

# APLICAÇÕES DE INTELIGÊNCIA ARTIFICIAL E CIÊNCIA DE DADOS

Julio C. Uzinski  
Caio C. E. de Abreu  
Bruno R. de **Oliveira**  

---

organizadores



Pantanal Editora

2020

Julio Cezar Uzinski  
Caio Cesar Enside de Abreu  
Bruno Rodrigues de Oliveira  
(Organizadores)

**Aplicações de  
Inteligência Artificial e  
Ciência de Dados**



Pantanal Editora

2020

Copyright© Pantanal Editora  
Copyright do Texto© 2020 Os Autores  
Copyright da Edição© 2020 Pantanal Editora  
Editor Chefe: Prof. Dr. Alan Mario Zuffo  
Editores Executivos: Prof. Dr. Jorge González Aguilera  
Prof. Dr. Bruno Rodrigues de Oliveira

Diagramação: A editora  
Edição de Arte: A editora  
Revisão: Os autor(es), organizador(es) e a editora

#### Conselho Editorial

- Prof. Dr. Adaylson Wagner Sousa de Vasconcelos – OAB/PB
- Profa. Msc. Adriana Flávia Neu – Mun. Faxinal Soturno e Tupanciretã
- Profa. Dra. Albys Ferrer Dubois – UO (Cuba)
- Prof. Dr. Antonio Gasparetto Júnior – IF SUDESTE MG
- Profa. Msc. Aris Verdecia Peña – Facultad de Medicina (Cuba)
- Profa. Arisleidis Chapman Verdecia – ISCM (Cuba)
- Prof. Dr. Bruno Gomes de Araújo - UEA
- Prof. Dr. Caio Cesar Enside de Abreu – UNEMAT
- Prof. Dr. Carlos Nick – UFV
- Prof. Dr. Claudio Silveira Maia – AJES
- Prof. Dr. Cleberton Correia Santos – UFGD
- Prof. Dr. Cristiano Pereira da Silva – UEMS
- Profa. Ma. Dayse Rodrigues dos Santos – IFPA
- Prof. Msc. David Chacon Alvarez – UNICENTRO
- Prof. Dr. Denis Silva Nogueira – IFMT
- Profa. Dra. Denise Silva Nogueira – UFMG
- Profa. Dra. Dennyura Oliveira Galvão – URCA
- Prof. Dr. Elias Rocha Gonçalves – ISEPAM-FAETEC
- Prof. Me. Ernane Rosa Martins – IFG
- Prof. Dr. Fábio Steiner – UEMS
- Prof. Dr. Gabriel Andres Tafur Gomez (Colômbia)
- Prof. Dr. Hebert Hernán Soto Gonzáles – UNAM (Peru)
- Prof. Dr. Hudson do Vale de Oliveira – IFRR
- Prof. Msc. Javier Revilla Armesto – UCG (México)
- Prof. Msc. João Camilo Sevilla – Mun. Rio de Janeiro
- Prof. Dr. José Luis Soto Gonzales – UNMSM (Peru)
- Prof. Dr. Julio Cezar Uzinski – UFMT
- Prof. Msc. Lucas R. Oliveira – Mun. de Chap. do Sul
- Prof. Dr. Leandro Argente-Martínez – ITSON (México)
- Profa. Msc. Lidiane Jaqueline de Souza Costa Marchesan – Consultório em Santa Maria
- Prof. Msc. Marcos Pisarski Júnior – UEG
- Prof. Dr. Mario Rodrigo Esparza Mantilla – UNAM (Peru)
- Profa. Msc. Mary Jose Almeida Pereira – SEDUC/PA
- Profa. Msc. Nila Luciana Vilhena Madureira – IFPA
- Profa. Dra. Patrícia Maurer
- Profa. Msc. Queila Pahim da Silva – IFB
- Prof. Dr. Rafael Chapman Auty – UO (Cuba)
- Prof. Dr. Rafael Felipe Ratke – UFMS
- Prof. Dr. Raphael Reis da Silva – UFPI

- Prof. Dr. Ricardo Alves de Araújo – UEMA
- Prof. Dr. Wéverson Lima Fonseca – UFPI
- Prof. Msc. Wesclen Vilar Nogueira – FURG
- Profa. Dra. Yilan Fung Boix – UO (Cuba)
- Prof. Dr. Willian Douglas Guilherme – UFT

#### Conselho Técnico Científico

- Esp. Joacir Mário Zuffo Júnior
- Esp. Maurício Amormino Júnior
- Esp. Tayronne de Almeida Rodrigues
- Esp. Camila Alves Pereira
- Lda. Rosalina Eufrausino Lustosa Zuffo

#### Ficha Catalográfica

<b>Dados Internacionais de Catalogação na Publicação (CIP)</b> (eDOC BRASIL, Belo Horizonte/MG)	
A642	<p>Aplicações de inteligência artificial e ciência de dados [recurso eletrônico] / Organizadores Julio Cezar Uzinski, Caio Cesar Enside de Abreu, Bruno Rodrigues de Oliveira. – Nova Xavantina, MT: Pantanal, 2020. 100p.</p> <p>Formato: PDF            Requisitos de sistema: Adobe Acrobat Reader            Modo de acesso: World Wide Web            ISBN 978-65-88319-26-0            DOI <a href="https://doi.org/10.46420/9786588319260">https://doi.org/10.46420/9786588319260</a></p> <p>1. Inteligência artificial. 2. Automação. 3. Processamento de dados. I. Uzinski, Julio Cezar. II. Abreu, Caio Cesar Enside de. III. Oliveira, Bruno Rodrigues de.</p> <p style="text-align: right;">CDD 001.535</p>
<b>Elaborado por Maurício Amormino Júnior – CRB6/2422</b>	

O conteúdo dos livros e capítulos, seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva do(s) autor (es). O download da obra é permitido e o compartilhamento desde que sejam citadas as referências dos autores, mas sem a possibilidade de alterá-la de nenhuma forma ou utilizá-la para fins comerciais.

#### **Pantanal Editora**

Rua Abaete, 83, Sala B, Centro. CEP: 78690-000.  
 Nova Xavantina – Mato Grosso – Brasil.  
 Telefone (66) 99682-4165 (Whatsapp).  
<https://www.editorapantanal.com.br>  
[contato@editorapantanal.com.br](mailto:contato@editorapantanal.com.br)

## APRESENTAÇÃO

Ciência de dados (*Data science*) e Inteligência artificial (*Artificial intelligence*) podem ser consideradas como algumas das áreas da ciência mais importantes de nosso tempo. Ademais, não estão apenas no mundo acadêmico ou no futuro como no filme *The terminator*, estão ao nosso redor nesse exato momento em situações simples, por exemplo, são empregadas para sugerir qual o próximo emoji que você enviará em uma conversa no seu aplicativo de bate papo favorito. Ciência de dados se refere a um campo de estudo muito amplo que se vale de métodos científicos para se obter informações e percepções de conjuntos de dados, enquanto que a inteligência artificial refere-se ao que chamamos de inteligência das máquinas. As aplicações dessas ferramentas estão relacionadas às tecnologias que nos envolvem em nosso cotidiano, e.g., as redes sociais ou comércio eletrônico, bem como aplicações que escapam ao senso comum, como em cibersegurança ou astrofísica.

Tais abordagens, às vezes, são utilizadas juntas, de forma intercalada ou separadas. Uma pesquisa poderá contar com ambas as ferramentas ou apenas com uma delas. Nesse livro, o leitor encontrará aplicações diversas dessas ferramentas em diferentes perspectivas. Em seis capítulos são apresentados resultados de pesquisas de dezesseis pesquisadores sobre temas diversos que dão uma demonstração do poder da Inteligência Artificial e da Ciência de Dados.

O primeiro capítulo traz uma comparação dos resultados da Transformada Wavelet (WT) e do Filtro Savitzky-Golay (SG) na estimação da tendência em séries temporais. Os resultados são obtidos a partir de uma série temporal artificial e uma série de dados reais, i.e, a cotação da moeda americana (Dólar US\$) frente à moeda brasileira (Real R\$).

Considerando que a principal abordagem em Aprendizado de Máquinas é a construção de algoritmos que, por meio das experiências e aprendizado, possam ser melhorados automaticamente, o segundo capítulo desse livro traz uma avaliação de um algoritmo usando Redes Neurais Artificiais LSTM. Essa abordagem é justificada pela necessidade de reconhecimento de padrões de comportamento de presença de elementos em um ambiente para realizar a predição de ações no local por meio dos dados coletados fornecendo uma base para o desenvolvimento de um sistema doméstico inteligente.

Já no terceiro capítulo, uma metodologia para descrição de Casos de Uso (Uc) no desenvolvimento de sistemas de informação é apresentada. A motivação para a criação desta metodologia é que muitos UC's são descritos com pouca riqueza de detalhes, o que prejudica a qualidade dos sistemas de informação.

O quarto capítulo traz uma aplicação bastante diversa de todas as outras: um estudo dos aspectos e causas da obesidade por meio do algoritmo de agrupamento Farthest First para agrupar

peças com informações demográficas, socioeconômicas, de biotipo e de conjunto de ações observáveis recorrentes.

No quinto capítulo é desenvolvida uma aplicação utilizando regressão logística como um mapeamento do espaço de atributos não-linearmente separável baseado em um tipo especial de funções denominadas de Golden Wavelets. Nesse texto, os autores buscam tratar a matemática envolvida no método de aprendizado de máquina denominado de Regressão Logística e propor a utilização de funções wavelets para mapeamento do espaço de atributos.

Para encerrar esse e-book com chave de ouro, o sexto capítulo, apresenta uma classificação de cenas acústicas utilizando a Transformada Wavelet e Aprendizagem de Máquina. Os autores dedicam a redação à apresentação dos principais métodos de Classificação de Cenas Acústicas (CCA) e os principais atributos utilizados para esta classificação, exemplos de aplicações, e uma investigação adicional centrada na utilização da Transformada Wavelet Contínua (TWC).

Caríssimo(a) leitor(a), convidamos você à leitura de Aplicações de Inteligência Artificial e Ciência de Dados, e esperamos que a mesma seja um deleite ilimitado e que contribua de alguma forma com suas aspirações.

**Julio Cezar Uzinski**


## SUMÁRIO

<b>Apresentação</b> .....	4
<b>Capítulo I</b> .....	7
Obtenção de Tendências em Série Temporal empregando a Transformada Wavelet e o filtro Savitzky-Golay.....	7
<b>Capítulo II</b> .....	24
Aplicação de Redes Neurais Artificiais LSTM em domótica inteligente .....	24
<b>Capítulo III</b> .....	32
Uma Metodologia Simples para Descrição de Casos de Uso no Desenvolvimento de Sistemas de Informação.....	32
<b>Capítulo IV</b> .....	49
Aspectos e causas da obesidade: uma análise utilizando algoritmo de agrupamento <i>Fartbest First</i> .....	49
<b>Capítulo V</b> .....	59
Mapeamento de espaço de atributos não-linearmente separável baseado na Golden wavelet: aplicação na regressão logística .....	59
<b>Capítulo VI</b> .....	85
Classificação de Cenas Acústicas Utilizando a Transformada Wavelet Contínua e Aprendizagem de Máquina .....	85
<b>Índice Remissivo</b> .....	99

# Mapeamento de espaço de atributos não-linearmente separável baseado na Golden wavelet: aplicação na regressão logística

Recebido em: 10/09/2020

Aceito em: 29/09/2020

 10.46420/9786588319260cap5

Bruno Rodrigues de Oliveira<sup>1\*</sup> 

Fabício Ely Gossler<sup>2</sup> 

Marco Aparecido Queiroz Duarte<sup>3</sup> 

## INTRODUÇÃO

Os termos inteligência artificial e aprendizado de máquina têm aparecido com frequência nos noticiários recentes, devido às incríveis aplicações que a cada dia surgem empregando as técnicas computacionais desenvolvidas nestas disciplinas. As aplicações vão desde diagnóstico de doenças, tal como a COVID-19 por meio de análises de raio-X (Basu et. al, 2020) até carros autônomos que utilizam visão computacional por meio do processamento e reconhecimento de imagens (Schwartz et. al, 2018).

Embora os primeiros estudos sobre estes temas remontem a década de 1950, nos últimos anos ocorreu uma explosão nas aplicações dessas técnicas, em parte devido à enorme massa de dados disponível atualmente e também à evolução dos hardwares utilizados para processamento.

Uma das metodologias mais empregadas atualmente são os algoritmos de aprendizado profundo (*Deep Learning*), que são essencialmente Redes Neurais Artificiais com uma estrutura de aprendizado não-supervisionado na etapa de extração de atributos e aprendizado supervisionado na etapa de classificação, possuindo elas, geralmente, milhares de parâmetros a estimar e várias camadas na rede, daí a denominação “profundo”. Embora os hardwares disponíveis no mercado deem conta do processamento necessário para implementar essas redes, o custo financeiro ainda é elevado. Por isso, muitos métodos clássicos de aprendizado de máquina, e que demandam menor custo computacional, ainda são empregados, principalmente nas situações onde um hardware mais poderoso não está acessível. Além disso, muitos destes métodos clássicos possuem poder explicativo inerente aos modelos aprendidos, enquanto que as redes neurais são vistas como “caixa preta”.

---

<sup>1</sup> Editora Pantanal.

<sup>2</sup> Rua Clevelândia 1924, Sibipiruna. Chapadão do Sul-MS. E-mail: fabricioely08@gmail.com.

<sup>3</sup> Universidade Estadual de Mato Grosso do Sul (UEMS), Unidade Universitária de Cassilândia. Rodovia MS 306, Cassilândia-MS. E-mail: marco@uems.br.

\*Autor de correspondência: bruno@editorapantanal.com.br



Um destes métodos clássicos é a Regressão Logística, que pode ser considerado um dos métodos mais simples de aprendizado de máquina, principalmente por haver poucos hiperparâmetros para serem ajustados. No entanto, uma das desvantagens desse método é que ele não resolve problemas muito complexos, quando é exigida uma fronteira de decisão não-linear. Para driblar essa desvantagem, uma das soluções é encontrar funções que mapeiam o espaço de atributos não-linearmente separável para um espaço linearmente separável.

Sendo assim, o texto ora apresentado tem duplo propósito: primeiro abordar os mecanismos matemáticos envolvidos no método de aprendizado de máquina denominado de Regressão Logística, e segundo, a proposição da utilização de funções wavelets para mapeamento do espaço de atributos, sendo mais especificamente utilizadas as funções Golden wavelets, que foram propostas recentemente e geraram resultados superiores às funções clássicas, relativamente às aplicações de classificação de padrões de arritmias cardíacas (Gossler, 2016).

Os resultados obtidos mostram que as Golden wavelets são adequadas para o mapeamento do espaço de atributos, melhorando significativamente a eficiência na classificação.

## APRENDIZADO DE MÁQUINA

Conceitualmente, o aprendizado de máquina é uma das subáreas da inteligência artificial e é constituído de métodos matemático/computacionais que geram modelos de um certo ambiente. A função de modelagem é executada empregando os conjuntos de dados apresentados ao algoritmo de aprendizado de máquina. Esses dados, por sua vez, são informações representativas do ambiente.

Os métodos de aprendizado de máquina podem ser enquadrados em duas grandes categorias: aprendizado supervisionado e aprendizado não-supervisionado. Para a primeira categoria, duas subcategorias principais são: classificação e regressão (Kubat, 2015; Fisher; Lenz, 1996).

Para a classificação ou reconhecimento de padrões, duas etapas essenciais são implementadas, a saber: treinamento/indução e teste/validação. Na primeira etapa, emprega-se um conjunto de dados  $T = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$ , com  $N$  instâncias extraídas do ambiente, sendo  $\mathbf{x}_n$  o  $n$ -ésimo vetor de atributos e  $y_n$  um rótulo que designa a qual classe pertence esse vetor.  $\mathbf{x}_n$  também recebe a denominação de variável preditora e  $y_n$  de variável predita.

Cada algoritmo de aprendizado de máquina utiliza os dados do conjunto  $T$  de uma maneira peculiar, entretanto, ao final do processo de indução sobre esse conjunto, o algoritmo fornece um modelo  $\hat{h}(\mathbf{x}, \beta) = \hat{y}$ , o qual é uma estimativa do modelo real  $h$  do ambiente (Haykin, 2007). O parâmetro  $\beta$  é relativo a cada algoritmo empregado, e seu tipo pode ser um simples escalar ou um vetor.

Em linhas gerais, o modelo  $\hat{h}$  recebe um vetor de atributos  $\mathbf{x}$  desconhecido e retorna um rótulo estimado  $\hat{y}$ . Assim sendo, na etapa de teste ou validação um conjunto  $T' = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N'}$  é utilizado para verificar o quão próximo o modelo estimado  $\hat{h}$  está do modelo real  $h$ . Para tanto, pode-se empregar a função (1) para calcular os erros de predição no caso de classificação binária

$$E = \sum_{n=1}^{N'} \begin{cases} 1, & \text{se } y_n \neq \hat{y}_n \\ 0, & \text{se } y_n = \hat{y}_n \end{cases} \quad (1)$$

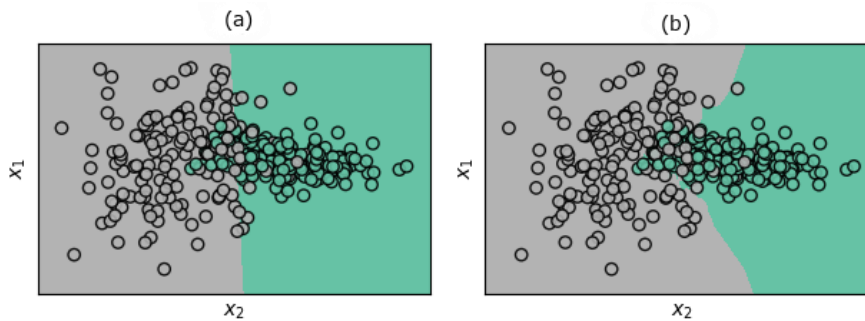
Deste modo, se  $E = N'$  então  $\hat{h}$  erra todas as predições, mas, por outro lado, se  $E = 0$ , então  $\hat{h}$  é uma estimativa perfeita do modelo do ambiente. Se  $E = N'/2$ , então a máquina induzida erra e acerta a mesma quantidade de vezes.

É importante notar nesse ponto que os conjuntos  $T$  e  $T'$  não foram restringidos a serem diferentes, pois, matematicamente, eles podem ser iguais. No entanto, na prática, para dados reais, aconselha-se que o teste/validação seja implementado tanto sobre o conjunto  $T$  quanto sobre o conjunto  $T' \neq T$ . Tal abordagem é relevante, pois o método de aprendizado pode se superajustar aos padrões. Sendo assim, validar o modelo inferido com dados desconhecidos o força a prever padrões não apresentados na fase de indução.

Outra questão de relevância na verificação da adequação do modelo estimado  $\hat{h}$  está relacionada ao conjunto de teste/validação. Quando é verificado que o modelo  $\hat{h}$  não é uma boa estimativa do modelo real  $h$  devido à quantidade de erros de predição, é comum voltar a fase de ajustes dos hiperparâmetros do modelo. Esses não são os parâmetros  $\beta$ , os quais são estimados a partir dos dados, mas são parâmetros ajustáveis ao nível do projetista, ou seja, são parâmetros fixos que variam de algoritmo para algoritmo. Após ajustar os hiperparâmetros, testa-se novamente o novo modelo encontrado. Quando esta ação é repetida, pode ocorrer que o modelo também fique superajustado aos hiperparâmetros. Neste caso, aconselha-se a utilização de um outro conjunto de validação, distinto do conjunto de teste empregado nas fases de ajustes do modelo.

O modelo  $\hat{h}$ , que é uma estimativa do modelo real  $h$  do ambiente, também pode ser encarado como uma fronteira de decisão à qual estão associadas superfícies de decisão. Quando os vetores de atributos  $\mathbf{x}_n$  do conjunto  $T$  estão em  $\mathbb{R}^2$  essa fronteira é uma curva; no  $\mathbb{R}^3$  ela é um plano; e no  $\mathbb{R}^Q$  com  $Q > 3$ , a fronteira de decisão é um hiperplano. Muitos algoritmos de aprendizado de máquina geram fronteiras de decisão apenas lineares, como o Perceptron e a Regressão Logística, que será abordada na próxima seção, enquanto outros têm capacidade de conceber fronteiras também não-lineares, como as Redes Neurais Artificiais e as Máquinas de Vetores Suporte (Haykin, 2007).

Na Figura 1 constam exemplos de superfícies de decisão com fronteiras: linear (Figura 1 (a)) e não-linear (Figura 1 (b)), onde os círculos representam as instâncias  $\mathbf{x}_n \in \mathbb{R}^2$  e as cores designam a classe à qual cada instância pertence.



**Figura 1.** Exemplos de superfícies de decisão: (a) com fronteira linear e (b) com fronteira não-linear. Fonte: os autores.

Fica claro analisando a Figura 1 que a superfície de decisão não-linear resultará em menos erros de predição, pois a superfície verde inclui mais exemplos dessa classe. Muito embora, essa fronteira não-linear possa estar superajustada aos dados de treinamento/indução, acarretando em um resultado inferior sobre o conjunto de teste.

## REGRESSÃO LOGÍSTICA

As técnicas de regressão permitem inferir a relação entre uma variável aleatória dependente (predita) e uma ou mais variáveis aleatórias independentes (preditoras) sem pressuposições a respeito dos processos geradores dos dados analisados (Loesch e Hoeltgebaum, 2012), ou seja, os modelos (equações) obtidos por meio destas técnicas consideram apenas as amostras fornecidas e alguns deles, como a regressão linear, alguma restrição sobre a distribuição de probabilidade dos dados.

A regressão logística é uma das técnicas de regressão específicas para aqueles fenômenos onde a variável predita tem uma distribuição de Bernoulli, ou seja, ela é uma variável binária. Os modelos  $\hat{h}$  inferidos são utilizados para explicar características dos fenômenos estudados, para classificação de dados e também para previsões. Esta técnica é extensivamente aplicada na área médica para classificação de doenças, inclusive este foi um dos primeiros problemas para o qual esta técnica de regressão teve sucesso, onde foram realizados experimentos para verificação de quais fatores mais influenciavam o surgimento de doenças cardiovasculares (Fávero et al., 2009). A capacidade explicativa da Regressão Logística a torna uma ferramenta importante, por exemplo, na determinação dos fatores que caracterizam um grupo de indivíduos em doentes ou saudáveis (Loesch; Hoeltgebaum, 2012). Nesse contexto, os atributos individuais  $\mathbf{x}_{kn}$  podem significar valores aferidos a partir de exames médicos.

Frisa-se que, embora o nome seja “Regressão” Logística, esse método de fato é utilizado para classificação e não regressão. Ficará mais claro adiante porque se utiliza a expressão “regressão”.

A seguir, baseado em Loesch e Hoeltgebaum (2012), Bishop (2006) e Fávero et al. (2009), com as devidas adaptações, a técnica de regressão logística é pormenorizadamente explicada.

Seja  $\mathbf{x}_n \in \mathbb{R}^K$  uma realização do vetor aleatório  $\mathbf{X}$  com distribuição de probabilidade qualquer e  $y_n \in \{0, 1\}$  uma realização da variável aleatória  $Y$  com distribuição de Bernoulli, e  $T = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$  um conjunto de  $N$  amostras aleatórias. Neste contexto, considera-se um problema de reconhecimento de padrões binário, ou seja, há apenas duas classes disponíveis, pois  $y_n \in \{0, 1\}$ .

Sem perda de generalidade, supõe-se que a probabilidade de ocorrência do valor categórico (rótulo) 1, dado o vetor  $\mathbf{x}_n$ , é igual a  $p_n$ , e de ocorrência do valor 0, conseqüentemente é  $q_n = 1 - p_n$ , isto é:

$$P(y_n = 1 | \mathbf{x}_n) = p_n, \tag{2}$$

$$P(y_n = 0 | \mathbf{x}_n) = 1 - p_n. \tag{3}$$

Visto que a variável  $y_n$  é de Bernoulli e a função densidade de probabilidade de uma distribuição de Bernoulli para uma variável aleatória  $y$  é dada pela expressão  $f_y(y) = p^y(1-p)^{1-y}$  dada uma probabilidade  $p$ , as equações (2) e (3) podem ser reescritas como uma única função de probabilidade:

$$P(y_n | \mathbf{x}_n) = p_n^{y_n} (1 - p_n)^{1-y_n}. \tag{4}$$

A razão das probabilidades, dada por

$$R(p_n) = \frac{p_n}{1-p_n} = \frac{P(y_n=1|\mathbf{x}_n)}{P(y_n=0|\mathbf{x}_n)}, \tag{5}$$

é denominada chance ou possibilidade, e se traduz como a quantidade de vezes que o valor  $y_n = 1$  é mais provável de ocorrer que o valor  $y_n = 0$ , dado o vetor aleatório  $\mathbf{x}_n$ .

Com o intuito de obter uma relação entre as variáveis predictoras e preditas, considera-se que a probabilidade da variável predita  $y_n$  ser igual a 1 é dada por  $p_n$ . Disto segue que:

$$\begin{aligned} g(y_n) &= p_n \\ &= \frac{1}{1 + \frac{1-p_n}{p_n}} \\ &= \frac{1}{1 + \frac{e^{\ln(1-p_n)}}{e^{\ln p_n}}} \\ &= \frac{1}{1 + e^{-y_n}}, \end{aligned} \tag{6}$$

onde  $e$  é o número de Euler e  $\ln x$  é o logaritmo de  $x$  na base  $e$ , também chamado de logaritmo natural.

Partindo do resultado da equação (6) verifica-se que

$$\frac{p_n}{1-p_n} = \frac{\frac{1}{1+e^{-y_n}}}{1-\frac{1}{1+e^{-y_n}}} = \frac{\frac{1}{1+e^{-y_n}}}{\frac{1+e^{-y_n}-1}{1+e^{-y_n}}} = \frac{1}{e^{-y_n}} = e^{y_n}.$$

Computando o logaritmo natural de ambos lados, a variável predita é dada por:

$$\begin{aligned} y_n &= \ln\left(\frac{p_n}{1-p_n}\right), \\ &= \ln R(p_n). \end{aligned} \tag{7}$$

Levando em conta que  $R(p_n)$ , conforme equação (5), representa o quanto  $y_n = 1$  é mais provável de ocorrer que  $y_n = 0$ , então  $R(q_n)$  representa a chance inversa. A equação (7) pode então ser escrita como

$$y_n = \ln\left(\frac{p_n}{q_n}\right), \tag{8}$$

que representa o logaritmo natural da razão entre a probabilidade de uma amostra aleatória  $(\mathbf{x}_n, y_n)$  pertencer ao grupo  $y_n = 1$  pela probabilidade de pertencer ao grupo  $y_n = 0$ . Como a função logística multivariada  $g(y_n)$ , equação (6), varia assintoticamente entre 0 e 1, quando  $y_n$  tende a  $+\infty$  e  $-\infty$ , respectivamente, então é uma boa candidata para modelar as probabilidades consideradas.

Uma das premissas da Regressão Logística é que a variável predita  $y_n$  seja modelada como uma combinação linear das variáveis predictoras  $\mathbf{x}_n$  (Fávero et al., 2009), resultando em um hiperplano:

$$y_n = \beta_0 + \sum_{k=1}^K \beta_k x_{kn}, \tag{9}$$

onde  $\boldsymbol{\beta} = [\beta_0, \beta_1, \dots, \beta_K]$  é um vetor de parâmetros desconhecidos, que devem ser aprendidos (estimados) pelo algoritmo. Eles ponderam a importância relativa das variáveis aleatórias na explicação do valor predito  $y_n$ , exceto  $\beta_0$  que determina onde o hiperplano cruza os eixos;  $x_{kn}$  representa a  $k$ -ésima componente do vetor  $\mathbf{x}_n$ .

Essa equação é um modelo de regressão linear que estima o valor  $y_n$  a partir da combinação linear dos atributos  $x_{kn}$ . Entretanto, e este é um ponto importante, na equação (9), a variável  $y_n$  poderá assumir valores diferentes de 0 e 1, pois as componentes do vetor aleatório  $\mathbf{x}_n$  são reais. Por isso, o valor predito deve ser modulado pela função logística  $g(y_n)$  expressa na equação (6), garantindo assim que se tenha um valor de probabilidade, isto é, um valor que esteja entre 0 e 1,

já que esses valores são o mínimo e o máximo, respectivamente, retornados pela função  $g(y_n)$ . Desse modo, essa função fornece a probabilidade da variável  $y_n$  ser igual a 1, que é equivalente a considerar um corte  $\alpha$  nas imagens da função, da seguinte maneira:

$$y_n = \begin{cases} 1, & \text{se } g(y_n) \geq \alpha \\ 0, & \text{se } g(y_n) < \alpha \end{cases} \quad (10)$$

sendo que um valor comum é  $\alpha = 0,5$  (Singh, 2019).

Note que, para prever o valor da variável binomial  $y_n$  a função logística receberá a combinação linear dos parâmetros desconhecidos  $\beta$  e das componentes do vetor aleatório  $\mathbf{x}_n$ , de acordo a equação (9). Ou seja, a função logística implementada é de fato igual a:

$$g\left(\beta_0 + \sum_{k=1}^K \beta_k x_{kn}\right) = \frac{1}{1 + e^{-\beta_0 - \sum_{k=1}^K \beta_k x_{kn}}}. \quad (11)$$

Logo, para um dado corte  $\alpha \in ]0, 1[$  e um vetor aleatório predictor  $\mathbf{x}_n$ , a variável predita  $y_n$  é obtida calculando:

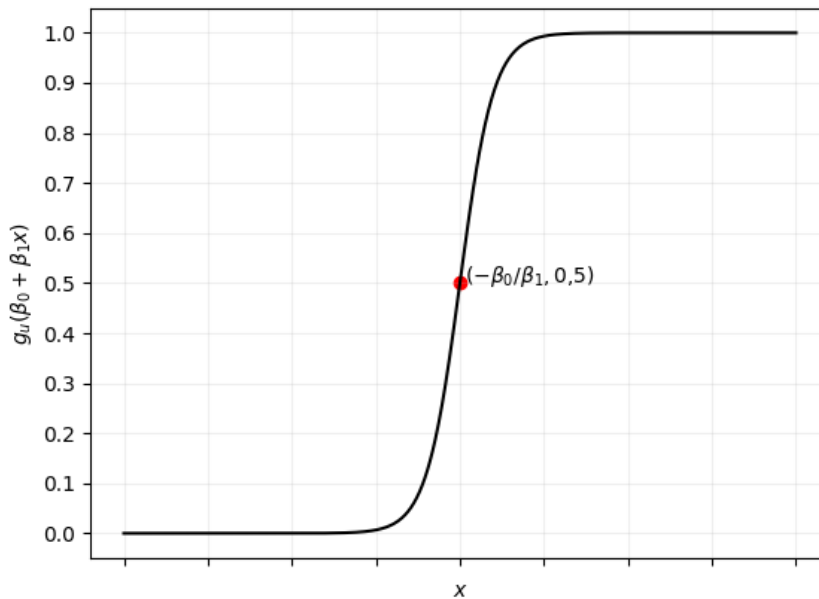
$$y_n = \begin{cases} 1, & \text{se } \frac{1}{1 + e^{-\beta_0 - \sum_{k=1}^K \beta_k x_{kn}}} \geq \alpha \\ 0, & \text{se } \frac{1}{1 + e^{-\beta_0 - \sum_{k=1}^K \beta_k x_{kn}}} < \alpha \end{cases}, \quad (12)$$

Isso significa que as variáveis predictoras que retornam valores da função logística acima ou igual a  $\alpha$  serão atribuídas à predição 1, e caso contrário a predição será 0.

Na Figura 1 está ilustrada uma função logística univariada, que é do tipo sigmoide, isto é

$$g_u(\beta_0 + \beta_1 x) = \frac{1}{1 + e^{-\beta_0 - \beta_1 x}}. \quad (13)$$

com  $\beta_1 > 0$ .



**Figura 2.** Ilustração de uma função logística univariada. Em vermelho está destacado o ponto de inflexão da curva. Fonte: os autores.

Observa-se que a curva ilustrada na Figura 2 tende assintoticamente a 0 e 1, como já mencionado anteriormente, devido a componente exponencial da função logística. O ponto de inflexão da curva é dado pela derivada segunda da função logística igualada a zero:

$$0 = \frac{d^2 g_u(\beta_0 + \beta_1 x)}{dx^2} = \frac{2\beta_1^2 e^{-2(\beta_0 + \beta_1 x)}}{(1 + e^{-\beta_0 - \beta_1 x})^3} - \frac{\beta_1^2 e^{-\beta_0 - \beta_1 x}}{(1 + e^{-\beta_0 - \beta_1 x})^2} \tag{14}$$

cuja solução real é dada por  $\beta_0 + \beta_1 x = 0$ . Logo,

$$x = -\frac{\beta_0}{\beta_1} \tag{15}$$

é a abscissa do ponto de inflexão. Substituindo este valor na função logística  $g_u(\beta_0 + \beta_1 x)$ , resulta que a imagem correspondente é 0,5. Isto significa que a mudança de concavidade da curva logística ocorre no ponto  $(-\beta_0/\beta_1; 0,5)$ , conforme ilustrado na Figura 1, ponto em vermelho. Assim, o valor de  $\beta_0$  desloca a curva horizontalmente proporcionalmente a  $\beta_1^{-1}$ , o que é facilmente entendido ao tomar  $\beta_1 = 1$ . Por outro lado,  $\beta_1$  influencia na inclinação desta curva, pois a derivada primeira é dada por

$$\beta_1 \frac{e^{-\beta_0 - \beta_1 x}}{(1 + e^{-\beta_0 - \beta_1 x})^2} \tag{16}$$

Afim de estimar os parâmetros  $\beta$  da função logística  $g(\beta_0 + \sum_{k=1}^K \beta_k x_{kn})$ , com probabilidade  $p_n$ , é empregado o método da máxima verossimilhança, considerando que  $p_n = 1/(1 + e^{-\beta_0 - \sum_{k=1}^K \beta_k x_{kn}})$  de acordo com equação (7) e que a partir da equação (4) tem-se que:

$$\begin{aligned}
 P(y_n | \mathbf{x}_n) &= p_n^{y_n} (1 - p_n)^{1 - y_n} \\
 &= \left( \frac{1}{1 + e^{-\beta_0 - \sum_{k=1}^K \beta_k x_{kn}}} \right)^{y_n} \left( 1 - \frac{1}{1 + e^{-\beta_0 - \sum_{k=1}^K \beta_k x_{kn}}} \right)^{1 - y_n} \\
 &= \frac{e^{y_n(\beta_0 + \sum_{k=1}^K \beta_k x_{kn})}}{e^{\beta_0 + \sum_{k=1}^K \beta_k x_{kn}} + 1}.
 \end{aligned}$$

Portanto, a função de verossimilhança é

$$L(\boldsymbol{\beta} | \mathbf{y}) = \prod_{n=1}^N P(y_n | \mathbf{x}_n, \boldsymbol{\beta}) = \prod_{n=1}^N \frac{e^{y_n(\beta_0 + \sum_{k=1}^K \beta_k x_{kn})}}{e^{\beta_0 + \sum_{k=1}^K \beta_k x_{kn}} + 1}. \quad (17)$$

Aplicando o logaritmo natural com o intuito de simplificar a função de verossimilhança, obtém-se uma função de  $\boldsymbol{\beta}$ :

$$\begin{aligned}
 E(\boldsymbol{\beta}) &= \ln L(\boldsymbol{\beta} | \mathbf{y}) \\
 &= \ln \prod_{n=1}^N \frac{e^{y_n(\beta_0 + \sum_{k=1}^K \beta_k x_{kn})}}{e^{\beta_0 + \sum_{k=1}^K \beta_k x_{kn}} + 1} \\
 &= \sum_{n=1}^N \ln e^{y_n(\beta_0 + \sum_{k=1}^K \beta_k x_{kn})} - \ln (e^{\beta_0 + \sum_{k=1}^K \beta_k x_{kn}} + 1) \\
 &= \sum_{n=1}^N y_n \left( \beta_0 + \sum_{k=1}^K \beta_k x_{kn} \right) - \ln (e^{\beta_0 + \sum_{k=1}^K \beta_k x_{kn}} + 1)
 \end{aligned} \quad (18)$$

pois os parâmetros  $\boldsymbol{\beta}$  que maximizam a função (17) são os mesmos que maximizam (18), já que a função logaritmo é crescente.

Para encontrar o máximo da função  $\ln L(\boldsymbol{\beta} | \mathbf{y})$  calcula-se a derivada parcial, o gradiente, em relação ao vetor de parâmetros  $\boldsymbol{\beta}$ , dada por:

$$\nabla E = \frac{\partial E(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{n=1}^N \left[ \left( y_n - \frac{1}{e^{\beta_0 + \sum_{k=1}^K \beta_k x_{kn}} + 1} \right) \sum_{k=1}^K x_{kn} \right]. \quad (19)$$

É importante notar que o gradiente segue a direção dada pelo erro entre o valor real  $y_n$  e o valor predito  $\hat{y}_n = 1 / (e^{\hat{\beta}_0 + \sum_{k=1}^K \hat{\beta}_k x_{kn}} + 1)$ , vezes a soma dos atributos.

Igualando o gradiente  $\nabla E$  a zero, os parâmetros  $\boldsymbol{\beta}$  são obtidos pela solução da seguinte equação:

$$\sum_{n=1}^N \left[ \left( y_n - \frac{1}{e^{\hat{\beta}_0 + \sum_{k=1}^K \hat{\beta}_k x_{kn}} + 1} \right) \sum_{k=1}^K x_{kn} \right] = 0, \quad (20)$$

onde  $\hat{\boldsymbol{\beta}}$  é o vetor de parâmetros estimado.



As soluções da equação (20) são alcançadas utilizando métodos de otimização, tais como gradiente descendente, mínimos quadrados e Newton-Raphson. Dentre as soluções possíveis duas delas são obtidas ao considerar valores muito pequenos ou muito grandes para os parâmetros  $\hat{\beta}$ , quando as instâncias de treinamento pertencem restritamente a classe positiva ( $y_n = 1$ ) ou negativa ( $y_n = 0$ ), respectivamente.

Em outros termos, se

$$\lim_{\hat{\beta} \rightarrow 0} e^{\hat{\beta}_0 + \sum_{k=1}^K \hat{\beta}_k x_{kn}} = 0, \quad (21)$$

então:

$$\lim_{\hat{\beta} \rightarrow 0, y_n = 1} y_n - \frac{1}{e^{\hat{\beta}_0 + \sum_{k=1}^K \hat{\beta}_k x_{kn+1}}} = 0, \quad (22)$$

independentemente dos valores dos atributos  $x_{kn}$ .

Considerando as equações (12) e (22), e tomando  $\alpha = 0,5$ , toda instância  $x_n$  será atribuída a classe 1, quando  $\hat{\beta} \rightarrow 0$ . Assim, obtêm-se um modelo  $\hat{h}(x, \beta)$  que é enviesado, porque prevê todos os exemplos como pertencentes a classe 1.

Analogamente, se  $\hat{\beta} \rightarrow \infty$ , então

$$\lim_{\hat{\beta} \rightarrow \infty} e^{\hat{\beta}_0 + \sum_{k=1}^K \hat{\beta}_k x_{kn}} = \infty, \quad (23)$$

portanto,

$$\lim_{\hat{\beta} \rightarrow \infty, y_n = 0} y_n - \frac{1}{e^{\hat{\beta}_0 + \sum_{k=1}^K \hat{\beta}_k x_{kn}} + 1} = 0, \quad (24)$$

também é uma solução da equação (20). Para este caso, a equação (12) sempre retorna a classe 0.

Para evitar que o modelo aprendido preveja apenas uma das classes, utilizam-se técnicas de regularização, as quais restringem os valores que os parâmetros  $\hat{\beta}$  podem assumir, empregando para tanto uma penalização aos valores extremos.

As técnicas de regularização L1 e L2 são implementadas adicionando na função  $E(\beta)$ , equação (18), os termos de penalização  $\lambda \sum_{k=0}^K |\hat{\beta}_k|$  e  $\lambda \sum_{k=0}^K \hat{\beta}_k^2$ , respectivamente, onde  $\lambda > 0$  é um hiperparâmetro que relativiza a importância do termo de regularização. As equações resultantes são

$$E_{L1}(\beta) = E(\beta) + \lambda \sum_{k=0}^K |\hat{\beta}_k| \quad e \quad (25)$$

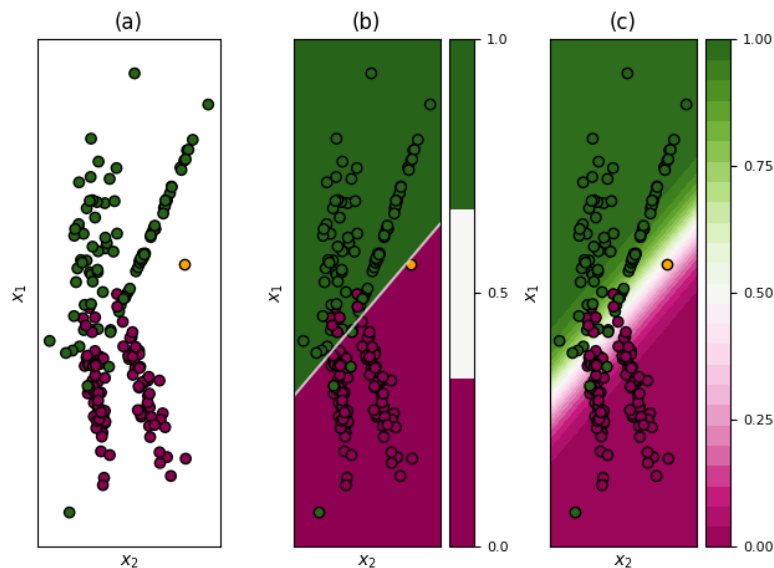
$$E_{L2}(\beta) = E(\beta) + \lambda \sum_{k=0}^K \hat{\beta}_k^2. \quad (26)$$

Para exemplificar a utilização do método de Regressão Logística para classificação de padrões, na Figura 3a estão ilustrados os dados de treinamento, onde os pontos verde e púrpura representam

duas classes distintas, e o ponto amarelo é uma instância não rotulada. Neste caso os vetores de atributos  $\mathbf{x}_n$  possuem apenas duas componentes, isto é,  $\mathbf{x}_n = (x_{1n}, x_{2n})$ .

Na Figura 3b a linha branca é a fronteira de decisão obtida ao empregar a Regressão Logística sobre os dados de treinamento da Figura 3a. Ela é modelada pela equação (9) após terem sido aprendidos os parâmetros  $\beta$ . Sobre essa fronteira a probabilidade de que um ponto pertença a classe verde ou a classe púrpura é a mesma, que é 0,5. Portanto, os pontos que estão localizados sobre a fronteira possuem classificação dúbia. Por outro lado, as superfícies de decisão verde e púrpura possuem probabilidade 1 e 0 para classificação dos exemplos das classes verde. Observe que a classe verde é a classe principal, logo, calcula-se a probabilidade  $p_n$  da instância pertencer a classe verde e a probabilidade  $q_n = 1 - p_n$  de não pertencer a classe verde. Por isso, a cor púrpura está associada a probabilidade 0, na escala de cores da Figura 3b.

Neste exemplo é evidente que algumas instâncias não serão corretamente classificadas.



**Figura 3.** Exemplo de aplicação da Regressão Logística sobre o conjunto de testes (a), gerando as superfícies de decisão (b) e (c). Fonte: os autores.

Na Figura 3c são apresentadas diversas fronteiras de decisão, cada uma com probabilidade maior, quanto mais longe estão da fronteira de decisão com probabilidade de 0,5. Nota-se que quanto mais distante da fronteira de decisão o exemplo estiver, maior será a probabilidade de que esse exemplo pertença à classe que a superfície determina.

Por fim, a instância (ponto amarelo) desconhecida é classificada como pertencente a classe púrpura, pois está localizada abaixo da fronteira de decisão, na superfície correspondente a essa classe.

Uma limitação inerente à Regressão Logística na classificação de padrões é sua capacidade de gerar bons resultados apenas para problemas linearmente separáveis, como aquele ilustrado na Figura

3. Devido a fronteira de decisão gerada ser linear, em ambientes onde as classes se distribuem conforme a Figura 4 (a), por exemplo, a aplicação da Regressão Logística não é apropriada, porque não é possível obter uma fronteira de decisão que separe as classes adequadamente.

Nestas situações uma das abordagens empregadas é a implementação de uma função  $\varphi: \mathbf{x}_n \mapsto \mathbf{z}_n$ , que mapeia o espaço de atributos não-linearmente separável para um espaço linearmente separável. Nesse novo espaço, a regressão logística pode ser empregada, mas agora sobre os vetores de dados mapeados, isto é, sobre  $\mathbf{z}_n$ .

Esta abordagem, aliás, é o foco da pesquisa apresentada neste texto. As funções que serão utilizadas para o mapeamento do espaço de atributos serão apresentadas na próxima seção.

## FUNÇÕES WAVELETS

As “funções wavelets” são funções com características especiais utilizadas na Transformada Wavelet para análise e processamento de sinais. Dentre as várias aplicações desta transformada, aquelas que mais ganharam repercussão estão associadas à compressão e remoção de ruído em sinais e imagens digitais (Mallat, 2009). Para remoção de ruído de sinais de eletrocardiograma (ECG), por exemplo, o método de anulamento dos coeficientes menos significantes utilizando a análise multirresolução wavelet supera os métodos comumente empregados na literatura especializada (Oliveira et al., 2018).

A Transformada de Fourier de um sinal  $f(t) \in L^2(\mathbb{R})$  é definida por Oppenheim, Willisky e Nawab (1997) como a integral imprópria:

$$F_f(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt, \quad (27)$$

onde  $j$  é a unidade imaginária e  $L^2(\mathbb{R})$  é o espaço das funções de quadrado integrável.

Uma função  $\psi(t) \in L^2(\mathbb{R})$ , com norma euclidiana unitária, i.e.,  $\|\psi(t)\|_2 = 1$ , é denominada de **função wavelet** se a condição de admissibilidade dada na equação (28) for satisfeita (Mallat, 2009).

$$C_\psi = 2\pi \int_{-\infty}^{\infty} \frac{|F_\psi(\omega)|^2}{|\omega|} d\omega < \infty \quad (28)$$

A fim de garantir a convergência da integral na equação (28) impõe-se  $F_\psi(0) = 0$  (Daubechies, 1992). Esta restrição, tomando a equação (27), implica que

$$0 = F_\psi(0) = \int_{-\infty}^{\infty} \psi(t)e^{-j0t} dt = \int_{-\infty}^{\infty} \psi(t)dt. \quad (29)$$

A equação (29), que impõe média nula as funções wavelets e a restrição de que  $\psi(t)$  pertença ao espaço das funções de quadrado integrável evidenciam o comportamento ondulatório característico das funções wavelets: a área da curva da função, acima e abaixo do eixo  $t$ , devem ser a mesma e, quando  $t \rightarrow \pm\infty$  então  $\psi(t)$  decai rapidamente, anulando-se. Portanto, a “ondinha” não oscila

indefinidamente, mas num intervalo curto, sendo este seu intervalo suporte. Isto garante que as wavelets sejam bem localizadas no tempo, implicando que sua energia está concentrada em uma região estreita.

Uma família de funções wavelets é obtida inserindo parâmetros de escala  $s$  e deslocamento  $\tau$  à função  $\psi(t)$ , obtendo assim versões escalonadas e transladadas no tempo, conforme equação (30), onde  $|s|^{-1/2}$  é o termo de normalização da função.

$$\psi_{s,\tau}(t) = |s|^{-1/2} \psi\left(\frac{t-\tau}{s}\right). \quad (30)$$

A função wavelet  $\psi(t)$  é denominada de “wavelet mãe” e esta tem energia unitária, pois as “wavelets filhas” são restringidas à normalização  $\|\psi_{s,\tau}(t)\|_2 = \|\psi(t)\|_2 = 1$ , implicando que:

$$1 = \|\psi(t)\|_2 = \int_{-\infty}^{\infty} |\psi(t)|^2 dt. \quad (31)$$

A equação (30) possibilita localizar, em um vetor, estruturas temporais tão grandes ou tão pequenas quanto se deseja, bastando para isto escolher adequadamente o parâmetro de escala  $s$  e transladar a wavelet resultante por toda a extensão do vetor. Esta, aliás, é a característica da análise Wavelet que a torna uma ferramenta mais adequada que a Transformada de Fourier de Tempo Curto para análise de transitórios.

Quando o parâmetro de escala é  $0 < s < 1$ ,  $\psi_{s,\tau}(t)$  é uma versão comprimida de  $\psi(t)$  ao longo do eixo horizontal. Ao deslocar esta versão ao longo do vetor, multiplicando-a por cada seguimento determinado pela janela wavelet, o resultado é que aqueles seguimentos com maiores oscilações serão melhor localizados do que quando se considera uma versão de  $\psi(t)$  mais expandida, que é obtida escolhendo-se  $s > 1$ . É claro que quando a wavelet é escalonada seu suporte é alterado. Assim, se a wavelet  $\psi(t)$  tem um suporte de tamanho  $N$  então  $\psi_{s,\tau}(t)$  terá um suporte de tamanho  $|s|N$ . Quando esta wavelet é deslocada, então seu suporte fixa-se próximo a sua localização temporal, ou seja, em  $t = \tau$  (Kaiser, 1994).

Além do parâmetro de escala, a quantidade de momentos nulos de uma wavelet também afeta a forma de onda da função. Para uma função wavelet  $\psi(t) \in L^2(\mathbb{R})$ , o  $g$ -ésimo momento nulo é dado pela integral

$$m_g = \int_{-\infty}^{\infty} t^g \psi(t) dt, \quad g = 0, 1, 2, \dots, G - 1. \quad (32)$$

O teorema do momento para a Transformada de Fourier de uma função  $\psi$  consiste do seguinte resultado (Vetterli e Kovacevic, 2007):

$$\left. \frac{\partial^g F_\psi(\omega)}{\partial \omega^g} \right|_{\omega=0} = (-j)^g m_g. \quad (33)$$

Considerando o par da Transformada de Fourier,  $\psi(t) \leftrightarrow F_\psi(\omega)$  e o valor da transformada na origem, estabelecido anteriormente para a condição de admissibilidade, a saber,  $F_\psi(0) = 0$ , e substituindo estes na expressão (33), se tem:

$$\int_{-\infty}^{\infty} t^g \psi(t) dt = 0, g = 0, 1, 2, \dots, G - 1. \quad (34)$$

Se o valor da integral na equação (34) se anula para  $G - 1$  momentos, então diz-se que a função wavelet  $\psi$  tem  $G$  momentos nulos. Esse resultado é utilizado pelo Teorema 1 para definir as funções wavelets como resultantes das  $G$ -ésimas derivada de uma função de suavização  $\theta(t)$ .

**Teorema 1.** Uma wavelet  $\psi(t)$ , com suporte compacto, tem  $G$  momentos nulos se, e somente se, existe uma função  $\theta(t)$  com decaimento rápido, tal que

$$\psi(t) = (-1)^G \frac{d^G \theta(t)}{dt^G}. \quad (35)$$

Por meio do Teorema 1 pode-se estabelecer a seguinte notação  $\psi_{s,\tau,G}(t) = (-1)^G \frac{d^G}{dt^G} \left[ |s|^{-1/2} \psi \left( \frac{t-\tau}{s} \right) \right]$ , para uma família de funções wavelets na escala  $s$ , deslocamento  $\tau$  e  $G$  momentos nulos. Entretanto, no contexto de classificação de padrões, os vetores de atributos não estão indexados temporalmente, ou seja, a ordem com que os atributos estão inseridos no vetor não é relevante para a tarefa de classificação. Por isso, a partir de agora o parâmetro de deslocamento será desconsiderado. Logo, as funções wavelets serão implementadas apenas em relação aos parâmetros de escala e momentos nulos, isto é,  $\psi_{s,G}(t) = (-1)^G \frac{d^G}{dt^G} \left[ |s|^{-1/2} \psi \left( \frac{t}{s} \right) \right]$ .

Dentre as diversas funções wavelets que são expressas como a derivada de uma função de suavização, conforme equação (35), recentemente uma nova família foi proposta por Lamblém et. al (2016), a qual é denominada Golden wavelets em referência ao famoso número de ouro. Essas funções apresentam resultados superiores às funções wavelets clássicas como a Mexican Hat, na análise da regularidade de sinais com descontinuidades (Gossler et al., 2018), e às wavelets Gaussianas e Daubechies (para alguns momentos nulos) na aplicação de extração de atributos para reconhecimento de arritmias cardíacas (Gossler et al., 2016).

Essas funções são obtidas pela derivada do quociente entre polinômios com coeficientes de Fibonacci, que se expressam por meio da igualdade

$$p_k(t) = t p_{k-1}(t) + H_{k-1}, k \geq 1, \quad (36)$$

onde  $H_{k-1}$  é o  $k$ -ésimo termo da sequência de Fibonacci  $\{1, 1, 2, 3, 5, 8, 13, 21, \dots\}$ , que é formada pela soma dos dois termos anteriores.

Um exemplo de Golden wavelet é obtido considerando a função de suavização  $\theta(t)$  como o quociente entre os polinômios  $p_0(t) = 1$  e  $p_2(t) = t^2 + t + 2$  (Gossler et al., 2018), isto é:

$$\theta(t) = \frac{1}{t^2 + t + 2}, \quad (37)$$

e aplicando a derivação, equação (35):

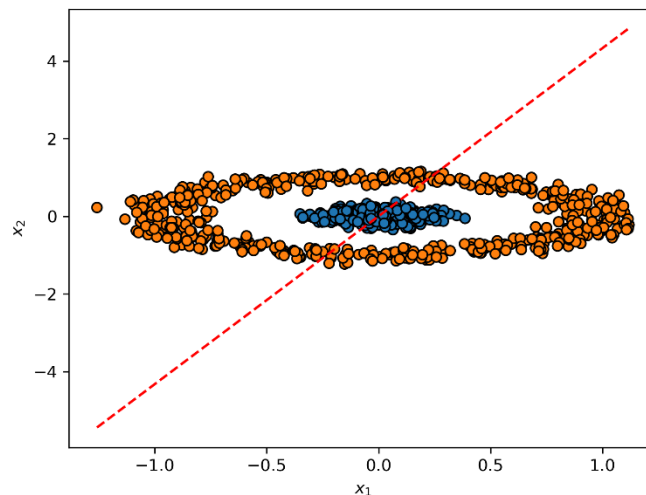
$$\psi(t) = -\frac{2t + 1}{(t^2 + t + 2)^2}, \quad (38)$$

Resultando em uma wavelet com um momento nulo. Para obter wavelets com mais momentos nulos basta derivar sucessivamente a equação (38). Na análise comparativa apresentada em Gossler et al. (2018), estuda-se a função Golden Hat, que possui quatro momentos nulos.

Na seção ulterior as soluções de problemas de classificação não-linearmente separáveis são abordadas e proposta uma técnica baseada nas funções Golden wavelets para mapeamento do espaço de atributos.

## RESULTADOS E DISCUSSÕES

O primeiro problema investigado está ilustrado na Figura 4, onde duas classes, amarelo e azul, são representadas por dois atributos  $x_1$  e  $x_2$ , onde a linha vermelha tracejada é uma fronteira de decisão aprendida pela Regressão Logística, cuja expressão é dada por  $y_n = 1,11 \times 10^{-5} - 0,012x_1 - 0,002x_2$ .



**Figura 4.** Exemplo de um problema não-linearmente separável. A linha vermelha tracejada representa a fronteira de decisão obtida empregando uma Regressão Logística. Fonte: os autores.

Este é claramente um problema do tipo não-linearmente separável, porque nenhuma fronteira de decisão linear aprendida é capaz de separar as classes adequadamente em superfícies de decisão que contenham a maior quantidade possível de instância da classe relativa a ela, lembrando que essas fronteiras devem particionar o plano em dois outros planos disjuntos. Deste modo, a Regressão

Logística não fornecerá bons resultados de classificação. Mais especificamente, neste exemplo ocorrem 511 erros de predição, de um total de 1.000 instâncias utilizadas para treinamento/teste. Observe que neste caso não foi selecionado outro conjunto para testar o modelo induzido, pois não se faz necessário nesse momento.

Um erro de cerca de 50% em um problema binário balanceado, como o exemplo em discussão, significa que a fronteira de decisão aprendida é irrelevante, porque pode-se simplesmente escolher uma classe ou outra para cada instância apresentada, que o resultado seria análogo.

Com o objetivo de tornar linearmente separável o espaço de atributos ilustrado na Figura 4 é proposto o mapeamento a seguir, equação (39), baseado na Golden wavelet descrita na equação (38):

$$\mathbf{z}_n = \varphi(\mathbf{x}_n) = [\psi_{s,G}(x_{1n}, x_{2n})]^T, \quad (39)$$

para todo  $n$ , recordando que  $\mathbf{x}_n = (x_{1n}, x_{2n})$  é o  $n$ -ésimo padrão que carrega informações características de um certo ambiente, e  $\mathbf{z}_n$  o respectivo padrão mapeado, sendo  $G$  e  $s$  a quantidade de momentos nulos e a escala considerada, respectivamente, para as funções wavelets;  $\psi_{s,G}(x_{1n}, x_{2n})$  significa que a função wavelet  $\psi_{s,G}$  é aplicada em cada coordenada, ou alternativamente, podemos também pensar na representação bidimensional dada na equação (40)

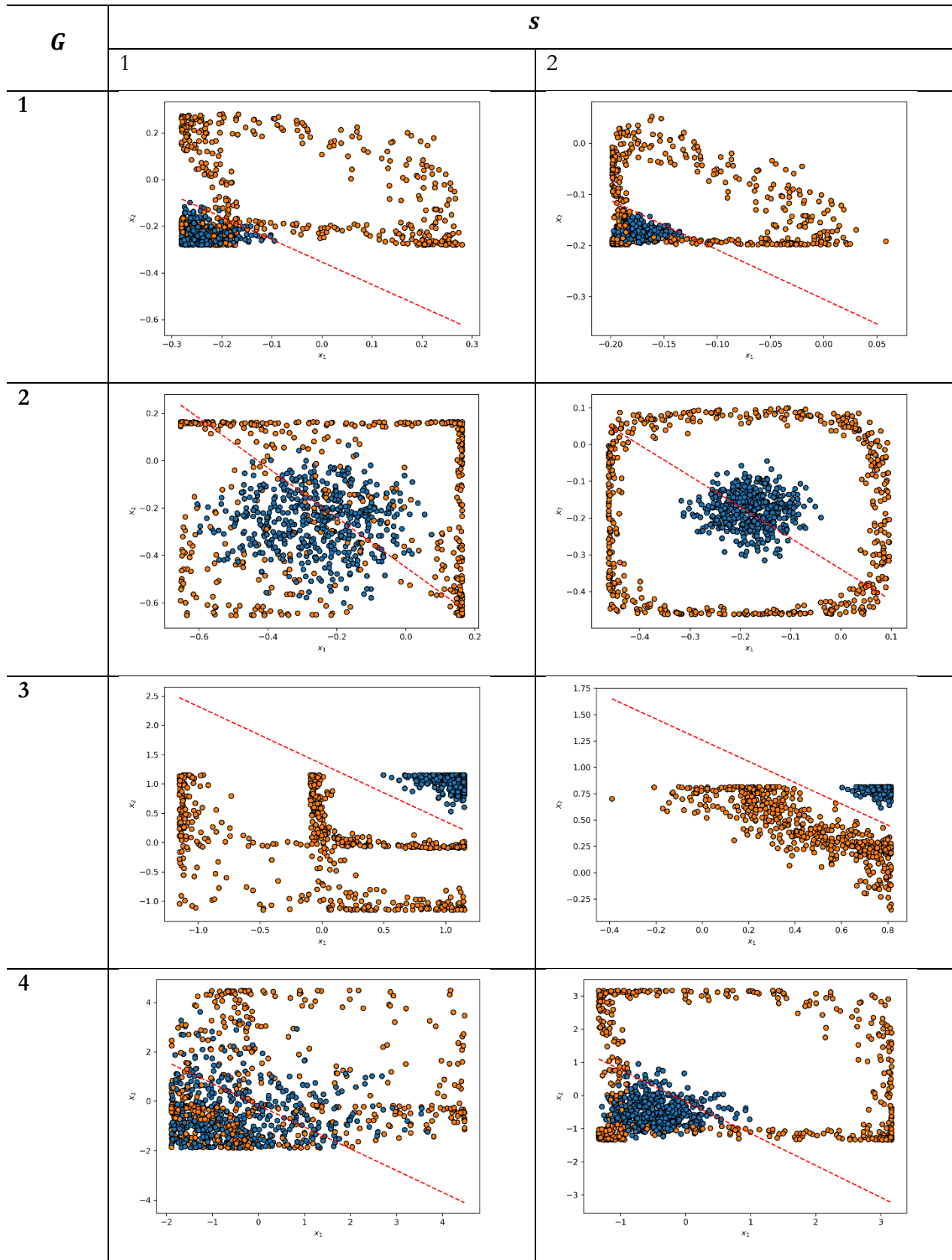
$$\psi_{s,G}(x_1, x_2) = \left( (-1)^G \frac{d^G}{dt^G} \left[ |s|^{-1/2} \psi \left( \frac{x_1}{s} \right) \right], (-1)^G \frac{d^G}{dt^G} \left[ |s|^{-1/2} \psi \left( \frac{x_2}{s} \right) \right] \right). \quad (40)$$

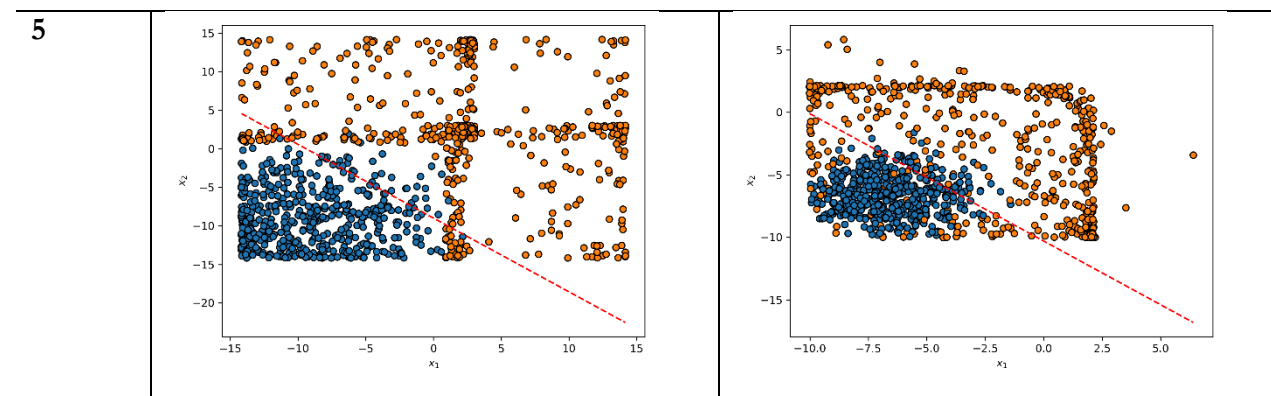
No

Quadro 1 estão ilustradas as fronteiras de decisão, linhas vermelhas tracejadas, nos espaços de atributos mapeados pelas Golden wavelets para distintos momentos nulos  $G$  e escalas  $s$ . Na Tabela 1 estão exibidos os erros de estimação, calculados pela equação (1), gerados ao tomar os espaços de atributos mapeados apresentados no Quadro 1.



**Quadro 1.** Fronteiras de decisão (linha vermelha tracejada) considerando funções Golden wavelets com diferentes momentos nulos ( $G$ ) e em distintas escalas ( $s$ ), para o problema ilustrado na Figura 4. Fonte: os autores.





**Tabela 1.** Erros de estimação considerando funções Golden wavelets com diferentes momentos nulos ( $G$ ) e em distintas escalas ( $s$ ). Fonte: Os autores.

$G$	$s$	
	1	2
1	168	202
2	393	461
3	0	0
4	392	235
5	92	140

Observa-se na Tabela 1 que apenas quando são implementadas as Golden wavelets  $\psi_{1,3}$  e  $\psi_{2,3}$  é que os espaços de atributos mapeados se tornam perfeitamente (erro nulo) linearmente separáveis. As demais possibilidades também geram espaços linearmente separáveis, mas com maiores erros de predição, como por exemplo, a utilização das Golden wavelet  $\psi_{1,5}$ . O pior resultado é obtido implementando a Golden wavelet  $\psi_{2,2}$  e, pela fronteira respectiva do

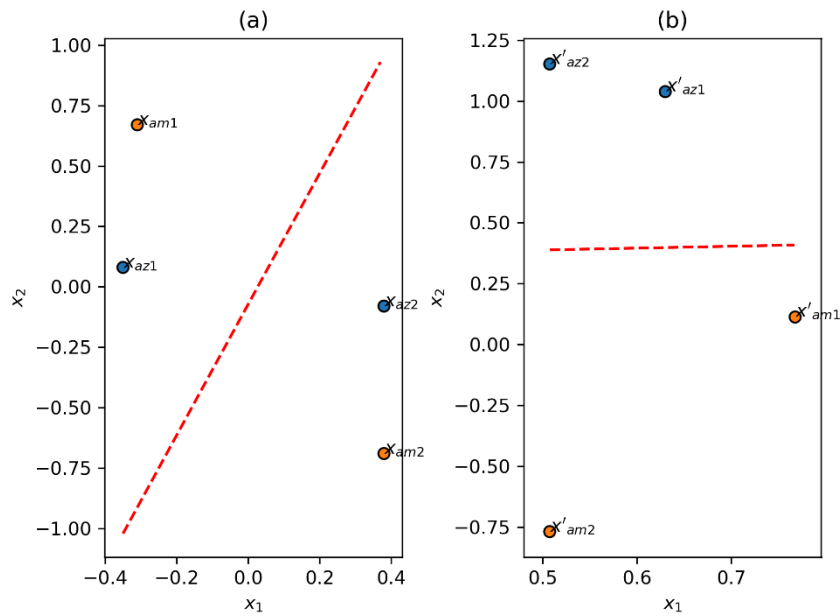
Quadro 1, nota-se que o padrão de dispersão dos atributos neste caso é análogo ao do problema original, Figura 4, exceto que ocorre uma dispersão maior dos atributos.

Em resumo, as Golden wavelets, equação (38), com 3 momentos nulos, são adequadas para transformação do espaço de atributos daqueles problemas similares ao ilustrado na Figura 4. Para compreender porque estas wavelets são adequadas é necessário analisar sua expressão analítica, que é dada por:

$$\psi_{s,3}(t) = -\frac{6\left(2\frac{t}{s} + 1\right)\left[2\left(\frac{t}{s}\right)^2 + 2\frac{t}{s} - 3\right]}{\left[2\left(\frac{t}{s}\right)^2 + \frac{t}{s} + 2\right]^4}. \quad (41)$$

Para exemplificar, tomam-se quatro instâncias:  $x_{am1} = (-0,31; 0,67)$ ,  $x_{am2} = (0,38; -0,69)$ ,  $x_{az1} = (-0,35; 0,08)$  e  $x_{az2} = (0,38; -0,04)$  das classes amarelo e azul, respectivamente, as quais estão localizadas na mesma superfície de decisão gerada pela Regressão Logística, significando que, pelo modelo aprendido, pertencem a mesma classe, ou seja,  $x_{am1}$  e  $x_{az1}$  estão no lado superior da fronteira de decisão da Figura 4, e  $x_{am2}$  e  $x_{az2}$ , no lado inferior (Veja Figura 5 (a)).

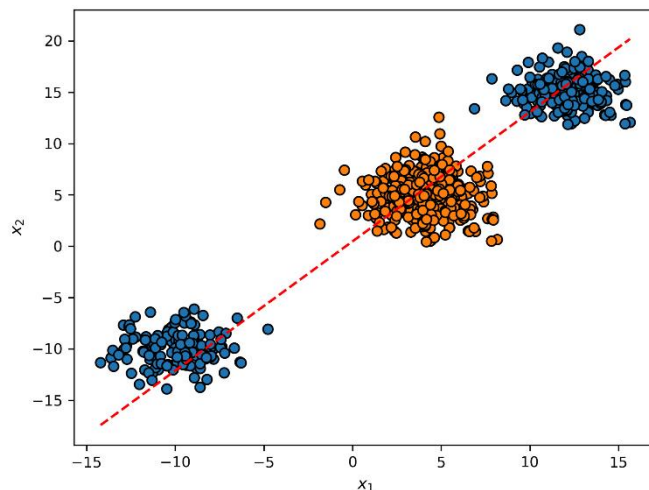
O mapeamento realizado pela Golden wavelet  $\psi_{1,3}$ , mapeia as instâncias  $x_{am1}$ ,  $x_{am2}$ ,  $x_{az1}$  e  $x_{az2}$  nas localizações:  $x'_{am1} = (0,76; 0,11)$ ,  $x'_{am2} = (0,50; -0,76)$ ,  $x'_{az1} = (0,63; 1,03)$  e  $x'_{az2} = (0,50; 1,51)$ , respectivamente. Na Figura 5 estão ilustradas as novas localizações destas instâncias bem como as fronteiras de decisões geradas antes do mapeamento, Figura 5 (a), e após o mapeamento, Figura 5 (b), representadas pela linha tracejada vermelha.



**Figura 5.** Exemplo de mapeamento para instâncias específicas. Superfícies: (a) não-linearmente separável e (b) linearmente separável após mapeamento utilizando a Golden wavelet da equação (40). As linhas tracejadas vermelhas representam as fronteiras de decisão. Fonte: os autores.

Um segundo problema de classes não-linearmente separáveis está ilustrado na Figura 6. Nesse, embora as instâncias das classes estejam distribuídas em núcleos pouco dispersos, há um núcleo da classe amarela entre dois núcleos da classe azul, o que impede que uma reta seja aprendida para separação das classes. A linha tracejada exibida na Figura 6, separa as classes com um erro de 287 instâncias de um total de 700, correspondendo a 41% de erro. Uma fronteira de decisão adequada neste exemplo seria uma fronteira circular em torno das instâncias da classe amarela. Assim, as instâncias dentro da fronteira pertenceriam a classe amarela e aquelas que estivessem fora a classe azul.

Para resolver este problema pela Regressão Logística, novamente o mapeamento baseado na Golden wavelet, equação (38), é empregado, mas desta vez apenas os resultados para as escalas 1 e 3 são exibidos, no Quadros 2 e Tabela 2.

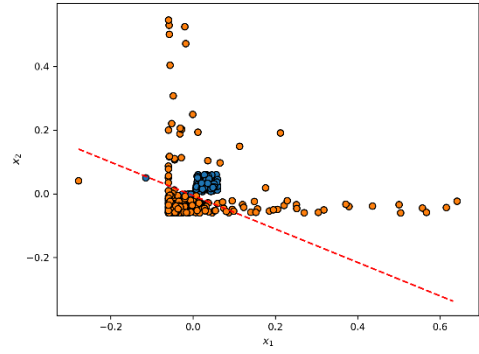
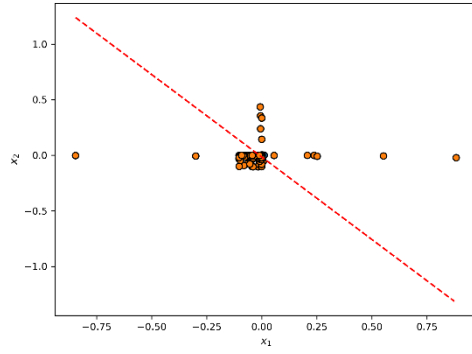


**Figura 6.** Um problema não-linearmente separável. A classe azul se distribui em dois núcleos, tornando impossível encontrar uma reta que separe ela da classe amarela. Fonte: os autores.

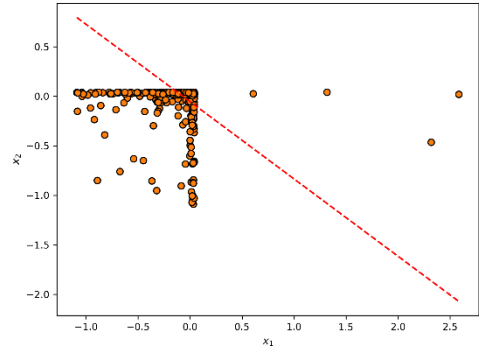
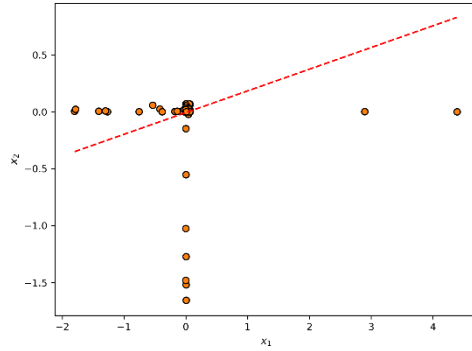
**Quadro 2.** Fronteiras de decisão (linha vermelha tracejada) considerando funções Golden wavelets com diferentes momentos nulos ( $G$ ) e em distintas escalas ( $s$ ), para o problema ilustrado na Figura 6. Fonte: os autores.

$G$	$s$	
	1	3
1		
2		

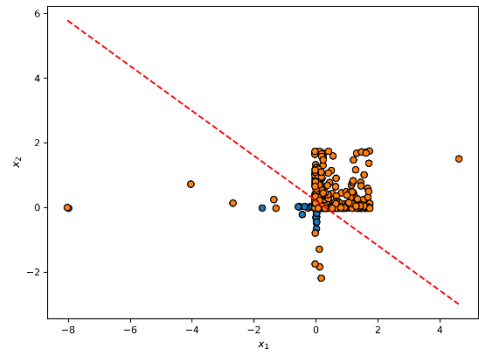
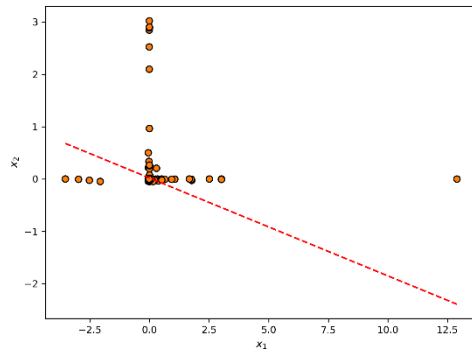
3



4



5



**Tabela 2.** Erros de estimação considerando funções Golden wavelets com diferentes momentos nulos ( $G$ ) e em distintas escalas ( $s$ ). Fonte: os autores.

$G$	$s$	
	1	3
1	115	2
2	143	75
3	135	131
4	312	182
5	317	131

Pelos resultados apresentados, é claro que utilizar a Golden wavelet  $\psi_{3,1}$ , que resulta em um erro de apenas duas instâncias, é a melhor opção. Na imagem correspondente no Quadro 2 vê-se como as classes foram bem separadas ao tomar esse mapeamento.

Nos experimentos anteriores foram fixados apenas alguns valores para a escala e momentos nulos, no entanto, pode-se variar tais parâmetros conforme se queira, tendo em mente que a escala  $s$  é um número real positivo e a quantidade de momentos nulos é um inteiro positivo. Além disso, a escala dilata ou comprime a wavelet, tendo conseqüentemente um resultado de dilatação ou contração do espalhamento das instâncias, enquanto que, a quantidade de momentos nulos está relacionada a forma de onda da função, pois esta varia conforme a ordem das derivadas. Pois isso, o resultado no espaço mapeado é muito mais distinto ao variar o parâmetro  $G$ , do que ao variar a escala  $s$ .

A solução de mapeamento apresentada baseada nas Golden wavelets pode ser entendida como um método de aprendizado de máquina não-supervisionado, analogamente ao método Análise de Componentes Principais, pois tais métodos representam transformações do espaço de entrada de dados.

Devido a essa característica, o método de mapeamento proposto deve empregar um esquema de aprendizagem seletivo, isto é, sobre o conjunto de treinamento testa-se as funções wavelets para transformação do espaço de atributos não-linearmente separável para um espaço linearmente separável. Após, na fase de testes, emprega-se aquela função wavelet que gerou o menor erro.

Além disso, a escolha da função wavelet também é uma tarefa importante, pois pode-se constatar pelos experimentos que a forma de onda da função afeta o resultado do mapeamento. As funções Golden wavelets são adequadas para este propósito, pois muitas formas de ondas podem ser geradas bastando tomar distintas razões entre polinômios com coeficientes de Fibonacci (Gossler, 2016).

Geralmente, espera-se que a performance não seja a mesma sobre os conjuntos de indução e teste, mas tal resultado é comum mesmo nos esquemas de aprendizagem supervisionada.

Em síntese, este ensaio apresentou uma proposta de utilização das funções Golden wavelets para mapeamento do espaço de atributos não-linearmente separáveis para espaços linearmente separáveis, e algumas aplicações utilizando o método Regressão Logística para obtenção das fronteiras de decisão. A abordagem proposta pode, evidentemente, ser aplicada para qualquer algoritmo que gere fronteiras de decisão lineares.

Em trabalhos futuros pretende-se investigar, em problemas reais, como o mapeamento proposto pode melhorar os resultados de performance, e quais das diversas funções Golden wavelets são mais adequadas a resolução destes problemas.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Basu S, Mitra S, S N (2020). Deep Learning for Screening COVID-19 using Chest X-Ray Images. *Cold Spring Harbor Laboratory Press*, DOI: 10.1101/2020.05.04.20090423.
- Bishop CM (2006). *Pattern Recognition and Machine Learning*. New York: Springer. 758p.
- Daubechies I (1992). Ten Lectures on Wavelets. Philadelphia: Society for Industrial and Applied Mathematics, 61, 343p.
- Fávero LP, Belfiore P, Silva FL da, Chan BL (2009). *Análise de Dados: Modelagem Multivariada para Tomada de Decisão*. São Paulo: Elsevier, Campus. 544p.
- Fisher M, Lenz HJ (1996). *Learning from Data: Artificial Intelligence and Statistics V*. Springer: New York. 443p.
- Gossler FE, Oliveira BR, Duarte MAQ, Lamblém RL, Villarreal F (2018). A comparative study between mexican and golden hat wavelets. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 6(2).
- Gossler FE, Oliveira BR, Duarte MAQ, Lamblém RL, Villarreal F (2016). A wavelet generated from Fibonacci-coefficient polynomials and its application in cardiac arrhythmia classification. Proceedings of XIX ENMC - National Meeting on Computational Modeling and VII ECTM - Meeting on Materials Science and Technology, October, João Pessoa.
- Gossler FE (2016). *Wavelets e polinômios com coeficientes de Fibonacci*. Dissertação de Mestrado. Programa de pós-graduação em Engenharia Elétrica (UNESP-Ilha Solteira). Disponível em: <<https://repositorio.unesp.br/handle/11449/148776>>.
- Haykin S (2007). *Redes Neurais: princípios e práticas*. 2 ed. Bookman: São Paulo. 900p.
- Kaiser G (1994). *A Friendly Guide to Wavelets*. Boston: Birkhäuser. 300p.
- Kubat M (2015). *An Introduction to Machine Learning*. Springer: New York. 296p.
- Lamblém RL, Gosler FE, Duarte MAQ, Villarreal F (2016). *Polinômios com coeficientes de Fibonacci e wavelets*. CNMAC, 5(1): 1-6.



- Loesch C., Hoeltgebaum M (2012). *Métodos estatísticos multivariados*. São Paulo: Editora Saraiva. 267p.
- Mallat S (2009). *A Wavelet Tour of Signal Processing: The Sparse Way*. 3 ed. New York: Elsevier. 805p.
- Oliveira BR de, Duarte MAQ, Abreu CCE de, Vieira Filho J (2018). *A wavelet-based method for power-line interference removal in ECG signals*. Res. Biomed. Eng, 34(1): 73-86. DOI: 10.1590/2446-4740.01817.
- Oppenheim AV, Willisky AS, Nawab SH (1997). *Signals and Systems*. 2 ed. New Jersey: Prentice Hall. 957p.
- Schwarting W, Alonso-Mora J, Rus D (2018). *Planning and Decision-Making for Autonomous Vehicles*. Annual Review of Control, Robotics, and Autonomous Systems, (1):187-210. DOI: 10.1146/annurev-control-060117-105157.
- Singh P (2019). *Machine Learning with PySpark: With Natural Language Processing and Recommender Systems*. Apress: New York.
- Vetterli M, Kovacevic J (2007). *Wavelets and Subband Coding*. New Jersey: Prentice Hall PTR.

## ÍNDICE REMISSIVO

### A

agrupamento, 49, 53, 54  
aprendizado de máquina, 8, 25, 28, 29, 59, 60, 61, 80  
Aprendizagem de Máquina, 50, 57, 83

### C

caso de uso, 41, 42, 44, 46  
CCA, 84, 85, 86, 87, 88, 95  
cenário, 39, 43, 44, 45, 85  
Cenas Acústicas, 83, 84, 95

### Ch

*checklist*, 34, 39, 41

### C

Ciência de Dados, 33  
classificação, 31, 39, 50, 56, 59, 60, 61, 62, 63, 68, 69, 72, 73, 74, 84, 85, 88, 90, 94  
clusterização, 50, 51, 54  
coeficiente, 10, 11, 89, 94  
comportamento, 8, 9, 22, 26, 30, 32, 37, 56, 70

### D

*Deep Learning*, 31, 32, 59, 81, 87, 90, 94  
desenvolvimento de SI, 35, 36, 39  
domótica, 25, 26

### E

Engenharia de Requisitos, 33, 47, 48  
escalograma, 89, 90  
espaço de atributos, 59, 60, 70, 73, 74, 76, 80, 81  
espectro, 86, 89  
espectrograma, 85, 86

### F

*Farthest First*, 49, 50, 53, 54, 56

filtro, 8, 9, 10, 11, 15, 16, 17, 18, 19, 20, 21, 22, 89

### G

GWS, 89, 90, 91, 92, 93, 94

### I

Inteligência Artificial, 25

### M

*machine learning*, 30, 31, 84, 87  
mapeamento, 48, 60, 70, 73, 74, 77, 78, 80, 81, 88  
metodologia, 33, 34, 35, 36, 37, 39, 41, 42, 47  
MFCCs, 84, 86, 87, 90, 91, 93, 94

### O

obesidade, 49, 50, 55, 56, 57, 58

### P

polinômio, 10, 11, 16, 20, 21  
prototipação, 33, 35, 39, 41, 47

### Q

qualidade, 33, 34, 35, 39, 41, 47, 57

### R

Redes Neurais  
  Artificiais, 25, 26  
  LSTM, 26  
regressão logística, 59, 62, 63, 70  
requisitos, 28, 33, 34, 35, 39, 41, 48

### S

Savitzky-Golay, 8, 9, 10, 15, 16, 17, 18, 19, 20, 21, 22, 23  
sensores, 26, 27, 29  
Série Temporal, 8, 9  
Sistema de Informação, 47, 48

*Software*, 47, 48, 95

superfície de decisão, 62, 77

## **T**

tendência, 8, 9, 10, 14, 15, 16, 17, 18, 19, 21,  
22, 57, 58

Transformada Wavelet, 8, 9, 10, 13, 70, 83, 84,  
87, 88, 90, 95



**Pantanal Editora**

Rua Abaete, 83, Sala B, Centro. CEP: 78690-000

Nova Xavantina – Mato Grosso – Brasil

Telefone (66) 99682-4165 (Whatsapp)

<https://www.editorapantanal.com.br>

[contato@editorapantanal.com.br](mailto:contato@editorapantanal.com.br)

ISBN 978-658831926-0



9

786588

319260