias(\hat{h}(\boldsymbol{x}))]^2 + Var(\varepsilon) \tag{1.1}$$

sendo $Var(\hat{h}(\boldsymbol{x}))$ a varianza de $\hat{h}(\boldsymbol{x})$, $Bias(\hat{h}(\boldsymbol{x}))$ o nesgo de $\hat{h}(\boldsymbol{x})$ e ε o erro irreducible ou ruído independente de h que verifica que $y = h(\boldsymbol{x}) + \varepsilon$ e $E[\varepsilon] = 0$.

Demostración. Como $y = h + \varepsilon$ e $E[Z + V] = E[Z] + E[V]$ para calquera par de variables aleatorias Z e V, pódese escribir a seguinte igualdade:

$$\begin{aligned} E[(y - \hat{h})^2] &= E[(h + \varepsilon - \hat{h})^2] = E[(h + \varepsilon - \hat{h} + E[\hat{h}] - E[\hat{h}])^2] \\ &= E[(h - E[\hat{h}])^2] + E[\varepsilon^2] + E[(E[\hat{h}] - h)^2] + 2E[(h - E[\hat{h}])\varepsilon] \\ &+ 2E[(E[\hat{h}] - h)\varepsilon] + 2E[(h - E[\hat{h}])(E[\hat{h}] - \hat{h})]. \end{aligned} \tag{1}$$

Tanto h como $E[\hat{h}]$ son deterministas, é dicir, non dependen do conxunto de adestramento, entón $E[h] = h$, $E[E[\hat{h}]] = E[\hat{h}]$. Deste xeito, $E[(h - E[\hat{h}])] = h - E[\hat{h}]$. Ademais, tendo en conta que h e ε son independentes, pódese reescribir a ecuación (1) como segue:

$$\begin{aligned} E[(y - \hat{h})^2] &= (h - E[\hat{h}])^2 + E[\varepsilon^2] + E[(E[\hat{h}] - \hat{h})^2] + 2(h - E[\hat{h}])E[\varepsilon] \\ &+ 2E[\varepsilon]E[E[\hat{h}] - \hat{h}] + 2E[(E[\hat{h}] - \hat{h})(h - E[\hat{h}])]. \end{aligned} \tag{2}$$

Desenvolvendo o último sumando obtense que se anula:

$$\begin{aligned} E[(E[\hat{h}] - \hat{h})(h - E[\hat{h}])] &= E[E[\hat{h}]h] - E[E[\hat{h}]^2] - E[\hat{h}h] + E[\hat{h}E[\hat{h}]] \\ &= E[\hat{h}]h - E[\hat{h}]^2 - E[\hat{h}]h + E[\hat{h}]^2 = 0. \end{aligned}$$

Adicionalmente, cúmprese que $\mathrm{Var}\,(Z) = E[Z^2] - E[Z]^2$ para calquera variable aleatoria Z, ou equivalentemente, que $E[Z^2] = \mathrm{Var}\,(Z) + E[Z]^2$. Como ademais $E[\varepsilon] = 0$, derívase da ecuación (2) o seguinte:

$$\begin{aligned}
E[(y-\hat{h})^2] &= (h-E[\hat{h}])^2 + E[\varepsilon^2] + E[(E[\hat{h}]-\hat{h})^2] \\
&= (h-E[\hat{h}])^2 + Var[\varepsilon] + E[\hat{h}]^2 - 2E[\hat{h}]^2 + E[\hat{h}^2] \\
&= (h-E[\hat{h}])^2 + Var[\varepsilon] + E[\hat{h}]^2 - 2E[\hat{h}]^2 + Var(\hat{h}) + E[\hat{h}]^2 \\
&= (h-E[\hat{h}])^2 + Var[\varepsilon] + Var[\hat{h}] \\
&= Bias[\hat{h}]^2 + Var[\varepsilon] + Var[\hat{h}].
\end{aligned}$$

□

Xeralmente, para métodos máis flexibles a varianza aumentará mentres que o nesgo diminuirá. Entón, tendo en conta a ecuación (1.1), haberá que axustar a flexibilidade dos métodos para que o seu erro se aproxime o máximo posible a $Var(\varepsilon)$, tratando deste xeito de minimizar a suma dos outros dous sumandos.

Clasificación

A **clasificación** é un tipo de aprendizaxe supervisada na que as variables de saída son cualititativas ou categóricas. Deste xeito, existe un conxunto finito de d valores que poden tomar estas.

Pola propia natureza destas últimas, non é posible calcular os erros de adestramento e de test empregando mínimos cadrados, coma no caso da regresión. Por este motivo, xorden outras aproximacións para calcular estes erros: **a taxa de erro de adestramento** (equivalente ao ECM_A) e a **taxa de erro de test** (equivalente ao ECM_T).

Definición 1.7. Dado $A = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$, o conxunto de adestramento empregado para calcular h, función de clasificación, defínese a **taxa de erro de adestramento** como

$$TE_A = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

sendo $\hat{y}_i$ a predición asociada a $\boldsymbol{x}_i$ e I a función indicadora.

Definición 1.8. Sexa f, a función de clasificación calculada a partir dun conxunto de adestramento A e $B = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_m, y_m)\}$, verificando $A \cap B = \emptyset$, o conxunto de observacións de validación con clase coñecida ás que se lles aplica h. Defínese a **taxa de erro de proba** como

$$TE_P = \frac{1}{m}\sum_{i=1}^{m} I(y_i \neq \hat{y}_i)$$

sendo $\hat{y}_i$ a predición asociada a x_i, e I a función indicadora.

Novamente, no caso da clasificación para a taxa de erro de test, existe un compromiso entre o nesgo e a varianza que se pode derivar do caso de regresión, mais, a súa demostración é dunha maior complexidade e excede os propósitos deste traballo.

De maneira natural, xorde a *taxa de acerto de adestramento* TA_A e a de *test* TA_T como as complementarias das de erro. Estas medidas, tal e como están definidas permiten coñecer a taxa de acerto sobre as predicións realizadas sobre o conxunto de adestramento e o de validación, respectivamente. Ademais da anterior, tamén existen outras métricas coas que medir a precisión e rendemento de clasificadores, que se introducirán en máis detalle na Sección 4.2.

O problema sobre o que versa este traballo, entra dentro dun de clasificación no que se quere estimar o tipo de combustible de masas forestais. Concretamente este parte de dous preditores, a altura e a intensidade, ambos cuantitativos, cos que estimar a variable resposta, que é cualitativa e fai referencia ao tipo de combustible. Desta forma, distínguense sete categorías ou clases distintas que se representan coas seguintes etiquetas $\{1, 2, 3, 4, 5, 6, 7\}$. Así, podemos considerar que $\boldsymbol{X}$ é unha matriz de dimensión $n \times p$, con $p = 2$ e que $\boldsymbol{Y}$ é un vector de dimensión n cuxas compoñentes toman valores no conxunto $\{1, .., 7\}$, habendo polo tanto $d = 7$ clases distintas.

Por este motivo, tendo en conta a natureza do problema de estudo, nos seguintes capítulos introduciranse distintos métodos de clasificación que se poden aplicar a este último.

Capítulo 2

Métodos de clasificación clásicos

Un método de clasificación ou clasificador fai referencia á técnica empregada para construír un modelo que se axusta a un problema de clasificación. Formalmente, pódese definir como se segue:

Definición 2.1. Unha técnica de clasificación ou clasificador enténdese como unha función $h : \boldsymbol{X} \rightarrow \boldsymbol{Y}$ que asigna unha etiqueta ou clase $\hat{h}(\boldsymbol{x}) = \hat{Y}$ ao vector de preditores $\boldsymbol{x}$ de $\boldsymbol{X}$ (Carrizosa and Romero Morales (2013)).

Polo tanto, unha técnica de clasificación non é máis que a estimación da función h de clasificación. Con este obxectivo, pártese dun conxunto de adestramento $A = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$, co que se estima un modelo que permita predicir a variable resposta.

Nos problemas de clasificación é frecuente o uso de **variables ficticias** ou ***dummy variables*** para a construción dos clasificadores. Estas variables empréganse, xeralmente, como indicadores de pertenza a unha clase. Deste xeito, para cada unha das posibles categorías do problema, constrúese unha variable ficticia que indica se pertence ou non a esa clase.

Por exemplo, nun problema de clasificación binario no que as observacións unicamente poden pertencer a dúas clases diferenciadas, como poden ser *ser femia* ou *ser macho* dentro dun conxunto de animais, pódense introducir dúas variables resposta ficticias y_{i1} e y_{i2}, definidas da seguinte forma:

$$y_{i1} = \begin{cases} 1 & \text{se o } i\text{-ésimo animal é femia,} \\ 0 & \text{se o } i\text{-ésima animal non é femia.} \end{cases}$$

$$y_{i2} = \begin{cases} 1 & \text{se o } i\text{-ésimo animal é macho,} \\ 0 & \text{se o } i\text{-ésimo animal non é macho.} \end{cases}$$

Deste xeito, as variables y_{i1} e y_{i2} empréganse para indicar se un animal é femia ou macho, sendo estas categorías excluíntes.

A introdución destas d variables, unha por cada clase distinta, é extrapolable a calquera problema de clasificación. Ademais, permiten flexibilizar e simplificar a construción dos clasificadores.

Existen diversas técnicas de clasificación ou clasificadores que se poden empregar para obter unha resposta categórica. Dúas delas, que son amplamente coñecidas e empregadas son o **clasificador de Bayes** e **K Puntos Próximos**.

2.1. Clasificador de Bayes

O **clasificador de Bayes** é un método de clasificación que se fundamenta no *Teorema de Bayes*. Para definir este clasificador, introduciranse primeiro unha serie de conceptos, comezando coa distribución de probabilidade a priori.

Definición 2.2. Defínese a *distribución de probabilidade a priori* $\pi_k = Pr(\boldsymbol{Y} = k)$ como a probabilidade de que un determinado elemento sexa da clase k sen ter en conta as posibles covariables.

A efectos prácticos, π_k aproximarase pola fracción do número de puntos de clase k no total do conxunto de adestramento, isto é, á frecuencia relativa da clase no conxunto de adestramento. Supoñendo que este é $A = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$, a anterior estimación representarase como:

$$\pi_k \approx \hat{\pi_k} = \frac{1}{n} \sum_{i=1}^{n} I(y_i = k),$$

sendo I a variable indicadora de pertenza á clase.

Adicionalmente, tal e como está definida a probabilidade a priori é directo observar que $\sum_{k=1}^{d} \hat{\pi_k} = 1$.

Definición 2.3. Defínese a *distribución de probabilidade a posteriori* $Pr(Y = k|X = \boldsymbol{x})$ como a probabilidade de que unha observación sexa da clase k condicionado a que as covariables tomen o valor $\boldsymbol{x}$, con $k = \{1, ..., d\}$.

Notación 2.4. Para simplificar a notación, faremos uso da abreviación $p_k(\boldsymbol{x}) = Pr(Y = k|X = \boldsymbol{x})$ para refirirnos á *probabilidade a posteriori* de que unha observación tal que $X = \boldsymbol{x}$ pertenza á clase k. Como este valor será aproximado cos datos do conxunto de adestramento, esta estimación denotarase como $\hat{p_k}(\boldsymbol{x})$.

Tendo en conta a anterior definición, introdúcese a *función de densidade* empregada no clasificador.

Definición 2.5. Defínese $f_k(\boldsymbol{x}) \equiv Pr(\boldsymbol{x}|\boldsymbol{Y} = k)$ como a *función de densidade* de $\boldsymbol{x}$ para unha observación con clase k.

A función de densidade, ao igual que a distribución de probabilidade a priori, tamén se calcula empregando o conxunto de adestramento. Deste xeito, o valor de $f_k(\boldsymbol{x})$ será relativamente maior cantas máis observacións con variables de entradas similares ás de $\boldsymbol{x}$ haxa no conxunto de adestramento coa clase k, e viceversa.

A anterior idea é a base do clasificador que imos construir. Para perfilalo, empregarase o Teorema de Bayes.

Teorema 2.6. *Teorema de Bayes*

Dados Y, conxunto de $d \geq 2$ sucesos excluíntes e con probabilidade distinta de cero, e X, outro suceso calquera, verifícase que:

$$Pr(Y = k|X = \boldsymbol{x}) = \frac{\pi_k Pr(X = \boldsymbol{x}|Y = k)}{\sum_{l=1}^{d} \pi_l Pr(X = \boldsymbol{x}|Y = l)} = \frac{\pi_k f_k(\boldsymbol{x})}{\sum_{l=1}^{d} \pi_l f_l(\boldsymbol{x})} \tag{2.1}$$

sendo d o número total de clases do problema.

A ecuación (2.1) suxire unha forma de calcular $p_k(x)$ en función de π_k e $f_k(x)$, modelándose a distribución dos preditores $\boldsymbol{X}$ de forma independente para cada unha das clases da variable resposta. Deste xeito, o modelo fará uso de tantas variables ficticias resposta como clases teña o problema considerado. Así, considerarase que unha observación é da clase coa que obtén unha probabilidade $p_k(x)$ máis alta.

Tendo en conta que o denominador da ecuación (2.1) se corresponde con $f(\boldsymbol{x}) \equiv Pr(\boldsymbol{x})$, é directo observar que este non depende da categoría k considerada. Polo tanto, para unha observación $\boldsymbol{x}$ unicamente será preciso coñecer cal é a clase coa que se obtén un maior numerador. Esta é coñecida como *máximo a posteriori* ou *MAP* e é calculada como $\hat{Y}$. Por outro lado, a regra coa que se obtén é a *regra do máximo a posteriori*, que se define como se segue:

$$\hat{Y} = \max_{k \in \{1,..,d\}} \{\pi_k f_k(\boldsymbol{x}) / k \in \{1, .., d\}\}. \tag{2.2}$$

Definición 2.7. Un modelo de clasificación que implemente a regra (2.2) coñecerase como **clasificador de Bayes** (Berrar (2019)).

Para poder construir este clasificador é necesario estimar a distribución de probabilidades a priori $\hat{\pi}_k$. Resulta sinxelo obtela a través do conxunto de adestramento. Pola contra, o cálculo da función de densidade $f_k(\boldsymbol{x})$ resulta máis complexo, especialmente cando hai máis dunha variable preditora.

2.1.1. Estimación da función de densidade

A función de densidade $f_k(\boldsymbol{x})$ depende das variables preditoras consideradas. No caso de que haxa máis dunha variable de entrada, poderase ter en conta a seguinte definición.

Definición 2.8. Sexan Z e V dúas variables aleatorias continuas, a *función de densidade conxunta* $f_{Z,V}(z,v)$ pódese escribir coma:

$$f_{Z,V}(z,v) = f_{Z|V}(z|v) \cdot f_V(v) = f_{V|Z}(v|z) \cdot f_Z(z), \tag{2.3}$$

sendo $f_{Z|V}(z|v)$ e $f_{V|Z}(v|z)$ as densidades condicionadas de Z dado $V = v$ e de V dado $Z = z$, respectivamente, e $f_Z(z)$ e $f_V(v)$ as funcións de densidade marxinal de Z e V respectivamente.

A través da ecuación (2.3) séguese que, no caso de que ambas variables sexan independentes, a función de densidade conxunta se podería calcular da seguinte maneira:

$$f_{Z,V}(z,v) = f_Z(z) \cdot f_V(v). \tag{2.4}$$

A igualdade (2.4) é debido a que, como as variables son independentes, a súa función de densidade condicionada correspóndese coa marxinal da variable considerada. Isto permite calcular dunha forma máis sinxela a función de densidade conxunta.

Tanto a igualdade (2.3) como a (2.4) son extrapolables a máis de dúas variables aleatorias, mais o caso de estudo deste traballo emprega unicamente dúas variables.

En calquera dos casos, habendo unha ou máis variables de entrada implicadas, a estimación correcta da densidade é un proceso fundamental para este método de clasificación. No caso de que se coñeza a familia da distribución da variable, é directo obter a súa función de densidade asociada. Mais, o anterior, non é o común na práctica, polo que nestes casos é preciso empregar técnicas para estimar a densidade como a *estimación de densidade de tipo kernel* (KDE) (Zambom and Dias (2013)).

Definición 2.9. Dadas $(z_1, ..., z_n)$ observacións independentes e identicamente distribuídas asociadas a unha variable aleatoria Z cunha distribución con función de densidade descoñecida f. Defínese a súa *estimación de densidade kernel* como

$$\hat{f}_h(z) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{z - z_i}{h}\right),$$

sendo K unha función núcleo ou kernel, $h > 0$ un parámetro de suavidade coñecido como ancho de banda.

Unha función kernel ou núcleo é unha función continua, simétrica e definida positiva. A súa definición formal é a que se presenta a continuación.

Definición 2.10. Sexa Z unha variable aleatoria, defínese unha función kernel ou núcleo $K : Z \to \mathbb{R}$ como unha función non negativa que verifica

$$\int_{-\infty}^{+\infty} K(z)dz = 1, \qquad K(z) = K(-z),$$

sendo z unha observación de Z.

Existen distintas funcións núcleo aplicables á estimación da densidade, entre elas atópanse as presentadas no Cadro 2.1.

Nome	**Función núcleo**
Uniforme	$K(z) = \frac{z}{2}, \lvert z\rvert \leq 1$
Epanechnikov	$K(z) = \frac{3}{4}(1 - \lvert z\rvert), \lvert z\rvert \leq 1$
Normal ou Gaussiana	$K(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}$
Sigmoide	$K(z) = \frac{2}{\pi} \frac{1}{e^z + e^{-z}}$

Cadro 2.1: Cadro con algunhas funcións núcleo.

A través do uso de calquera das anteriores no KDE poderase obter unha estimación da función de densidade dunha variable aleatoria con distribución descoñecida.

2.1.2. Clasificador de Bayes inxenuo

A variante do clasificador de Bayes que asume que as variables son independentes entre si, coñécese como **Naive Bayes classifier** ou **clasificador de Bayes inxenuo**. Este modelo é

adecuado cando existen moitas variables preditoras, xa que a estimación das densidades resulta máis complexa. Deste xeito, as funcións de densidade conxunta das clases calcularanse como:

$$f_k(X) = \prod_{i=1}^{p} f_k^i(X_i),$$

sendo $f_k^i(X_i)$ a función de densidade do preditor X_i para a clase k. O principal problema deste clasificador é que a anterior suposición non é sempre certa, o que pode conlevar a un maior erro de predición. A pesar diso, o custo computacional no clasificador así como a súa complexidade diminúe considerablemente, especialmente cando se emprega un número elevado de preditores. Isto é debido a que soamente é necesario calcular as densidades marxinais de cada variable de entrada (Hastie et al. (2017)).

Adicionalmente, para estimar a función de densidade conxunta é necesario dispoñer dunha cantidade elevada de observacións, polo que *Naive Bayes* é unha boa elección en casos nos que non se dispoña dun conxunto de adestramento amplo.

Aínda que a suposición deste clasificador aumenta o nesgo, tamén reduce a varianza. Isto fai que traballe ben na práctica, grazas ao compromiso entre nesgo e varianza.

2.2. K Puntos Próximos (KNN)

O clasificador de Bayes é unha técnica de clasificación que se fundamenta no estudo da clase coa que cada observación obtén a maior probabilidade. Para datos reais, non é doado obter a distribución de X dado Y, o que fai que, en ocasións, a implementación do clasificador de Bayes non sexa asequible. **K-Nearest Neighbors** ou **K-Veciños Próximos** (KNN) é un método que permite lidiar con este problema. Baséase no estudo da distribución da probabilidade a posteriori de Y nun entorno local de X, para posteriormente clasificar cada observación en función da distribución estimada máis alta.

O anterior clasificador, dado un enteiro positivo $\mathcal{K}$ e unha observación $\boldsymbol{x}_0$, toma os $\mathcal{K}$ puntos do conxunto de adestramento máis próximos a $\boldsymbol{x}_0$. Os índices asociados a estes conforman o conxunto $\mathcal{N}_0$. Desta forma, calcúlase a probabilidade condicional da clase k como a fracción dos puntos asociados a $\mathcal{N}_0$ cuxas respostas son iguais a k:

$$\hat{p_k}(\boldsymbol{x}_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = k), \tag{2.5}$$

sendo I a función indicadora de pertenza á clase k.

Deste xeito, obterase unha probabilidade condicionada para cada clase existente. Así, de xeito análogo ao clasificador de Bayes, asignaráselle a cada punto a clase coa que a probabilidade sexa maior.

Na Figura 2.1 (James et al. (2021)) amósase un exemplo ilustrativo do comportamento deste clasificador para un problema con dous preditores e dúas clases, representadas con cores distintas. Para iso partirase dun conxunto de adestramento formado con seis puntos de cada clase, tal e como se amosa no gráfico da esquerda. Neste caso aplícase o algoritmo con $\mathcal{K} = 3$, polo que é necesario determinar os tres veciños máis próximos a cada punto, como se exemplifica co círculo verde. No da dereita, por outro lado, preséntase en negro a fronteira de decisión entre as dúas clases tras aplicar o algoritmo. Deste xeito, calquera punto enmarcado entre estas dúas liñas sería etiquetado coa clase correspondente á cor laranxa, mentres que o resto pertencerían á outra categoría.

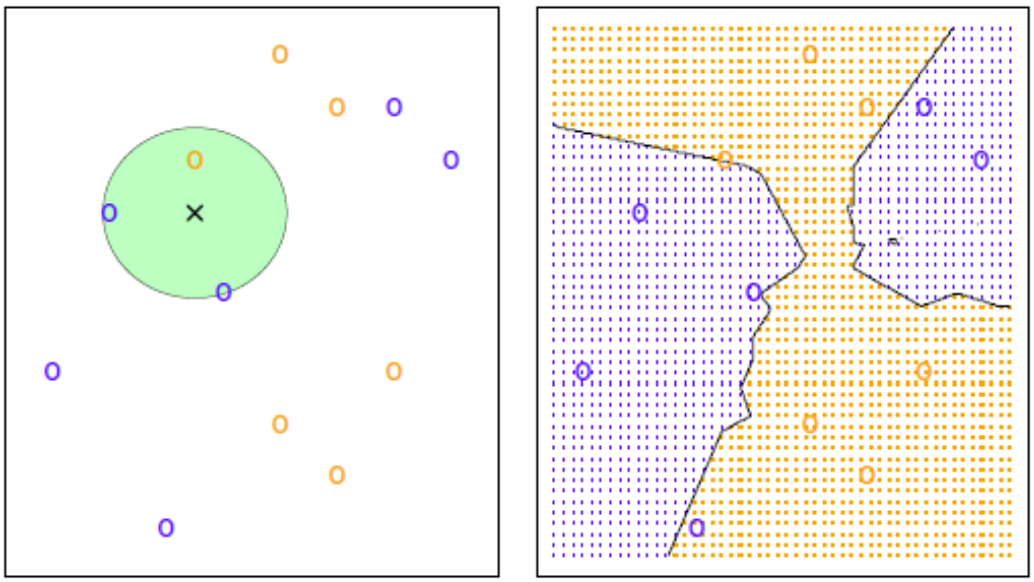

Figura 2.1: Exemplo ilustrativo de KNN. Na esquerda preséntase o conxunto de adestramento, habendo seis observacións de cada cor. Na dereita amósase o resultado de aplicar o algoritmo para $\mathcal{K} = 3$.

2.2.1. Determinación da distancia métrica

No método de K-Veciños Próximos a elección dunha métrica cobra vital importancia xa que a distancia entre os datos é a base para a selección dos $\mathcal{K}$ veciños máis cercanos a un punto. Como consecuencia, analizaranse distintas distancias presentando as súas características, vantaxes e inconvintes para, finalmente, determinar cal é a máis adecuada.

As distancias que se presentarán son a **euclídea**, a de **Manhattan**, a de **Chebyshov** e a de **Mahalanobis**.

Distancia euclidiana

A **distancia euclídea** é a noción de distancia máis empregada nos espazos euclidianos. A súa definición é a seguinte.

Definición 2.11. Dados $\boldsymbol{x}_0$, $\boldsymbol{x}_1$ observacións de $\boldsymbol{X}$, defínese a *distancia euclidiana* entre $\boldsymbol{x}_0$ e $\boldsymbol{x}_1$ como:

$$d_E(\boldsymbol{x}_0, \boldsymbol{x}_1) = \sqrt{\sum_{i=1}^{p} (x_{0i} - x_{1i})^2} = \sqrt{(\boldsymbol{x}_0 - \boldsymbol{x}_1)^T (\boldsymbol{x}_0 - \boldsymbol{x}_1)}.$$

Para poder facer uso desta distancia ao comparar múltiples variables, será necesario estandarizar os datos previamente. Desta maneira, elimínanse as unidades e pesos de ambas medidas de forma igualitaria. No caso contrario, podería ocorrer que unha das variables tivera máis peso no cálculo da distancia se, por exemplo, esta tomara valores máis grandes e/ou espaciados que as outras.

Outra desvantaxe é que, no caso de que haxa correlación entre as variables aleatorias, a distancia euclidiana non é a mellor opción xa que existe información redundante no seu cálculo.

Distancia de Manhattan

A **distancia de Manhattan**, tamén coñecida como distancia de L_1 ou do taxista, é un tipo de métrica que se fundamenta na suma das diferenzas absolutas das coordenadas dos puntos. A súa definición formal é a seguinte.

Definición 2.12. Dados $\boldsymbol{x}_0$, $\boldsymbol{x}_1$ observacións de $\boldsymbol{X}$, defínese a *distancia de Manhattan* entre $\boldsymbol{x}_0$ e $\boldsymbol{x}_1$ como:

$$d_1(\boldsymbol{x}_0, \boldsymbol{x}_1) = \|\boldsymbol{x}_0 - \boldsymbol{x}_1\|_1 = \sum_{i=1}^{p} |x_{0i} - x_{1i}|.$$

Ao igual que a euclidiana, a distancia de Manhattan tampouco ten en conta a correlación das variables e é preciso normalizar os datos antes de calculala.

Distancia de Chebyshov

A **distancia de Chebyshov** ou **métrica máxima** é unha métrica que se pode exemplificar nun taboleiro de xadrez. Baixo este contexto, esta distancia fai referencia ao número mínimo de movementos que o rei necesita para moverse entre dúas celas.

Definición 2.13. Dados $\boldsymbol{x}_0$, $\boldsymbol{x}_1$ observacións de $\boldsymbol{X}$, defínese a *distancia de Chebyshov* entre $\boldsymbol{x}_0$ e $\boldsymbol{x}_1$ como:

$$d_C(\boldsymbol{x}_0, \boldsymbol{x}_1) = \max_{i \in \{1,\dots,p\}} (|x_{0i} - x_{1i}|).$$

De novo, esta distancia tampouco ten en conta a correlación entre variables e, para calculala, é necesario estandarizar os datos. A principal diferenza con todas as anteriores é que soamente se terá en conta a diferenza absoluta máis alta, co que se podería estar perdendo información do resto de variables. De todas formas, isto podería ser unha vantaxe ou inconvinte dependendo da propia natureza do problema de clasificación.

Distancia de Mahalanobis

A **distancia de Mahalanobis** é unha métrica que, ao contrario que as anteriores distancias, ten en conta a correlación entre as variables aleatorias. Tal e como está definida, elimina a redundacia provocada por esta correlación, xa que emprega a inversa da matriz de variaza-covarianza.

Definición 2.14. Dados $\boldsymbol{x}_0$, $\boldsymbol{x}_1$ observacións de $\boldsymbol{X}$ e Σ a matriz de varianza-covarianza asociada ás variables aleatorias, defínese a *distancia de Mahalanobis* entre $\boldsymbol{x}_0$ e $\boldsymbol{x}_1$ como:

$$d_M(\boldsymbol{x}_0, \boldsymbol{x}_1) = \sqrt{(\boldsymbol{x}_0 - \boldsymbol{x}_1)^T \Sigma^{-1} (\boldsymbol{x}_0 - \boldsymbol{x}_1)}.$$

Como xa se indicou, a anterior distancia ten en conta a correlación entre as variables aleatorias. Desta forma, evalúase a distancia asignando diferentes pesos ou factores de importancia ás características dos datos.

Adicionalmente, a distancia de Mahanalobis estandariza os datos directamente, grazas á inclusión da inversa da matriz de varianza-covarianza no seu cálculo. Desta forma non é necesario realizar unha estandarización dos datos a priori coma no caso da distancia euclídea. De feito, unicamente no caso de que as variables sexan incorreladas e estén estandarizadas, ambas distancias serían idénticas.

Outra vantaxe da distancia de Mahalanobis é que axusta a distribución dos datos, permitindo que a distancia entre puntos semellantes sexa pequena. Isto pode mellorar o rendemento en termos de precisión de algoritmos de clasificación (De Maesschalck et al. (2000), Xiang et al. (2008)).

Como consecuencia das anteriores propiedades, a representación dunha distancia de Mahalanobis constante dende un centroide é un elipsoide p-dimensional, sendo p a dimensión do espazo. De feito, os eixos destas elipsoides corresponderíanse coas direccións dos autovalores da matriz de varianza-covarianza Σ.

En contraste, no caso da distancia euclídea, en lugar de elipsoides, as liñas de nivel corresponderanse con p-bólas. Se os datos están estandarizados, a representación desta corresponderase con elipsoides con eixos pararelos aos das coordenadas (Srivastava and Rao (2016)).

As anteriores propiedades pódense observar na Figura 2.2, onde se representan liñas coa mesma distancia ao centro da mostra empregada, provinte das observacións LiDAR, en función da distancia de Mahalanobis e a distancia euclídea estandarizada.

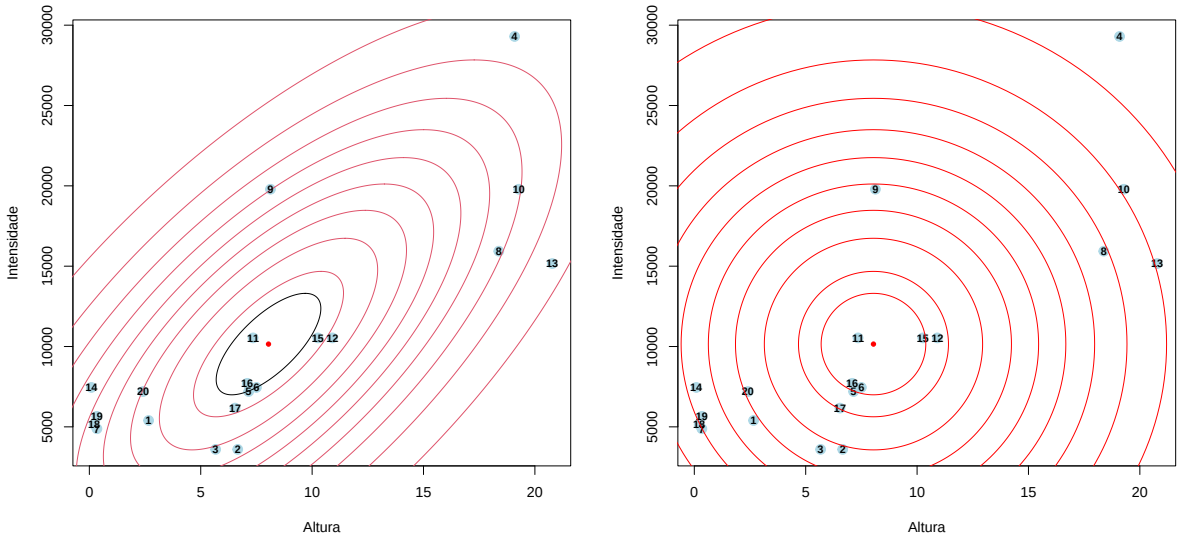

Figura 2.2: Gráficas coas curvas de nivel equivalentes para a distancia de Mahalanobis (esquerda) e a euclidiana (dereita) con respecto á media da mostra de puntos representados (en vermello), do problema de clasificación da vexetación a través de LiDAR.

A representación das curvas de nivel das dúas distancias correspóndese con elipses, cuxos eixos son distintos en cada caso. Deste xeito, é visible como inflúe a correlación das variables na distancia de Mahalanobis. Concretamente, analizando o punto 1 obsérvase que, no caso da distancia euclídea, este se atopa entre a cuarta e a quinta curva de nivel mentres que para a de Mahalanobis este punto está entre a terceira e a cuarta.

Polo tanto, ao entrar en consideración a correlación das variables na determinación da distancia, pode cambiar a proximidade entre os puntos. A través do uso de dendogramas[1], que agrupan puntos en función da súa proximidade, pódense observar as diferenzas entre as dúas distancias (Figura 2.3).

[1] Un dendograma é un tipo de representación gráfica en forma de árbore que agrupa os datos en subcategorías que se van dividindo progresivamente ata acadar o nivel de detalle desexado. A anterior agrupación realízase en

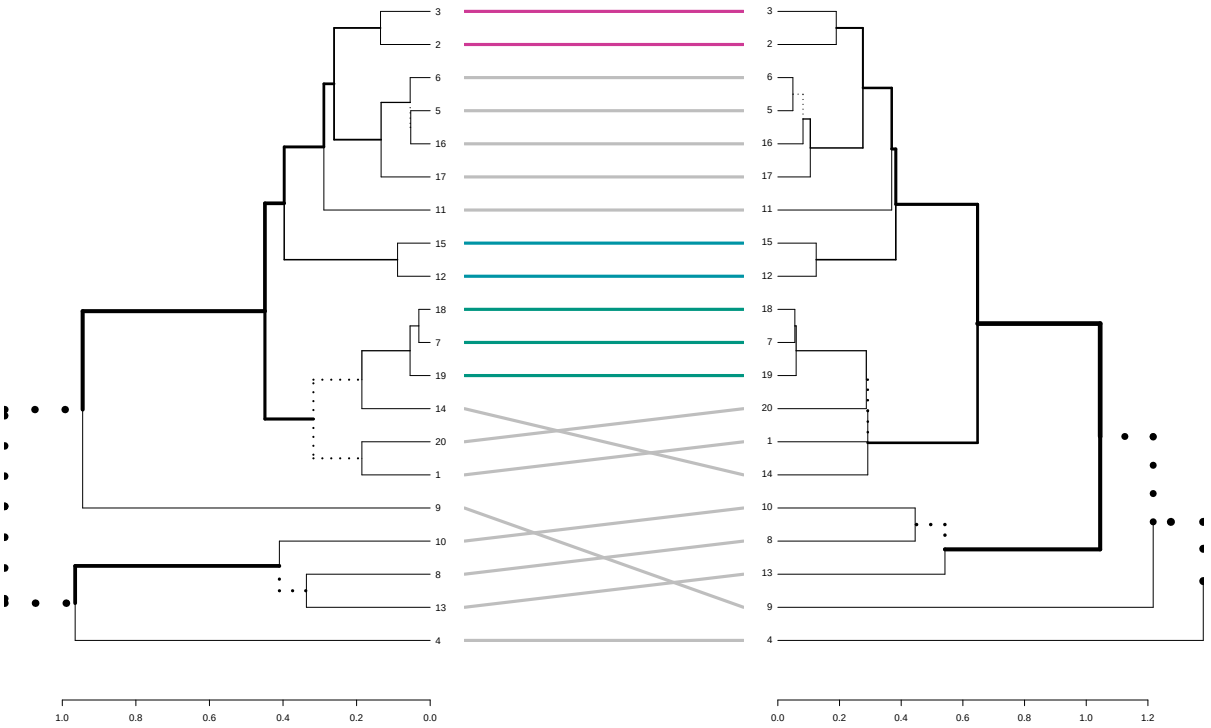

Figura 2.3: Comparación dos dendogramas de agrupación en función da proximidade dos puntos para a distancia de Mahalanobis á esquerda e da euclídea estandarizada á dereita.

Na Figura 2.3 agrúpanse os puntos en función da súa cercanía. Deste xeito, estarán máis próximos aqueles que o estén no diagrama xerárquico. Obsérvanse diferenzas nos dendogramas xerados coas dúas distancias. Un exemplo é o asociado aos puntos 8, 10 e 13. Por un lado, para a distancia de Mahalanobis os puntos 8 e 13 están máis próximos entre si que co 10, mentres que para a distancia euclidiana son o 10 e o 8 os que están máis cerca.

A principal desvantaxe da distancia de Mahalanobis é que é preciso calcular a matriz de varianza-covarianza Σ que, ademais, pode non ser invertible. Neste último caso habería que estudar formas de estimala como por exemplo a través da matriz pseudoinversa.

Tendo en conta todo o exposto anteriormente resulta convinte analizar as variables aleatorias do problema, para determinar cal é a distancia que se pode axustar mellor ás súas características. Deste xeito, dependendo do problema considerado, será máis edecuado aplicar unha distancia ou outra.

canto á proximidade ou similitude entre os datos, permitindo coñecer así as relacións entre eles.

2.2.2. Determinación do valor de $\mathcal{K}$

Un aspecto crítico á hora de facer uso do método de $\mathcal{K}$ Puntos Próximos é determinar un valor adecuado de $\mathcal{K}$. Para valores moi pequenos, como por exemplo $\mathcal{K} = 1$, o clasificador é bastante flexible tendo ademais moi pouco nesgo pero unha varianza moi alta. A medida que o valor de $\mathcal{K}$ vai aumentando, increméntase o nesgo e diminúe a varianza e a flexibilidade. Este fenómeno pódese observar na Figura 2.4, na que se representan as taxas de erro para un problema de clasificación (James et al. (2021)).

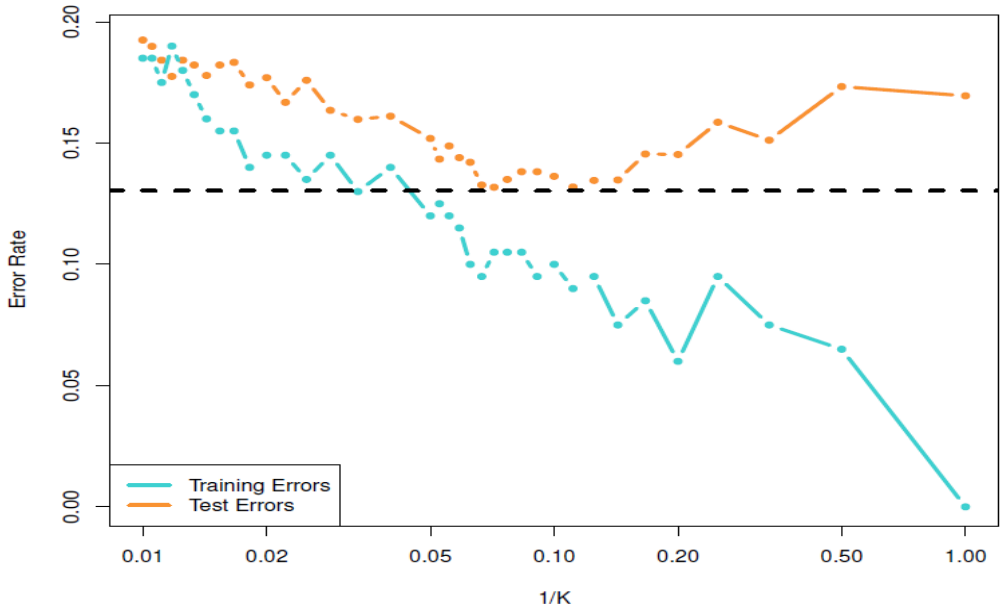

Figura 2.4: Taxa de erro de adestramento (azul, con 200 observacións) e de test (laranxa, 5000 observacións) de *KNN* fronte ao nivel de flexibilidade $\frac{1}{\mathcal{K}}$ (en escala logarítmica). A liña descontinua representa a taxa de erro do clasificador de Bayes.

A medida que o valor de $\frac{1}{\mathcal{K}}$ aumenta, tamén o fai a flexibilidade, é dicir, o modelo adáptase máis aos datos de adestramento. Por este motivo, pódese observar unha certa tendencia decrecente no erro de adestramento a medida de que o nivel de flexibilidade aumenta. Non ocorre o mesmo coa taxa de erro de test, xa que a súa gráfica non segue a mesma tendencia sempre. De feito, pódese observar que esta ten unha forma en U, debido ao compromiso entre nesgo e varianza. Por este motivo, é preferible seleccionar un valor de $\mathcal{K}$ intermedio que minimice a taxa de erro de test.

De todas formas, o valor óptimo de $\mathcal{K}$ depende de cada problema concreto. Para determinalo, pódense obter as taxas de erro de adestramento e proba para diferentes valores de $\mathcal{K}$, analizando con cal se acada un valor máis baixo. Con este obxectivo, pódense empregar técnicas de validación cruzada coma as estudadas no anterior capítulo.

Outro aspecto a ter en conta no método de $\mathcal{K}$ Veciños Próximos é a resolución de desempates á hora de determinar a clase coa maior presenza nos veciños dun punto. No caso dun problema de clasificación binaria, é dicir, no que soamente hai dúas clases distintas, pódese evitar o empate escollendo un valor de $\mathcal{K}$ impar. Cando o número de clases é maior, non é posible adoptar esta aproximación xa que non evita o empate. Deste xeito, é necesario determinar un procedemento para a súa resolución. Algunhas aproximacións son as seguintes.

- **Desempate aleatorio**. Baséase en seleccionar de foma aleatoria a clase da obvervación entre aquelas que están empatadas.
- **Clase máis cercana das empatadas**. Esta solución asígnalle á observación a clase (pertencente ás empatadas) co punto máis próximo. Se, por exemplo, existe un empate entre as clases 1, 4 e 6, e o punto máis próximo á observación das anteriores categorías ten como variable resposta 4, tomarase esta última como a clase da observación.
- **Promedio de distancias**. Esta aproximación fundaméntase no cálculo da distancia media dos veciños das clases empatadas. Deste xeito tomarase aquela cuxa distancia promedio á observación sexa menor.

Capítulo 3

Modelos Aditivos Xeneralizados

Ademais de métodos máis clásicos, existen outras aproximacións que se basean na adaptación de métodos de regresión ao caso da clasificación. Unha delas son os **Modelos Aditivos Xeralizados** (GAM), desenvolvidos por Trevor Hastie e Robert Tibshirani (Hastie and Tibshirani (1990)). Estes permiten combinar as propiedades dos Modelos Lineais Xeneralizados (GLMs) cos dos modelos aditivos, sendo ademais unha técnica eficaz para modelar as relacións non lineais entre múltiples variables (De Bock et al. (2010)).

Neste capítulo presentaranse unha serie de conceptos necesarios para entender o contexto dos GAMs e así introducir estes últimos. Adicionalmente, indicaranse as bases para a súa estimación e a súa adaptación a problemas de clasificación.

3.1. Introdución aos GAMs

Os modelos xeneralizados aditivos xorden da combinación dos GLMs e o modelos aditivos. Por un lado, os primeiros son unha xeneralización dos modelos de regresión lineal. Estes permiten predicir unha variable aleatoria seguindo unha distribución da *familia exponencial* a partir dun conxunto de variables de entrada.

Definición 3.1. A *familia exponencial* comprende aquelas distribucións da probabilidade cuxa función de densidade se pode escribir da seguinte forma:

$$f_\theta(y) = \exp\left[\frac{y\theta - b(\theta)}{a(\gamma)} + c(y,\gamma)\right],$$

sendo b, a e c funcións arbitrarias, γ un parámetro de escala arbitrario e θ o *parámetro canónico ou de localización* da distribución.

Tal e como está definida, a familia exponencial inclúe a distribución de Poisson, binomial, gamma ou a normal, entre outras. A expresión desta última como distribución da familia exponencial pódese derivar de forma sinxela. Partindo dunha distribución $N(\mu, \sigma^2)$, obtense que

$$\begin{aligned} f_\mu(y) &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y-\mu)^2}{2\sigma^2}\right] \\ &= \exp\left[\frac{-y^2 + 2y\mu - \mu^2}{2\sigma^2} - \log(\sigma\sqrt{2\pi})\right] \\ &= \exp\left[\frac{y\mu - \mu^2/2}{\sigma^2} - \frac{y^2}{2\sigma^2} - \log(\sigma\sqrt{2\pi})\right], \end{aligned} \tag{3.1}$$

con $\theta = \mu, b(\theta) = \theta^2/2 \equiv \mu^2/2, a(\gamma) = \gamma = \sigma^2$ e $c(\gamma, y) = -y^2/(2\varphi) - \log(\sqrt{\gamma 2\pi}) \equiv -y^2/\left(2\sigma^2\right) - \log(\sigma\sqrt{2\pi})$ (Wood, 2017, p. 103).

Adicionalmente, os GLMs permiten introducir un determinado grao de non linealidade na estrutura do modelo. De forma máis concreta, a súa definición formal é a seguinte.

Definición 3.2. Dadas $X_1, .., X_p$ un conxunto de variables preditoras e Y unha variable aleatoria seguindo unha distribución exponencial, defínese un *modelo xeneralizado lineal* como un da forma

$$g(Y) = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \varepsilon, \tag{3.2}$$

sendo $\beta_0, .., \beta_p$ uns parámetros inicialmente descoñecidos, g a *función de enlace* que é monótona e infinitamente diferenciable e ε o erro irreducible asociado ao modelo con $E[\varepsilon] = 0$ e $\text{Var}(\varepsilon) = \sigma^2$.

Así, dados $\boldsymbol{X}$ matriz de observacións dun problema de regresión e $\boldsymbol{Y}$ baixo as condicións da anterior definición, a ecuación 3.2 pódese reescribir da seguinte maneira:

$$g(\hat{y}_i) = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{ji}, \tag{3.3}$$

sendo $\hat{y}_i$ a predición de y_i e $\hat{\beta}_j$ con $j = \{1, .., p\}$ as estimacións de cada un dos parámetros β_j.

A *función de enlace* ou función link, debe ser seleccionada con anterioridade para cada problema, aínda que é habitual escollela en función da distribución da variable resposta. Así, esta permite modelar as relacións non lineais entre ámbalas variables preditoras e a resposta.

Por outro lado, os modelos aditivos son aqueles que permiten encapsular o comportamento das variables de entrada en funcións independentes entre si, que se suman para obter o modelo final.

Definición 3.3. Dadas $X_1, .., X_p$ un conxunto de variables preditoras e Y unha variable aleatoria, defínese un *modelo aditivo* como un que segue a seguinte estrutura

$$Y = \alpha + \sum_{j=1}^{p} \phi_j(X_j) + \varepsilon, \tag{3.4}$$

sendo α un parámetro inicialmente descoñecido, $\phi_1, .., \phi_p$ as funcións suaves de cada unha das variables preditoras e ε o erro irreducible asociado ao modelo con $E[\varepsilon] = 0$ e $\text{Var}(\varepsilon) = \sigma^2$.

As funcións suaves para cada modelo son únicas salvo constante. Estas modelan o comportamento das variables preditoras sobre a variable resposta de forma independente. Posteriormente súmanse para obter a predición de Y, o que define o carácter aditivo do modelo. Concretamente, estas permiten definir modelos non paramétricos ou semi paramétricos, é dicir, aqueles nos que a distribución dos preditores non se asume de antemán.

Os *Modelos Aditivos Xeneralizados* ou *Generalized Additive Models* (GAM) combinan as características dos GLMs e os modelos aditivos. Por un lado, inclúen a función de enlace, o que permite flexibilizar as suposicións sobre a relación existente entre a resposta e os preditores. Por outro lado, a inclusión das funcións suaves fai posible adaptarse mellor aos datos posibilitando empregar modelos non paramétricos. De xeito máis formal, a súa definición é a seguinte.

Definición 3.4. Dadas $X_1, ..., X_p$ variables aleatorias independentes e Y a variable aleatoria de saída tal que asociada seguindo unha distribución exponencial, defínese un *modelo aditivo xeneralizado* como un da forma

$$g(Y) = \alpha + \sum_{j=1}^{p} \phi_j(X_j) + \varepsilon, \tag{3.5}$$

sendo α un parámetro inicialmente descoñecido, ϕ_j as funcións suaves para as variables independentes X_j, g a función de enlace que é monótona e infinitamente diferenciable e ε o erro irreducible asociado ao modelo con $E[\varepsilon] = 0$ e $\text{Var}(\varepsilon) = \sigma^2$.

Dados $\boldsymbol{X}$ matriz contendo as observacións e $\boldsymbol{Y}$ a resposta seguindo as condicións da anterior definición, a ecuación 3.5 pódese reescribir como

$$g(\hat{y}_i) = \hat{\alpha} + \sum_{j=1}^{p} \hat{\phi}_j(x_{ji}), \tag{3.6}$$

con $i = \{1, ..., n\}$ e sendo g a función enlace, $\hat{\alpha}$ e $\hat{\phi}_j$ as estimacións do parámetro α e das funcións ϕ_j, respectivamente.

3.2. Consideracións para a estimación de GAMs

Unha vez presentadas as principais características dos GAMs, introduciranse unha serie de consideracións para estimalos. Concretamente, indicarase unha forma de representar as funcións suaves e, posteriormente, os fundamentos nos que se sustenta a estimación do modelo.

3.2.1. Representación das funcións suaves

Para problemas non paramétricos ou semi paramétricos, será necesario definir e estimar as funcións suaves para as variables preditoras que o precisen. Existen diversas aproximacións para realizar este proceso, entre as que destaca o uso de **funcións base**.

Esta aproximación fundaméntase en tomar p conxuntos $\mathcal{B}_j = \{b_j^1(X_j), ..., b_j^{m_j}(X_j)\}$ de m_j funcións aplicables a cada unha das variables de entrada X_j como base para a estimación da función suaves asociada a $\phi_j(X_j)$. Así, esta escribirase como unha combinación lineal do anterior conxunto, é dicir,

$$\hat{\phi}_j(X_j) = \beta_j^0 + \sum_{t=1}^{m_j} \beta_j^t b_j^t(X_j), \tag{3.7}$$

sendo β_j^t o parámetro asociado a cada función base con $t = \{1, .., m_j\}$, m_j o número de funcións base elixidas e β_j^0 outro parámetro adicional . Así, para estimar cada función suave será necesario calcular o valor de cada un dos anteriores parámetros en función dun conxunto de observacións de referencia.

As funcións base poderán ser de distintos tipos, como por exemplo polinómicas ou a trozos (splines). Concretamente, neste último caso existen unha serie de consideracións que é necesario ter en conta. A primeira é a selección dos nodos ou puntos do dominio nos que se cambia de función. O spline é máis flexible naquelas rexións que conteñen moitos nodos, especialmente se están definidas por funcións polinómicas. Polo tanto, é preferible posicionar estes nodos nas rexións nas que se vaia a producir un cambio ou unha variación maior e reducir a súa presenza naquelas que sexan máis estables.

Adicionalmente, para o caso de splines polinómicos, tamén é preciso adaptalos ao nivel de suavidade desexado para as funcións suaves $\phi_j(X_j)$. Este reflíctese nos *graos de liberdade* das funcións base e, consecuentemente, nas restriccións que se lles impoñen. Un exemplo pode ser esixirlles continuidade ou derivabilidade en todos os seus puntos.

As anteriores consideracións así como outras posibles funcións e splines aplicables á estimación das funcións suaves, desenvólvense en maior detalle en (James et al., 2021, p. 290-306).

3.2.2. Estimación do modelo

A estimación do modelo baséase na estimación dos parámetros que o compoñen. Partindo da representación das funcións suaves con funcións base introducida no anterior apartado, estes parámetros serán os β_j^t asociados a cada unha das ϕ_j, con $t = \{1, ..., m_j\}$, ademais de α da ecuación 3.5.

Por simplicidade, prescindirase de α por ser esta unha constante que se pode incluír sen perda de xeneralidade nalgunha das constantes β_j^0 das funcións suaves. Deste xeito, a estimación do modelo baséase na estimación dos $\boldsymbol{\beta}_j = (\beta_j^0, \beta_j^1, ..., \beta_j^{m_j})$, con $j = \{1, ..., p\}$.

Para estimar os anteriores parámetros, empregarase o **método de máxima verosimilitude** apoiándose no conxunto de adestramento. Como $Y_i \sim f_{\hat{y}_i}(y_i)$ son mutuamente independentes, a función de verosimilitude asociada a $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_p$ será a seguinte:

$$L(\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_p) = \prod_{i=1}^{n} f_{\hat{y}_i}(y_i). \tag{3.8}$$

Así, é directo derivar a función de log-verosimilitude:

$$l(\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_p) = \sum_{i=1}^{n} \log\left\{f_{\hat{y}_i}\left(y_i\right)\right\} = \sum_{i=1}^{n} \left\{y_i\hat{y}_i - b_i\left(\hat{y}_i\right)\right\} / a_i(\gamma) + c_i\left(\gamma, y_i\right). \tag{3.9}$$

O estimador de máxima verosimilitude (EMV) será o que maximice as ecuacións 3.8 e 3.9. É directo observar que neste caso o EMV é o que aproxima máis y_i e $\hat{y}_i$ cos datos dispoñibles.

Tendo en conta as anteriores consideracións, xorde a *devianza* como medida de bondade que permite determinar o nivel de axuste dun modelo aditivo xeneralizado aos datos de adestramento.

Definición 3.5. Defínese a *devianza* dun modelo aditivo xeneralizado como

$$D(\hat{\boldsymbol{\beta}}_{\mathbf{1}}, ..., \hat{\boldsymbol{\beta}}_{\boldsymbol{p}}) = 2\gamma\{l(\boldsymbol{\beta}_{\mathbf{1}}^{\boldsymbol{max}}, ..., \boldsymbol{\beta}_{\boldsymbol{p}}^{\boldsymbol{max}}) - l(\hat{\boldsymbol{\beta}}_{\mathbf{1}}, ..., \hat{\boldsymbol{\beta}}_{\boldsymbol{p}})\}, \tag{3.10}$$

sendo $\hat{\boldsymbol{\beta}}_{\mathbf{1}}, ..., \hat{\boldsymbol{\beta}}_{\boldsymbol{p}}$ as estimacións dos parámetros, $\beta_1^{max}, ..., \boldsymbol{\beta}_p^{max}$ os valores que maximizan a función de máxima verosimilitude e γ o parámetro de escalado da distribución.

Concretamente, $\boldsymbol{\beta}_{\mathbf{0}}^{\boldsymbol{max}}, .., \boldsymbol{\beta}_{\boldsymbol{j}}^{\boldsymbol{max}}$ correspóndense con aqueles valores cos que se verifica $y_i = \hat{y}_i$. Deste xeito, a minimización da anterior medida equivale á maximización das ecuacións 3.8 e 3.9.

Proposición 3.6. *No caso de que $Y_i \sim N(\hat{y}_i, \sigma^2)$ a devianza equivale á estimación de mínimos cadrados baseada na minimización do erro de mínimos cadrados EMC dos modelos de regresión lineais, isto é,*

$$EMC = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = D(\hat{\boldsymbol{\beta}}_{\mathbf{1}}, ..., \hat{\boldsymbol{\beta}}_{\boldsymbol{p}}). \tag{3.11}$$

Demostración. Tendo en conta que a distribución normal equivale a unha exponencial da forma 3.1, cúmprese que

$$
\begin{aligned}
l(\hat{\boldsymbol{\beta}}_{\mathbf{1}}, ..., \hat{\boldsymbol{\beta}}_{\boldsymbol{p}}) &= \sum_{i=1}^{n} \frac{y_i\hat{y}_i - \hat{y_i}^2/2}{\sigma^2} - \frac{y_i^2}{2\sigma^2} - \log(\sigma\sqrt{2\pi}) = \sum_{i=1}^{n} \frac{-y_i^2 + 2y_i\hat{y}_i - \hat{y_i}^2}{2\sigma^2} - \log(\sigma\sqrt{2\pi}) \\
&= \sum_{i=1}^{n} -\frac{(y_i - \hat{y}_i)^2}{2\sigma^2} - \log(\sigma\sqrt{2\pi}).
\end{aligned} \tag{1}
$$

Concretamente, para o caso de $\boldsymbol{\beta_0^{max}}, .., \boldsymbol{\beta_j^{max}}$, como $y_i = \hat{y}_i$, obtense que

$$
l(\boldsymbol{\beta_1^{max}}, ..., \boldsymbol{\beta_p^{max}}) = \sum_{i=1}^{n} -\frac{(y_i - y_i)^2}{2\sigma^2} - \log(\sigma\sqrt{2\pi}) = \sum_{i=1}^{n} -\log(\sigma\sqrt{2\pi}) = -n\log(\sigma\sqrt{2\pi}). \tag{2}
$$

Tendo en conta as ecuacións 1 e 2 a devianza pódese reescribir como

$$
\begin{aligned}
D(\hat{\boldsymbol{\beta}}_{\mathbf{1}}, ..., \hat{\boldsymbol{\beta}}_{\boldsymbol{p}}) &= 2\sigma^2\{l(\boldsymbol{\beta_1^{max}}, ..., \boldsymbol{\beta_p^{max}}) - l(\hat{\boldsymbol{\beta}}_{\mathbf{1}}, ..., \hat{\boldsymbol{\beta}}_{\boldsymbol{p}})\} \\
&= 2\sigma^2\{-n\log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n} -\frac{(y_i - \hat{y}_i)^2}{2\sigma^2} - \log(\sigma\sqrt{2\pi})\} \\
&= 2\sigma^2 \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{2\sigma^2} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = EMC,
\end{aligned}
$$

tal e como se quería demostrar.

□

O principal problema da estimación a través da minimización da devianza é que o modelo pode chegar a adaptarse demasiado ao conxunto de adestramento e non ser capaz de xeneralizar toda a información, isto é, que se produza sobreaxuste. Ademais, este fenómeno repercutirá negativamente na suavidade das funcións que compoñen o modelo. Para evitar o anterior problema e as súas consecuencias, pódese engadir á expresión 3.10 un novo sumando que permita penalizar esta falta de suavidade (James et al., 2021, cap. 7.7):

$$
\overbrace{D(\hat{\boldsymbol{\beta}}_{\mathbf{1}}, ..., \hat{\boldsymbol{\beta}}_{\boldsymbol{p}})}^{devianza} + \overbrace{\sum_{j=1}^{p} \lambda_j \int \phi_j''(x_{ij})^2 dx}^{penalización}, \tag{3.12}
$$

sendo $\lambda_1, ..., \lambda_j$ os parámetros de penalización da suavidade que permiten controlar o compromiso entre o axuste aos datos de referencia e a suavidade do modelo.

Como se pode observar na ecuación 3.12, o termo de penalización está composto pola suma das integrais das derivadas segundas ó cadrado de cada unha das funcións suaves do modelo, ponderadas a través duns parámetros. Concretamente, $\int \phi_j''(x)^2 dx$ permite penalizar curvaturas excesivamente elevadas, tanto positivas como negativas.

De feito, se a función ϕ_j é moi suave, a súa derivada ϕ'_j será moi próxima a ser constante o que fará que $\int \phi''_j(x)^2 dx$ tome valores pequenos. De forma análoga, se ϕ_j é unha función moi cambiante, $\int \phi''_j(x)^2 dx$ tomará valores máis grandes. Polo tanto, este termo de penalización incentivará a construción de funcións ϕ_j máis suaves.

O nivel de suavidade destas funcións dependerá dos valores de λ_j. Por un lado, con $\lambda_j = 0$, o termo de penalización correspondente non terá ningún peso en 3.12, sendo este igual a $D(\hat{\boldsymbol{\beta}}_{\mathbf{1}}, ..., \hat{\boldsymbol{\beta}}_{\boldsymbol{p}})$. Por outro lado, se $\lambda_j \to \infty$ este termo de penalización terá un maior impacto na expresión, polo que a función asociada estará perfectamente suavizada. Para valores de λ_j intermedios obteranse funcións que aproximan os puntos de adestramento ao mesmo tempo que manteñen a súa suavidade nun determinado nivel.

Tendo en conta todo o anterior, a elección dos parámetros de penalización da suavidade cobra unha vital importancia na construción do modelo. Unha forma de determinar os valores que mellor se axustan é empregando técnicas de validación cruzada. Estas fundamentaranse en minimizar a expresión 3.12, comparando así para que valores dos parámetros se obteñen mellores resultados.

A pesar de ser unha aproximación sinxela e escalable, o emprego de técnicas de validación cruzada pode ter un alto custo computacional. De feito, a estimación dos parámetros de penalización da suavidade é a operación para a construción do modelo máis esixente nestes termos. Ademais, está directamente ligada ao número de variables de entrada do problema xa que cada unha destas conleva o cálculo dun parámetro de suavizado independente (Hastie and Tibshirani (1990)).

A minimización da devianza ou da súa adaptación a un termo de penalización da suavidade, serven como regra base para a definición de algoritmos que estimen os valores dos parámetros dos GAMs. Dous deles son *iteratively re-weighted least squares* (IRLS) e a súa variante coa penalización de suavizado *penalized iteratively re-weighted least squares* (PIRLS). Ambos son algoritmos iterativos fundamentados na minimización das expresións 3.10 e 3.12 a través do uso de distintas aproximacións numéricas para representar o modelo como un GLM e posteriormente resolvelo (Wood, 2017, p. 105-107, 251).

Por outro lado, o algoritmo *backfitting* tamén permite axustar modelos aditivos xeneralizados. Esta aproximación flexibiliza a representación das funcións suaves, permitindo casi a totalidade de técnicas de representación e modelado das funcións suaves. Este desenvólvese en máis detalle en (Wood, 2017, p. 318-320).

3.3. Aplicación a problemas de clasificación

Aínda que os anteriores apartados están máis enfocados a problemas de regresión, os GAMs tamén poden ser empregados cando a variable de resposta é cualitativa. Para iso, é necesario adaptar esta última ao anterior modelo. A aproximación máis común para realizar isto fundaméntase na idea de modelar a probabilidade de que Y pertenza a unha determinada categoría en lugar da propia variable resposta directamente.

A **regresión loxística** permite modelar a anterior probabilidade. Esta parte do suposto de que a variable resposta é binaria (por simplicidade considerarase que pode tomar os valores 0 ou 1) e que ademais segue unha distribución binomial. O anterior modelo permite coñecer a relación entre $p(\boldsymbol{x}_i) = Pr(Y = 1|X = \boldsymbol{x}_i)$ e $\boldsymbol{x}_i$ a través da función logit:

$$\text{logit}(p(\boldsymbol{x}_i)) = \ln\left(\frac{p(\boldsymbol{x}_i)}{1 - p(\boldsymbol{x}_i)}\right) = \ln(p(\boldsymbol{x}_i)) - \ln(1 - p(\boldsymbol{x}_i)). \tag{3.13}$$

Deste xeito, o modelo xeneralizado aditivo da ecuación 3.5 pode reescribirse empregando logit como función de enlace, é dicir,

$$\ln\left(\frac{p(\boldsymbol{x}_i)}{1 - p(\boldsymbol{x}_i)}\right) = \beta_0 + \sum_{j=1}^{p} \phi_j(x_{ji}). \tag{3.14}$$

Así, todos os procedementos e características presentados en apartados anteriores consérvanse no referente a variables cualitativas.

Adicionalmente, é posible extender a anterior aproximación para problemas de clasificación nos que existen máis de dúas categorías. Para realizar este proceso, dividirase o problema en múltiples subproblemas de clasificación binarios. Estes poderán ser tratados a través de dúas metodoloxías distintas: *One Versus One* (OVO) e *One Versus All* (OVA)Raziff et al. (2017). Todo o anterior proceso é coñecido como **regresión loxística multinomial**.

3.3.1. One Versus One

Esta técnica fundaméntase en dividir o problema en tantos subproblemas como posibles combinacións de dúas categorías existan. O número de modelos xerados a través de procedemento é de $\frac{d(d-1)}{2}$, que depende directamente da cantidade de clases do problema d. Para a construción dos anteriores, soamente se empregarán as observacións do conxunto de adestramento que pertenzan a unha das dúas clases consideradas. Deste xeito, será posible comparar a probabilidade dous a dous de que unha observación pertenza a unha das categorías seleccionadas.

Para cada un dos subproblemas xerados, ademais de seleccionar o conxunto de adestramento segundo as clases consideradas, tamén é preciso adaptar o GAM. A pesar de que este mantería a estrutura da expresión 3.14, a probabilidade que estima debe adaptarse ao subproblema concreto. Así, considerando $d_1, d_2 \in \{1, .., d\}$ os índices das súas categorías dun subproblema dado, deberá tomarse $p(\boldsymbol{x}_i) = Pr(Y = d_1|X = \boldsymbol{x}_i)$ ou ben $p(\boldsymbol{x}_i) = Pr(Y = d_2|X = \boldsymbol{x}_i)$. En calquera dos dous casos, $1 - p(\boldsymbol{x}_i)$ fará referencia á probabilidade de que $\boldsymbol{x}_i$ pertenza á outra clase.

A categoría coa que se etiquetará cada observación será aquela que saia elixida nun maior número de subproblemas, é dicir, a que obteña a maior probabilidade máis veces. No caso de empate entre clases, analizaranse os resultados obtidos nos subproblemas nas que estén implicadas para determinal cal é a que ten unha probabilidade maior ca o resto.

3.3.2. One Versus All

OVA é unha técnica de adaptación de problemas multiclase a binarios no que se xeran tantos subproblemas como categorías. Como o seu nome indica, fundaméntase en contrastar a probabilidade de pertencer a unha determinada categoría e a de formar parte de calquera outra.

Ao igual ca no caso de OVO, construirase un modelo para cada un dos d subproblemas xerados. Deste xeito, para o k-ésimo problema, no que se compara a probabilidade da clase k fronte ao resto, será necesario considerar $p(\boldsymbol{x}_i) = Pr(Y = k|X = \boldsymbol{x}_i)$ na ecuación 3.14.

Neste caso, o conxunto de adestramento empregado para obter o modelo estará en todos os casos formado pola totalidade das observacións de referencia. Así, será posible estimar a probabilidade de pertencer a cada unha das categorías de todas as observacións. A estas asignaráselles aquela clase coa que se estime unha probabilidade maior.

O principal inconvinte desta técnica é que a súa precisión pode verse afectada se o número de observacións de cada clase non está balanceado. Pola propia natureza desta metodoloxía, se por exemplo unha clase ten unha cantidade de observacións notablemente menor no conxunto de adestramento, esta terá máis predisposición a ter unha probabilidade menor ca outras que teñan máis observacións. Polo tanto, é recomendable analizar a distribución das clases no conxunto de adestramento para determinar se o anterior pode ser un inconvinte á hora de realizar predicións.

Capítulo 4

Aplicación dos métodos de clasificación

Este capítulo centrarase na aplicación e análise do rendemento dos métodos de clasificación estudados neste traballo. Con este obxectivo primeiro indicarase a forma de particionar o conxunto de referencia para analizar o seu rendemento, as métricas empregadas e finalmente, os resultados obtidos coa aplicación dos anteriores métodos.

4.1. Particionado dos datos

Para o problema de clasificación de masas forestais disponse dun conxunto de referencia formado por 7519 puntos con clase coñecida. Cada unha destas enmárcase nun dos sete tipos de combustible do modelo *Prometheus*. Concretamente, a distribución dos datos é a que se presenta Cadro 4.1.

Tipo 1	**Tipo 2**	**Tipo 3**	**Tipo 4**	**Tipo 5**	**Tipo 6**	**Tipo 7**
503	1149	1021	925	1158	1301	1462

Cadro 4.1: Táboa coa distribución en clases dos puntos de referencia.

Tal e como se observa, non hai o mesmo número de puntos de cada clase, de feito, entre algunhas delas existe unha diferencia notable. Polo tanto, pódese considerar que o conxunto de referencia non é balanceado, aspecto que se deberá ter en consideración para realizar unha división dos datos equitativa.

Como xa se introduciu no Capítulo 1, unha aproximación moi estendida para estimar o rendemento dun modelo é particionando os datos en dous conxuntos disxuntos: o conxunto de adestramento e o de validación. Mentres que o primeiro será empregado na construción do modelo, o segundo permitirá probalo sobre novos datos e analizar a súa eficacia e rendemento.

Para realizar este particionado de datos, presentáronse distintas técnicas de validación cruzada. Entre elas, a validación cruzada de k-iteracións destaca como a aproximación idónea para este caso de estudo. De feito, esta reduce a dependencia das métricas de rendemento dos conxuntos de datos empregados para adestramento e test ao mesmo tempo que mantén un custo computacional máis axustado que, por exemplo, a validación cruzada clásica.

Tendo en conta o anterior, implementarase a validación cruzada de 5-iteracións, dividindo o conxunto de refencia en cinco subgrupos co mesmo número de elementos. O anterior realizouse garantizando a presenza da mesma cantidade de puntos de cada clase en cada un deles. A súa implementación preséntase no Apéndice A.

4.2. Métricas de rendemento

Para analizar o rendemento dos distintos modelos, será necesario definir as métricas que se empregarán. Estas, inicialmente, introduciranse para o caso de clasificación binaria para posteriormente xeneralizalas ao caso de modelos multiclase.

4.2.1. Métricas para clasificación binaria

Nun problema binario, existen catro escenarios distintos que se poden producir na realización de predicións. Supoñendo un problema coas clases "positiva" e "negativa", os posibles escenarios recóllense no Cadro 4.2.

	Predición positiva	Predición negativa
Clase positiva	Verdadeiro Positivo (VP)	Falso Negativo (FN)
Clase negativa	Falso Positivo (FP)	Verdadeiro Negativo(VN)

Cadro 4.2: Táboa cos posibles escenarios da clasificación binaria.

Tendo en conta o Cadro 4.2, xorde a primeira métrica, denominada *accuracy*.

Definición 4.1. Dado un modelo de clasificación binario, defínese a súa precisión como

$$acc = \frac{VP + VN}{VP + FN + FP + VN},$$

sendo VP, VN, FN e FP o número de situacións dos tipos do Cadro 4.2.

A anterior medida pode levar a resultados erróneos, especialmente cando o número de observacións do conxunto de test non están balanceadas. Por exemplo, se o 20 % das observacións son dunha clase e o 80 % da outra, un clasificador podería devolver sempre a segunda clase obtendo

un *accuracy* de 0,8 a pesar de non realizar predición ningunha. Deste xeito, introdúcense dúas novas medidas que servirán como base para estimar de forma máis precisa o rendemento dun modelo: o *recall* ou exhaustividade e a precisión.

Definición 4.2. Dado un modelo de clasificación binario, defínese a súa exhaustividade como

$$rcl = \frac{VP}{VP + FN},$$

sendo VP e FN o número de verdadeiros positivos e de falsos negativos respectivamente.

A anterior medida pódese interpretar como a exactitude ou rendemento do modelo para predicir a clase positiva. Por outro lado, a precisión permite medir a proporción de positivos que realmente pertencen a esa clase.

Definición 4.3. Dado un modelo de clasificación binario, defínese a súa precisión como

$$pcs = \frac{VP}{VP + FP},$$

sendo VP e FP o número de verdadeiros e falsos positivos respectivamente.

Unha medida que permite resumir todo o anterior e medir o rendemento dun modelo é o *F-1 Score*. Concretamente, esta defínese como se segue:

Definición 4.4. Dado un modelo de clasificación binario, defínese o *F1-Score* como

$$F1 = 2 \cdot \frac{pcs \cdot rcl}{pcs + rcl},$$

sendo *pcs* e *rcl* a precisión e a exhaustividade do modelo, respectivamente.

A anterior métrica pódese entender como a media harmónica entre *pcs* e *rcl* dun modelo, sendo un compromiso entre ambas cantidades (Grandini et al. (2020)).

4.2.2. Adaptación á clasificación multiclase

No caso de problemas de clasificación con máis de dúas clases, é preciso adaptar as anteriores medidas. Todas elas, basearanse na *matriz de confusión* asociada ás predicións feitas. Esta pódese entender como unha extensión dos tipos de erros do Cadro 4.2.

Definición 4.5. Dado un modelo de clasificación, defínese a *matriz de confusión* asociada ás súas predicións como

$$\boldsymbol{MC} = \overbrace{\begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1d} \\ c_{21} & c_{22} & \cdots & c_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ c_{d1} & c_{d2} & \cdots & c_{dd} \end{pmatrix}}^{\text{Clases reais}} \Bigg\} \text{Predicións}$$

sendo d o número de categorías distintas e $c_{ij} = \sum_{k=1}^{m} I(y_k = i) \cdot I(\hat{y}_k = j)$ con m o número de observacións e $i, j = \{1, ..., d\}$ indicadores de clases.

É directo observar que $\sum_{k=1}^{d} c_{ik}$, a suma da fila i, representa o número de observacións etiquetadas coa clase i-ésima. Da mesma maneira, $\sum_{k=1}^{d} c_{kj}$, a suma da columna j, presenta aquelas cuxa categoría real é a j-ésima. Así, a intersección das anteriores filas e columnas c_{ij} indican o número de observación etiquetadas coa categoría i cando a súa clase real é a j.

A matriz de confusión servirá como base para a adaptación ao caso multiclase das métricas presentadas no anterior apartado. Concretamente, estas obteranse reducindo o problema a un de clasificación binaria para cada unha das clases. Así considerarase a categoría seleccionada como "positiva"e o resto conxuntamente como a "negativa". De forma equivalente ao Cadro 4.2, dada unha clase k, existirán catro posibles situacións:

	Predición k	Predición $\neq k$
Clase k	$VP_k = c_{kk}$	$FN_k = \sum_{i \neq k} c_{ik}$
Clase $\neq k$	$FP_k = \sum_{j \neq k} c_{kj}$	$VN_k = \sum_{i \neq k} \sum_{j \neq k} c_{ij}$

Cadro 4.3: Táboa cos posibles escenarios para unha clase k.

Tendo en conta o cadro 4.3, a *accuracy* para o caso multiclase, definirase da seguinte forma.

Definición 4.6. Dado un modelo de clasificación multiclase cunha matriz de confusión asociada MC, defínese a súa *accuracy* como

$$acc = \left(\sum_{k=1}^{d} \frac{VP_k + VN_k}{VP_k + FN_k + FP_k + VN_k} \right) / d = \frac{\sum_{k=1}^{d} VP_k + VN_k}{m \cdot d},$$

sendo m o número total de predicións.

Para o caso do resto de medidas, existen dúas aproximacións para calculalas, coñecidas como promedios *macro* e o *micro*. O primeiro deles baséase en obter as métricas de forma independente para cada clase e a partir destas últimas calcular a medida global, a través do seu promedio. Isto

permite que as métricas de todas as clases teñan o mesmo peso no cómputo global, independentemente do número de observacións de cada unha delas. Neste contexto, defínense a precisión e exhaustividade parciais da seguinte forma (Grandini et al. (2020)).

Definición 4.7. Dado un modelo de clasificación multiclase cunha matriz de confusión asociada MC, defínese a súa exhaustividade parcial con respecto da clase k como

$$rcl_k = \frac{VP_k}{VP_k + FN_k},$$

con $k = \{1, ..., d\}$.

Definición 4.8. Dado un modelo de clasificación multiclase cunha matriz de confusión asociada MC, defínese a súa precisión parcial con respecto da clase k como

$$pcs_k = \frac{VP_k}{VP_k + FP_k},$$

con $k = \{1, ..., d\}$.

O *F1-Score* neste caso, coñécese como *macro F1-Score*. A súa definición é a seguinte.

Definición 4.9. Dado un modelo de clasificación multiclase defínese o *macro F1-Score* como

$$maF1 = \left(\sum_{k=1}^{d} \frac{2 \cdot pcs_k \cdot rcl_k}{pcs_k \cdot rcl_k} \right) / d,$$

sendo pcs_k e rcl_k a precisión e exhaustividade parciais de cada clase.

Por outro lado, a aproximación *micro* fundaméntase en calcular directamente de xeito global as medidas, sen cuantificar previamente as súas parciais. Á diferenza da aproximación *macro*, o peso de cada clase no cálculo das métricas estará directamente ligado á proporción de observacións desa categoría, o que pode ser un inconvinte en caso de conxuntos de datos con clases non balanceadas.

Definición 4.10. Dado un modelo de clasificación multiclase cunha matriz de confusión asociada MC, defínese a súa *micro-exhaustividade* como

$$miRcl = \frac{\sum_{k=1}^{d} VP_k}{\sum_{k=1}^{d} (VP_k + FP_k)}.$$

Definición 4.11. Dado un modelo de clasificación multiclase cunha matriz de confusión asociada MC, defínese a súa *micro-precisión* como

$$miPcs = \frac{\sum_{k=1}^{d} VP_k}{\sum_{k=1}^{d} (VP_k + FP_k)}.$$

Tendo en conta as anteriores definicións, introdúcese o *micro F1Score* como se segue:

Definición 4.12. Dado un modelo de clasificación multiclase cunha matriz de confusión asociada MC, defínese a *micro F1-Score* como

$$miF1 = \frac{2 \cdot miRcl \cdot miPcs}{miRcl + miPcs},$$

sendo $miPcs$ e $miRcl$ a micro-precisión e micro-exhaustividade respectivamente.

Proposición 4.13. *A micro-precisión, a micro-exhaustividade e a micro-F1Score son equivalentes.*

Demostración. Tendo en conta as anteriores definicións, pódese escribir que

$$miRcl = \frac{\sum_{k=1}^{d} VP_k}{\sum_{k=1}^{d}(VP_k + FP_k)} = \frac{\sum_{k=1}^{d} c_{kk}}{\sum_{j=1}^{d}\sum_{i=1}^{d} c_{ji}} = \frac{\sum_{k=1}^{d} c_{kk}}{m},$$

sendo m o número total de predicións. Por outro lado,

$$miPcs = \frac{\sum_{k=1}^{d} VP_k}{\sum_{k=1}^{d}(VP_k + FP_k)} = \frac{\sum_{k=1}^{d} c_{kk}}{\sum_{i=1}^{d}\sum_{j=1}^{d} c_{ij}} = \frac{\sum_{k=1}^{d} c_{kk}}{m}.$$

Tendo en conta as anteriores expresións, a *micro F1-Score* calcúlase como:

$$miF1 = 2 \cdot \frac{miRcl \cdot miPcs}{miRcl + miPcs} = 2 \cdot \frac{miPcs^2}{2 \cdot miPcs} = \frac{\sum_{k=1}^{d} c_{kk}}{m},$$

tal e como se quería demostrar.

□

Adicionalmente, é directo observar que as anteriores medidas equivalen á *taxa de acerto de proba* introducida no Capítulo 1.

4.3. Resultados e comparativa

Para analizar o rendemento dos clasificadores estudados, implementáronse no software estatístico R (R Core Team (2020)) e calculáronse o *accuracy*, a *taxa de erro de proba* e a *macro-F1Score*. As anteriores obtivéronse como o promedio das obtidas en cada unha das $k = 5$ iteracións da técnica de validación cruzada empregada.

Para obter as distintas métricas e medir o rendemento dos clasificadores, fíxose uso do conxunto de referencia dispoñible para o problema de clasificación da vexetación. Este segue o modelo

Prometheus e conta con $d = 7$ clases que indican o tipo de combustible da vexetación. Neste caso as variables preditorias empregadas son a intensidade e a altura real de cada un dos puntos ($p = 2$). Tendo en conta o anterior, implementáronse os seguintes clasificadores:

- **Naive Bayes**. Baseado no cálculo da función de densidade conxunta da intensidade e a altura dos puntos de referencia. Foi implementado empregando funcións de tipo kernel para a estimación da función de densidade. Con este propósito, fíxose uso do paquete *naiveBayes* (Majka (2019)) de R.
- **KNN**. Realizouse a súa implementación para nove valores distintos $\mathcal{K}$, empregando a distancia euclidiana (DE) e a de Mahalanobis (DM) para o cálculo dos veciños e a técnica de desempate aleatorio. Neste caso farase uso da función *knn* do paquete *class* (Venables and Ripley (2002)) de R.
- **GAM**. Adaptouse á clasificación multiclase a través da aproximación *OVO* (regresión loxística multinomial) e estimáronse as funcións de suavizado a través da combinación lineal de funcións base. Todo isto realizouse coa función *gam* do paquete *mgcv* (Wood (2011)) de R.
- **Clasificador aleatorio**. Adicionalmente implementouse un clasificador que lle asigna a cada clase a probabilidade en función da súa proporción no conxunto de test. Este servirá como base para determinar o rendemento dos métodos estudados.

O código asociado aos anteriores métodos e ao cálculo das métricas para cada un deles atópase no apéndice A. Concretamente, os resultados obtidos son os presentados no Cadro 4.4.

	Taxa acerto	***Accuracy***	**Macro F1-Score**
Naive Bayes	0,510	0,860	0,493
KNN (DE) $K = 15$	0,662	0,904	0,633
KNN (DM) $K = 15$	0,661	0,903	0,631
GAM	0,621	0,892	0,596
Aleatorio	0,153	0,758	0,143

Cadro 4.4: Resultados da aplicación dos clasificadores.

Tal e como se observa, os métodos estudados obteñen resultados notablemente mellores que o clasificador aleatorio. Todos eles teñen unha taxa de acerto e macro F1-Score superior ou moi próximo a 0,5. Ademais, a accuracy é superior a 0,8 nos métodos estudados. De todas formas, esta medida é convinte analizala en referencia á obtida co clasificador aleatorio, xa que depende directamente do número de clases implicadas, que neste problema é 7. Concretamente, esta

tenderá a ser máis elevada por ser a media aritmética das accuracies parciais para cada clase con respecto ao resto de categorías. De todas formas, polos valores das outras dúas métricas, pódese considerar que o rendemento dos clasificadores é globalmente bo.

Por un lado, *Naive Bayes* é o método dos estudados que en xeral obtén peores resultados, aínda que continúa sendo aceptable. O anterior pode estar ligado á necesidade de estimar a distribución da densidade para a súa implementación, o que pode levar a errores se non se empregan estimadores axeitados.

Por outro lado, os outros dous métodos presentan globalmente mellores resultados, sendo estes bastante parellos. O método GAM a pesar de ter métricas cun valor máis baixo que as implementacións de KNN, obtén un rendemento notablemente bo, sendo en todos os casos moi próximo ou superior a 0,6. De novo, neste caso o seu rendemento podería mellorarse estudando as características que deben ter as funcións de suavizado, axustando os seus graos de liberdade ou impoñendo ou quitando restricións.

Finalmente, o método de K Veciños Próximos, a pesar da súa sinxeleza, é o que obtén a puntuación máis alta nas tres métricas en calquera das súas implementacións. A diferenza entre o uso da distacia euclídea e da de Mahalanobis é practicamente nula nas medidas de rendemento, aínda que é nesta última na que se obteñen uns resultados lixeiramente peores. Isto indica que a correlación das variables de entrada non é un aspecto suficientemente representativo no conxunto de referencia.

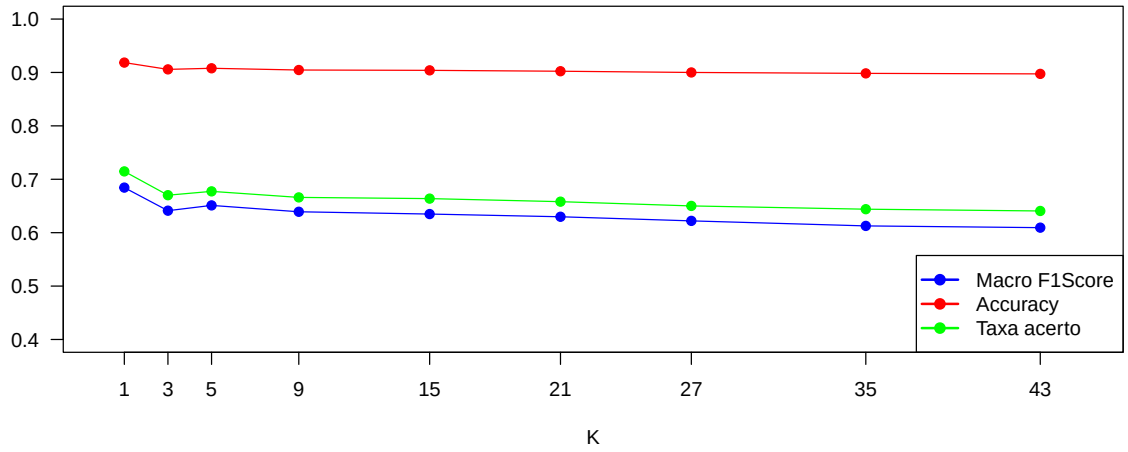

Figura 4.1: Variación das métricas en función do valor de K empregando a distancia euclídea. En vermello preséntase a accuracy, en azul a macro F1-Score e en verde a taxa de acerto.

Ademais de tomar $K = 15$, tamén se calcularon as métricas para máis valores deste parámetro. Estes represéntanse no gráfico da Figura 4.1 en función do valor deste parámetro para a distancia euclídea, sendo para a de Mahalanobis practicamente idénticos.

Na Figura 4.1 pódese apreciar como se obteñen mellores resultados nas métricas para valores de K máis pequenos. Este fenómeno pode ter relación cos datos de referencia empregados e como se obtiveron. De todas formas, ao clasificar novas observacións non é seguro que esta situación se repita. Por este motivo é preferible empregar un valor de K intermedio que teña un bo compromiso entre nesgo e varianza. Neste caso, podería seleccionarse $K = 15$ tendo en conta a anterior consideración e que segue mantendo uns valores elevados en todas as métricas.

4.4. Exemplos de aplicación

Os anteriores clasificadores aplicáronse sobre novas nubes de puntos LiDAR, etiquetándoas segundo o modelo *Prometheus*. Os resultados son visibles nas Figuras 4.2, 4.3 e 4.4, representando as cores verdes os tipos do 1 ao 4 de máis claro a máis escuro, e os tipos 5, 6 e 7 amarelo, laranxa e vermello, respectivamente. Adicionalmente, os puntos marróns fan referencia a puntos do chan que non son vexetación.

As Figuras 4.2, 4.3 e 4.4 obtivéronse facendo uso do programa de visualización *Olivia*, que tamén inclúe unha ferramenta de reetiquetado de puntos (Blanco (2022)). Esta permite corrixir e mellorar clasificacións previas, cambiando as estiquetas de puntos previamente seleccionados.

Polo tanto, os clasificadores implementados neste traballo non só serven para etiquetar puntos, senón que tamén dan soporte a anterior ferramenta. De feito, a través dela é posible obter clasificacións máis axustadas á realidade e incluso incrementar o conxunto de puntos de referencia empregado para a construción dos clasificadores. Así, estes métodos, constitúen unha base da que partir para obter de forma iterativa melloras na fiabilidade e calidade das clasificacións.

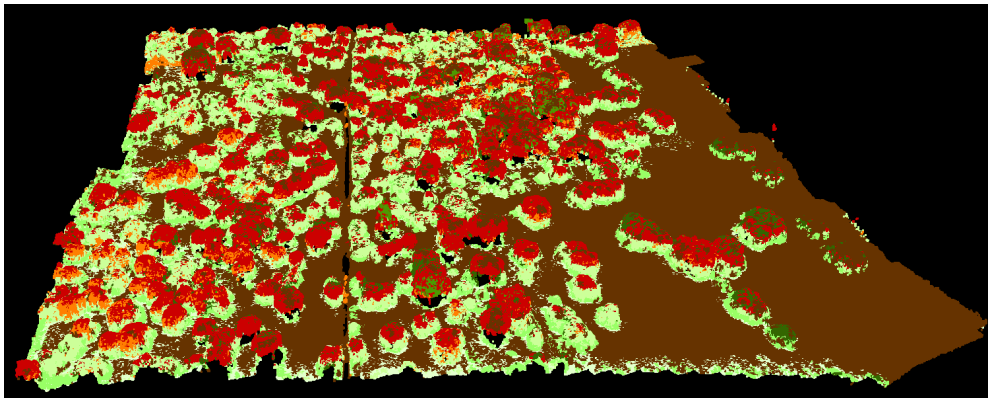

Figura 4.2: Clasificación dunha nube de puntos empregando Naive Bayes.

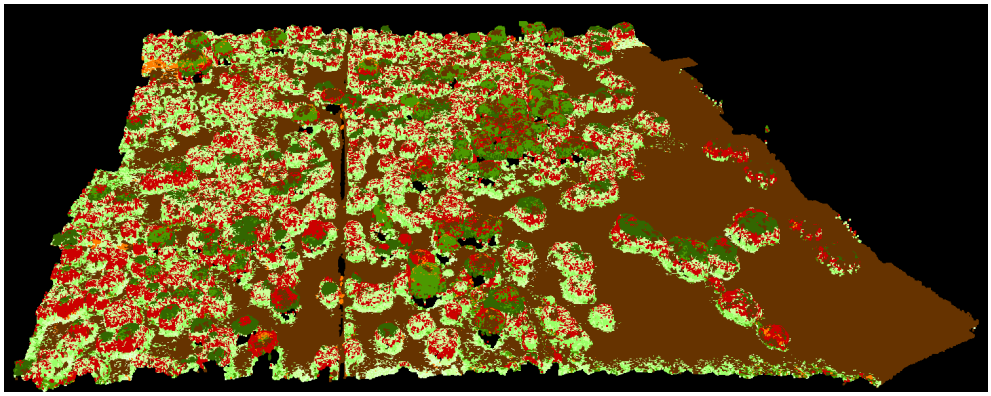

Figura 4.3: Clasificación dunha nube de puntos usando KNN coa distancia euclídea e $K = 15$.

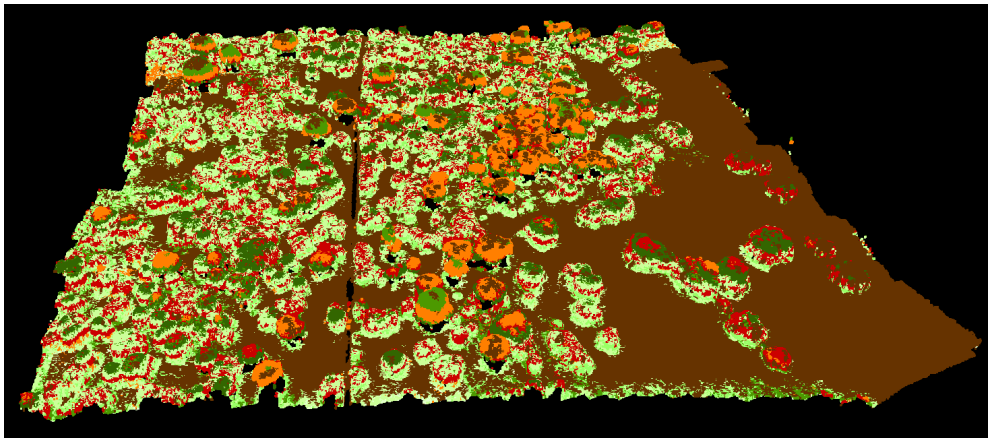

Figura 4.4: Clasificación dunha nube de puntos usando un modelo aditivo xeneralizado.

Anexo A

ANEXO I: Código R

Neste capítulo presentarase o código R empregado para obter as métricas de rendemento indicadas no Capítulo 4.

A.1. Particionado de datos

O primeiro paso foi ler e almacenar as observacións de referencia e dividilas nos 5 grupos que se empregará para a validación cruzada de 5 iteracións.

```
#Abrimos os ficheiros cos datos e almacenámolos
type1<-read.table("puntos/1.csv",header = T,sep = ' ')
type2<-read.table("puntos/2.csv",header = T,sep = ' ')
type3<-read.table("puntos/3.csv",header = T,sep = ' ')
type4<-read.table("puntos/4.csv",header = T,sep = ' ')
type5<-read.table("puntos/5.csv",header = T,sep = ' ')
type6<-read.table("puntos/6.csv",header = T,sep = ' ')
type7<-read.table("puntos/7.csv",header = T,sep = ' ')
numberClass<-rep(0,7)

types<-list(type1,type2,type3,type4,type5,type6,type7)

#Xuntamos todos os puntos no mesmo data frame e engadimos a columna tipo
trainPoints<-cbind(type1[,3:4],Type=rep(1,nrow(type1)))
#obtemos o numero de elementos de cada tipo dos grupos
numberClass[1]<-round(nrow(trainPoints[trainPoints$Type==1,])/k)
```

```
for(i in 2:7){
  trainPoints<-rbind(trainPoints,cbind(types[[i]][,3:4],
  Type=rep(i,nrow(types[[i]]))))
  numberClass[i]<-round(nrow(trainPoints[trainPoints$Type==i,])/k)
}

set.seed(123)
k<-5  #Numero folds
d<-7  #Numero de clases

trainPoints[ , 'fold'] <- NA
total<-nrow(trainPoints)

for(i in 1:k){
  for( j in 1:d){
    if(i!=k){
          points<-trainPoints$Type==j & is.na(trainPoints$fold)
          index<-1:total
          index<-index[points]
          index<-sample(index,numberClass[j])
          trainPoints[index,4]<-i
        }else{
          trainPoints[is.na(trainPoints$fold) & trainPoints$Type==j,4]<-i
        }
  }
}
```

A.2. Naive Bayes

A continuación preséntase a implementación do clasificador Naive Bayes especificada no Capítulo 4.

```
library(naivebayes)

trainPoints$Type <- as.factor(trainPoints$Type)
```

```
acerto<-rep(0,5)
accuracyBayes<-rep(0,k)
precisionBayes<-matrix(0,k,7)
recallBayes<-matrix(0,k,7)
maF1Bayes<-rep(0,k)

#NAIVE BAYES CONXUNTO
for(i in 1:k){
  #Obter clasificacions para cada fold
  index<-1:nrow(trainPoints)
  nb.fit <- naive_bayes(Type ~ z + I, data=trainPoints[trainPoints$fold!=i,-4],
        usekernel = T)
  nb.predict <- predict(nb.fit, trainPoints[trainPoints$fold==i,1:2])

  #Matriz de confusión
  (tab1 <- table(nb.predict, trainPoints[trainPoints$fold==i,3]))
  total<-sum(tab1)

  #Taxa de acerto test
  acerto[i] <- sum(diag(tab1)) / total

  for(j in 1:d){
    vp<-tab1[j,j]                                #Verdadeiros positivos
    fp<-sum(tab1[j,-j])          #Falsos positivos
    fn<- sum(tab1[-j,j])         #Falsos negativos
    vn<-total-vp-fp-fn                   #Verdadeiros negativos

    accuracyBayes[i]=accuracyBayes[i]+vp+vn
    precisionBayes[i,j]<-vp/(vp+fp)
    recallBayes[i,j]<-vp/(vp+fn)
  }
  accuracyBayes[i]=accuracyBayes[i]/(total*d)
  maF1Bayes[i]<-mean(2*precisionBayes[i,]*recallBayes[i,]/
                (precisionBayes[i,]+recallBayes[i,]))
}

mean(accuracyBayes)
```

```
mean(acerto)
mean(maF1Bayes)
```

A.3. K Veciños Próximos

Implementouse o algoritmo de K Veciños Próximos con $K = \{1, 3, 5, 9, 15, 21, 27, 35, 43\}$. O código asociado ao clasificador que emprega a distancia euclídea preséntase a continuación.

```
library(class)

#Estandarizamos datos
standardized.X <- scale (trainPoints[,1:2])
K<-c(1,3,5,9,15,21,27,35,43)
acertoKNN <- matrix(0,length(K),k)
maF1KNN<-matrix(0,length(K),k)
accuracyKNN<-matrix(0,length(K),k)

row.names(errorsKNN)<-paste("K:", as.character(K), collapse = NULL)
row.names(maF1KNN)<-paste("K:", as.character(K), collapse = NULL)
row.names(accuracyKNN)<-paste("K:", as.character(K), collapse = NULL)

for(i in 1:k){
  precisionKNN<-matrix(0,k,7)
  recallKNN<-matrix(0,k,7)

  for(j in 1:length(K)){
    knn.pred <- knn ( standardized.X[trainPoints$fold!=i,],
        standardized.X[trainPoints$fold==i,], trainPoints[trainPoints$fold!=i,3],
           k = K[j])
    #Matriz confusion
    (tab1 <- table(knn.pred, trainPoints[trainPoints$fold==i,3]))
    total<-sum(tab1)
    #Taxa de acerto test
    acertoKNN[j,i] <- sum(diag(tab1)) / total

    for(l in 1:d){
```

```
      vp<-tab1[l,l]          #Verdaderos positivos
      fp<-sum(tab1[l,-l])    #Falsos positivos
      fn<- sum(tab1[-l,l])   #Falsos negativos
      vn<-total-fp-vp-fn     #Verdaderos negativos

      accuracyKNN[j,i]= accuracyKNN[j,i]+vp+vn
      precisionKNN[i,l]<-vp/(vp+fp)
      recallKNN[i,l]<-vp/(vp+fn)
    }

  maF1KNN[j,i]<-mean(2*precisionKNN[i,]*recallKNN[i,]/
               (precisionKNN[i,]+recallKNN[i,]))
  accuracyKNN[j,i] = accuracyKNN[j,i]/(total*7)
  }
}

rowMeans(acertoKNN)
rowMeans(maF1KNN)
rowMeans(accuracyKNN)
```

Para o caso da distancia de Mahalanobis, o código é análogo ao anterior, mais é preciso executar antes os seguintes comandos para computar a anterior distancia.

```
pc <- prcomp(as.matrix(trainPoints[,1:2]))
data <- scale(pc$x)
```

Deste xeito, empregarase a variable *data* en lugar de *standarized.X* para a realización das predicións.

A.4. GAM

Nesta sección preséntase o código asociado á implementación do GAM indicado no Capítulo 4.

```
#GAM-OVO
library(mgcv)
```

```
trainPoints$Type <- as.numeric(trainPoints$Type)
combinations <- matrix(0,d*(d-1)/2,2)
cont<-1
for(c1 in 1:d){
  if(c1<d){
    for(c2 in (c1+1):d){
      combinations[cont,1] <- c1
      combinations[cont,2] <- c2
      cont= cont +1
    }
  }
}

acertoGAM <-rep(0,k)
accuracyGAM<-rep(0,k)
precisionGAM<-matrix(0,k,7)
recallGAM<-matrix(0,k,7)
maF1GAM<-rep(0,k)

for(i in 1:k){
  fold <- trainPoints[trainPoints$fold==i,-4]
  rest <- trainPoints[trainPoints$fold!=i,-4]

  predictions<-matrix(0,nrow(fold),d)

  for(j in 1:nrow(combinations)){
    c1<-combinations[j,1]
    c2<-combinations[j,2]

    trainPts <- rest[rest$Type==c1 |  rest$Type==c2,]

    trainPts[trainPts$Type==c1,3] <- rep(1,sum(trainPts$Type==c1))
    trainPts[trainPts$Type==c2,3] <- rep(0,sum(trainPts$Type==c2))

    #FASE DE TRAINING
    gam.fits <- gam (Type ~ s(z) + s(I),  data = trainPts,
                family = binomial, method = "ML")
```

```
    #TEST
    predicts <- predict.gam(gam.fits,newdata = fold[,1:2], type="response")
    predictions[predicts>0.5,c1] = predictions[predicts>0.5, c1] +1
    predictions[predicts<=0.5,c2] = predictions[predicts<=0.5, c2] +1
  }
  #Obtemos a columna maxima de cada predicion
  types<-max.col(predictions)

  #Calculamos erros asociados
  (tab1 <- table(types, fold[,3]))
  #Taxa de ACERTO test
  acertoGAM[i] <- sum(diag(tab1)) / sum(tab1)
  total<-sum(tab1)

  for(l in 1:7){
    vp<-tab1[l,l]                #Verdaderos positivos
    fp<-sum(tab1[l,-l])  #Falsos positivos
    fn<- sum(tab1[-l,l]) #Falsos negativos
    vn<-total-vp-fp-fn   #Verdaderos negativos

    accuracyGAM[i] = accuracyGAM[i]+vp+vn
    precisionGAM[i,l]<-vp/(vp+fp)
    recallGAM[i,l]<-vp/(vp+fn)
  }
    maF1GAM[i]<-mean(2*precisionGAM[i,]*recallGAM[i,]/
                (precisionGAM[i,]+recallGAM[i,]))
    accuracyGAM[i] = accuracyGAM[i]/(d*total)
}

mean(acertoGAM)
mean(maF1GAM)
mean(accuracyGAM)
```

A.5. Clasificador aleatorio

Para obter as métricas do clasificador aleatorio soamente foi necesario calcular a matriz de confusión asociada tendo en conta os pesos de cada clase no conxunto de referencia. O código asociado é o que se presenta a continuación.

```
library(plyr)
v<-count(trainPoints,"Type")$freq
v<-as.matrix(v)
tab1<-v%*%t(v)

accuracy<-0
precision<-rep(0,d)
recall<-rep(0,d)
total<-sum(tab1)
for(l in 1:7){
  vp<-tab1[l,l]
  fp<-sum(tab1[l,-l])
  fn<- sum(tab1[-l,l])
  vn<-total-vp-fp-fn

  accuracy = accuracy+vp+vn
  precision[l]<-vp/(vp+fp)
  recall[l]<-vp/(vp+fn)
}

accuracy<-accuracy/(d*total)
maF1<-mean(2*precision*recall/(precision+recall))
taxaAcerto<-sum(diag(tab1))/total
```

Bibliografía

Arroyo, L. A., Pascual, C., and Manzanera, J. A. (2008). Fire models and methods to map fuel types: The role of remote sensing. *Forest Ecology and Management*, 256(6):1239–1252.

ASPRS (2008). *LAS Specification Version 1.2*. ASPRS.

Berrar, D. (2019). Bayes' theorem and Naive Bayes classifier. *Encyclopedia of Bioinformatics and Computational Biology*, 1:403–412.

Blanco, A. (2022). Ferramenta de clasificación da vexetación semiautomática a través de LiDAR. Traballo de Fin de Grao. Escola Técnica de Enxeñaría. Universidade de Santiago de Compostela.

Carrizosa, E. and Romero Morales, D. (2013). Supervised classification and mathematical optimization. *Computers & Operations Research*, 40:150–165.

De Bock, K. W., Coussement, K., and Van den Poel, D. (2010). Ensemble classification based on generalized additive models. *Computational Statistics & Data Analysis*, 54(6):1535–1546.

De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D. (2000). The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1–18.

García-Cimarras, A., Manzanera, J. A., and Valbuena, R. (2021). Analysis of mediterranean vegetation fuel type changes using multitemporal LiDAR. *Forests*, 12(3).

Grandini, M., Bagli, E., and Visani, G. (2020). Metrics for multi-class classification: An overview.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized Additive Models*. Chapman & Hall, 1st edition.

Hastie, T. J., Tibshirani, R. J., and Friedman, J. (2017). *The Elements of Statistical Learning*. Springer, 2nd edition.

James, G., Witten, D., Hastie, T. J., and Tibshirani, R. J. (2021). *An Introduction to Statistical Learning*. Springer, 2nd edition.

Majka, M. (2019). *naivebayes: High Performance Implementation of the Naive Bayes Algorithm in R*. R package version 0.9.7.

Martínez, J., Lorenzo, O. G., Vilariño, D. L., Pena, T. F., Cabaleiro, J. C., and Rivera, F. F. (2018). A developer-friendly "Open LiDAR Visualizer and Analyser" for point clouds with 3D stereoscopic view. *IEEE Access*, 6:63813–63822.

Novo, A., González-Jorge, H., Martínez-Sánchez, J., and Lorenzo, H. (2020). Remote sensing approach to evaluate post-fire vegetation structure. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B3-2020:1031–1038.

Quirós Rosado, E. (2015). *Introducción a la Fotogrametría y Cartografía aplicadas a la Ingeniería Civil*. Universidad de Extremadura.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ramasubramanian, K. and Singh, A. (2016). *Machine Learning Using R*. Apress.

Raziff, A. R. A., Sulaiman, M., Mustapha, N., and Perumal, T. (2017). Single classifier, OvO, OvA and RCC multiclass classification method in handheld based smartphone gait identification. *AIP Conference Proceedings*, 1891(1):020009.

Srivastava, N. and Rao, S. (2016). Learning-based text classifiers using the Mahalanobis distance for correlated datasets. *International Journal of Big Data Intelligence*, 3:18–27.

Venables, W. and Ripley, B. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0.

Wood, S. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)*, 73(1):3–36.

Wood, S. (2017). *Generalized Additive Models: An introduction with R*. CRC Press, 2nd edition.

Xiang, S., Nie, F., and Zhang, C. (2008). Learning a Mahalanobis distance metric for data clustering and classification. *Pattern Recognition*, 41(12):3600–3612.

Zambom, A. Z. and Dias, R. (2013). A review of kernel density estimation with applications to econometrics. *International Econometric Review*, 5(1):20 – 42.

Zamora-Martínez, M. C. (2017). La tecnología LiDAR, herramienta útil para el estudio de la biodiversidad. *Revista mexicana de ciencias forestales*, 8(39).