UNIVERSIDADE D
COIMBRA

Carlos Roberto Bastos Lacerda

CLASSIFICAÇÃO SUPERVISIONADA

Árvores de Decisão e Florestas Aleatórias

Dissertação no âmbito do Mestrado em Matemática, Ramo de Estatística, Otimização e Matemática Financeira, orientada pelo investigador doutorado Luís Miguel Dias Pinto e apresentada ao Departamento de Matemática da Faculdade de Ciências e Tecnologia.

Setembro de 2023

Classificação Supervisionada
Árvores de Decisão e Florestas Aleatórias

Carlos Roberto Bastos Lacerda

UNIVERSIDADE D
COIMBRA

Mestrado em Matemática

Master in Mathematics

Dissertação de Mestrado | MSc Dissertation

Setembro 2023

Agradecimentos

Esta dissertação de mestrado, de longo período de desenvolvimento, teve contributo essencial de múltiplas pessoas e instituições para sua concreção, os quais gostaria de agradecer, com profundo reconhecimento, nesta secção.

Agradeço ao Doutor Luís Miguel Dias Pinto, pelo suporte e orientação ao longo deste trajeto, pelo aconselhamento durante a investigação, de rigor exemplar e grande simpatia, o qual evidencio minha imensa gratidão.

À minha mãe Verônica Souza e meu padrasto Fernando Sabino, por todo o apoio, força e carinho em minha vida, cujos sem suas presenças muito do que alcancei não seria possível, em especial esta tese. A toda minha família, à minha querida avó Milsa Bastos, como meus amigos e colegas de curso que me acompanharam e incentivaram neste percurso.

À coordenadora de mestrado, Doutora Sílvia Alexandra Alves Barbeiro, pela sua primordial assistência e encaminhamento deste mestrado. À banca examinadora, composta pelo Doutor José Luís Esteves dos Santos, pelo Doutor Stéphane Louis Clain e também pelo Doutor Luís Pinto, por disporem a revisar este trabalho, colocarem ponderações pertinentes e contribuírem para o desfecho deste processo.

À Universidade de Coimbra, ao Departamento de Matemática e a seu corpo docente e de funcionários, por dedicarem os meios, bases teórico-práticas e esforços que possibilitaram este momento.

Expresso meu agradecimento a todos que auxiliaram e foram importantes para o completar deste caminho, edificando-me como pessoa, sem os quais esta jornada não teria se concretizado.

Resumo

Esta tese de mestrado introduz os métodos árvores de decisão e florestas aleatórias para resolver o problema de data mining denominado por classificação supervisionada. Consideremos um dataset de pacientes covid-19 (ou objetos) classificados em duas classes consoante a evolução para óbito ou recuperado. Partindo de um conjunto de características (ou atributos) dos pacientes, como a idade e a pré-existência de outras doenças, o objetivo do problema de classificação supervisionada é encontrar uma função (ou classificador) que estabelece uma relação entre atributos dos pacientes e as respetivas classes. A utilidade fundamental de um classificador reside na possibilidade de classificar um novo objeto, por exemplo, prever a evolução de um novo paciente covid-19. O método árvores de decisão distingue-se pela sua interpretabilidade e performance competitiva, particularmente quando utilizado na técnicas ensemble floresta aleatória.

A tese está organizada da seguinte forma. O primeiro capítulo apresenta o problema de classificação supervisionada, seguindo-se dois capítulos dedicados aos principais fundamentos teóricos dos métodos estatísticos árvores de decisão e floresta aleatória. O quarto capítulo ilustra o potencial prático dos métodos usando um conjunto de dados públicos de pacientes com covid-19, e terminamos no capítulo cinco com algumas conclusões.

Abstract

This master's thesis introduces decision trees and random forest methods to solve the data mining problem of supervised classification. Let us consider a dataset of covid-19 patients (or objects) classified into two classes based on whether they died or recovered. From a set of patient characteristics (or attributes), such as age and the pre-existence of other diseases, supervised classification aims at developing a function (or classifier) that establishes a relationship between patient attributes and the respective classes. The primary utility of a classifier is the ability to classify a new object, e.g., predicting the evolution of a new covid-19 patient. The decision tree method is known for its interpretability and competitive performance, particularly when combined with ensemble techniques like random forest.

This thesis is organized as follows. The first chapter introduces the supervised classification problem, followed by two chapters on the theoretical foundations of decision trees and random forests. The fourth chapter illustrates the practical potential of the methods using a public dataset of covid-19 patients, and we finish in chapter five with some conclusions.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Classificação supervisionada

1.1 Introdução

De modo geral o data mining pode ser descrito como o processo de extração de informação relevante de um conjunto de dados. A informação extraída depende por vezes de relações complexas, de difícil perceção direta mesmo para especialistas da área, sendo necessário recorrer a métodos e modelos matemáticos. O conceito, e algumas das ferramentas utilizadas em data mining não são recentes, porém, este campo de investigação tem atualmente uma grande visibilidade devido à quantidade de informação armazenada. Apesar do seu potencial, métodos de data mining (e.g., neural networks) tendem a sofrer de falta de interpretabilidade, ou seja, é praticamente impossível analisar e compreender os pressupostos subjacentes aos resultados obtidos. A ausência de interpretabilidade inviabiliza qualquer validação pelos especialistas, dificultando a aceitação destas ferramentas. Em áreas como medicina e finanças, é pouco provável que decisões críticas sejam tomadas tendo por base resultados que não se compreendem. A nível europeu, as orientações éticas para uma inteligência artificial (IA) de confiança destacam a interpretabilidade como um fator primordial [20]. É referido o seguinte: "A explicabilidade técnica exige que as decisões tomadas por um sistema de IA possam ser compreendidas e rastreadas por seres humanos... Sempre que um sistema de IA tenha um impacto significativo na vida das pessoas, deverá ser possível solicitar uma explicação adequada do respetivo processo de tomada de decisões."

Este seminário é focado no processo de data mining conhecido por classificação supervisionada. O problema consiste em encontrar modelos matemáticos que façam a distinção de objetos em classes. Damos particular destaque aos problemas de classificação binária (i.e., com duas classes), por exemplo, classificar pacientes covid-19 em recuperados ou óbitos. Para a definição de um modelo de classificação supervisionada é necessário um conjunto de dados que inclua características (ou atributos) dos objetos, bem como informação sobre a respetiva classe. Considerando novamente o exemplo dos pacientes covid-19, este conjunto deve conter uma lista de pacientes, a classe de cada um, óbito ou recuperado, bem como uma série de atributos considerados relevantes, por exemplo, a idade e doenças pré-existentes. De referir que a etapa de escolha e pré-processamento dos atributos dos objetos pode ter um impacto significativo na performance dos modelos de classificação. Após a denominada fase de treino, os modelos podem ser utilizados para classificar novos objetos com classe desconhecida. Além disso, se o modelo de classificação for interpretável, é possível estabelecer relações potencialmente

interessantes entre os atributos dos objetos e a respetiva classe. Mantendo o exemplo covid-19, além de classificar um novo paciente covid-19 em recuperado ou óbito, é possível identificar quais são as doenças pré-existentes que representam um risco acrescido de óbito. Este tipo de informação poderá ser útil no combate à doença. Neste seminário, abordamos o método de classificação árvores de decisão. Uma das principais vantagens deste método é precisamente a excelente interpretabilidade. Outra vantagem das árvores de decisão é a sua capacidade para lidar diretamente com atributos quantitativos (ou numéricos), com valores em $\mathbb{R}$, e com atributos qualitativos (ou categóricos), com valores simbólicos. Por exemplo, a idade de um paciente covid-19 é um atributo numérico e a diabetes é um atributo simbólico, tem o valor sim se o paciente é diabético e o valor não no caso contrário.

Além de neural networks e árvores de decisão, alguns dos métodos de classificação supervisionada mais utilizados incluem naive Bayes e svm (support vector machine). Em [17], é analisado um problema de classificação onde se pretende prever se um paciente é saudável ou portador de doença arterial coronária. O algoritmo árvore de decisão proposto apresentou melhor performance que métodos alternativos baseados em neural networks, svm e naive Bayes. A possibilidade de a árvore de decisão ser facilmente interpretada pela comunidade médica é destacado como outro ponto positivo do algoritmo.

1.2 Modelo de Bayes

Seja Ω o domínio dos objetos a classificar. Cada objeto de Ω é representado por um vetor de atributos $x = (x_1, \ldots, x_p)$, com $x_j \in X_j$, $j = 1, \ldots, p$, e $X = X_1 \times \ldots \times X_p$ o espaço dos atributos.

Definição 1. *Um atributo X_j é designado por numérico se $X_j = \mathbb{R}$ e por categórica se X_j assume um conjunto finito de valores sem uma ordem natural.*

Seja $\mathbb{L} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ o denominado conjunto treino com $y_n \in Y$, $n = 1, \ldots, N$, e $Y = \{c_1, \ldots, c_J\}$ um conjunto de finito de classes. O problema de classificação supervisionada pode ser encarado como o problema de encontrar uma função $\phi : X \longrightarrow Y$ num conjunto de treino $\mathbb{L}$, tal que a sua previsão $\phi(x)$ seja a melhor possível em determinado sentido.

Definição 2. *Um classificador é uma função $\phi : X \longrightarrow Y$, com X o espaço dos atributos e Y um conjunto finito de classes.*

Do ponto de vista estatístico podemos ver X e Y como variáveis aleatórias e definir $P(X = x, Y = y)$ como a probabilidade de X tomar o valor x e Y o valor y quando retiramos de modo aleatório um objeto do universo Ω. Neste sentido, a noção de encontrar a melhor previsão possível $\phi(x)$ pode ser definida como encontrar um classificador que minimiza o erro esperado da previsão.

Definição 3. *O erro esperado da previsão, também denominado por erro de generalização ou erro teste, de um classificador $\phi_{\mathbb{L}}$ é definido por*

$$E(\phi_{\mathbb{L}}) = \mathbb{E}_{X,Y}(L(Y, \phi_{\mathbb{L}}(X))), \tag{1.1}$$

com $\mathbb{L}$ é o conjunto treino utilizado para construir o classificador $\phi_{\mathbb{L}}$, L uma função perda que mede a discrepância entre os dois argumentos e $\mathbb{E}$ a esperança matemática ou valor esperado.

Note-se que o erro de generalização (1.1) está definido no domínio de todos os objetos possíveis Ω. De facto, o objetivo é obter um classificador que seja preciso não apenas no conjunto treino $\mathbb{L}$ mas também no conjunto de todos os dados possíveis não observados $X \times Y \setminus \mathbb{L}$. Em problemas de classificação uma função perda comum é a denominada função zero-um

$$L(Y, \phi_{\mathbb{L}}(X)) = \mathbb{I}(Y \neq \phi_{\mathbb{L}}(X)), \tag{1.2}$$

com $\mathbb{I}$ a função indicadora, ou seja, a função (1.2) atribui o valor zero a classificações corretas e penaliza de modo idêntico com o valor um as classificações incorretas. Neste caso, o erro de generalização (1.1) reduz-se à probabilidade do modelo efetuar uma classificação incorreta

$$E(\phi_{\mathbb{L}}) = \mathbb{E}_{X,Y}(\mathbb{I}(Y \neq \phi_{\mathbb{L}}(X))) = P(Y \neq \phi_{\mathbb{L}}(X)) \tag{1.3}$$

Vamos assumir, em termos teóricos, que a distribuição de probabilidade $P(X,Y)$ é conhecida. Neste caso, o erro (1.3) de um classificador ϕ_B pode ser reescrito como

$$\mathbb{E}_{X,Y}(L(Y, \phi_B(x))) = \mathbb{E}_X(\mathbb{E}_{Y|X}(L(Y, \phi_B(x)))), \tag{1.4}$$

e o classificador que minimiza (1.4) é dado por

$$\begin{aligned} \phi_B(x) &= \arg\min_{y \in Y} \mathbb{E}_{Y|X=x}(L(Y,y)) \\ &= \arg\min_{y \in Y} \mathbb{E}_{Y|X=x}(\mathbb{I}(Y,y)) \\ &= \arg\min_{y \in Y} P(Y \neq y | X = x) \\ &= \arg\max_{y \in Y} P(Y = y | X = x). \end{aligned} \tag{1.5}$$

Ou seja, o melhor classificador possível consiste em escolher sistematicamente a classe mais provável $y \in \{c_1, \ldots, c_J\}$ dado $X = x$. O classificador ϕ_B é denominado na literatura por classificador de Bayes.

Na prática, a distribuição de probabilidade $P(Y|X)$ é, em geral, desconhecida, ou seja, a obtenção do classificador ótimo de Bayes é impraticável, restando a opção de tentar encontrar o melhor classificador possível. Assumindo que o comportamento de um classificador é controlado pelo denominado vetor de hiperparâmetros θ, o objetivo é encontrar o valor θ^* para o qual o classificador possui o menor erro de generalização. Porém, outra questão se coloca, nomeadamente o cálculo do erro de generalização. Em geral existe um único conjunto de dados que tem de ter utilizado para treinar/otimizar o classificador e para estimar o erro de generalização. Uma opção imediata e simples que tem, contudo, o defeito de fornecer uma estimativa demasiado otimista do erro de generalização, consiste em utilizar o mesmo conjunto de dados para treinar/otimizar o classificador e estimar o erro de generalização. A estimativa obtida desta forma é designada por erro de resubstituição e, para a função perda zero-um (1.2), é definido por

$$\bar{E}(\phi_{\mathbb{L}}) = \frac{1}{N} \sum_{(x,y) \in \mathbb{L}} \mathbb{I}(Y \neq \phi_{\mathbb{L}}(X)), \tag{1.6}$$

com N a dimensão do conjunto treino $\mathbb{L}$.

Um protocolo alternativo mais adequado consiste em separar de modo aleatório o conjunto de dados disponíveis em conjuntos de treino, validação e teste. Os hiperparâmetros do classificador são fixos analisando o erro no conjunto de validação do modelo treinado no conjunto treino. Após a etapa de treino/otimização, uma estimativa para o erro de generalização (ou erro teste) é calculada utilizando o conjunto teste. É também usual combinar esquemas de k-fold cross-validation com a estratégia de divisão em treino, validação e teste. Por exemplo, a etapa treino/validação em esquema k-fold cross-validation consiste em dividir o conjunto reservado ao treino e à validação em k conjuntos disjuntos, $\mathbb{L}_1, \ldots, \mathbb{L}_k$ e estimar o erro de validação como a média do erro de validação nos conjuntos $\mathbb{L}_k$ dos modelos treinados nos restantes conjuntos $\mathbb{L} \setminus \mathbb{L}_k$. Esta abordagem tem a vantagem de todos os pares (x, y) nos conjuntos treino e validação serem utilizados na estimativa do erro de validação.

1.3 Medidas de erro

Consideremos um problema de classificação binária com a classe 0, designada por classe negativa, e a classe 1, designada por classe positiva. Seja N o número total de objetos e N_0 e N_1 o número de objetos na classe 0 e 1. Utilizamos a Tabela 1.1 para introduzir as noções de verdadeiro positivo (VP), verdadeiro negativo (VN), falso positivo (FP) e falso negativo (FN). Note-se que N_0=VN+FP e N_1=VP+FN.

		Classe real		Total
		Classe 1	Classe 0	
Classe prevista	Classe 1	VP	FP	VP+FP
	Classe 0	FN	VN	FN+VN
	Total	VP+FN	FP+VN	N

Tabela 1.1 Matriz de confusão para um problema de classificação binária.

De acordo com (1.6), o erro associado à função perda zero-um é igual ao número de classificações incorretas a dividir pelo número de objetos, ou seja, $(FN+FP)/N$. Algumas das medidas alternativas usualmente utilizadas são:

- Acurácia: mede as classificações corretas; (VP+VN)/N;
- Sensibilidade: mede as classificações corretas da classe positiva; VP/N_1;
- Especificidade: mede as classificações corretas da classe negativa; VN/N_0;

A acurácia reflete o número de classificações corretas e fornece uma visão geral da qualidade do classificador, particularmente quando o dataset é balanceado (i.e., quando N_0 está próximo de N_1). A análise da sensibilidade e da especificidade é igualmente relevante em determinados contextos. Consideremos o problema de classificar a gravidade de determinada doença, com a classe 1 a representar doença grave e a classe 0 doença pouco grave. Neste contexto é desejável que a sensibilidade seja elevada, ou seja, que o método não cometa o erro de classificar doença grave como pouco grave (FN). No caso de um classificador de investimento financeiro, com a classe 1 a indicar

bom investimento e a classe 0 mau investimento, é desejável que a especificidade seja elevada, ou seja, que o método não cometa o erro de classificar mau investimento como bom (FP).

A noção do balanceamento do dataset introduzida no parágrafo anterior é uma questão importante e um campo bastante ativo de investigação. O problema, algo comum em situações reais, surge quando a classe mais importante é uma das classes menos representadas. Quando os dados têm uma distribuição assimétrica, ou seja, quando existem diferenças significativas na frequência relativa das várias classes, os classificadores tendem a privilegiar as classes mais representativas. Este comportamento afeta significativamente a sua performance. Uma das abordagens possíveis para lidar com este problema consiste na utilização de funções perda pesadas, isto é, que penalizam de forma mais acentuada classificações incorretas da classe menos representada. Outra abordagem consiste em utilizar técnicas de amostragem para balancear as classes (e.g., geração de amostras sintéticas das classes sub-representadas) [11]. Outro problema recorrente das bases de dados é a existência de informação incompleta, o que levou ao desenvolvimento de vários métodos de imputação [12, 21].

Capítulo 2

Árvores de decisão

2.1 Introdução

Consideremos o problema de classificação apresentado na Figura 2.1 à esquerda. Existem duas classes de objetos, a classe 0 representada por pontos a azul e a classe 1 representada por pontos a laranja. Os objetos possuem dois atributos: x1, no eixo dos xx, e x2, no eixo dos yy. O objetivo do método árvores de decisão é estabelecer regras sobre os atributos $x1$ e $x2$ que permitam classificar os objetos na classe 0 ou na classe 1. Uma possibilidade é representada pelas linhas tracejadas a preto, note-se

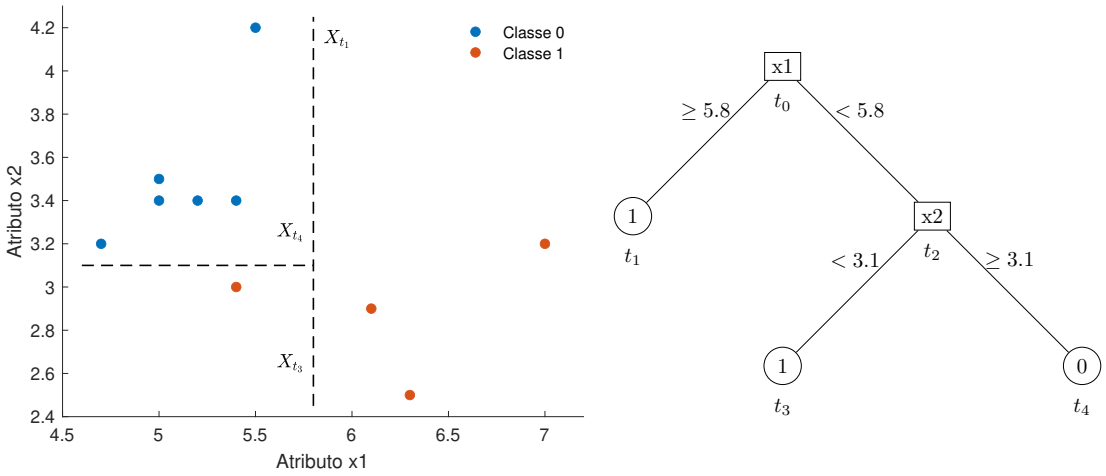

Figura 2.1. Problema de classificação com duas classes (à esquerda). As linhas a tracejado ilustram a partição $X_{t_4} \cup (X_{t_1} \cup X_{t_3})$ induzida pelos nós folha t_4, t_3 e t_1 na árvore de decisão à direita.

que a partição induzida pelas linhas permite separar todos os objetos na classe 0 ou na classe 1. Ou seja, neste caso o classificador não comete nenhum tipo de erro, não existem objetos da classe 0 classificados na classe 1 ou objetos da classe 1 classificados na classe 0. A árvore associada a esta partição é apresentada na Figura 2.1 à direita.

Tal como ilustrado na Figura 2.1 à direita, uma árvore de decisão é uma estrutura constituída por nós $(t_0, \ldots, t_4)$ e arestas que conectam nós adjacentes. Os nós internos (representados por quadrados) indicam um conjunto de testes sobre os atributos dos objetos, as arestas indicam o resultado dos testes, e os nós folha (representadas por círculos), indicam a classe atribuída aos objetos em resultado dos

sucessivos testes. Analisemos o seu funcionamento. No nó raiz t_0 realizamos o teste $x1 \geq 5.8$ ou $x1 < 5.8$ (linha vertical no gráfico à esquerda). O ramo à esquerda do nó t_0 representa o resultado $x1 \geq 5.8$, e o valor 1 no nó folha adjacente t_1 (i.e., o filho esquerdo de t_0) indica que todos os objetos que satisfazem esta regra são classificados na classe 1. Por outro lado, o ramo à direita do nó raiz t_0 representa o resultado $x1 < 5.8$. Este ramo está ligado ao nó interno t_2 (i.e., o filho direito de t_0) onde realizamos o teste $x2 < 3.1$ ou $x2 \geq 3.1$ (linha horizontal no gráfico à esquerda). O ramo à direta do nó t_2 representa o resultado $x2 \geq 3.1$, e o valor 0 no nó folha adjacente t_4 indica que todos os objetos que satisfazem o conjunto de regras $x1 < 5.8$ e $x2 \geq 3.1$ são classificados na classe 0. Por outro lado, o ramo à esquerda do nó t_2 representa o resultado $x2 < 3.1$, e o valor 0 no nó folha adjacente t_3 indica que todos os objetos que satisfazem o conjunto de regras $x1 < 5.8$ e $x2 < 3.1$ são classificados na classe 0.

2.2 O algoritmo CART

Analisando o problema de classificação na Figura 2.1, é evidente que existem regras alternativas, do tipo das apresentadas na árvore de decisão, que permitem obter uma classificação igualmente correta. Entre todas as árvores de decisão existentes, faz sentido favorecer a árvore menos complexa com menor número de nós. Este critério favorece a interpretabilidade da árvore e a sua capacidade de generalização, isto é, a capacidade da árvore classificar novos objetos. Independentemente do problema, é amplamente reconhecido que a capacidade de generalização de um modelo matemático diminui com o aumento do número de parâmetros. Modelos mais complexos são mais propensos a sofrer de overfitting, fenómeno caracterizado por um ajuste excessivo aos dados de treino com diminuição da capacidade de generalização. Porém, tal como demonstrado em [19], o problema de encontrar a árvore que minimiza o número de nós necessários para classificar um conjunto de objetos é um problema NP-completo, recorrendo-se a métodos heurísticos para encontrar árvores de decisão quase ótimas.

Um dos algoritmos heurísticos mais utilizados na construção de árvores de decisão é o CART [7], um algoritmo que realiza sucessivas divisões localmente ótimas no espaço dos atributos. De um modo vago, a estratégia heurística utilizada no CART consiste em escolher as divisões que localmente maximizam o decréscimo de uma medida de impureza. As divisões são definidas apenas por um atributo e uma regra de divisão desse atributo. Por exemplo, na árvore da Figura 2.1 à direita, a divisão associada ao nó t_0 é definida pelo atributo $x1$ e pela divisão do espaço dos atributos em dois sub-conjuntos, $\{(x1,x2) : x1 \geq 5.8\}$ e $\{(x1,x2) : x1 < 5.8\}$. Outros algoritmos heurísticos clássicos para a construção de árvores de decisão são o ID3 [32] e o C4.5 [34]. Na sua formulação original duas das diferenças entre estes algoritmos são o tipo de árvore construída, binária ou não, e a medida de impureza utilizada. Alternativas aos métodos heurísticos, baseadas em métodos de otimização mais robustos, são apresentadas em [1, 31].

Pseudocódigo recursivo

Consideremos um problema de classificação com conjunto treino $\mathbb{L} = \{(x_1, c_1), \ldots, (x_N, c_N)\}$. Na primeira iteração o algoritmo CART procura a divisão ótima s^* que maximiza o decréscimo de uma

medida de impureza, cria o nó raiz t_0 associado ao atributo ótimo e particiona o conjunto treino $\mathbb{L}$ em $\mathbb{L}_{t_E} \cup \mathbb{L}_{t_D}$. O conjunto $\mathbb{L}_{t_E}$ assume o papel de conjunto treino da sub-árvore esquerda e $\mathbb{L}_{t_D}$ o papel de conjunto treino da sub-árvore direta. Este processo repete-se recursivamente para a sub-árvore esquerda e para a sub-árvore direita. Quando um determinado critério de paragem é satisfeito, é criado um nó folha sendo atribuída uma determinada classe.

Algoritmo 1 Pseudocódigo recursivo do algoritmo CART.

```
1:  function CART_RECURSIVO(conjunto treino 𝕃)
2:      if critério de paragem then
3:          criar nó folha e atribuir classe
4:          return nó folha
5:      else
6:          encontrar divisão s* que maximiza o decréscimo da medida de impureza
7:          criar nó t associado a s*
8:          definir a partição 𝕃 = 𝕃_{t_E} ∪ 𝕃_{t_D}
9:          sub_árvore_esq(t) = CART_RECURSIVO(conjunto treino 𝕃_{t_E})
10:         sub_árvore_dir(t) = CART_RECURSIVO(conjunto treino 𝕃_{t_D})
11:         return nó t
```

Nas secções seguintes discutimos as componentes principais do Algoritmo 1, nomeadamente: atribuição de uma classe a um nó folha, critérios de paragem e o processo de divisão, que engloba a noção de medida de impureza e o critério de divisão.

Regra de atribuição de classes

Numa árvore de decisão ϕ, minimizar o erro de generalização com função perda zero-um (1.4) é equivalente a minimizar o erro de generalização local associado a todos os nós folha. Denotando por $\tilde{\phi}$ o conjunto dos nós folha, vem

$$E(\phi) = \sum_{t \in \tilde{\phi}} P(X \in X_t)(1 - P(Y = y | X \in X_t)). \tag{2.1}$$

Assim, o erro mínimo é obtido atribuindo a cada nó folha a classe mais provável de X_t, com X_t o sub-conjunto associado ao nó folha t,

$$\bar{y}_t = \arg\max_{c \in y} P(Y = c | X \in X_t). \tag{2.2}$$

Como a distribuição de probabilidade $P(X,Y)$ é desconhecida, estimamos o valor de $P(Y = c | X \in X_t)$ pelo valor da proporção N_{t_c}/N_t, com N_{t_c} o número de objetos da classe c em $\mathbb{L}_t$ e N_t o número de objetos em $\mathbb{L}_t$. Utilizando esta estimativa em (2.2), obtemos

$$\bar{y}_t = \arg\max_{c \in y} \frac{N_{t_c}}{N_t}, \tag{2.3}$$

o que equivale a dizer que atribuímos aos nós folha a classe mais representada no conjunto $\mathbb{L}_t$.

Considerando em (2.1) a estimativa utilizada em (2.2) para $P(Y = y|X \in X_t)$ e estimando $P(X \in X_t)$ pela proporção N_t/N, obtemos

$$\begin{aligned}\bar{E}(\phi) &= \sum_{t\in\tilde{\phi}} \frac{N_t}{N}\left(1 - \frac{N_{c_t}}{N_t}\right) \\ &= \frac{1}{N}\sum_{t\in\tilde{\phi}} N_t - N_{c_t} \\ &= \frac{1}{N}\sum_{x,y\in\mathbb{L}} \mathbb{I}(y \neq \phi(x)) \\ &= \bar{E}_{\mathbb{L}}(\phi). \end{aligned} \tag{2.4}$$

A igualdade (2.4) significa que quando se considera a regra de atribuição da classe mais representada, minimizar o erro de generalização equivale a minimizar o erro de resubstituição (1.6). Outra propriedade desta regra de atribuição é que o erro de resubstituição de uma árvore diminui sempre que se dividi um nó folha.

Proposição 1. *Para qualquer divisão não vazia de um nó folha $t \in \tilde{\phi}$ em dois nós t_E e t_D, a árvore ϕ' obtida definindo a classe dos novos nós de acordo com (2.3) verifica*

$$\bar{E}_{\mathbb{L}}(\phi) \geq \bar{E}_{\mathbb{L}}(\phi'),$$

existindo igualdade se $\bar{y}_t = \bar{y}_{t_E} = \bar{y}_{t_D}$.

Demonstração. Obtemos sucessivamente,

$$\begin{aligned}\bar{E}_{\mathbb{L}}(\phi) &\geq \bar{E}_{\mathbb{L}}(\phi') \\ \sum_{t\in\tilde{\phi}} p(t)(1 - p(\bar{y}_t|t)) &\geq \sum_{t\in\tilde{\phi}'} p(t)(1 - p(\bar{y}_t|t)) \\ p(t)(1 - p(\bar{y}_t|t)) &\geq p(t_E)(1 - p(\bar{y}_{t_E}|t_E)) + p(t_D)(1 - p(\bar{y}_{t_D}|t_D)) \\ \frac{N_t}{N}(1 - \max_{c\in y}\frac{N_{ct}}{N_t}) &\geq \frac{N_{t_E}}{N}(1 - \max_{c\in y}\frac{N_{ct_E}}{N_{t_E}}) + \frac{N_{t_D}}{N}(1 - \max_{c\in y}\frac{N_{ct_D}}{N_{t_D}}) \\ N_t - \max_{c\in y} N_{ct} &\geq N_{t_E} - \max_{c\in y} N_{ct_E} + N_{t_D} - \max_{c\in y} N_{ct_D} \\ \max_{c\in y} N_{ct} &\leq \max_{c\in y} N_{ct_E} + \max_{c\in y} N_{ct_D} \\ \max_{c\in y}(N_{ct_E} + N_{ct_D}) &\leq \max_{c\in y} N_{ct_E} + \max_{c\in y} N_{ct_D}\end{aligned}$$

com a última desigualdade a ser verdadeira uma vez que $\max_{c\in y} N_{ct_E}$ (t_D) é maior ou igual que o termo esquerdo N_{ct_E} (que o termo direito N_{ct_E}) no lado esquerdo da equação. A igualdade verifica-se se as classes maioritárias são as mesmas em t, t_E e t_D.

■

A proposição anterior garante que o erro de resubstituição de uma árvore é minimal quando não é possível efetuar mais divisões. Em particular, se existir apenas um objeto em cada folha, o erro será igual a zero.

Critérios de paragem

O erro de resubstituição não é, em geral, um bom indicador do erro de generalização. Pelo contrário, a construção de árvores profundas/complexas que, segundo a Proposição 1.6, garantem a diminuição do erro no conjunto treino traduz-se normalmente numa diminuição da sua performance num conjunto teste independente. De modo a evitar este fenómeno, é necessário definir critérios de paragem adequados que controlem o crescimento excessivo da árvore. A escolha ideal deve apresentar um balanço entre árvores profundas, associadas a overfitting, e árvores rasas, associadas a underfitting (i.e., quando o modelo não consegue capturar as relações existentes no conjunto treino).

Em árvores de decisão os critérios de paragem são em geral definidos sobre os principais hiperparâmetros da árvore, nomeadamente, o número mínimo de objetos a partir do qual um nó é declarado nó folha, o número máximo de nós e a profundidade máxima da árvore. Estes hiperparâmetros podem ser otimizados conforme o dataset utilizando um esquema treino-validação-teste. Estratégias que visam reduzir a complexidade de uma árvore são denominadas pruning, podendo ser enquadradas em dois grupos, pre-pruning (implementadas na forma de critérios de paragem), ou post-pruning. O post-pruning consiste em construir uma árvore complexa e em seguida remover nós até encontrar uma árvore que apresente um equilíbrio entre performance e complexidade.

Regras de divisão

A regra de divisão utilizada no Algoritmo 1 depende do tipo de atributo selecionado, numérico ou categórico.

Definição 4. *Para árvores binárias denotamos por divisão s de um nó t uma partição de X_t em dois sub-conjuntos não vazios, disjuntos, associados aos dois nós filho, t_E e t_D.*

O número de divisões binárias para um conjunto com N elementos cresce exponencialmente segundo a fórmula $2^{N-1}-1$. Assim, de modo a obter um problema computacionalmente viável, é desejável reduzir o espaço de possibilidades. Se o atributo X_j associada à regra de divisão ótima s^* é numérico a partição binária de $\mathbb{L}_t$ é definida por

$$\mathbb{L}_{t_E}^{\mu} = \{(x,y) : (x,y) \in \mathbb{L}_t, x_j > \alpha\}, \qquad \mathbb{L}_{t_D}^{\mu} = \{(x,y) : (x,y) \in \mathbb{L}_t, x_j \leq \alpha\} \tag{2.5}$$

com α um threshold adequado. Neste caso o número de partições possíveis é igual a $N^* - 1$, com N^* o cardinal do conjunto dos valores únicos de $X_j \in \mathbb{L}_t$. Sejam x_j e x_{j+1} dois valores consecutivos e distintos de X_j. Qualquer threshold $\alpha \in [x_j, x_{j+1}[$ induz a mesma partição no conjunto L_t, sendo equivalentes em termos de decréscimo da medida de impureza (ver Definição 7). Em generalização, contudo, as partições não são equivalentes, pois a partição induzida em X_t é distinta para cada valor de α. Na prática, a opção usual é definir α como o ponto médio entre x_j e x_{j+1}.

Por outro lado, se o atributo ótimo X_j categórico e estamos perante um problema de classificação multi-classe (i.e., com várias classes) é necessário recorrer a estratégias heurísticas para contornar a

exploração exaustiva das $2^{N-1}-1$ partições binárias. Porém, para problemas de classificação binária e medidas de impureza adequadas, é possível demonstrar que é suficiente analisar $N-1$ partições [7].

Teorema 1. *Consideremos um problema de classificação binária com classes c_0 e c_1, Seja X_j uma variável categórica com valores simbólicos $b_1,\ldots,b_L$ ordenados pelo valor estimado de probabilidade de b_l ser de classe c_1*

$$p(c_1|t,X_j=b_1)\leq\ldots\leq p(c_1|t,X_j=b_L),$$

Para medidas de impureza adequadas (ver Definição 6 na próxima secção), então um dos $L-1$ sub-conjuntos de $\mathbb{B}=\{b_1,\ldots,b_h\}$, $h=1,\ldots,L-1$ define uma partição binária de $\mathbb{L}_t$ em

$$\mathbb{L}_{t_E}^{\mathbb{B}}=\{(x,y):(x,y)\in\mathbb{L}_{t,s^*},x_j\in\mathbb{B}\},\qquad \mathbb{L}_{t_D}^{\mathbb{B}}=\{(x,y):(x,y)\in\mathbb{L}_{t,s^*}x_j\in\mathbb{B}\} \tag{2.6}$$

que maximiza o decréscimo da medida de impureza (ver Definição 7 na próxima secção).

Na prática, o teorema anterior significa que uma variável categórica pode ser analisada como uma variável numérica após substituir os seus valores simbólicos b_l pelas estimativas $p(c_1|t,X_j=b_l)$.

Novos algoritmos de construção de árvores de decisão têm sugerido ao longo dos anos. Algumas das opções visam aumentar a flexibilidade das fronteiras de classificação, substituindo as regras de decisão lineares univariadas (2.5), (2.6) por regras de decisão lineares e não lineares multivariadas [26, 36].

Medida de impureza

A maximização do decréscimo da medida de impureza guia o processo de divisão dos nós de uma árvore binária e é uma componente chave do Algoritmo 1. Em termos intuitivos, a medida de impureza deve favorecer divisões que agreguem os objetos da mesma classe. Ou seja, o valor de uma medida de impureza $i(t)$ deve diminuir com a homogeneidade do conjunto $\mathbb{L}_t$ e aumentar com a heterogeneidade. Apresentamos em seguida a definição de medida de impureza introduzida em [7].

Definição 5. *Seja $F_i(p_1,\ldots,p_J)$ uma função estritamente côncava definida sobre $p_j\in\mathbb{R}$, $j=1,2,...,J$, com $p_j\geq 0$ e $\sum_{j=1}^{J}p_j=1$, e tal que*

- *F_i atinge o máximo no ponto $(\frac{1}{J},\ldots,\frac{1}{J})$;*
- *F_i atinge o mínimo nos pontos $(1,0,...0)=\ldots=(0,0,...1)$.*

Definição 6. *Para uma determinada função de impureza F_i, a medida de impureza $i(t)$ de um nó t, é definida por*

$$i(t)=F_i(p(c_1|t),\ldots,p(c_J|t)),$$

com $p(c_j|t)=N_{c_t}/N_t$ a proporção de objetos da classe c_t em $\mathbb{L}_t$.

Das condições impostas em (5) à função F_I resulta que a impureza de um nó é máxima quando as proporções $p(c_j|t)$ são todas iguais, ou seja, quando o conjunto $\mathbb{L}_t$ é heterogéneo com todas as classes igualmente representadas. Por outro lado, a impureza de um nó é mínima quando o conjunto $\mathbb{L}_t$ é homogéneo contendo apenas uma classe.

Durante a separação binária de um nó t, definimos o decréscimo da impureza como a diferença entre a impureza do nó t e a soma ponderada da impureza dos nós t_E e t_D.

Definição 7. *O decréscimo de impureza de uma divisão binária s de um nó t em dois nós filho, t_E e t_D é definida por*

$$\Delta i(s,t) = i(t) - p_E i(t_E) - p_D i(t_D), \tag{2.7}$$

com p_E (p_D) a proporção de objetos direcionadas de $\mathbb{L}_t$ para t_E (t_D), definida por N_{t_E}/N_t (N_{t_D}/N_t) com N_t a dimensão do subconjunto $\mathbb{L}_t$.

O teorema seguinte estabelece que o decréscimo de impureza é não negativo para qualquer divisão s.

Teorema 2. *Seja $i(t) = F_i(p(c_1|t),\ldots,p(c_J|t))$ uma medida de impureza satisfazendo as condições da Definição 6. Então, para qualquer divisão s,*

$$\Delta i(s,t) \geq 0,$$

verificando-se a igualdade se, e só se, $p(c_k|t_E) = p(c_k|t_D) = p(c_k|t)$, $k = 1,\ldots,J$.

Demonstração. Começamos por verificar que

$$\begin{aligned} p(c_k|t) &= \frac{N_{c_k t}}{N_t} \\ &= \frac{N_{c_k t_E} + N_{c_k t_D}}{N_t} \\ &= \frac{N_{t_E}}{N_t}\frac{N_{c_k t_E}}{N_{t_E}} + \frac{N_{t_D}}{N_t}\frac{N_{c_k t_D}}{N_{t_D}} \\ &= p_E p(c_k|t_E) + p_D p(c_k|t_D). \end{aligned}$$

Utilizando a concavidade estrita, vem

$$\begin{aligned} i(t) &= F_i(p(c_1|t),\ldots,p(c_J|t)) \\ &= F_i(p_E p(c_1|t_E) + p_D p(c_1|t_D),\ldots,p_E p(c_J|t_E) + p_D p(c_J|t_D)) \\ &\geq p_E F_i(p(c_1|t_E),\ldots,p(c_J|t_E)) + p_D F_i(p(c_1|t_D),\ldots,p(c_J|t_D)) \\ &= p_E i(t_E) + p_D i(t_D) \end{aligned}$$

com igualdade se, e só se, $p(c_k|t_E) = p(c_k|t_D) = p(c_k|t)$, $k = 1,\ldots,J$.

■

Múltiplas funções de impureza têm sido apresentadas na literatura [33], contudo, a função proposta no algoritmo CART original, o índice de Gini, é ainda uma das mais utilizadas nas diversas implementações do algoritmo. A denominada função de entropia é igualmente popular.

Definição 8. *A função de impureza de Gini*

$$F_{G,i}(p_1,\ldots,p_J) = \sum_{k=1}^{J} p_k(1-p_k), \tag{2.8}$$

satisfaz as condições da Definição 5.

Demonstração. A única condição não trivial é a prova do ponto máximo. Note-se que qualquer função côncava $F(q)$ verifica a desigualdade

$$F(\frac{1}{J}\sum_{j=1}^{J} q_j) \geq \frac{1}{J}\sum_{j=1}^{J} F(q_j),$$

para qualquer $q_j \geq 0$. Seja $q_j = p_j$ e $F(q) = q(1-q)$, utilizando o facto que $\sum_j p_j = 1$, vem

$$\begin{aligned} F(\frac{1}{J}\sum_{j=1}^{J} p_j) &= \frac{1}{J}\left(1-\frac{1}{J}\right) \\ &\geq \frac{1}{J}\sum_{j=1}^{J} p_j(1-p_j) \\ &= \frac{1}{J}F_{G,i}(p_1,\ldots,p_J). \end{aligned}$$

Logo,

$$F_{G,i}(p_1,\ldots,p_J) \leq 1-\frac{1}{J} = F_{G,i}(\frac{1}{J},\ldots,\frac{1}{J})$$

e a concavidade estrita de $F_{G,i}$ garante que $(\frac{1}{J},\ldots,\frac{1}{J})$ é o único ponto máximo.

■

O teorema seguinte estabelece que a aplicação sucessiva do critério de divisão que maximiza o decréscimo de impureza (2.7) garante a obtenção de uma árvore com impureza global mínima.

Teorema 3. *Seja $\tilde{\phi}$ o conjunto das folhas de uma árvore binária e*

$$I(t) = \sum_{t\in\tilde{\phi}} p(t)i(t), \tag{2.9}$$

a impureza global da árvore. Como $\Delta I(s,t) = p(t)\Delta i(s,t)$, a impureza global $I(t)$ é mínima se em todos os nós internos t for escolhida a divisão binária s^* que verifica

$$s^* = arg\ \max_s \Delta i(s,t).$$

É interessante destacar que a escolha da medida de impureza de Gini não é motivada pela diminuição do erro de resubstituição (2.4). Tal função de impureza é definida por

$$F_{R,i}(p_1,\ldots,p_J) = 1 - \max_{k=1,\ldots,J} p_k.$$

Apesar de ser uma escolha intuitiva, esta função possui algumas limitações. Consideremos um nó raiz com 40 objetos da classe 1 e 40 objetos da classe 2. Duas divisões possíveis são: primeira, 20 objetos da classe 1 e 0 objetos da classe 2 no nó filho esquerdo e 20 objetos da classe 1 e 40 objetos da classe 2 no nó filho direito; segunda, 30 objetos da classe 1 e 10 objetos da classe 2 no nó filho esquerdo e 10 objetos da classe 1 e 30 objetos da classe 2 no nó filho direito. A primeira divisão parece vantajosa,

pois o nó filho esquerdo apenas possui objetos da classe 1 e não necessita de mais divisões. Porém, como 20 objetos são incorretamente classificados em ambos os casos, o erro de resubstituição é igual.

A função de impureza de Gini surge associada ao erro cometido quando, para um objeto de um nó folha escolhido de modo aleatório, atribuímos a classe c_k com probabilidade $p(c_k|t)$. A probabilidade do objeto pertencer a uma classe $c_j \neq c_k$ é $p(c_j|t)$ e o erro cometido é o índice de Gini.

2.3 Consistência

A análise teórica das árvores de decisão permanece uma área ativa de investigação com muitos resultados ainda em aberto. Uma questão fundamental que tem sido analisada é a demonstração da consistência, ou seja, da convergência do erro de generalização para o erro ótimo de Bayes [2, 3, 6–8, 23, 28, 35]. No próximo Teorema 4 apresentamos, sem demonstrar, um resultado de consistência para árvores de decisão obtido em [28].

Uma forma alternativa de olhar para um problema de classificação é constatando que $Y = \{c_1, \ldots, c_J\}$ define uma partição no universo dos objetos Ω, isto é

$$\Omega = \Omega_{c_1} \cup \ldots \cup \Omega_{c_J},$$

onde Ω_{c_k} representa o conjunto de objetos da classe c_k. Neste sentido, a classificação induzida por uma árvore de decisão ϕ pode ser vista como uma partição no domínio dos atributos X,

$$X = X_{c_1} \cup \ldots \cup X_{c_J},$$

onde X_{c_k} é o vetor de atributos $x \in X$ tais que $\phi(x) = c_k$. Por exemplo, no problema de classificação na Figura 2.1, a árvore de decisão origina a partição

$$X = X_{c_0}^{\phi} \cup X_{c_1}^{\phi} = X_{t_4} \cup (X_{t_1} \cup X_{t_3}),$$

com $c_0 = 0$ e $c_1 = 1$ as duas classes do problema de classificação binária.

Teorema 4. *Seja $\mathbb{L}_n$ um conjunto de treino de dimensão n. Denotamos por $\mathbb{P}_n = \{X_{n1}, \ldots, X_{nM}\}$ a partição originada pelos nós folha de uma árvore de decisão ϕ_{L_n}. A classe dos nós folha é atribuída por maioria e admitimos que (X^n, Y^n) são independentes e identicamente distribuídas. Denotamos por $diam(A)$ o diâmetro, isto é, a distância máxima na norma Euclidiana de quaisquer dois pontos de um conjunto $A \subset \mathbb{R}^d$,*

$$diam(A) = \sup_{x, x^* \in A} ||x - x^*||,$$

e por $X_n(x)$ o conjunto de um ponto $x \in X \subset \mathbb{R}^d$, isto é, $X_n(x) = X_{ni}$ se $x \in X_{ni}$. Se,

(i) para qualquer $\gamma > 0$ e $\delta \in (0,1)$

$$\lim_{n \longrightarrow \infty} \inf_{S \subset \mathbb{R}^d : \mu(S) \geq 1-\delta} \mu(\{x : diam(X_n(x) \cap S > \gamma\}) = 0$$

quase certamente, com μ a distribuição de X,

(ii) qualquer conjunto da partição $\mathbb{P}_n$ contém pelo menos k_n pontos, com

$$\lim_{n \longrightarrow \infty} \quad \frac{k_n}{\log n} = \infty,$$

então, $\phi_{\mathbb{L}}$ é consistente (fortemente), isto é,

$$E(\phi_{\mathbb{L}_n}) \longrightarrow E(\phi_B),$$

quase certamente.

Capítulo 3

Técnicas ensemble

3.1 Introdução

No contexto de árvores de decisão, as denominadas técnicas ensemble merecem especial destaque devido à sua performance. Estas técnicas tendem a reduzir o overfitting, uma das principais limitações das árvores de decisão, e assentam no pressuposto que vários classificadores de baixa complexidade são mais eficientes que um único classificador complexo.

Duas das técnicas ensemble mais comuns são o bagging e o boosting [18]. O bagging (acrónimo para bootstrap-aggregating) consiste em treinar vários classificadores utilizando bootstrapping e obter a classificação final com votação por maioria [4, 22]. No boosting o classificador base é aplicado sequencialmente, atribuindo em cada iteração um peso superior aos objetos incorretamente classificados na iteração anterior. A classificação final é obtida por combinação linear tendo em conta a performance individual de cada classificador [10, 13, 16]. Um método clássico de bagging assente em árvores de decisão é a denominada floresta aleatória [4, 5]. O extremely randomized trees [16] e o XGBoost [10] são outros exemplos de bagging e boosting com árvores de decisão.

Técnicas ensemble baseadas em árvores de decisão têm um potencial enorme, tendo sido utilizadas, por exemplo, na análise estatística que culminou na deteção do bosão de Higgins [9]. Outro exemplo de aplicação é apresentado em [37]. Para o problema de classificar crédito bancário em função do risco de incumprimento associado, baixo ou alto, métodos ensemble baseados em árvores de decisão apresentaram melhor desempenho que métodos alternativos baseados em neural networks, svm e classificadores lineares.

3.2 Floresta aleatória

A floresta aleatória é uma técnica do tipo ensemble onde a classificação final é obtida por votação por maioria de um determinado número de árvores de decisão treinadas utilizando amostras bootstrap do conjunto de treino $\mathbb{L}$. Outra característica primordial da floresta aleatória é que durante o treino das árvores de decisão, a divisão ótima s^* é obtida considerando um conjunto aleatório de atributos. Ou seja, enquanto que na construção clássica de árvores de decisão o processo de divisão consiste em examinar todos os atributos e escolher aquele que maximiza o decréscimo de impureza (Teorema 3, página 12), na floresta aleatória, a maximização do decréscimo de impureza é efetuada apenas sobre

um conjunto aleatório de atributos. Deste modo, além dos parâmetros associados à construção das árvores de decisão, os parâmetros mais importantes na construção de uma floresta aleatória forest são o número de árvores que a constituem e o número de atributos que é escolhido de modo aleatório em cada divisão.

Apesar da interpretabilidade das árvores de decisão ser perdida na floresta aleatória, é ainda assim possível obter alguma informação relevante, nomeadamente, sobre os atributos do problema de classificação. Um exemplo, é a denominada impureza de decréscimo médio, uma medida obtida considerando a média sobre todas as árvores do decréscimo de impureza nas divisões associadas a cada atributo. Esta medida fornece uma estimativa sobre quais os atributos mais relevantes na construção da floresta aleatória. Uma análise aprofundada sobre medidas de importância dos atributos é apresentada em [27]. Um resultado interessante obtido pelos autores revela que, em determinados pressupostos, adicionar atributos definidos como irrelevantes ao conjunto treino não altera a impureza de decréscimo médio dos restantes atributos. Este resultado sugere que a floresta aleatória é robusta na presença de atributos irrelevantes.

Intuitivamente um dos motivos para o sucesso da floresta aleatória, e em geral das técnicas ensemble, reside no facto de muitos modelos de classificação dependerem de algoritmos de minimização local, por exemplo, as heurísticas do tipo greedy nas árvores de decisão. Neste sentido, diversos classificadores construídos com pontos de partida distintos têm maior probabilidade de evitar a convergência para mínimos locais que um classificador único. Por outro lado, numa perspetiva estatística, assumindo que as decisões de cada um dos modelos simples não estão correlacionados, uma regra de classificação que agregue vários modelos tende a diminuir o risco de tomar uma decisão baseada em falsos pressupostos. O objetivo dos processos aleatórios introduzidos na floresta aleatória, quer na escolha da divisão ótima quer na escolha do conjunto de treino, é precisamente descorrelacionar as suas árvores de decisão.

3.3 Decomposição viés-variância

Uma abordagem formal para analisar o comportamento das técnicas ensemble é a denominada decomposição viés-variância do erro de generalização. A componente do viés está relacionada com o erro do modelo, ou seja, com o seu desvio em relação ao modelo ótimo de Bayes, enquanto que a componente da variância está relacionada com a sensibilidade do modelo a variações no conjunto treino. Este tipo de análise é particularmente indicada para problemas de regressão, onde a utilização da função de perda quadrática permite obter uma decomposição viés-variância simples e intuitiva. Para problemas de classificação, baseados na função perda zero-um, a obtenção deste tipo de decomposição é mais complexa. O teorema seguinte, apresenta uma tentativa de decomposição viés-variância para problemas de classificação binária [14].

Teorema 5. *Consideremos um problema de classificação binária com função perda zero-um. Seja* $P(\phi_{\mathbb{L}}(x) \neq \phi_B(x))$ *a probabilidade para* $X = x$ *de o classificador* $\phi_{\mathbb{L}}$ *efetuar uma classificação diferente daquela obtida com o modelo ótimo de Bayes* ϕ_B*. Temos que*

$$P(\phi_{\mathbb{L}}(x) \neq \phi_B(x)) = P(\hat{p}(Y = \phi_B(x)) < 0.5)$$

e vamos assumir que a estimativa $\hat{p}(Y = \phi_B(x))$ segue uma lei Normal. Então, o erro de generalização de $\phi_{\mathbb{L}}$ em $X = x$, pode ser decomposto em

$$P(Y \neq \phi_{\mathbb{L}}(x)) = P(Y \neq \phi_B(x)) + \Phi\left(\frac{0.5 - \mathbb{E}_{\mathbb{L}}(\hat{p}_{\mathbb{L}}(Y = \phi_{\mathbb{B}}(x)))}{\sqrt{\mathbb{V}_{\mathbb{L}}(\hat{p}_{\mathbb{L}}(Y = \phi_{\mathbb{B}}(x)))}}\right)(2P(\phi_B(x) = Y) - 1), \tag{3.1}$$

com Φ a função de distribuição cumulativa da lei normal padrão.

No teorema anterior, a suposição da normalidade da estimativa $\hat{p}(Y = \phi_B(x))$ é certamente demasiado forte para classificadores do tipo árvores de decisão. Por outro lado, no contexto da floresta aleatória esta estimativa pode ser considerada razoável, pois ela é obtida considerando a média de múltiplas árvores de decisão aleatórias.

Uma análise da decomposição (3.1) permite estabelecer algumas conclusões interessantes. Consideremos o caso em que o valor esperado da estimativa $\hat{p}(Y = \phi_B(x))$ é superior a 0.5. Ou seja, o modelo tem um viés relativamente baixo estimando, em média, a verdadeira classe maioritária em mais de 50% das vezes. Nestas condições, a diminuição da variância da estimativa implica a diminuição do erro de generalização. Além disso, se $\mathbb{V}_{\mathbb{L}}(\hat{p}_{\mathbb{L}}(Y = \phi_{\mathbb{B}}(x))) \to 0$, então $\Phi \to 0$ e o erro de generalização converge para o seu valor mínimo, isto é, o erro irredutível de Bayes $P(Y \neq \phi_B(x))$. Por outro lado, se o valor esperado da estimativa $\hat{p}(Y = \phi_B(x))$ é inferior a 0.5, a diminuição da variância pode aumentar o erro de generalização. Além disso, se $\mathbb{V}_{\mathbb{L}}(\hat{p}_{\mathbb{L}}(Y = \phi_{\mathbb{B}}(x))) \to 0$, então $\Phi \to 1$ e o erro de generalização converge para o seu valor máximo. Ou seja, assumindo que o viés se mantém constante, ou pelo menos não aumenta demasiado, podemos diminuir o erro de generalização de um classificador diminuindo a sua variância.

O método floresta aleatória, procura precisamente diminuir o erro de generalização das árvores de decisão através da diminuição da sua variância. Devido à metodologia utilizada na sua construção (por exemplo, durante o processo de divisão), as árvores de decisão tendem a apresentar elevada variância. De facto, duas árvores de decisão treinadas em dois subconjuntos aleatórios de um conjunto de treino podem ser consideravelmente diferentes. Uma justificação plausível para o facto da floresta aleatória reduzir, em geral, a variância das árvores de decisão está relacionada com o comportamento da variância da média amostral. Seja $X_1, \ldots, X_n$ uma amostra aleatória de uma variável aleatória X com variância σ^2, ou seja, X_i, $i = 1, \ldots, n$, são independentes e seguem a lei de X. Então, a variância da média amostral é σ^2/n. Ou seja, a variância da média amostral é inferior à variância de cada uma das variáveis X_i. Recordamos que a floresta aleatória consiste, idealmente, de um conjunto de árvores de decisão descorrelacionadas.

Para problemas de regressão, é possível obter estimativas sobre a variância de modelos do tipo floresta aleatória que permitem obter uma justificação mais formal [15]. Seja $\psi_{\mathbb{L},\theta_1,\ldots,\theta_M}$ um modelo do tipo ensemble formado por M modelos individuais aleatórios $\phi_{\mathbb{L},\theta_m}$, $m = 1, \ldots, M$. A variância, em $X = x$, do modelo ensemble verifica

$$\mathbb{V}(x) = \rho(x)\sigma_\theta^2(x) + \frac{1 - \rho(x)}{M}\sigma_\theta^2(x),$$

com $\sigma_\theta^2(x)$ a variância do modelo aleatório $\phi_{\mathbb{L},\theta}$ e $\rho(x)$ o coeficiente de correlação entre as previsões de dois modelos aleatórios. Ou seja, quando $M \to \infty$, a variância do ensemble é igual a $\rho(x)\sigma_\theta^2(x)$. Logo, assumindo que o processo de aleatorização descorrelaciona os modelos individuais, ou seja, assumindo

que $\rho(x) < 1$, a variância do ensemble é inferior à variância de um modelo individual. Além disso, se $\rho(x) \to 0$ então $\mathbb{V}(x) \to 0$ quando $M \to \infty$. Por outro lado, se $\rho(x) \to 1$, ou seja, se o processo de aleatorização não descorrelaciona os modelos individuais, temos que $\mathbb{V}(x) = \sigma_\theta^2(x)$, e o ensemble dos modelos individuais não reduz a sua variância. Em conclusão, quanto mais descorrelacionados estiverem os modelos individuais menor a variância do ensemble.

Por outro lado, quando o crescimento de uma árvore de decisão não é controlado, as árvores de decisão apresentam elevada complexidade com tendência a possuir baixo viés. A floresta aleatória tende a preservar o baixo de viés das árvores de decisão porém, o facto de as árvores serem treinadas com recurso a bootstrapping pode originar o seu aumento, particularmente para conjuntos de treino com dimensão reduzida. De igual modo, a restrição do número de atributos a analisar em cada divisão da árvore pode igualmente contribuir para aumentar o seu viés. Ou seja, apesar de em geral a diminuição da variância compensar o possível aumento do viés, existe sempre um trade-off viés-variância.

3.4 Consistência

Naturalmente a escassez de resultados sobre a consistência de árvores de decisão estende-se à floresta aleatória, particularmente para a versão original do método. Dois mecanismos fundamentais, mas que dificultam consideravelmente a análise da floresta aleatória, são o bootstraping e o algoritmo de escolha dos atributos durante o processo de divisão. Assim, grande parte dos resultados de consistência envolvem versões simplificadas destes mecanismos.

Em [6], é provado que uma floresta aleatória é consistente para uma versão bastante simplificada do algoritmo assente nos seguintes pressupostos: ausência de bootstraping, a escolha do atributo a dividir segue uma distribuição uniforme, a divisão dos atributos considerados relevantes ocorre no seu ponto médio, a divisão dos atributos considerandos irrelevantes ocorre de modo aleatório e as árvores de decisão são balanceadas.

Em [3], é demonstrado que técnicas ensemble do tipo bagging com votação por maioria são consistentes desde que os modelos individuais sejam consistentes. Seja n a dimensão do conjunto de treino $\mathbb{L}_n$ e vamos assumir que cada subconjunto de treino é selecionado de modo aleatório de acordo com uma lei binomial de parâmetros n e q_n. Ou seja, cada amostra é selecionada de modo aleatório, sem substituição, de $\mathbb{L}_n$, com probabilidade q_n. O resultado de consistência é demonstrado assumindo que $nq_n \to \infty$, quando $n \to \infty$. É interessante realçar que este resultado é independente do número de modelos individuais no ensemble. Nestas circunstâncias, podemos concluir a consistência de uma floresta aleatória assumindo, por exemplo, que as suas árvores de decisão satisfazem as condições do Teorema 4.

Capítulo 4

Aplicação: dataset México covid-19

Nesta secção aplicamos árvores de decisão para analisar um problema de classificação. Recorremos a um dataset disponibilizado em acesso aberto pelo governo mexicano referente à evolução de pacientes covid-19 [30]. O nosso objetivo é analisar a capacidade de uma árvore de decisão em classificar o desfecho de cada paciente, em particular, se o paciente evoluiu para recuperado ou óbito.

4.1 Descrição e pré-processamento do dataset

O dataset possui além da informação sobre o desfecho de cada paciente, recuperado ou óbito, um conjunto de atributos relacionados com a evolução do seu estado, por exemplo, se o doente desenvolveu pneumonia, mas também sobre a existência prévia de outras doenças como diabetes ou hipertensão. Existe igualmente um atributo com informação sobre o resultado de um teste de covid-19, os valores possíveis são: positivo, inconclusivo, negativo ou resultado desconhecido. Para a nossa análise mantivemos apenas os pacientes com resultado de teste positivo ou negativo, descartando os pacientes com resultado inconclusivo ou desconhecido. Tal como apresentado na Tabela 4.1, o dataset possui mais de 100.000 pacientes, com 50% representando recuperados e 50% óbitos.

	Recuperados	Óbitos	Total
N.º Pacientes	54236	54236	108472

Tabela 4.1 Dimensão do dataset.

Outro dos atributos presentes no dataset diz respeito ao sistema de saúde do paciente. Este atributo está dividido em 13 classes que refletem a organização do sistema de saúde mexicano [24]. Para o nosso estudo decidimos agrupar as classes em apenas duas, sistema de saúde público ou privado. No total, o dataset é constituído por 12 atributos, sendo a sua descrição apresentada na Tabela 4.2.

Nas Figuras 4.1 e 4.2, apresentamos uma descrição gráfica do dataset. O gráfico à esquerda na Figura 4.1, apresenta a percentagem de pacientes possuidores do atributo indicado na legenda do eixo dos xx. Por exemplo, a primeira barra vertical indica que 97% dos pacientes têm um sistema de saúde público, SSaúde (Pub). Naturalmente, este valor significa que apenas 3% dos pacientes possuem um sistema de saúde privado. De modo idêntico, a segunda barra vertical indica que 19%

Atributo	Descrição	Valor
SSaúde	indica qual o sistema de saúde do paciente	Público ou Privado
Ventilador	indica se o paciente foi ventilado	Sim ou Não
Pneumonia	indica se o paciente desenvolveu pneumonia	Sim ou Não
Diabetes	indica se o paciente tem diabetes	Sim ou Não
Asma	indica se o paciente tem asma	Sim ou Não
Hipertensão	indica se o paciente tem hipertensão	Sim ou Não
Cardiovascular	indica se o paciente tem doenças cardiovasculares	Sim ou Não
Obesidade	indica se o paciente tem obesidade	Sim ou Não
DRC	indica se o paciente tem doença renal crónica	Sim ou Não
Tabagismo	indica se o paciente é fumador	Sim ou Não
Grau Covid	indica o nível de gravidade do teste covid-19	1 (mais grave) ou 2

Tabela 4.2 Descrição dos atributos existentes no dataset.

dos pacientes foram ligados a um ventilador mecânico, Ventilador (S), ou de modo equivalente, que 81% dos pacientes não foi ligado a um ventilador.

No gráfico à direita na Figura 4.1, apresentamos a percentagem de recuperados e de óbitos para as faixas etárias indicadas na legenda do eixo dos xx. Por exemplo, para a faixa etária $0-10$, verificamos que 83% dos pacientes são recuperados e 17% são óbitos. Por outro lado, na faixa etária $91-110$, observamos que 88% dos pacientes são óbitos e 12% são recuperados. O valor percentual apresentado no eixo dos xx indica a percentagem de pacientes que pertence a cada uma das faixas etárias, por exemplo, a faixa etária $0-10$ corresponde a 1% do total de pacientes.

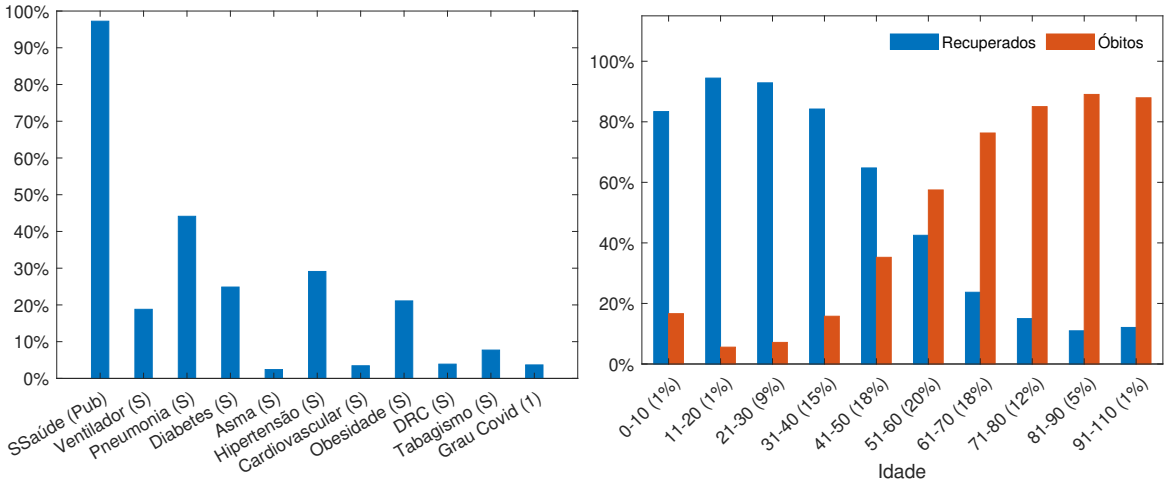

Figura 4.1. O gráfico à esquerda ilustra a distribuição dos pacientes por cada um dos atributos. Por exemplo, 44% dos pacientes desenvolveram pneumonia, Pneumonia (S). O gráfico à direita ilustra a distribuição dos pacientes por faixas etárias, bem como a percentagem de recuperados e óbitos em cada uma das faixas. Por exemplo, 1% dos pacientes pertencem à faixa etária $11-20$ e nesta faixa etária temos 94% de recuperados e 6% de óbitos.

Da análise do gráfico à esquerda na Figura 4.1, destacamos a elevada percentagem de pacientes com sistema de saúde público (97%) e a baixa percentagem de pacientes (4%) portadores de covid-19 de grau 1, o mais grave. Destaque também para o facto de 44% dos pacientes terem desenvolvido

pneumonia e de 19% dos pacientes terem sido ligados a um ventilador mecânico. Do conjunto das doenças pré-existentes, as mais comuns entre os pacientes são hipertensão 29%, diabetes 25%, e obesidade 21%. As restantes, por ordem decrescente de prevalência, são tabagismo 8%, doença renal crónica (DRC) 4%, doença cardiovascular 3% e asma 2%.

Da análise do gráfico à direita na Figura 4.1, destacamos o facto de a percentagem de óbitos superar claramente a percentagem de recuperados para idades superiores a 60 anos. Além disso, é na faixa etária $51-60$ em que a percentagem de óbitos ultrapassa pela primeira vez a percentagem de recuperados. Destaque para o facto de apenas 2% dos pacientes estarem na faixa etária $0-20$, enquanto 62% pertence à faixa etária $21-60$ e 36% à faixa etária $61-110$. A idade média dos pacientes é de 52 anos.

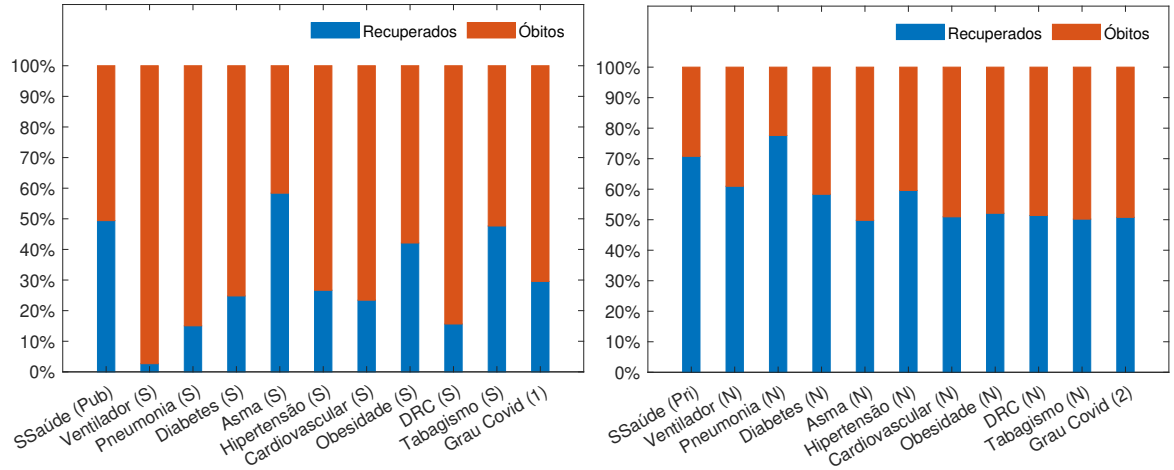

Figura 4.2. Distribuição da percentagem de recuperados e óbitos em função do valor de cada atributo (exceto para o atributo idade, ver Figura 4.1 à direita).

De modo idêntico ao apresentado na Figura 4.1 à direita para o atributo idade, a Figura 4.2 ilustra a percentagem de recuperados e óbitos em função do valor de cada um dos restantes atributos. Por exemplo, a primeira barra na Figura 4.2 à esquerda revela que quando o paciente tem um sistema de saúde público, SSaúde (Pub), a percentagem de recuperados e de óbitos é idêntica (50%). Por outro lado, a primeira barra na Figura 4.2 à direita, revela que quando o paciente tem um sistema de saúde privado, SSaúde (Pri), a percentagem de recuperados é superior à de óbitos, 70% contra 30%.

Da análise da Figura 4.2 à esquerda, destacamos a elevada percentagem de óbitos para pacientes ventilados 97%, que desenvolvem pneumonia 85%, com diabetes 75%, hipertensão 73%, doença cardiovascular 77% ou doença renal crónica (DRC) 84%. Em pacientes portadores de asma, obesidade ou tabagismo a diferença entre a percentagem de óbitos e recuperados é menor, apresentando percentagens de óbitos iguais a 42%, 58% e 53%, respetivamente.

Da análise da Figura 4.2 à direita, destacamos que a percentagem de recuperados é claramente superior à de óbitos em pacientes com sistema de saúde privado, SSaúde (Pri), 71%, não ventilados 61%, sem desenvolvimento de pneumonia 78%, sem diabetes 58% e sem hipertensão 60%. A percentagem de recuperados e óbitos é praticamente igual quando o paciente tem covid-19 em grau 2 ou quando não possui asma, doença cardiovascular, obesidade, DRC ou tabagismo.

4.2 Treino-Validação-Teste

Nesta secção utilizamos os atributos descritos na Tabela 4.2 para treinar e testar uma árvore de decisão. O objetivo é classificar os pacientes em recuperados ou óbitos. Utilizamos um esquema treino-validação-teste de acordo com o apresentado na Tabela 4.3. Os conjuntos treino-validação são utilizados para otimizar o número máximo de nós, um dos hiperparâmetros mais importantes numa árvore de decisão. O conjunto de teste é mantido à margem deste processo de treino, sendo utilizado para medir a performance do método. Ou seja, representa um conjunto de dados completamente novo para a árvore, pretendendo simular o comportamento da árvore em ambiente real.

	Treino-Validação (80%)		Teste (20%)	Total
	Treino (60%)	Validação (20%)		
N.º Pacientes	69423	17355	21694	108472

Tabela 4.3 Percentagem de dados utilizados no esquema de treino-validação-teste.

Para medir o desempenho da árvore recorremos às medidas de acurácia, sensibilidade e especificidade. Denotamos por VP os verdadeiros positivos, ou seja, os óbitos corretamente classificados como óbitos, por VN os verdadeiros negativos, ou seja, os recuperados corretamente classificado como recuperados, por FP os falsos positivos, ou seja, os recuperados classificados como óbitos, e por FN os falsos negativos, ou seja, os óbitos classificados como recuperados. Neste contexto a sensibilidade representa a capacidade do método em classificar os óbitos e a especificidade a capacidade em classificar os recuperados. A acurácia representa a capacidade de o método obter classificações corretas, independentemente de serem óbitos ou recuperados.

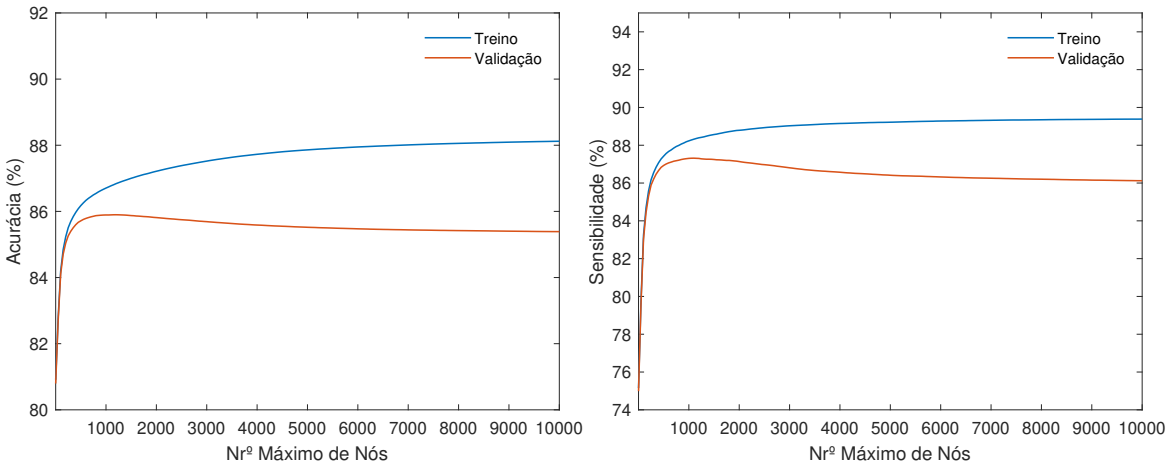

Figura 4.3. Comportamento da acurácia (à esquerda) e da sensibilidade (à direita) da árvore de decisão em função do número máximo de nós.

Durante o processo de otimização calculamos a acurácia, a sensibilidade e a especificidade nos conjuntos treino e validação. O gráfico na Figura 4.3, à esquerda, mostra a evolução da acurácia em função do número de nós. Observamos que a acurácia no conjunto treino (linha a azul) aumenta com o aumento do número de nós, atingindo 88% para o máximo de 10000 nós. A acurácia no conjunto de

validação (linha laranja) também apresenta um crescimento inicial, porém, para aproximadamente 1000 nós, essa tendência é invertida, existindo um decréscimo com o aumento sucessivo do número de nós. Este comportamento, de aumento da acurácia no conjunto de treino e de diminuição no conjunto de validação, apesar de pouco significativo, ilustra o fenómeno de overfitting. Ou seja, o método efetua um ajuste excessivo ao conjunto treino perdendo capacidade de generalização para o conjunto de validação. O comportamento da sensibilidade (gráfico na Figura 4.3, à direita) e da especificidade (não ilustrado) em função do número de nós é semelhante ao da acurácia.

Ponderando o comportamento da acurácia, da sensibilidade e da especificidade com a interpretabilidade, optamos por considerar uma árvore com no máximo 20 nós. Treinamos a árvore no conjunto de dados reservados para o treino (80% do total de dados), e utilizamos o conjunto de teste (20% do total de dados) para avaliar o seu desempenho. A árvore de decisão obtida é exibida na Figura 4.4 e os resultados no conjunto teste são apresentados na Tabela 4.4.

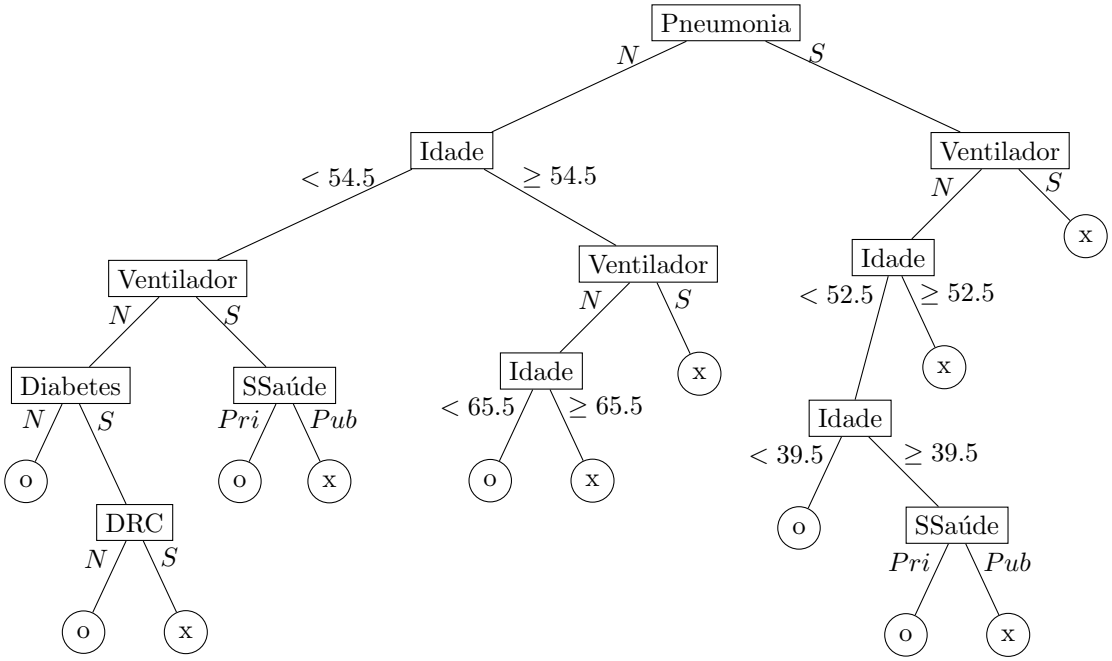

Figura 4.4. Árvore de decisão obtida na etapa de treino. O símbolo 'x' representa uma classificação de óbito e o símbolo 'o' uma classificação de recuperado.

VP	VN	FP	FN	Sens	Esp	Acc
9454	9173	1678	1389	87.19%	84.54%	85.86%

Tabela 4.4 Performance da árvore de decisão no conjunto teste.

O valor da sensibilidade apresentado na Tabela 4.4 significa que 87.19% dos óbitos foram corretamente classificados como óbitos pela árvore de decisão. Por outro lado, o valor da especificidade significa que 84.54% dos recuperados foram corretamente classificados como recuperados. Estes valores revelam que o método tem uma performance equilibrada, com valores semelhantes de sensibil-

idade e especificidade. O valor da acurácia significa que em 85.86% das vezes a árvore efetua uma classificação correta.

Tirando partido da interpretabilidade da árvore de decisão, é possível estabelecer relações potencialmente relevantes para a compreensão da evolução da covid-19. Da análise da Figura 4.4 concluímos que:

- A doença é potencialmente fatal quando a idade do paciente é superior a 65 anos.
- Quando o paciente desenvolve pneumonia e necessita de ventilação mecânica, a doença é potencialmente fatal para qualquer idade.
- Com idade superior a 53 anos ou 40 anos e sistema de saúde público, a doença é potencialmente fatal quando o paciente desenvolve pneumonia.
- Com idade inferior a 55 anos a doença é potencialmente fatal caso o paciente seja diabético e doente renal crónico (DRC) ou caso necessite de ventilação e possua sistema de saúde público.

Em relação ao risco acrescido para pacientes com mais de 65 anos, é interessante referir que de acordo com dados da agência Centros de Controle e Prevenção de Doenças dos Estados Unidos da América, o risco de óbito por covid-19 é pelo menos 10 vezes superior para pacientes com mais de 65 anos quando comparado com pacientes de qualquer outra faixa etária [29]. A nossa conclusão de que a diabetes e a doença renal crónica são um fator de risco acrescido está igualmente de acordo com alguns estudos publicados na literatura. Em [25], os autores concluem que o risco de óbito aumenta cerca de 8 vezes quando comparamos pacientes saudáveis com menos de 60 anos com pacientes possuidores de doença renal crónica ou diabetes.

De modo a avaliar o método floresta aleatória, seguimos a metodologia treino-validação-teste utilizada anteriormente e efetuamos uma otimização Bayesiana sobre os hiperparâmetros: número máximo de nós, número de árvores de decisão e número de atributos a selecionar de modo aleatório em cada nó da árvore. A combinação ótima foi obtida com 147 nós, 10 árvores e 5 atributos. A performance da floresta aleatória no conjunto teste é apresentada na Tabela 4.5. Comparando os

VP	VN	FP	FN	Sens	Esp	Acc
9454	9173	1678	1389	88.87%	84.25%	86.56%

Tabela 4.5 Performance da floresta aleatória no conjunto teste.

resultados da Tabela 4.5 com os resultados da Tabela 4.4 para a árvore de decisão, verificamos que existe uma ligeira melhoria na acurácia, de 85.86% para 86.56%, traduzindo a melhoria na sensibilidade de 87.17% para 88.87%, e o valor idêntico de especificidade, 84.54% contra 84.25%. De acordo com a discussão teórica anterior, uma possível explicação para o facto da floresta aleatória apresentar uma apenas uma ligeira melhoria em relação à árvore de decisão pode estar relacionada com a ausência de um overfitting significativo pela árvore de decisão, tal como ilustrado na Figura 2.3. Estes resultados sugerem que seria necessário considerar atributos adicionais para melhor a performance destes métodos.

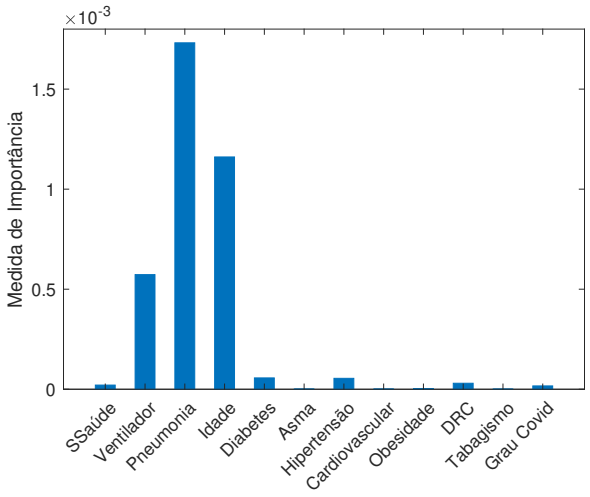

Figura 4.5. Medida de importância dos atributos utilizados na construção da floresta aleatória.

Finalmente, tal como ilustrado na Figura 4.5., a análise da floresta aleatória revela que os atributos mais significativos são pneumonia, idade e ventilador. Com uma importância significativamente inferior, surgem os atributos diabetes, hipertensão e doença real crónica. Esta hierarquia é semelhante à obtida através da análise da árvore de decisão na Figura 4.4.

Capítulo 5

Conclusão

Esta tese de mestrado é dedicada aos métodos de classificação supervisionada denominados por árvores de decisão e floresta aleatória. Iniciamos a tese com uma introdução teórica do problema de classificação, definindo noções essenciais como o erro de generalização e o modelo ótimo de Bayes. O capítulo seguinte é dedicado ao método árvores de decisão, apresentamos o algoritmo CART para a indução de árvores de decisão e descrevemos e fundamentamos do ponto de vista teórico as suas componentes cruciais, como a medida de impureza e as propriedades de consistência. O capítulo seguinte é dedicado ao método floresta aleatória, uma técnica ensemble que consiste no agrupamento de diversas árvores de decisão aleatórias. Analisamos o comportamento deste método recorrendo à decomposição viés-variância do erro de generalização e apresentamos alguns resultados de consistência.

Ilustramos o funcionamento das árvores de decisão com recurso a um dataset público de pacientes covid-19. Além da boa performance, com valores de acurácia, sensibilidade e especificidade superiores a 85% num conjunto teste com mais de 20.000 pacientes, a excelente interpretabilidade do método permitiu identificar alguns fatores de risco para óbito em pacientes covid-19. No mesmo dataset, a floresta aleatória apresenta uma performance ligeiramente superior ao método árvores de decisão.

Dos vários tópicos que foram ignorados, destacamos os problemas de classificação multi-classe e os problemas não balanceados. Estes problemas, comuns em aplicações práticas, são desafiantes para os métodos de classificação. A abordagem do problema de regressão seria igualmente interessante. Apesar de possuir uma fundamentação teórica distinta, a formulação do problema de regressão é idêntica ao problema de classificação no contexto de árvores de decisão e florestas aleatórias.

Bibliografia

[1] D. Bertsimas and J. Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, 2017.

[2] G. Biau. Analysis of a random forests model. *The Journal of Machine Learning Research*, 13(1):1063–1095, 2012.

[3] G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9(9), 2008.

[4] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[6] L. Breiman. Consistency for a simple model of random forests. 2004.

[7] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees. 1984.

[8] M. D. Cattaneo, R. Chandak, and J. M. Klusowski. Convergence rates of oblique regression trees for flexible function libraries, 2022.

[9] S. Chatrchyan et al. Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Physics Letters B*, 716(1):30–61, sep 2012.

[10] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[11] D. A. Cieslak and N. V. Chawla. Learning decision trees for unbalanced data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 241–256. Springer, 2008.

[12] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona. A survey on missing data in machine learning. *Journal of Big Data*, 8(1), oct 2021.

[13] Y. Freund, R. E. Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

[14] J. Friedman. A recursive partitioning decision rule for nonparametric classification. *IEEE Transactions on Computers*, C-26(4):404–408, 1977.

[15] P. Geurts. Contributions to decision tree induction: bias/variance tradeoff and time series classification. 2002.

[16] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[17] M. M. Ghiasi, S. Zendehboudi, and A. A. Mohsenipour. Decision tree-based diagnosis of coronary artery disease: CART model. *Computer Methods and Programs in Biomedicine*, 192:105400, aug 2020.

[18] S. González, S. García, J. D. Ser, L. Rokach, and F. Herrera. A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Information Fusion*, 64:205–237, dec 2020.

[19] L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5(1):15–17, may 1976.

[20] Independent High-Level Expert Group on Artificial Intelligence. Ethics guidelines for trustworthy AI. Technical report, 2019.

[21] P. Khosravi, A. Vergari, Y. Choi, Y. Liang, and G. V. d. Broeck. Handling missing data in decision trees: A probabilistic approach. *arXiv preprint arXiv:2006.16341*, 2020.

[22] J. Kittler, M. Hatef, R. P. Duin, and J. Matas. On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence*, 20(3):226–239, 1998.

[23] J. M. Klusowski. Universal consistency of decision trees in high dimensions. *arXiv preprint arXiv:2104.13881*, 2021.

[24] L. C. Krasniak, S. de Camargo Catapan, G. de Almeida Raschke Medeiros, and M. C. M. Calvo. Análise do seguro popular de saúde mexicano: uma revisão integrativa da literatura. *Saúde em Debate*, 43(spe5):273–285, 2019.

[25] R. M. Lana et al. Identificação de grupos prioritários para a vacinação contra COVID-19 no brasil. *Cadernos de Saúde Pública*, 37(10), 2021.

[26] Y. Li, M. Dong, and R. Kothari. Classifiability-based omnivariate decision trees. *IEEE Transactions on Neural Networks*, 16(6):1547–1560, nov 2005.

[27] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts. Understanding variable importances in forests of randomized trees. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, pages 431–439. Curran Associates, Inc., 2013.

[28] G. Lugosi and A. Nobel. Consistency of data-driven histogram methods for density estimation and classification. *The Annals of Statistics*, 24(2):687–706, 1996.

[29] Centers for Disease Control and Prevention - CDC. Risk for covid-19 infection, hospitalization, and death by age group, 2022. URL https://www.cdc.gov/coronavirus/2019-ncov/covid-data/investigations-discovery/hospitalization-death-by-age.html. Acedido em: 15-12-2022.

[30] Secretaría de Salud México. Información referente a casos covid-19 en México, 2020. URL https://datos.gob.mx/busca/dataset/informacion-referente-a-casos-covid-19-en-mexico. Acedido em: 01-12-2022.

[31] M. Norouzi, M. Collins, M. A. Johnson, D. J. Fleet, and P. Kohli. Efficient non-greedy optimization of decision trees. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[32] J. R. Quinlan. Induction of decision trees. machine learning 1. *№*, 1:1–81, 1986.

[33] L. Rokach and O. Maimon. Decision trees. In *Data mining and knowledge discovery handbook*, pages 165–192. Springer, 2005.

[34] S. L. Salzberg. C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993, 1994.

[35] E. Scornet, G. Biau, and J.-P. Vert. Consistency of random forests. *The Annals of Statistics*, 43(4):1716–1741, 2015.

[36] F. Wang, Q. Wang, F. Nie, Z. Li, W. Yu, and F. Ren. A linear multivariate binary decision tree classifier based on k-means splitting. *Pattern Recognition*, 107:107521, nov 2020.

[37] G. Wang, J. Hao, J. Ma, and H. Jiang. A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications*, 38(1):223–230, jan 2011.