VIVIANE COSTA SILVA

ANÁLISE COMPARATIVA ENTRE MODELOS DE REGRESSÃO DISTRIBUCIONAL E OS PRINCIPAIS ALGORITMOS DE APRENDIZADO DE MÁQUINA NA PREDIÇÃO DE DADOS METEOROLÓGICOS

LAVRAS – MG

2024

VIVIANE COSTA SILVA

ANÁLISE COMPARATIVA ENTRE MODELOS DE REGRESSÃO DISTRIBUCIONAL E OS PRINCIPAIS ALGORITMOS DE APRENDIZADO DE MÁQUINA NA PREDIÇÃO DE DADOS METEOROLÓGICOS

Dissertação apresentada à Universidade Federal de Lavras como parte das exigências do Programa de Pós-Graduação em Estatística e Experimentação Agropecuária, para obtenção do título de Mestre.

Prof. DSc. Luiz Ricardo Nakamura
Orientador

Prof. DSc. Geraldo Magela da Cruz Pereira
Coorientador

LAVRAS – MG
2024

Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Silva, Viviane Costa
Análise comparativa entre modelos de regressão distribucional e os principais algoritmos de aprendizado de máquina na predição de dados meteorológicos /
Viviane Costa Silva. – 2024.
91 p. : il.

Dissertação(mestrado acadêmico)–Universidade Federal de Lavras, 2024.
Orientador: Prof. DSc. Luiz Ricardo Nakamura.
Coorientador: Prof. DSc. Geraldo Magela da Cruz Pereira.
Bibliografia.

1. Aprendizado de Máquina. 2. Árvores Aleatórias. 3. GAMLSS. I. Nakamura, Luiz Ricardo. II. Pereira, Geraldo Magela

VIVIANE COSTA SILVA

ANÁLISE COMPARATIVA ENTRE MODELOS DE REGRESSÃO DISTRIBUCIONAL E OS PRINCIPAIS ALGORITMOS DE APRENDIZADO DE MÁQUINA NA PREDIÇÃO DE DADOS METEOROLÓGICOS

COMPARATIVE ANALYSIS OF DISTRIBUTIONAL REGRESSION MODELS AND MAJOR MACHINE LEARNING ALGORITHMS FOR METEOROLOGICAL DATA PREDICTION

Dissertação apresentada à Universidade Federal de Lavras como parte das exigências do Programa de Pós-Graduação em Estatística e Experimentação Agropecuária, para obtenção do título de Mestre.

APROVADA em 19 de fevereiro de 2024.

Prof. Dr. Paulo Henrique Sales Guimarães	UFLA
Prof. Dr. Tiago Almeida de Oliveira	UEPB
Prof. Dr. Thiago Gentil Ramires	UTFPR

Prof. DSc. Luiz Ricardo Nakamura
Orientador

Prof. DSc. Geraldo Magela da Cruz Pereira
Co-Orientador

LAVRAS – MG
2024

Este trabalho de pesquisa é inteiramente dedicado aos meus pais Lurdinez e Evangelista e as minhas irmãs Vanessa, Vaniara e Vitória (In Memorian). Os maiores incentivadores das realizações dos meus sonhos. Muito obrigada!

AGRADECIMENTOS

Primeiramente, expresso minha profunda gratidão ao Bom Deus, cuja presença sustentou-me ao longo desses dois anos de mestrado. A Ele, toda honra e glória. Agradeço à amada Nossa Senhora do Perpétuo Socorro por sua intercessão constante em meu favor.

Aos meus pais, Lurdinez Costa e Evangelista Silva, merecem uma gratidão especial por seu amor, incentivo e apoio incansável. Cada esforço para me dar o melhor que podiam não passou e nunca passará despercebido. Agradeço também às minhas irmãs, Vanessa Costa e Vaniara Costa, verdadeiras incentivadoras dos meus estudos, por suas palavras e atitudes de amor, por não me deiarem nunca me sentir sozinha.

Ao meu orientador, Profº Drº Luiz Ricardo Nakamura, que vai além do papel de orientador, sendo também um inspirador e amigo. Agradeço por compartilhar seus conhecimentos e por sua humanidade nos meus momentos difíceis. Foi o orientador certo no momento certo, serei eternamente grata por tudo que fez por mim.

Meu co-orientador, Profº Drº Geraldo Magela da Cruz Pereira, que demonstrou uma generosidade e paciência incomparáveis ao transmitir seus conhecimentos. Suas orientações foram fundamentais para o meu desenvolvimento profissional.

À banca examinadora, Profº. Drº. Paulo Henrique Sales Guimarães (UFLA), Profº. Drº. Tiago Almeida de Oliveira (UEPB), Profº. Drº. Thiago Gentil Ramire (UTFPR), agradeço pelas valiosas contribuições que tornaram este trabalho digno de aprovação.

Aos colegas de pós-graduação, agradeço por compartilharem conhecimento, tirarem dúvidas e facilitarem minha adaptação a uma nova cidade. Que nossa amizade continue contribuindo para a ciência e que pendure por toda a vida.

Um agradecimento especial a José Adeilton (Alemão), Doralice Teixeira, Eduardo Gomes e Regimário Moura que em um momento de precisão me ajudaram e não mediram esforços para que eu vinhesse para Minas Gerais, serei eternamente grata por sua generosidade e bondade para comigo, Deus os recompesem com muitas bençãos em suas vidas.

Agradeço aos amigos, Adilson Teixeira, Yohana Hoffmann, Mateus Peixoto, Rafaella Santos, Maria Beatriz, Luiz Otávio, Pe. Sergio Fernando, Pe. Sandro Sebastião, Douglas Silva, Hérica Faria, Jocilene Pereira, Lucas Ferreira e Diana Rebaza por todas as vezes que precisei de um ombro amigo e por estarem presente na minha vida.

Aos professores do Departamento de Estatística e Experimentação Agropecuária, meu reconhecimento pelas excelentes aulas e pela contribuição para minha formação.

A Universidade Federal de Lavras, na pessoa do Profº Drº Tales Jesus Fernandes, coordenador do Departamento de Estatistica e Experimentação Agropecuária, expresso a essa Instituição minha gratidão por permitir que trilhasse o árduo, mas satisfatório caminho acadêmico do mestrado.

Por fim, e não menos importante, agradeço o apoio financeiro da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) durante esses dois anos de estudo.

Confido fide et spe

RESUMO

Os modelos de regressão univariados remotam ao século XIX e visam compreender como um conjunto de variáveis explicativas influencia ou explica uma variável resposta. Embora seja comum encontrar trabalhos que comparem metodologias flexíveis de aprendizado de máquina com modelos de regressão convencionais, essa comparação pode não ser adequada, devido às pressuposições rigorosas e a restrição de flexibilidade dos modelos de regressão usuais. Assim, esta dissertação propõe verificar e comparar o desempenho dos modelos de regressão distribucional, inicialmente propostos como modelos aditivos generalizados para locação, escala e forma (GAMLSS), que são uma abordagem mais moderna e flexível, com outros algoritmos de aprendizado de máquina comumente empregados na literatura, a saber: *random forest*, *support vector regression*, *extreme gradient boosting* e *prophet*, para conjuntos de dados meteorológicos. Em um primeiro artigo, já publicado em um periódico, foi destacada a necessidade de utilizar os GAMLSS na modelagem da temperatura média diária em um período de um ano na cidade de Florianópolis – SC. Esse estudo mostrou que modelos de regressão menos complexos não seriam adequados para explicar completamente a resposta, devido às diferentes estruturas de regressão construídas na sua distribuição. No segundo artigo, comparamos a performance preditiva dos GAMLSS com os quatro outros algoritmos de *machine learning* mencionados. Utilizamos dados provenientes de uma estação meteorológica automática na cidade de Florianópolis – SC, coletados ao longo de 10 anos (de 30 de março de 2013 a 28 de março de 2023). Os GAMLSS baseados na distribuição Box-Cox *t* apresentaram resultados mais satisfatórios na maioria das métricas utilizadas para a comparação dos modelos ajustados, provando ser uma alternativa interessante e robusta para o ajuste e predição de dados meteorológicos.

Palavras-chave: Aprendizado de Máquina; Árvores Aleatórias; Aumento Extremo de Gradiente; GAMLSS; Profeta; Regressão por Vetores de Suporte.

ABSTRACT

Univariate regression models date back to the 19th century and aim to comprehend how a set of explanatory variables influences or explains a response variable. While it is common to encounter papers comparing flexible machine learning methodologies with conventional regression models, such a comparison may not be suitable due to the stringent assumptions and limited flexibility of typical regression models. Therefore, this dissertation proposes to assess and compare the performance of distributional regression models, initially proposed as generalised additive models for location, scale, and shape (GAMLSS), which represent a more modern and flexible approach, with other commonly employed machine learning algorithms in the literature, namely: random forest, support vector regression, extreme gradient boosting, and prophet, for meteorological datasets. In our first article, already published in a journal, the need to use GAMLSS in modelling daily average temperature over a one-year period in the city of Florianópolis, Brazil, was emphasized. This study demonstrated that less complex regression models would not be suitable for fully explaining the response due to the different regression structures built into its distribution. In the second paper, we compare the predictive performance of GAMLSS with the four other mentioned machine learning algorithms. We used data from an automatic weather station in the city of Florianópolis, Brazil, collected over 10 years (from 30 March 2013 to 28 March 2023). GAMLSS based on the Box-Cox *t* distribution returned more satisfactory results in most metrics used for comparing the fitted models, proving to be an interesting and robust alternative for fitting and predicting meteorological data.

Keywords: Extreme Gradient Boosting; GAMLSS; Machine Learning; Prophet; Random Forest; Regressão; Support Vector Regression.

INDICADORES DE IMPACTO

O estudo demonstra um potencial para impactar a sociedade, especialmente nas áreas de meteorologia e climatologia. Ele destaca a necessidade de métodos mais flexíveis e modernos na modelagem de fenômenos complexos, como a temperatura média diária em uma região específica. Ao demonstrar que modelos de regressão menos complexos não são adequados para explicar completamente a resposta devido às diferentes estruturas de regressão presentes na distribuição dos dados, o estudo enfatiza a importância de abordagens mais avançadas, como os Modelos Aditivos Generalizados para Localização, Escala e Forma (GAMLSS). Portanto, o estudo tem o potencial de contribuir significativamente para o avanço da ciência meteorológica, fornecendo métodos mais eficazes e precisos para prever as condições climáticas. Isso pode beneficiar diversos setores, incluindo agricultura, aviação, energia e gestão de desastres naturais.

IMPACT INDICATORS

The study shows a potential to impact society, notably in meteorology and climatology. It highlights the need for more flexible and modern techniques in modelling complex phenomena, such as the daily average temperature in a specific location. By showing that simpler regression models are not sufficient for fully explaining the response due to the different regression structures present in the data distribution, the study emphasises the importance of more advanced approaches, such as Generalised Additive Models for Location, Scale, and Shape (GAMLSS). Therefore, the study has the potential to considerably contribute to the advancement of meteorological science by introducing more effective and accurate methods for predicting climatic conditions. This can benefit various sectors, including agriculture, aviation, energy, and natural disaster management.

SUMÁRIO

PRIMEIRA PARTE - UM PANORAMA GERAL

1 INTRODUÇÃO

Os modelos de regressão remotam de meados do século XIX, quando Francis Galton propôs o uso de regressão linear para descrever a relação entre a altura dos pais e a altura dos filhos (SOUSA; ALVES, 2016). Tais modelos buscam compreender como um conjunto de variáveis explanatórias influencia ou explica uma determinada variável resposta. Em geral, estima-se uma equação de regressão, da qual é possível realizar predições e/ou previsões sobre o comportamento da resposta de interesse. Esta metodologia, apesar de possuir pressuposições bastante rigorosas – como a necessidade de que a resposta, dada as covariáveis incluídas no modelo, tenha distribuição normal – ainda é bastante empregada atualmente, como, por exemplo, em Rath et al. (2020), Dimitriadou e Nikolakopoulos (2022) ou Brix et al. (2023).

Ainda, não é incomum encontrarmos trabalhos que comparam metodologias bastante flexíveis de aprendizado de máquinas (*machine learning*) – como, por exemplo, *random forest* (BREIMAN, 2001), *support vector machine* (VAPNIK, 1999), *extreme gradient boosting* (CHEN; GUESTRIN, 2016) – com estes modelos de regressão usuais, como pode ser visto em Markovics e Mayer (2022) ou Baturynska e Martinsen (2021). Entretanto, a comparação de técnicas tão complexas, com um modelo de regressão, com pressuposições bastante rigorosas e flexibilidade restrita, talvez não seja a mais adequada.

Duas das generalizações mais conhecidas dentro das classes de modelos de regressão que, de alguma maneira, flexibilizam as restrições impostas pelo modelo de regressão linear clássico, são os modelos lineares generalizados (GLM), propostos por Nelder e Wedderburn (1972), e os modelos aditivos generalizados (GAM), desenvolvidos por Hastie e Tibshirani (1990). Nos GLM, respostas não-Gaussianas podem ser consideradas desde que pertençam à família exponencial, ao passo que os GAM adicionam ainda a possibilidade de inclusão de relacionamentos não-lineares complexos por meio de funções de suavização (WOOD, 2017). Alguns trabalhos que comparam os GLM ou GAM com outras técnicas de aprendizado de máquina podem ser vistos em Song et al. (2021) ou Thottakkara et al. (2016).

Em determinadas situações práticas, a flexibilidade obtida pelos GLM ou GAM podem não ser suficientes. O grande entrave destes modelos é a necessidade de que a distribuição da resposta seja pertencente à família exponencial e, mais importante, que apenas uma estrutura de regressão para a média é considerada, isto é, outros momentos como variância, assimetria e curtose não são modelados diretamente. Neste sentido, os modelos aditivos generalizados para locação, escala e forma (GAMLSS) (RIGBY; STASINOPOULOS, 2005), também denominados de modelos de regressão distribucional (HELLER et al., 2022), são uma interessante alternativa, uma vez que todo e qualquer parâmetro da distribuição da variável resposta (não necessariamente pertencente à família exponencial) pode ser modelado explicitamente por meio de diferentes estruturas de regressão. Com isso, esses modelos são adequados em situações em que se deseja compreender como cada covariável influencia em cada momento da distribuição, sendo possível descrever comportamentos extremamente complexos, com alta assimetria e diferentes graus de curtose (ROQUIM et al., 2021).

Os GAMLSS vêm sendo aplicados, com sucesso, em diferentes áreas do conhecimento, como, por exemplo, ciências atuariais (NAKAMURA et al., 2017), ciências imobiliárias (DE BASTIANI et al., 2018), ciências médicas (RAMIRES et al., 2018), agricultura de precisão (RIGHETTO et al., 2019), ciências florestais (OLIVEIRA et al., 2019), segurança pública (RAMIRES et al., 2019), nas ciências naturais (HE et al., 2021), psicologia (TIMMERMAN et al., 2021), ecologia marinha (COSTA et al., 2022) e ciências animais (NAKAMURA et al., 2022).

No que tange à comparação dos GAMLSS com outros modelos de aprendizado de máquina altamente flexíveis, um estudo pioneiro foi desenvolvido por Vieira (2021). Em seu traba-

lho, o autor compara a utilização dessa classe de modelos de regressão com técnicas, como, por exemplo, *k* vizinhos mais próximos, árvores de decisão, modelos de aprendizado por *ensemble* e redes neurais. Todavia, apesar de inovador, seu estudo limita-se à comparação de modelos para respostas categóricas.

Assim, neste trabalho, buscaremos verificar e comparar a performance dos GAMLSS e outros modelos flexíveis de aprendizado de máquina, a saber: *random forest*, *support vector regression*, *extreme gradient boosting* e *prophet* em respostas contínuas, mais especificamente em dados meteorológicos. Para tal, esta dissertação está dividida em duas partes – uma contendo um panorama geral das metodologias em estudo e a segunda contendo a escrita de dois artigos científicos.

Com relação aos artigos científicos, o primeiro, já publicado em Silva et al. (2023), apresenta a necessidade da utilização dos GAMLSS na modelagem de dados meteorológicos, tendo como resposta a temperatura média diária em um período de um ano na cidade de Florianópolis – SC, onde é possível notar que a utilização de um modelo de regressão menos complexo não seria suficiente para explicar completamente a resposta em estudo dada as diferentes estruturas de regressão construídas. O segundo artigo tem como proposta utilizar a mesma variável resposta, considerando um período de 10 anos, e comparar o ajuste e poder preditivo obtido pelos GAMLSS com os outros algoritmos de *machine learning* supracitados.

2 REFERENCIAL TEÓRICO

Neste capítulo, descreveremos brevemente as metodologias de análise de séries temporais e modelos de regressão, com ênfase maior nos GAMLSS e em Machine Learning, criando um contexto inicial para a compreensão de ambos. Em seguida, apresentamos os métodos de estimação, inferências, seleção de modelos e diagnósticos de resíduos. Também discutimos as principais métricas utilizadas para comparar modelos de Machine Learning e GAMLSS.

2.1 Séries Temporais

Segundo Morettin e Toloi (2022), uma série temporal $Y_t = \{t_1, t_2, \ldots, t_n\}$, onde Y_t é o valor da série temporal no instante de tempo t e os $\{t_1, t_2, \ldots, t_n\}$ índices ou valores de tempo associados às observações na série temporal, é definida como uma coleção de observações sequenciais em um determinado tempo t, ou seja, são dados observados em diferentes instantes do tempo. Esta pode ser discreta, quando acontece em um tempo específico, como $(Y_t = t_1, \ldots, t_n)$, ou contínua, quando as observações ocorrem continuamente no tempo $(Y_t = t : t_1 < t < t_2)$.

De acordo com Sáfadi (2004), para compreensão do comportamento das séries temporais é necessário ter conhecimento dos componentes característicos, sendo eles, tendência que pode ser entendida como o comportamento que a série apresenta a longo prazo, incluindo crescimento e/ou decrescimento, com vários possíveis padrões, a sazonalidade que mostra flutuações ocorridas em períodos e a componente aleatória ou erro,que são flutuações não identificadas.

2.1.1 Modelo Autorregressivo (AR)

Os processos autoregressivos (AR) são um tipo de modelo estatístico amplamente utilizado em análise de séries temporais. Como o nome sugere, eles envolvem regressões em si mesmos, o que significa que o valor atual de uma série temporal é modelado como uma combinação linear dos valores passados da mesma série. Um processo autorregressivo de ordem p, AR(p), é dado por,

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \ldots + \phi_p Z_{t-p} + \varepsilon_t \tag{2.1}$$

em que, ϕ_i, $i = 1, 2, \ldots, p$ são parâmetros do modelo e ε_t é o ruído branco no tempo t (MORETTIN, 2017).

2.1.2 Modelo de Médias Móveis (MA)

O modelo de médias móveis MA(q) assume que a série modelada é gerada através de uma combinação linear de q sinais de ruídos $\varepsilon(t-i)$, aleatórios e independentes entre si, sua fórmula é dada por,

$$Z_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q} \tag{2.2}$$

em que há q defasagens na média móvel e $\theta_1, \theta_2, \ldots, \theta_q$ $(q \neq 0)$ são parâmetros (MORETTIN, 2017).

2.1.3 Modelo autorregressivo de médias móveis (ARMA)

A combinação de modelos autorregressivos (AR) e de médias móveis (MA), resulta na combinação de um modelo autorregressivo de médias móveis (ARMA), sendo esse representa-

ção adequada com um número menor de parâmetros, formando uma classe de modelos muito úteis e parcimoniosos. O modelo autorregressivo de médias móveis, ARMA, é descrito por,

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \ldots + \phi_p Z_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q} \tag{2.3}$$

sendo que os $\phi' s$ são parâmetros autoregressivos e os $\theta' s$, parâmetros das médias móveis, com $\psi \neq 0$, $\theta \neq 0$ e $\sigma^2 \varepsilon > 0$ (EHLERS, 2007).

2.1.4 Modelo autorregressivo integrado de médias móveis (ARIMA)

O modelo autorregressivo integrado de médias móveis (ARIMA) é uma generalização do modelo ARMA. Quando as séries não apresentam estacionariedade, a série pode se tornar estacionária ao aplicarmos d diferenças aos dados, na qual no máximos 2 diferenças podem ser aplicadas (BOX et al., 2015), da qual a equação é dada por,

$$W_t = \phi_1 W_{t-1} + \phi_2 W_{t-2} + \ldots + \phi_p W_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q} \tag{2.4}$$

em que $W_t = \nabla^d Z_t$.

2.2 Modelos de Regressão

Análise de regressão é uma técnica estatística utilizada para investigar a relação existente entre variáveis a partir da construção de uma equação (um modelo). Essa técnica, proposta inicialmente em 1895 por Sir Francis Galton (RODGERS; NICEWANDER, 1988), pode ser utilizada com vários objetivos como, por exemplo: i) descrever a relação entre variáveis para entender um processo ou fenômeno; ou ii) prever o valor de uma variável a partir do conhecimento dos valores das outras variáveis.

Matematicamente, podemos definir as relações entre as variáveis em estudo por meio de um modelo de regressão linear como

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_r X_{ir} + \varepsilon_i, \tag{2.5}$$

em que Y_i, $i = 1, \ldots, n$, representa a variável resposta dos dados observados, $X_{i1}, \ldots, X_{ir}$ são as variáveis explanatórias, $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_r)^\top$ é o vetor de parâmetros e ε_i são os erros associados ao modelo.

Ao estabelecer o modelo (2.5), pressupomos que:

a) A relação entre as variáveis explanatórias e a variável resposta é linear;
b) Os valores das variáveis explicativas são fixos, isto é, $X_{i1}, \ldots, X_{ir}$ não são considerados variáveis aleatórias;
c) Os erros são independentes e identicamente distribuídos, com uma distribuição normal de média zero e variância constante σ^2, isto é $\varepsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$.

Ainda, o modelo (2.5) pode ser reescrito em sua forma matricial como

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon,$$

em que $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\top$ é o vetor da variável resposta com dimensão n, $\mathbf{X}$ é uma matriz de delineamento com dimensão $n \times p$, sendo $p = r+1$, $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_r)^\top$ é o vetor de parâmetros e $\boldsymbol{\varepsilon} \overset{\text{iid}}{\sim} N(\mathbf{0}, \boldsymbol{I}\sigma^2)$ é o vetor de erros n-dimensional (FIGUEIRA, 2006).

2.2.1 Modelos Lineares Generalizados (GLM)

Um dos pressupostos mais rigorosos de se trabalhar com os modelos de regressão linear apresentados equação (2.5), é que a variável resposta em estudo é suposta seguir uma distribuição normal. Entretanto, em muitas situações do cotidiano, nem sempre esta suposição é válida e, assim, outros modelos que consigam flexibilizar esta característica são necessários. Neste sentido, Nelder e Wedderburn (1972) propuseram os modelos lineares generalizados (GLM), nos quais é possível ampliar o leque de distribuições de probabilidade que podem ser associadas à variável resposta, sendo essas distribuições pertencentes, necessariamente, à família exponencial, tornando menos rígida a relação entre a variável resposta e as variáveis explanatórias (PAULA, 2004).

Conforme Nelder e Wedderburn (1972), um GLM é composto por três componentes:

a) Componente aleatório: especifica a distribuição da variável resposta Y, que deve pertencer, necessariamente, à família exponencial na forma canônica, isto é, uma distribuição cuja função (densidade) de probabilidade pode ser escrita como

$$f(y_i; \mu_i, \phi) = \exp\left\{\phi^{-1}\left[y_i\mu_i - b(\mu_i)\right] + c(y_i, \phi)\right\},\ i = 1, \ldots, n,$$

em que $b(\cdot)$ e $c(\cdot)$ são funções conhecidas, ϕ é um parâmetro de dispersão e $E(Y) = \mu$. Alguns exemplos de distribuições discretas que podem ser escritas desta maneira são a Poisson e a binomial, ao passo que, para distribuições contínuas, temos, por exemplo, a gama e a distribuição normal. Isto é, o modelo (2.5) é um caso particular dos GLM;

b) Componente sistemático: contempla as variáveis explicativas, que são inseridas como uma combinação linear no modelo, vinculada à média da variável resposta por meio de uma função de ligação;

c) Função de ligação: especifica a relação entre os componentes aleatório e sistemático.

Assim, formalmente, os GLM são definidos como

$$\begin{aligned} &Y_i \overset{\text{iid}}{\sim} \mathcal{FE}(\mu_i, \phi) \\ &\eta_i = g(\mu_i) = X\beta = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_r X_{ir}, \end{aligned}$$

em que $\mathcal{FE}$ indica que a distribuição da variável resposta pertence à família exponencial, η_i é o preditor linear e $g(\cdot)$ é uma função de ligação. A Tabela 2.1 apresenta algumas das funções de ligação mais comuns.

Tabela 2.1 – Funções de ligação canônicas

Distribuição	Ligação
Normal	μ
Binomial	$\log \frac{\mu}{\mu-1}$
Poisson	$\log \mu$
Gama	μ^{-1}
Normal Inversa	μ^{-2}

Fonte: Do Autor (2023).

Segundo Turkman e Silva (2000), as funções de ligações produzem quase sempre estatísticas desejáveis para o modelo e por vezes uma facilidade na interpretação, porém nem sempre é com elas que vamos trabalhar, o pesquisador deve considerar a adaptabilidade e adequabilidade do modelo.

2.2.2 Modelos Aditivos Generalizados (GAM)

Uma limitação dos GLM é considerar que a relação da função da média é linear às variáveis explicativas. Com isso, Hastie e Tibshirani (1990) desenvolveram os modelos aditivos generalizados (GAM), uma generalização dos GLM, envolvendo a soma de funções suavizadas das covariáveis. O GAM pode ser escrito como

$$\begin{aligned} Y_i &\overset{\text{iid}}{\sim} \mathcal{FE}(\mu_i, \phi) \\ \eta = g(\mu) &= X\beta + s_1(x_1) + ... + s_J(x_J) \\ &= X\beta + \sum_{j=1}^{J} s_j(\boldsymbol{x}_j), \end{aligned}$$

em que $s_J(\cdot)$ são as funções parciais ou funções de suavização não-paramétricas aplicada à covariável x_j. Existem várias funções de suavização, dentre elas, *splines* cúbicas, polinômios fracionais e potência, ajuste não linear, P-*splines*, P-*splines* cíclicos, P-*splines* monótonos, P-*splines* encolhidos em zero, P-*splines* de coeficientes variantes (STASINOPOULOS et al., 2017).

Apesar da flexibilização alcançada com os GAM, ainda há dois potenciais problemas em sua utilização: i) apenas o parâmetro da média é modelado com base nas covariáveis; e ii) a distribuição da variável resposta ainda deve, necessariamente, pertencer à família exponencial.

2.2.3 Modelos Aditivos Generalizados para locação, escala e forma (GAMLSS)

Para superar os entraves supracitados, os Modelos Aditivos Generalizados para Locação, Escala e Forma (GAMLSS) foram propostos por Rigby e Stasinopoulos (2005). Os GAMLSS são uma abordagem para aprendizado estatístico, sendo considerados como uma classe de modelos de regressão semiparamétricos que generalizam os modelos anteriormente apresentados. Nesta abordagem, qualquer distribuição $\mathcal{D}$ pode ser assumida para representar a resposta e todo e qualquer parâmetro pode ser modelado por meio de funções lineares e de suavização não lineares, isto é, são construídas diferentes estruturas de regressão para cada um dos parâmetros. Matematicamente, um GAMLSS pode ser definido como

$$\begin{aligned} Y &\sim \mathcal{D}(\boldsymbol{\theta}_k) \\ g_k(\boldsymbol{\theta}_k) &= \boldsymbol{X}_k\boldsymbol{\beta}_k + \sum_{j=1}^{J_k} s_{kj}\left(\boldsymbol{x}_{kj}\right), \end{aligned} \tag{2.6}$$

em que $\boldsymbol{\theta}_k = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p)^\top$ é o vetor de parâmetros associados à distribuição $\mathcal{D}$ da variável resposta, $g_k(\cdot)$, $k = 1, \ldots, p$, denotam as funções de ligação relacionadas aos k-parâmetros da distribuição, $\boldsymbol{X}_k$ é uma matriz de delineamento, $\boldsymbol{\beta}_k = \left(\beta_{1k}, \ldots, \beta_{J_k k}\right)^\top$ é o vetor de parâmetros e $s_{kj}(\cdot)$ é uma função de suavização que explica o relacionamento da covariável $\boldsymbol{x}_{kj}$. Conforme Righetto et al. (2019), se $\sum_{j=1}^{J_k} s_{kj}\left(\boldsymbol{x}_{kj}\right) = 0$, então o GAMLSS é reduzido à sua versão linear paramétrica.

No caso em que $k = 1, \ldots, 4$, o modelo (2.6) se reduz ao caso

$$\begin{aligned} Y &\sim \mathcal{D}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\nu}, \boldsymbol{\tau}) \\ g_1(\boldsymbol{\mu}) &= \boldsymbol{X}_1\boldsymbol{\beta}_1 + \sum_{j=1}^{J_1} s_{1j}(\boldsymbol{x}_{1j}) \\ g_2(\boldsymbol{\sigma}) &= \boldsymbol{X}_2\boldsymbol{\beta}_2 + \sum_{j=1}^{J_2} s_{2j}(\boldsymbol{x}_{2j}) \\ g_3(\boldsymbol{\nu}) &= \boldsymbol{X}_3\boldsymbol{\beta}_3 + \sum_{j=1}^{J_3} s_{3j}(\boldsymbol{x}_{3j}) \\ g_4(\boldsymbol{\tau}) &= \boldsymbol{X}_4\boldsymbol{\beta}_4 + \sum_{j=1}^{J_4} s_{4j}(\boldsymbol{x}_{4j}), \end{aligned} \tag{2.7}$$

em que, usualmente, $\boldsymbol{\mu}$ e $\boldsymbol{\sigma}$ são parâmetros de locação e escala, respectivamente, e $\boldsymbol{\nu}$ e $\boldsymbol{\tau}$ são parâmetros de forma.

Segundo Rigby e Stasinopoulos (2005), as funções de suavização dos GAMLSS podem, na maioria das situação, serem reescritas como $s(\boldsymbol{x}) = \boldsymbol{Z}\boldsymbol{\gamma}$, em que $\boldsymbol{Z}$ é a matriz de base dependendo dos valores de $\boldsymbol{x}$ e γ é um conjunto de parâmetros sujeitos à penalização $\boldsymbol{\lambda}\boldsymbol{\gamma}^\top \boldsymbol{G}\boldsymbol{\gamma}$, para uma matriz conhecida $\boldsymbol{G} = \boldsymbol{D}^\top\boldsymbol{D}$ em que $\boldsymbol{D}$ é uma matriz de diferenças e $\boldsymbol{\lambda}$ é um vetor ou escalar de hiperparâmetros que regula o grau de suavização necessário no ajuste.

Dessa forma, podemos reescrever o modelo (2.8) como

$$\begin{aligned} Y &\sim \mathcal{D}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\nu}, \boldsymbol{\tau}) \\ g_1(\boldsymbol{\mu}) &= \boldsymbol{X}_1\boldsymbol{\beta}_1 + \sum_{j=1}^{J_1} \boldsymbol{Z}_{1j}(\boldsymbol{\gamma}_{1j}) \\ g_2(\boldsymbol{\sigma}) &= \boldsymbol{X}_2\boldsymbol{\beta}_2 + \sum_{j=1}^{J_2} \boldsymbol{Z}_{2j}(\boldsymbol{\gamma}_{2j}) \\ g_3(\boldsymbol{\nu}) &= \boldsymbol{X}_3\boldsymbol{\beta}_3 + \sum_{j=1}^{J_3} \boldsymbol{Z}_{3j}(\boldsymbol{\gamma}_{3j}) \\ g_4(\boldsymbol{\tau}) &= \boldsymbol{X}_4\boldsymbol{\beta}_4 + \sum_{j=1}^{J_4} \boldsymbol{Z}_{4j}(\boldsymbol{\gamma}_{4j}), \end{aligned}$$

em que $\boldsymbol{\gamma}_{kj} \overset{\text{iid}}{\sim} N\left(0, \boldsymbol{G}_{kj}^{-}(\boldsymbol{\lambda}_{kj})\right)$ são os parâmetros de efeito aleatório e $\boldsymbol{G}_{kj}^{-}(\boldsymbol{\lambda}_{kj})$ a inversa (generalizada) da matriz simétrica $\boldsymbol{G}_{kj}(\boldsymbol{\lambda}_{kj})$ de ordem $q_{kj} \times q_{kj}$.

2.2.3.1 Distribuições da família GAMLSS

Em suma, como mencionado na Seção 2.2.3, qualquer distribuição $\mathcal{D}$ pode ser utilizada para representar a variável resposta, independentemente dela pertencer, ou não, à família exponencial. Podemos dividí-las em três grandes grupos: distribuições discretas, contínuas e mistas; dentro dos quais há distribuições com características que comportam dados fortemente assimétricos, bem como com caudas pesadas (leptocurtose) e leves (platicurtose). Há um pacote específico da família GAMLSS de distribuições denominado `gamlss.family` (STASINOPOU-

LOS et al., 2015), no software R (R Core Team, 2023), onde mais de 100 distribuições já estão implementadas. Algumas dessas distribuições são apresentadas nos Quadros 2.2 e 2.1.

Quadro 2.1 – Algumas distribuições discretas implementadas nos pacotes gamlss (com funções de ligação padrão).

Distribuição	Nomenclatura	μ	σ	ν
beta binomial	BB()	logit	log	-
binomial	BI()	logit	-	-
Delaporte	DEL()	log	log	logit
Negative Binomial type I	NBI()	log	log	-
Negative Binomial type II	NBII()	log	log	-
Poisson	PO()	log	-	-
Poisson inverse Gaussian	PIG()	log	log	-
Sichel	SI()	log	log	identity
Sichel (μ the mean)	SICHEL()	log	log	identity
zero inflated poisson	ZIP()	log	logit	-
zero inflated poisson (μ the mean)	ZIP2()	log	logit	-

Fonte: STASINOPOULOS:RIGBY(2008).

Quadro 2.2 Algumas distribuições contínuas implementadas nos pacotes gamlss (com funções de ligação padrão)

Distribuição	Nomenclatura	μ	σ	ν	τ
beta	BE()	logit	logit	-	-
Box-Cox Cole and Green	BCCG()	identity	log	identity	-
Box-Cox power exponential	BCPE()	identity	log	identity	log
Box-Cox-t	BCT()	identity	log	identity	log
exponential	EXP()	log	-	-	-
exponential Gaussian	exGAUS()	identity	log	log	-
exponential gen. beta type 2	EGB2()	identity	identity	log	log
gamma	GA()	log	log	-	-
generalized beta type 1	GB1()	logit	logit	log	log
generalized beta type 2	GB2()	log	identity	log	log
generalized gamma	GG()	log	log	identity	-
generalized inverse Gaussian	GIG()	log	log	identity	-
generalized y	GT()	identity	log	log	log
Gumbel	GU()	identity	log	-	-
inverse Gaussian	IG()	log	log	-	-
Johnson's SU (μ the mean)	JSU()	identity	log	identity	log
Johnson's original SU	JSUo()	identity	log	identity	log
logistic	LO()	identity	log	-	-
log normal	LOGNO()	log	log	-	-
log normal (Box-Cox)	LNO()	log	log	fixed	-
NET	NET()	identity	log	fixed	fixed
normal	NO()	identity	log	-	-
normal family	NOF()	identity	log	identity	-
power exponential	PE()	identity	log	log	-
reverse Gumbel	RG()	identity	log	-	-
skew power exponential type 1	SEP1()	identity	log	identity	log
skew power exponential type 2	SEP2()	identity	log	identity	log
skew power exponential type 3	SEP3()	identity	log	log	log
skew power exponential type 4	SEP4()	identity	log	log	log
shash	SHASH()	identity	log	log	log
skew t type 1	ST1()	identity	log	identity	log
skew t type 2	ST2()	identity	log	identity	log
skew t type 3	ST3()	identity	log	log	log
skew t type 4	ST4()	identity	log	log	log
skew t type 5	ST5()	identity	log	identity	log
t Family	TF()	identity	log	log	-
Weibull	WEI()	log	log	-	-
Weibull (PH)	WEI2()	log	log	-	-
Weibull (μ the mean)	WEI3()	log	log	-	-
zero adjusted IG	ZAIG()	log	log	logit	-

Fonte: STASINOPOULOS:RIGBY(2008).

Outrossim, há ainda a possibilidade de se implementar novas distribuições caso necessário (ROQUIM et al., 2021), além de ser possível construir novas distribuições a partir de

transformações daquelas já implementadas no pacote `gamlss.family` (STASINOPOULOS et al., 2015). As funções disponíveis são apresentadas a seguir:

a) `gen.Family()`: transformação logarítmica ou do tipo logit em qualquer distribuição que assuma que a variável resposta esteja no intervalo $-\infty < Y < \infty$;
b) `gen.trun()`: função pertencente ao pacote `gamlss.tr` que realiza o truncamento de qualquer distribuição;
c) `gen.cens()`: função pertencente ao pacote `gamlss.cens` que constrói versões censuradas de distribuições no intervalo $(0, \infty)$;
d) `gamlssZadj()`: função pertencente ao pacote `gamlss.inf` que constrói versões ajustadas em zero de distribuições no intervalo $(0, \infty)$;
e) `gen.Inf0to1()`: função pertencente ao pacote `gamlss.inf` que constrói versões inflacionadas em zero e/ou um de distribuições no intervalo $(0, 1)$.

Mais detalhes sobre as distribuições podem ser encontrados em (RIGBY et al., 2019).

2.2.3.2 Método de estimação dos parâmetros

Segundo Rigby et al. (2019), o processo de estimação dos parâmetros dos GAMLSS em sua versão linear paramétrica consiste em utilizar o método de máxima verossimilhança, isto é, maximizamos a função de máxima verossimilhança, dada por

$$\ell = \sum_{i=1}^{n} \log f(y_i \mid \mu_i, \sigma_i, \nu_i, \tau_i) \tag{2.8}$$

em que $f(\cdot)$ representa a função (densidade) de probabilidade da variável de resposta.

Para modelos que apresentam termos não paramétricos é necessário recorrer ao método da máxima verossimilhança penalizada, maximizando a função

$$\ell_p = \ell - \frac{1}{2} \sum_{k=1}^{p} \sum_{j=1}^{J_k} \lambda_{kj} \gamma_{kj}^{'} \boldsymbol{G}_{kj} \gamma_{kj} \tag{2.9}$$

em que $\ell = \sum_{i=1}^{n} \log f(y_i \mid \boldsymbol{\theta^i})$ é o logaritmo da função de verossimilhança dos dados condicionais a $\boldsymbol{\theta^i}$.

Rigby e Stasinopoulos (2005) sugerem dois algoritmos para estimar um GAMLSS para valores fixos de hiperparâmetros, a fim de maximizar a função de verossimilhança penalizada, são eles os algoritmos CG e RS.

O CG sendo uma generalização do algoritmo de Cole e Green (1992), requer informações sobre a primeira e segunda derivadas cruzadas da função de log-verossimilhança em relação aos parâmetros de distribuição $\boldsymbol{\theta} = (\mu, \sigma, \nu, \tau)$ para uma distribuição de quatro parâmetros. Por outro lado, o RS é uma generalização do algoritmo usado por Rigby e Stasinopoulos (1996) para ajustar um modelo aditivo de média e dispersão (MADAM). Este algoritmo não usa as derivadas cruzadas do logaritmo da função de verossimilhança.

2.2.3.3 Seleção das variáveis

Para o processo de seleção de variáveis a serem introduzidas em cada uma das estruturas de regressão dos parâmetros (μ, σ, ν, τ) existem diversas maneiras disponíveis na literatura (STASINOPOULOS et al., 2017). Conforme Ramires et al. (2021), a Estratégia A, implementada no pacote `gamlss`, na função `stepGAICAll.A()`, é a mais utilizada. Ela consiste em uma

abordagem de seleção de variáveis, por definição, que utiliza o critério de informação de Akaike (AKAIKE, 1974), mas é possível alterar e colocar outro critério.

Para uma distribuição de quatro parâmetros, conforme Stasinopoulos et al. (2017) e Nakamura et al. (2017), a Estratégia A funciona da seguinte maneira:

a) Ajusta-se um modelo para μ por meio do procedimento *forward* baseado no AIC, considerando σ, ν e τ como constantes;
b) Usando o mesmo procedimento, o modelo para o parâmetro σ é ajustado, considerando o modelo ajustado para o parâmetro μ no passo anterior e considerando os parâmetros restantes como constantes;
c) O modelo para o parâmetro ν é ajustado utilizando um procedimento *forward*, considerando os modelos de μ e σ obtidos nos passos (1) e (2) e τ como fixo;
d) O modelo para o parâmetro τ é considerado por meio de um procedimento *forward* considerado os três modelos ajustados nos passos (1), (2) e (3);
e) O algoritmo começa a retrogredir, reajustando o parâmetro ν, procedimento *backward*, dado todas as demais estruturas de regressão já ajustados;
f) Um procedimento *backward* é realizado para o modelo do parâmetro σ;
g) Finalmente, um último procedimento *backward* é realizado para o parâmetro μ.

Após os sete passos, o modelo final irá conter uma sub-seleção das covariáveis para cada parâmetro da distribuição que não é necessariamente igual.

2.3 Aprendizado de máquina

O termo aprendizado de máquina refere-se a um conjunto de técnicas que tratam da criação e avaliação de algoritmos que facilitam o reconhecimento de padrões, classificação e previsão, com base em modelos derivados de dados existentes (ADI et al, 2007). De acordo com Mitchell (1997) é importante ter uma boa compreensão do que é aprendizado de máquina. Para o autor, o aprendizado de máquina é um ramo da inteligência artificial que emprega uma variedade de ferramentas estatísticas, probabilísticas e de otimização para "aprender" com exemplos passados e depois utilizar os modelos treinados para a realização de classificação, predição, identificação de novos padrões ou prever tendências com novos dados.

O aprendizado de máquina é fundamentado em técnicas estatísticas e de probabilidade e, muitas vezes, é mais poderoso porque permite fazer inferências ou listar decisões a serem tomadas que não poderiam ser feitas com técnicas estatísticas tradicionais (MITCHELL, 1997). Por exemplo, muitos métodos estatísticos são baseados em regressão múltipla ou análise de correlação. Embora geralmente muito poderosos, essas abordagens fazem suposições de que as variáveis são independentes e que os dados podem ser modelados usando combinações lineares dessas variáveis. Na prática, podemos verificar diversas situações em que a modelagem por meio destes métodos não é a mais indicada. Como exemplo, podemos citar os sistemas biológicos que são fundamentalmente não lineares e seus parâmetros condicionalmente dependentes. Quando os relacionamentos não são lineares e as variáveis são interdependentes, os métodos estatísticos convencionais geralmente deixam de obter os melhores resultados. Este é um dos cenários em que os métodos de aprendizado de máquina tendem a se destacar.

Os conjuntos de treinamento e teste desempenham papéis cruciais no desenvolvimento de modelos de machine learning. O conjunto de treinamento é utilizado para ensinar o modelo, permitindo que ele aprenda padrões e relações nos dados. Em contrapartida, o conjunto de teste é reservado para avaliar a capacidade do modelo de generalizar para dados não vistos durante o treinamento. Manter esses conjuntos separados é essencial para evitar que o modelo se ajuste excessivamente aos dados de treinamento, garantindo assim uma avaliação mais precisa de sua

eficácia em situações do mundo real (UÇAR et al., 2020). Encontrar o equilíbrio adequado entre os conjuntos de treinamento e teste é fundamental para o desenvolvimento de modelos robustos e confiáveis.

Dois conceitos importantes na área de aprendizado de máquina são o *overfitting* e *underfitting*, particularmente em algoritmos de aprendizado supervisionado, como regressão e classificação. Eles estão relacionados ao desempenho do modelo em dados não vistos, ou seja, dados que não foram utilizados durante o treinamento.

O *overfitting* ocorre quando um modelo se ajusta excessivamente aos dados de treinamento, reunindo todas as peculiaridades específicas desse conjunto de dados. Logo, o modelo ajustado se torna altamente especializado nos dados de treinamento, mas não generaliza, ou seja, leva a um desempenho ruim em dados de teste ou em situações do mundo real. No caso do *underfitting* é o oposto, o modelo de treinamnto não consegue capturar os padrões e relações importantes nos dados, ele é muito simples ou tem capacidade insuficiente para aprender a complexidade dos dados. Esse tipo de situação se caracteriza por ter um desempenho abaixo do esperado nos dados de treinamento e uma falta de capacidade de generalização (LÓPEZ et al., 2022).

Ambos *overfitting* e *underfitting* são problemas indesejáveis, pois comprometem a capacidade do modelo de generalizar e fazer previsões precisas em novos dados. O objetivo é encontrar um equilíbrio entre complexidade e generalização, evitando um super aprofudamento ou a falta de capacidade do modelo. No caso do *overfitting*, uma possilidade de lidar com ele é usar técnicas de regularização, que imcluem uma penalização na função de custo do modelo para evitar parâmetros excessivamente grandes. Já no *underfitting*, é possível melhorar o desempenho do modelo através da coleta de mais dados, redução da dimensionalidade dos dados de entrada ou ajuste de hiperparâmetros (JABBAR; KHAN, 2015).

O aprendizado de máquina pode ser classificado em três tipos: 1) aprendizado supervisionado, 2) aprendizado não supervisionado e 3) aprendizado por reforço. Neste trabalho será abordado apenas o aprendizado supervisionado. Este tipo de aprendizado se baseia na utilização de um conjunto de dados rotulados para o treinamento de modelos preditivos. Neste procedimento, os métodos aprendem a partir do conjunto de dados de treinamento composto por variáveis de entradas e de saída, em seguida, os modelos treinados podem ser utilizados para a predição de saídas em um novo conjunto de variáveis de entradas. Os métodos de aprendizado de máquina podem também ser classificados de acordo com a variável de saída. Para saídas contínuas e categóricas são ajustados, respectivamente, modelos de regressão e de classificação.

2.3.1 Pré processamento de dados

Em *Machine Learning* uma parte muito importante, é a fase do pré processamento dos dados, onde logo após a coleta dos dados, eles são organizados. Algumas tecnicas utilizadas são:

a) *Feature selection*, utilizada para selecionar os atributos que se mostram mais relevantes para o modelo, essa técnica se mostra muito eficaz especialmente em dados de alta dimensão (LI et al., 2017);
b) *Feature engineering*, captura importantes comportamentos das variáveis de um conjunto de dados, visando melhorar a performance de um modelo (TURNER et al., 1999);
c) Normalização, no conjunto de dados pode conter variáveis em diferentes escalas e, assim sendo, recomenda-se padronizar esses dados para uma mesma escala. Para isso, usamos a técnica de *Z-score* (JAIN et al., 2005);

d) Redução de dimensionalidade, processo para diminuir o número de variáveis. Isso ajuda a simplificar os modelos, melhorar o desempenho e a visualização, eliminar características desnecessárias e tratar problemas como multicolinearidade (ALMEIDA; YAMAKAMI, 2011);
e) Divisão dos dados em treino e teste, em aprendizagem supervisionada, deve-se dividir os dados em dois *datasets* treinameno e teste. Os dados de treinamento são usados para criar o modelo e os dados de teste são usados para verificar a performance do modelo. Além disso, essa divisão deve ser feita de forma aleatória (SANTOS et al., 2019).

2.3.2 Otimização de hiperparâmetros

Cada algoritmo possui um conjunto de hiperparâmetros que podem ser alterados, a otimização dos hiperparâmetros em métodos de aprendizado de máquina é um processo importante para melhorar o desempenho e a precisão dos modelos. Os parâmetros do modelo são ajustados durante o processo de treinamento com base nos dados fornecidos, e seus valores são determinados automaticamente pelo algoritmo. Por outro lado, os hiperparâmetros do modelo não são aprendidos a partir dos dados e requerem uma configuração manual. Os hiperparâmetros têm o poder de influenciar significativamente os resultados do modelo, tornando crucial a escolha adequada desses valores (COSTA, 2022). Existem várias abordagens utilizadas para otimizar os hiperparâmetros, uma bastante utilizada é a Grid Search.

O *Grid Search* (ZHANG et al., 2018) é um algoritmo utilizado para encontrar a combinação ideal de hiperparâmetros de um modelo de aprendizado de máquina. Ele é chamado de "busca em grade"porque realiza uma busca exaustiva em uma grade pré-definida de valores para cada hiperparâmetro (SUN et al., 2021).

O processo de *Grid Search* envolve os seguintes passos, para cada hiperparâmetro que se deseja otimizar, é necessário especificar uma lista de possíveis valores. Esses valores podem ser escolhidos com base em conhecimento prévio, experiência ou tentativa e erro, em seguida, o algoritmo gera todas as possíveis combinações de valores dos hiperparâmetros dentro da grade definida. Para cada combinação de hiperparâmetros, o modelo de aprendizado de máquina é treinado e avaliado usando um conjunto de validação ou uma técnica de validação cruzada, após avaliar todas as combinações de hiperparâmetros, o *Grid Search* seleciona a combinação que obteve o melhor desempenho de acordo com a métrica escolhida. Após a seleção dos melhores hiperparâmetros, o modelo é treinado novamente utilizando todos os dados disponíveis (conjunto de treinamento + conjunto de validação) com essa configuração otimizada. O desempenho final do modelo é então avaliado usando um conjunto de teste separado e a métrica escolhida (PRIYADARSHINI; COTTON, 2021); (ZHANG et al., 2018).

2.3.3 Aprendizagem

Nesta fase, o modelo é construído com base nos dados fornecidos ao algoritmo. Diversas técnicas são empregadas, entre as quais se destaca a validação cruzada (*cross-validation*), um procedimento essencial para treinar e avaliar o desempenho do modelo. Segundo HAYKIN (), a validação cruzada envolve a divisão do conjunto de dados em várias partições, permitindo que o modelo seja treinado em uma parte dos dados e validado em outra, garantindo que o desempenho seja avaliado de maneira robusta e imparcial.

No contexto de séries temporais, o uso das técnicas tradicionais de validação cruzada, como o *K-Fold Cross-Validation*, não é adequado devido à presença de autocorrelação nos dados. Nesse cenário, a abordagem mais apropriada é a chamada *Time Series Cross-Validation* (Validação Cruzada para Séries Temporais), conforme mencionado no artigo de Deng (2023).

Nessa técnica, os dados são divididos em blocos de tempo, respeitando a ordem cronológica dos eventos. A validação é realizada de maneira sequencial, garantindo que as previsões sejam feitas para períodos futuros com base em informações históricas, simulando assim as condições do mundo real em que as previsões precisam ser feitas com base no passado.

2.4 Modelos de aprendizado

A escolha dos modelos de *machine learning* desempenha um papel fundamental na eficácia e precisão das previsões, especialmente quando se trata de fenômenos complexos como a variação da temperatura. Neste contexto, a seleção criteriosa de algoritmos é crucial para capturar a natureza dinâmica e muitas vezes não linear dos dados meteorológicos. Neste trabalho, optamos por utilizar uma abordagem diversificada, empregando Support Vector Regression (SVR), Random Forest, XGBoost e o modelo de previsão de séries temporais Prophet. Essa escolha não foi arbitrária, mas sim fundamentada em características específicas do problema em questão, visando melhor capturar a variabilidade e complexidade inerentes aos dados de temperatura, trabalhos relacionados a elementos climatológicos utilizando esses modelos podem ser observados em (BASAK et al., 2022); (PANG et al., 2017); (HE et al., 2022); (MA et al., 2020); (AGHELPOUR et al., 2019).

2.4.1 Random Forest (RF)

A *Random Florest* desenvolvida por (BREIMAN, 2001) é uma técnica não paramétrica e não linear, que não necessita de suposições quanto ao relacionamento entre a variável de saída e as variáveis de entrada. Para sua construção, inicialmente são selecionadas aleatoriamente B amostras bootstrap do conjunto de dados, em seguida, para cada uma destas amostras, cresce uma árvore de decisão T_b $(b = 1,\ldots,B)$ obtida pela divisão dos indivíduos recursivamente, e pelo sorteio aleatório de p variáveis de entrada, de modo que, a variável que melhor divide os dados é utilizada em cada nó (BREIMAN, 2001). Os passos do *Random Florest* são descritos em (HASTIE et al., 2009) como:

a) Para $b = 1,\ldots,B$;
- Faça uma amostra *bootstrap* Z^* de tamanho;
- Cresça uma árvore de floresta aleatória T_b para os dados de *bootstrap*, repetindo recursivamente as seguintes etapas para cada nó da árvore, até que o tamanho mínimo do nó n_min seja atingido;

a) Selecione m variáveis aleatoriamente das p variáveis;
b) Escolha a melhor variável/ponto de divisão entre os m;
c) Divida o nó em dois nós filhos.

Saída do conjunto de árvores $\{T_b\}_1^B$

Para a realização de predições é utilizada a expressão:

$$\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x). \tag{2.10}$$

em que, $\hat{f}_{rf}^B(x)$ é a predição do algoritmo *Random Forest* para um novo dado x, após ter sido treinado em B árvores de decisão, B é o número de árvores, $T_b(x)$ predição da b-ésima árvore de decisão para o dado x.

Este procedimento leva a um modelo de melhor desempenho, reduzindo a variância, quando comparado com a utilização de apenas uma árvore de decisão. A partir do momento em

que as árvores não sejam correlacionadas, a média de todas elas irá obter boas previsões para o conjunto de treinamento (SILVA; NETO, 2023).

Em aprendizado de máquina, os hiperparâmetros são parâmetros que controlam o comportamento e a performance de um algoritmo de aprendizado. Eles são chamados de "hiperparâmetros"para distingui-los dos parâmetros do modelo, e são aprendidos diretamente a partir dos dados durante o treinamento (SANTOS et al., 2019).

O algoritmo (RF) possui vários hiperparâmetros que devem ser definidos pelo usuários. Segundo Probst et al. (2019), Callens et al. (2020), Dasari et al. (2019) os hiperparâmetros mais comuns da *Random Florest* são,

a) Número de árvores ($n\_estimators$): Determina o número de árvores de decisão que serão combinadas para formar a floresta. Um valor maior geralmente leva a um modelo mais robusto, mas aumenta o tempo de treinamento;
b) Profundidade máxima das árvores (max_depth): Controla a profundidade máxima das árvores de decisão individuais. Uma árvore mais profunda pode aprender relações mais complexas nos dados, mas também pode levar a um modelo superajustado (*overfitting*);
c) Número mínimo de amostras em uma folha ($min\_samples\_leaf$): Especifica o número mínimo de amostras necessárias em um nó folha. Um valor maior promove uma regularização mais forte, evitando árvores com poucas amostras, o que pode levar a overfitting;
d) Número mínimo de amostras para dividir um nó ($min\_samples\_split$): Define o número mínimo de amostras necessárias para dividir um nó interno. Assim como o parâmetro anterior, isso também ajuda a controlar a regularização e evitar overfitting;
e) Número máximo de recursos considerados em cada divisão ($max\_features$): Determina o número máximo de recursos que são considerados em cada divisão de um nó. Isso permite controlar a aleatoriedade da Random Forest. Um valor menor pode reduzir a correlação entre as árvores individuais, tornando o modelo mais diversificado.

2.4.2 Support vector regression (SVR)

Inicialmente proposta por Vapnik (1999) como um método de aprendizado de máquina para variáveis respostas dicotômicas, a *support vector machine* (SVM), objetiva obter o hiperplano com a maior margem, que proporcione uma separação ótima entre as classes. Algumas das principais características das SVM que tornam seu uso atrativo são, boa capacidade de generalização, robustez em grandes dimensões, convexidade da função objetivo e teoria bem definida (LORENA; CARVALHO, 2003). Posteriormente, a SVM foi estendida para a modelagem de problemas de regressão e denotada como *support vector regression* (SVR) (XU et al., 2010). Neste caso, considerando o conjunto de dados $\{(x_1, y_1), \ldots, (x_t, y_t) \subset \mathbf{X} \times \mathbb{R}\}$, com $\mathbf{X}$ representado o espaço das variáveis de entrada e y_i um valor real, o intuito é encontrar o preditor que mais se aproxime dos dados, dado pela função $f(x) = \langle w, x \rangle + b$, com vetor de pesos $\mathbf{w} \in X$ e viés $b \in \mathbb{R}$.

A *support vector regression* é formulada pelo seguinte problema de otimização:

$$\text{minimizar:} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*)$$

$$\text{sujeito a:} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq \end{cases}$$

em que, w é um vetor de pesos, representando os coeficientes da função de regressão linear, C uma constante positiva de regularização, determina a quantidade de erros de treinamento que serão permitidos, ξ_i e ξ_i^* são variáveis de folga, m o número de pontos de treinamento no conjunto de dados, ε a margem de tolerância, $\langle w, x_i \rangle$ o produto interno entre o vetor de pesos w e o vetor de características x_i, representando a predição da função de regressão para o ponto x_i e y_i o valor alvo real associado ao ponto de treinamento x_i.

O algoritmo utiliza uma função de perda, que é projetada para minimizar a diferença entre as predições do modelo e os valores reais dos dados de treinamento, enquanto ao mesmo tempo controla a magnitude do erro permitido (LIN et al., 2007). Ignorando erros que estão além de uma certa distância dos valores considerados válidos, ou seja, erros são permitidos somente se forem menores do que ε.

2.4.3 Extreme gradient Boosting (XGBoost)

O *Extreme Gradient Boosting* (XGBoost), proposto por Chen e Guestrin (2016), é uma adaptação do algoritmo *Gradiente Boosting* (GBM) introduzido por Friedman et al. (2000). O GBM combina sequencialmente vários modelos base (*weak learners*), com o intuito de melhorar o desempenho preditivo do modelo final, pela correção das deficiências do modelo base anterior. Após cada iteração, mais peso é atribuído às amostras de treinamento que foram classificadas incorretamente nas rodadas anteriores. No final do processo, todos os modelos sucessivos são ponderados de acordo com seu desempenho e as saídas são combinadas usando votação para problemas de classificação ou média para problemas de regressão, criando o modelo final (NOBRE; NEVES, 2019). Este método é fundamentado na função de perda, no modelo de predição (árvore de decisão) e em um modelo aditivo que combina os modelos base visando minimizar a função de perda (FRIEDMAN et al., 2000). A principal diferença entre XGBoost e o GBM é que o XGBoost utiliza uma nova forma de regularização para controlar o *overfitting*.

Matematicamente, é possível escrever esse modelo de árvore da seguinte forma:

$$y = \sum_{\ell=1}^{L} f_\ell(x_i), f_\ell \in F$$

em que L é o número de árvores, f é uma função no espaço F e o espaço F são todos os valores das árvores de decisão (MARINHO et al., 2021).

O modelo XGboost possui diversos hiperparâmetros (MARINHO et al., 2021), (QIN et al., 2021), (BUDHOLIYA et al., 2022) que pode ajudar na otimização do modelo, tais como:

a) Número de árvores `n_estimators`: Determina o número de árvores de decisão que serão construídas no modelo. Um valor maior geralmente melhora o desempenho, mas aumenta o tempo de treinamento e pode levar ao *overfitting*;
b) Profundidade máxima da árvore `max_depth`: Limita a profundidade máxima das árvores de decisão. Um valor maior permite que as árvores capturem interações mais complexas, mas também pode levar ao*overfitting*;
c) Taxa de aprendizado `learning_rate`: Controla a taxa na qual o modelo aprende durante o treinamento. Um valor menor requer mais árvores para alcançar um desempenho semelhante, mas pode levar a um modelo mais robusto;
d) Regularização `reg_lambda, reg_alpha`: Os parâmetros `reg_lambda` e `reg_alpha` controlam a regularização L2 e L1, respectivamente. Eles ajudam a evitar o *overfitting*, adicionando penalidades à função objetivo com base nos pesos dos modelos;

e) Taxa de amostragem por árvore `learning_rate`: Define a taxa na qual os pesos das árvores são atualizados durante o treinamento. Um valor maior permite que as árvores se ajustem mais rapidamente aos erros, mas também pode levar a um ajuste excessivo.

2.4.4 Prophet

O *Prophet* (TAYLOR; LETHAM, 2018) é uma biblioteca de previsão de séries temporais desenvolvida pelo *Facebook's Core Data Science team*. Ele é baseado em um modelo ARIMA modificado que é capaz de lidar com dados não estacionários e sazonais (CAZEIRO; OLIVEIRA, 2023).Trata-se de um modelo de regressão aditiva, o que implica que o modelo é formado pela soma de diversos componentes que são opcionais. Esses componentes incluem uma curva de tendência de crescimento linear ou logístico, uma curva de sazonalidade anual, uma curva de sazonalidade semanal, uma curva de sazonalidade diária, considerações para feriados e eventos especiais, bem como curvas de sazonalidade adicionais especificadas pelo usuário, tais como aquelas referentes a horários específicos ou a trimestres. Sua formulação é dada por,

$$Y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

em que $g(t)$ representa a tendência, $s(t)$ representa a sazonalidade, $h(t)$ representa o efeito do feriado sobre o comportamento dos dados. O valor de ε_t é um erro ou alteração nos dados que não estão contidos no modelo (BASHIR et al., 2022).

O Prophet decompõe uma série temporal em três componentes:

a) Tendência: A tendência é a direção geral da série temporal ao longo do tempo. Ela é modelada por uma função linear ou quadrática;
b) Sazonalidade: A sazonalidade é a variação periódica da série temporal. Ela é modelada por um conjunto de funções sinusoidal;
c) Resíduos: Os resíduos são a parte da série temporal que não pode ser explicada pela tendência ou pela sazonalidade.

Segundo (SILVA et al., 2022) Prophet é capaz de lidar com séries temporais com múltiplos períodos sazonais. Ele também é capaz de lidar com séries temporais com observações não regularmente espaçadas. Além disso, ele é rápido de treinar, o que o torna adequado para o uso com grandes bases de dados.

2.5 Procedimentos para a avaliação do desempenho dos modelos

Os resíduos ordinários, ou brutos, são definidos pela diferença entre os valores observados e os estimados, isto é,

$$\hat{\boldsymbol{\varepsilon}} = \mathbf{y} - \hat{\mathbf{y}}.$$

Uma restrição em trabalharmos com esse tipo de resíduo é a dificuldade em os generalizarmos para outras distribuições além da normal, isto é, para além do caso dos modelos lineares clássicos. Assim, diferentes transformações vêm sendo propostas para sugerir resíduos que possuam boas propriedades para outros tipos de modelo. Neste sentido, os resíduos quantílicos (aleatorizados) normalizados, propostos por Dunn e Smyth (1996), parecem uma excelente alternativa para a classe dos GAMLSS, uma vez que, independente da distribuição assumida para a variável resposta, os verdadeiros resíduos sempre têm uma distribuição normal padrão, quando o modelo assumido é correto. Estes resíduos são definidos por

$$\hat{r}_i = \Phi^{-1}\left(\hat{u}_i\right),$$

em que $\Phi^{-1}(\cdot)$ é o inverso da função distribuição acumulada da normal padrão e os $(\hat{u_i})$ são denoominados resíduos quantílicos, definidos diferentemente para variáveis contínuas e discretas.

Para verificar as pressuposições do modelo baseados nos resíduos supracitados, usualmente utilizamos métodos gráficos, como, por exemplo, gráficos de resíduos versus os valores ajustados ou o *Worm plot* (RIGBY; STASINOPOULOS, 2005).

O gráfico *worm plot* (BUUREN; FREDRIKS, 2001) é uma ferramenta eficaz para detectar áreas em que um modelo não está bem ajustado aos dados. No *worm plot*, cada observação é representada no eixo vertical pela diferença entre a sua posição na distribuição teórica e na distribuição empírica. Quando as observações são disponibilizadas juntas, elas formam uma curva semelhante a uma minhoca, cuja forma indica o grau de afastamento dos dados da distribuição assumida, no caso a normal. Considerando a forma do gráfico como um todo, é possível identificar possíveis problemas nos resíduos e, consequentemente, no modelo ajustado conforme mostra o Quadro 2.3.

Quadro 2.3 – Diferentes formatos do *worm plot* e interpretações

Formato	Resíduos	Parâmetro ajustado
nível: acima da origem	média muito alta	locação subestimada
nível: abaixo da origem	média muito baixa	locação superestimada
reta: inclinação positiva	variância muito alta	escala subestimada
reta: inclinação negativa	variância muito baixa	escala superestimada
U	assimetria positiva	assimetria subestimada
U invertido	assimetria negativa	assimetria superestimada
S com curva esquerda pra baixo	leptocurtose	cauda muito leve
S com curva esquerda pra cima	platicurtose	cauda muito pesada

Fonte: Do Autor (2023)

A avaliação de modelos preditivos pode ser realizada com a aplicação do método de validação simples (hold-out), que propõe a divisão aleatória do conjunto de dados em duas partes disjuntas. A primeira parte, denotada como conjunto de treinamento, utilizada para ajuste dos modelos, e a segunda parte, chamada de conjunto de teste, utilizada para a avaliação da qualidade de ajuste dos modelos. O objetivo central ao utilizar esta divisão é o de avaliar a capacidade de generalização dos modelos. No conjunto de teste são calculadas as chamadas métricas de erro.

A literatura apresenta diversas métricas de erro. Estas medidas possibilitam quantificar e comparar o desempenho preditivo dos modelos de regressão. A seguir são apresentadas as métricas: Erro médio absoluto (MAE), erro médio absoluto (MAE), raiz do erro quadrático médio (RMSE), erro percentual absoluto médio (MAPE), erro Médio Absoluto Escalado (MASE) e erro Médio Absoluto Percentual Simétrico (SMAPE).

Segundo Ruezzene et al. (2021) o erro médio absoluto (MAE) é definido como a média da diferença absoluta entre os valores preditos $(\hat{y}_i)$ e os valores reais (y_i). O MAE é dado por:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|d_i|. \tag{2.11}$$

em que: $d_i = |y_i - \hat{y}_i|$ e N representa o número de observações no conjunto de teste. Segundo Fox (1981), esta estatística é mais robusta (menos afetada pela remoção de alguns outliers).

O erro quadrático médio (MSE) é obtido pela média dos quadrados dos erros, isto é:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2. \tag{2.12}$$

Uma extensão do MSE que também é muito utilizada na avaliação de modelos é a raiz do erro quadrático médio (RMSE) (DASARI et al., 2019), dada por:

$$RMSE = \sqrt{MSE}. \tag{2.13}$$

Estas duas estatísticas têm como característica comum penalizar mais erros maiores.

O Erro percentual absoluto médio (MAPE) representa a proporção da diferença média absoluta entre os valores preditos e os valores reais dividida pelo valor real. O MAPE é obtido por (FERRAZ et al., 2017):

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%. \tag{2.14}$$

O Erro Médio Absoluto Escalado (MASE) é uma medida de erro de previsão que compara a previsão de um modelo com uma previsão baseada em um modelo de tendência simples. O MASE é calculado dividindo o erro médio absoluto da previsão pelo erro médio absoluto da previsão baseada em um modelo de tendência simples (HYNDMAN; KOEHLER, 2006).

$$MASE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{Y}_i|}{\frac{1}{n-1}\sum_{i=2}^{n}|Y_i - Y_{i-1}|} \tag{2.15}$$

O Erro Médio Absoluto Percentual Simétrico (SMAPE) é uma medida de erro de previsão baseada em erros percentuais. É uma medida robusta e escalar, o que significa que é relativamente insensível a mudanças na distribuição dos dados e pode ser usada para comparar modelos de previsão com diferentes unidades de medida. É calculado dividindo a soma dos erros absolutos entre os valores reais e previstos pelos valores reais. O resultado é expresso em porcentagem (MAISELI, 2019).

$$SMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{2 \times |Y_i - \hat{Y}_i|}{|Y_i| + |\hat{Y}_i|} \times 100\% \tag{2.16}$$

SEGUNDA PARTE - ARTIGOS

ARTIGO 1 - Análise da temperatura de Florianópolis – SC utilizando uma abordagem GAMLSS

Redigido conforme as normas da revista *Sigmae journal* (versão publicada).

Análise da temperatura de Florianópolis (SC) utilizando uma abordagem GAMLSS

Viviane C. Silva[†1], Luiz R. Nakamura[2], Thiago G. Ramires[3], Geraldo M. C. Pereira[2]

[1] *Programa de Pós-Graduação em Estatística e Experimentação Agropecuária. Universidade Federal de Lavras (UFLA).*
[2] *Departamento de Estatística. Universidade Federal de Lavras (UFLA).*
[3] *Departamento de Matemática. Universidade Tecnológica Federal do Paraná (UTFPR).*

Resumo: *Compreender a variabilidade dos elementos climáticos na temperatura é relevante para atividades econômicas e para o cotidiano das pessoas. Ciente disso, o objetivo do estudo é analisar a temperatura média da cidade de Florianópolis – SC no período de um ano (01 de julho de 2021 a 30 de junho de 2022). Para isso, foram consideradas as seguintes variáveis explicativas candidatas: data da medição (tempo), temperatura em ponto de orvalho, precipitação total, pressão atmosférica, umidade e velocidade do vento. Para a modelagem, foram utilizados os modelos aditivos generalizados para locação, escala e forma (GAMLSS) por conta de sua flexibilidade para explicar o comportamento da variável resposta. A distribuição escolhida para representar a resposta foi a Box-Cox exponencial potência (BCPE), uma vez que ela é capaz de modelar variáveis que assumem valores positivos e apresentam diferentes graus de curtose. Para o processo de seleção de covariáveis em cada um dos parâmetros da distribuição, foi utilizado um processo de seleção de variáveis baseado no stepwise. Com base nos resíduos obtidos a partir do modelo final verificou-se que ele é adequado para explicar o conjunto de dados em questão.*

Palavras-chave: *Clima; Meteorologia; Regressão distribucional; Variabilidade.*

Abstract: *Understanding the variability of climate elements in the temperature is important for economic activities and people's daily lives. With this in mind, the main aim of this paper is to analyse the average temperature of Florianópolis, SC over a one-year period (1 July 2021 to 30 June 2022). The following explanatory variables were considered for this task: date (time), dew point temperature, total precipitation, atmospheric pressure, humidity, and wind speed. The generalised additive models for location, scale and shape (GAMLSS) were used due to their flexibility to explain the behaviour of the response variable. The Box-Cox power exponential (BCPE) distribution was chosen to explain the response since it can deal with positive variables with varying degrees of kurtosis. A stepwise-based method was performed to select covariates in each of the distribution's parameters. The residuals obtained from the final model were found to be adequate for explaining the data set.*

Keywords: *Climate; Meteorology; Distributional regression; Variability.*

Introdução

Compreender a variabilidade dos elementos climáticos na temperatura é relevante para atividades econômicas e para o cotidiano das pessoas. O clima se dá pelo comportamento e a atuação das condições da atmosfera em um dado local, consistindo em uma série de padrões climáticos que se sucedem e se repetem ciclicamente durante um período de meses ou anos. Nas ciências agrícolas, por exemplo, a variabilidade do clima pode afetar o rendimento nas colheitas como podemos observar em (BARLOW et al., 2015).

[†]Autora correspondente: viviane.silva3@estudante.ufla.br.

A caracterização da temperatura do ar de uma região pode ser realizada por meio da interpolação dos valores medidos em estações meteorológicas. Segundo Silva e Assunção (2004), a saúde humana, a energia e o conforto são afetados mais pelo clima do que por qualquer outro elemento do meio ambiente, a exemplo de doenças "induzidas pelo clima". Logo, o corpo humano sofre uma diminuição da sua resistência, por conta da mudança e temperaturas extremas. Não obstante, segundo Guimarães (2011), a temperatura tem forte influência no número de notificações de doenças respiratórias, tanto para idosos como para crianças.

Com as informações supracitadas, tem-se interesse específico no estudo da cidade de Florianópolis, capital do Estado de Santa Catarina. De modo geral, a cidade é frequentemente afetada pelas inúmeras mudanças no tempo, na pressão atmosférica, e, ainda, alta umidade relativa do ar, como pode-se observar nos trabalhos de Mendonça (2002) e Herrmann et al. (2009). Assim, o objetivo deste trabalho situa-se em uma análise da temperatura da cidade de Florianópolis – SC, relacionando-a com variáveis climáticas específicas. Para tal, foram considerados os modelos aditivos generalizados para locação, escala e forma (GAMLSS), propostos por Rigby e Stasinopoulos (2005).

Material e métodos

Conjunto de dados

O conjunto de dados utilizado neste trabalho foi obtido diretamente do Instituto Nacional de Meteorologia (INMET) e corresponde a 365 observações da estação meteorológica automática A806, em Florianópolis – SC, localizada na latitude -27,602530 e longitude -48,620096 a 4,87 metros de altitude, durante o período de 01 de julho de 2021 a 30 de junho de 2022. A temperatura média (em °C) na cidade é a variável resposta, e, para sua explicação, são consideradas as seguintes covariáveis candidatas: tempo (em dias), temperatura em ponto de orvalho (em °C), precipitação total (em mm), pressão atmosférica (em mB), umidade (em %) e velocidade do vento (em m.s^{-1}).

Modelagem estatística

Conforme descrito na seção de Introdução, neste trabalho, os GAMLSS serão empregados com o intuito de se explicar a temperatura média na cidade de Florianópolis. Os GAMLSS são modelos de regressão semi-paramétricos em que uma distribuição é escolhida para a resposta e diferentes estruturas de regressão são consideradas para explicar cada um de seus parâmetros, isto é, são selecionadas quais covariáveis afetam, por exemplo, a mediana ou a assimetria da distribuição da resposta. Os GAMLSS vêm recebendo grande destaque, teórico e prático, nos mais diversos campos do conhecimento, como, por exemplo, nas ciências médicas (RAMIRES et al., 2018), agrárias (RIGHETTO et al., 2019), atuariais (RAMIRES et al., 2021b), nos esportes (NAKAMURA et al., 2019) e na produção animal (NAKAMURA et al., 2022a).

A distribuição de probabilidade associada ao modelo que será utilizada neste trabalho é a Box-Cox exponencial potência (BCPE), proposta por Rigby e Stasinopoulos (2004). Algumas aplicações recentes dos GAMLSS baseados na distribuição BCPE podem ser vistas em Yamada et al. (2020), Costa et al. (2022) e Nakamura et al. (2022b).

Matematicamente, uma variável aleatória $Y > 0$ segue uma distribuição BCPE, definida pela transformação da variável aleatória Z dada por

$$Z = \begin{cases} \dfrac{1}{\sigma\nu}\left[\left(\dfrac{Y}{\mu}\right)^{\nu} - 1\right], & \text{se } \nu \neq 0 \\ \dfrac{1}{\sigma}\log\left(\dfrac{Y}{\mu}\right), & \text{se } \nu = 0 \end{cases},$$

em que $\mu > 0$, $\sigma > 0$, $-\infty < \nu < \infty$ e Z segue distribuição exponencial potência com parâmetro $\tau > 0$. Os parâmetros da distribuição BCPE são diretamente interpretáveis (RIGBY et al., 2019), característica desejável nos GAMLSS (RAMIRES et al., 2021a): μ é exatamente a mediana, σ é aproximadamente o coeficiente de variação, ν é o parâmetro de assimetria e τ o parâmetro relacionado à curtose.

Rigby e Stasinopoulos (2004) definem um GAMLSS baseado na distribuição BCPE como sendo

$$\begin{aligned}
\log(\boldsymbol{\mu}) &= \boldsymbol{X}_1\boldsymbol{\beta}_1 + \sum_{j=1}^{J_1} s_{j1}(\boldsymbol{x}_{j1}) \\
\log(\boldsymbol{\sigma}) &= \boldsymbol{X}_2\boldsymbol{\beta}_2 + \sum_{j=1}^{J_2} s_{j2}(\boldsymbol{x}_{j2}) \\
\boldsymbol{\nu} &= \boldsymbol{X}_3\boldsymbol{\beta}_3 + \sum_{j=1}^{J_3} s_{j3}(\boldsymbol{x}_{j3}) \\
\log(\boldsymbol{\tau}) &= \boldsymbol{X}_4\boldsymbol{\beta}_4 + \sum_{j=1}^{J_4} s_{j4}(\boldsymbol{x}_{j4}),
\end{aligned}$$

em que $\boldsymbol{X}_k$, $k = 1, \ldots, 4$, é uma matriz de delineamento, $\boldsymbol{\beta}_k = (\beta_{1k}, \ldots, \beta_{J_k k})^\top$ é o vetor de parâmetros e $s_{jk}(\cdot)$ é uma função de suavização que explica o relacionamento entre a covariável x_{jk} e o parâmetro da distribuição BCPE, que, neste trabalho, trata-se de um P-spline (EILERS e MARX, 1996).

No que tange à estimação dos GAMLSS, usualmente emprega-se o método da máxima verossimilhança penalizada, conforme disponível em Stasinopoulos e Rigby (2005). Ademais, diferentes estratégias podem ser adotadas com o intuito de se selecionar as estruturas de regressão (diferentes covariáveis) para cada um dos parâmetros. Conforme afimado por Ramires et al. (2021c), o protocolo mais utilizado para este fim é denominado Estratégia A, que consiste em uma metodologia baseada nos conhecidos procedimentos *stepwise* baseados em critérios de informação, como o de Akaike (AKAIKE, 1974). Mais detalhes sobre sua construção e aplicação podem ser encontrados em Nakamura et al. (2017) e Stasinopoulos et al. (2017).

Resultados e discussões

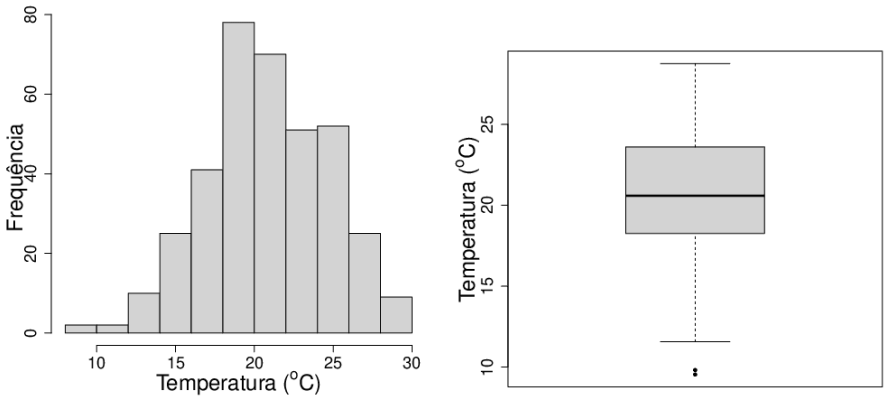
A Tabela 1 apresenta algumas medidas descritivas da variável resposta temperatura média (em °C) da cidade de Florianópolis, onde observamos que a resposta média e mediana durante o período de estudo é de 20,81 °C e 20,59 °C, respectivamente, com desvio padrão de 3,81 °C.

Tabela 1: Medidas descritivas acerca da temperatura média na cidade de Florianópolis

Média	Mediana	Desvio padrão	Assimetria	Curtose
20,81	20,59	3,81	-0,09	-0,39

A distribuição marginal da resposta (Figura 1) é simétrica (coeficiente de assimetria equivalente a -0,09) e possui caudas relativamente mais leves (platicúrtica) do que a distribuição normal (coeficiente de curtose igual a -0,39). Baseado nas características citadas, a distribuição BCPE, apresentada na seção de Material e Métodos, torna-se uma potencial e interessante alternativa para modelar o conjunto de dados em estudo.

Figura 1: Temperatura média na cidade de Florianópolis: (a) histograma e (b) gráfico de caixas



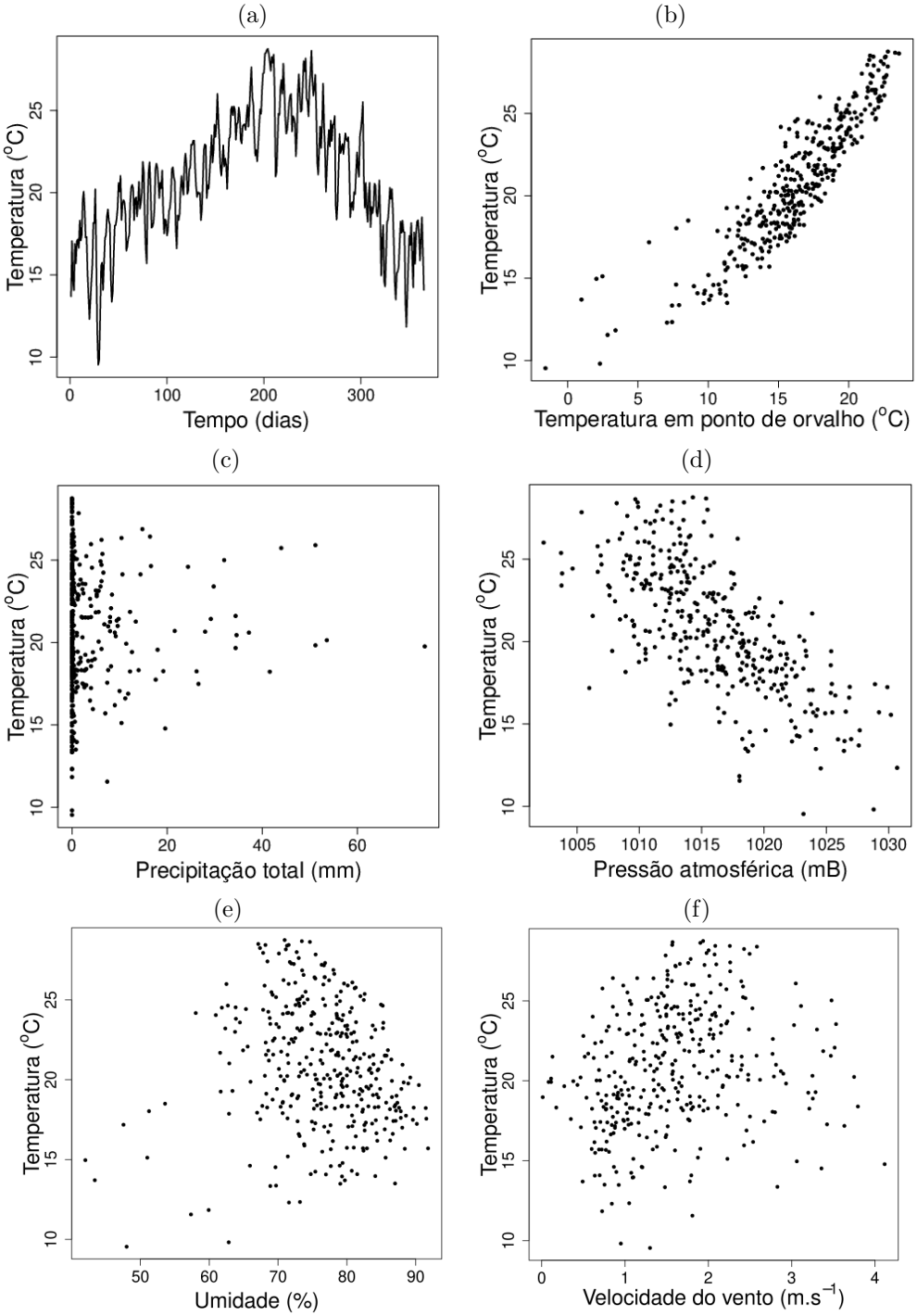
Fonte: Autores.

Uma vez escolhida a potencial distribuição a ser considerada no modelo, observam-se os relacionamentos, dois a dois, entre a resposta e cada uma das variáveis explicativas candidatas (Figura 2). Conforme pode-se observar no Painel (a), há um aumento na temperatura média da cidade de Florianópolis até, aproximadamente, o dia 200 e, depois, uma queda nesta característica. Tal resultado é esperado, uma vez que essas observações foram coletadas a partir do dia 01 de julho de 2021. Assim, as mais altas temperaturas médias são verificadas na estação de verão. O Painel (b) apresenta o relacionamento linear positivo entre a resposta e a temperatura em ponto de orvalho. A temperatura de ponto de orvalho é um bom indicador da quantidade de água existente numa parcela ou pacote de ar (TALAIA e VIGÁRIO, 2016). Os pontos de orvalho que estão concentrados no intervalo de 16 °C a 18 °C são os dias de melhor sensação no ser humano, enquanto que acima de 18 °C já há uma sensação desconfortável. Nos dados estudados, pode-se observar que essa temperatura de orvalho, passa desse limiar, uma vez que a cidade Florianópolis é considerada uma cidade fria.

Na Figura 2 (c), observa-se, conforme esperado, a quantidade excessiva de precipitação igual a zero. Correlação negativa pode ser verificada entre a resposta e a pressão atmosférica (Painel (d)). Segundo Jardim (2011), a pressão atmosférica do ar está sujeita a variações horárias, diárias, mensais, altitudinais e latitudinais. A relação entre temperatura média e umidade é apresentada no Painel (e), onde nota-se níveis, em geral, superiores a 60%, corroborando com o trabalho de Murara (2012), trazendo à população uma sensação de calor extremo que resulta na dificuldade de evaporação do suor do corpo humano. Segundo a Organização Mundial da Saúde (OMS) a umidade ideal para a saúde dos seres humanos deve estar entre 50 e 60% (CEPAGRI/UNICAMP, 2008).

A Figura 2 (f) apresenta o relacionamento entre a temperatura média e a velocidade do vento. Apesar da grande variabilidade, há uma relação linear positiva. Observa-se que a velocidade do vento na maioria das observações fica em torno de 0,5 $m.s^{-1}$ a 2,5 $m.s^{-1}$ (ou 1,8 $km.h^{-1}$ a 9 $km.h^{-1}$). Conforme a escala de Beaufort (WMO, 2008), os níveis do vento indicam a ocorrência de ar calmo a brisa fraca durante todo o estudo. Cabe ressaltar que para os meses de dezembro a março, estação de verão, a intensidade do vento extrapola os 3 $m.s^{-1}$ em determinados dias, sendo a velocidade máxima atingida na estação de, aproximadamente, 4,12 $m.s^{-1}$ (ou 14,83 $km.h^{-1}$).

Figura 2: Relacionamento entre a resposta temperatura média e variáveis explicativas: (a) tempo em dias; (b) temperatura em ponto de orvalho; (c) precipitação total; (d) pressão atmosférica; (e) umidade; e (f) velocidade do vento



Fonte: Autores.

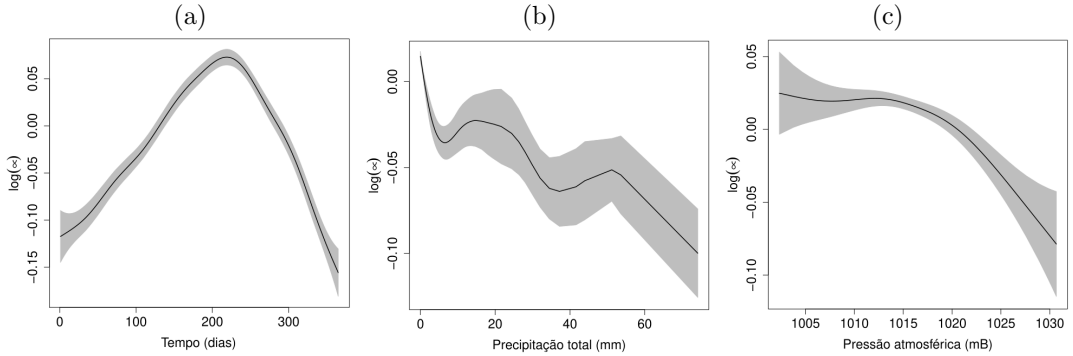
Para o processo de seleção das covariáveis em cada uma das estruturas de regressão foi utilizada a Estratégia A, com base no critério de informação de Akaike. O modelo final, após a aplicação do procedimento é dado a seguir.

$$\begin{aligned}
\log(\hat{\mu}) &= 11,1734 + s(\texttt{Tempo}) + s(\texttt{Precipitação}) + s(\texttt{Pressão}) + 0,0306\ \texttt{Orvalho} \\
\log(\hat{\sigma}) &= -2,0180 - 0,0832\ \texttt{Orvalho} + 0,1328\ \texttt{Vento} \\
\hat{\nu} &= -2,3982 + 0,0126\ \texttt{Tempo} + 0,6635\ \texttt{Precipitação} \\
\hat{\tau} &= 2,3555
\end{aligned}$$

Pode-se observar que funções de suavização foram necessárias somente para modelar $\hat{\mu}$. Conforme indicado por Ramires et al. (2019), os valores-p associados à covariáveis modeladas a partir de tais funções não devem ser interpretados. Nestes casos, apenas o efeito da função sobre o parâmetro da distribuição da resposta é avaliado graficamente (Figura 3).

O Painel (a) apresenta a influência da variável tempo em relação à mediana da temperatura média na cidade de Florianópolis. O comportamento observado coincide exatamente com o apresentado na Figura 2(a), isto é, as maiores temperaturas são observadas na estação de verão. O Painel (b) indica que diferentes valores de precipitação exercem um efeito negativo ou constante na mediana da temperatura média. O Painel (c) mostra que a mediana da temperatura média é constante até, aproximadamente, 1015 mB e, após este limiar, a mediana decresce. Além dessas três variáveis utilizadas para explicar $\hat{\mu}$, há ainda o efeito da temperatura em ponto de orvalho, estatisticamente significativa (valor-p <0,05): para cada 1 °C a mais observado nesta variável, espera-se um aumento de 0,0306 °C na temperatura média da cidade.

Figura 3: Relacionamento entre a resposta mediana e as covariáveis: (a) tempo; (b) precipitação total; e (c) pressão atmosférica



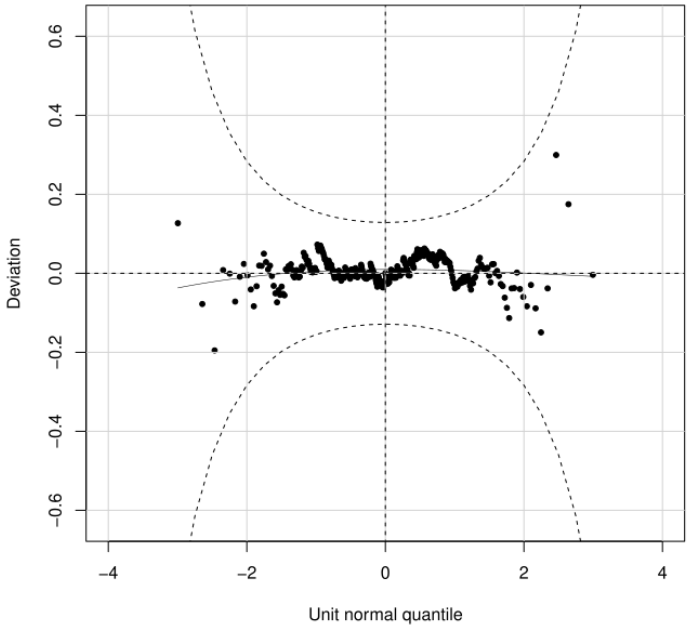
Fonte: Autores.

Em relação ao coeficiente de variação ($\hat{\sigma}$), duas covariáveis foram significativas (valor-p <0,05): i) a cada 1 °C a mais na temperatura em ponto de orvalho, há um decréscimo de 0,0832 unidades no coeficiente de variação da temperatura média na cidade de Florianópolis; ii) a cada 1 mm a mais de precipitação, há um aumento esperado de 0,1328 unidades no coeficiente de variação da resposta.

A variável tempo em $\hat{\nu}$ foi considerada não significativa (valor-p=0,1027). Entretanto, mesmo após esta constatação, ela foi mantida no modelo uma vez que, como aponta Lee et al. (2016), deve-se ter cautela em se remover variáveis após o processo de seleção. Em relação à variável precipitação (valor-p <0,05), a cada 1 mm a mais de chuva, espera-se um aumento de 0,6635 unidades na assimetria. Finalmente, a curtose ($\hat{\tau}$) foi modelada como uma constante, igual a 2,3555. Conforme pode ser visto em Rigby e Stasinopoulos (2004), quando $\hat{\tau} > 2$ a distribuição BCPE é platicúrtica.

A Figura 4 apresenta o *worm plot* (VAN BUUREN e FREDRIKS, 2001) construído a partir dos resíduos quantílicos normalizados (DUNN e SMYTH, 1996). Uma vez que os pontos estão todos dentro das bandas de 95% de confiança e nenhum padrão específico é observado, pode-se afirmar que o modelo ajustado, isto é, o GAMLSS baseado na distribuição BCPE provém um bom ajuste ao conjunto de dados em estudo.

Figura 4: Resíduos quantílicos normalizados obtidos do ajuste do GAMLSS baseado na distribuição BCPE



Fonte: Autores.

Considerações finais

Neste trabalho, o uso dos modelos aditivos generalizados para locação, escala e forma (GAMLSS) foi adequado para modelar os dados de temperatura da cidade de Florianópolis – SC demonstrando a versatilidade desta metodologia nas mais diversas áreas de conhecimento. A distribuição escolhida para representar a variável resposta, Box-Cox exponencial potência (BCPE), mostrou-se adequada para explicar o conjunto de dados em questão. O modelo não só foi vantajoso para modelar o parâmetro de locação (mediana da temperatura) como também os parâmetros de escala (coeficiente de variação) e forma (assimetria e curtose, sendo este segundo apenas modelado por uma constante). Assim, foi possível descrever e interpretar a natureza da variável resposta de uma forma mais objetiva. Finalmente, o gráfico *worm plot* construído a partir dos resíduos quantílicos normalizados nos propicia uma ferramenta para afirmar que o modelo em questão é razoável para descrever o conjunto de dados em estudo.

Agradecimentos

A primeira autora agradece o apoio financeiro recebido pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

Referências

AKAIKE, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, v. 19, p. 716-723, 1974.

BARLOW, J. W.; CHRISTY, B. P.; O'LEARY, G. J.; RIFFKIN, P. A.; NUTTALL, J. G. Simulating the impact of extreme heat and frost events on wheat crop production: a review. *Field Crops Research*, v. 171, p. 109–119, 2015.

CEPAGRI/UNICAMP. *Escala psicrométrica Unicamp para indicação de níveis de umidade relativa do ar prejudiciais à saúde humana*. 2008. Disponível em: http://www.cpa.unicamp.br/artigos-especiais/umidade-do-arsaude-no-inverno.html

COSTA, A. C. L.; OLIVEIRA, A. D. M.; CARACIOLO, J. P. S.; LUCENA, L. R. R.; LEITE, M. L. M. V. A GAMLSS approach to predicting growth of *Nopalea cochenillifera* Giant Sweet clone submitted to water and saline stress. *Acta Scientiarum. Agronomy*, v. 44, p. e54939, 2022.

DUNN, P. K.; SMYTH, G. K. Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, v. 5, p. 236-244, 1996.

EILERS, P. H. C.; MARX, B. D. Flexible smoothing with B-splines and penalties. *Statistical Science*, v. 11, p. 89–121, 1996.

GUIMARÃES, P. R. B.; BERGER, R.; PEREZ, F. L.; PIRES, P. T. L. Relações entre as doenças respiratórias e a poluição atmosférica e variáveis climáticas na cidade de Curitiba, Paraná, Brasil. *Floresta*, v. 42, p. 817–828, 2012.

HERRMANN, M. L. P.; CARDOZO, F. S.; BAUZYS, F.; PEREIRA, G. Freqüência dos desastres naturais no estado de santa catarina no período de 1980 a 2007. In: Encuentro de geógrafos de América Latina, 12. Montevideo, Uruguay. *Anais...*, 2009. p. 1-12. DVD. Disponível em: http://urlib.net/ibi/J8LNKAN8RW/36KNHCP.

JARDIM, C. H. Relações entre temperatura, umidade relativa do ar e pressão atmosférica em área urbana: comparação horária entre dois bairros no município de São Paulo-SP. *Revista Geografias*, v. 7, p. 128–142, 2011.

LEE, J. D.; SUN, D. L.; SUN, Y; TAYLOR, J. E. Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, v. 44, p. 907–927, 2016.

MURARA, P. G.; MENDONÇA, M.; BONETTI, C. O clima e as doenças circulatórias e respiratórias em Florianópolis/SC. *Hygeia*, v. 9, 2013.

MENDONÇA, M. *A dinâmica têmporo-espacial do clima subtropical na região conurbada de Florianópolis/SC*. Tese (Doutorado em Geografia Física). Departamento de Geografia da F.F.L.C.H./USP. São Paulo, 2002.

NAKAMURA, L. R.; CERQUEIRA, P. H. R.; RAMIRES, T. G.; PESCIM, R. R.; RIGBY, R. A.; STASINOPOULOS, D. M. A new continuous distribution on the unit interval applied to modelling the points ratio of football teams. *Journal of Applied Statistics*, v. 46, p. 416–431, 2019.

NAKAMURA, L. R.; RAMIRES, T. G.; RIGHETTO, A. J.; PESCIM, R. R.; ROQUIM, F. V.; SAVIAN, T. V.; STASINOPOULOS, D. M. Cattle reference growth curves based on centile estimation: A GAMLSS approach. *Computers and Electronics in Agriculture*, v. 192, p. 106572, 2022a.

NAKAMURA, L. R.; RAMIRES, T. G.; RIGHETTO, A. J.; SILVA, V.; KONRATH, A. C. Using the Box-Cox family of distributions to model censored data: a distributional regression approach. *Brazilian Journal of Biometrics*, v. 40, p. 407-414, 2022b.

NAKAMURA, L. R.; RIGBY, R. A.; STASINOPOULOS, D. M.; LEANDRO, R. A.; VILLEGAS, C.; PESCIM, R. R. Modelling location, scale and shape parameters of the Birnbaum-Saunders generalized t distribution. *Journal of Data Science*, v. 15, p. 221–237, 2017.

RAMIRES, T. G.; NAKAMURA, L. R.; RIGHETTO, A. J.; CARVALHO, R. J.; VIEIRA, L. A.; PEREIRA, C. A. B. Comparison between highly complex location models and GAMLSS. *Entropy*, v. 23, p. 469, 2021a.

RAMIRES, T. G.; NAKAMURA, L. R.; RIGHETTO, A. J.; KONRATH, A. C.; PEREIRA, C. A. B. Incorporating clustering techniques into GAMLSS. *Stats*, v. 4, p. 916–930, 2021b.

RAMIRES, T. G.; NAKAMURA, L. R.; RIGHETTO, A. J.; ORTEGA, E. M. M.; CORDEIRO, G. M. Predicting survival function and identifying associated factors in patients with renal insufficiency in the metropolitan area of Maringá, Paraná State, Brazil. *Cadernos de Saúde Pública*, v. 34, p. e00075517, 2018.

RAMIRES, T. G.; NAKAMURA, L. R.; RIGHETTO, A. J.; PESCIM, R. R.; MAZUCHELI, J.; CORDEIRO, G. M. A new semiparametric Weibull cure rate model: fitting different behaviors within GAMLSS. *Journal of Applied Statistics*, v. 46, p. 2744–2760, 2019.

RAMIRES, T. G.; NAKAMURA, L. R.; RIGHETTO, A. J.; PESCIM, R. R.; MAZUCHELI, J.; RIGBY, R. A.; STASINOPOULOS, D. M. Validation of stepwise-based procedure in GAMLSS. *Journal of Data Science*, v. 19, p. 96–110, 2021c.

RIGBY, R. A.; STASINOPOULOS, D. M. Smooth centile curves for skew and kurtotic data modelled using the Box-Cox power exponential distribution. *Statistics in Medicine*, v. 23, p. 3053-3076, 2004.

RIGBY, R. A.; STASINOPOULOS, M. D.; HELLER, G. Z.; DE BASTIANI, F. *Distributions for Modeling Location, Scale, and Shape: Using GAMLSS in R*. Boca Raton: CRC Press. 2019. 560 p.

RIGHETTO, A. J.; RAMIRES, T. G.; NAKAMURA, L. R.; CASTANHO, P. L. D. B.; FAES, C.; SAVIAN, T. V. Predicting weed invasion in a sugarcane cultivar using multispectral image. *Journal of Applied Statistics*, v. 46, p. 1–12, 2019.

SILVA, E. M.; ASSUNÇÃO, W. L. O clima da cidade de Uberlândia-MG. *Sociedade & Natureza*, v. 16, p. 91–107, 2004.

STASINOPOULOS, M. D.; RIGBY, R. A.; HELLER, G. Z.; VOUDOURIS, V.; DE BASTIANI, F. *Flexible Regression and Smoothing: Using GAMLSS in R*. Boca Raton: CRC

Press. 2017. 572 p.

TALAIA, M.; VIGÁRIO, C. *Temperatura de ponto de orvalho: um risco ou uma necessidade*, 2016. Disponível em: http://hdl.handle.net/10316.2/39909.

VAN BUUREN, S.; FREDRIKS, M. Worm plot: a simple diagnostic device for modelling growth reference curves. *Statistics in Medicine*, v. 20, p. 1259-1277, 2001.

WMO (World Meteorological Organization). *Guide to meteorological instruments and methods of observation*, 2008.

YAMADA, G.; JONES-SMITH, J. C.; CASTILLO-SALGADO, C.; MOULTON, L. H. Differences in magnitude and rates of change in BMI distributions by socioeconomic and geographic factors in Mexico, Colombia, and Peru, 2005–2010. *European Journal of Clinical Nutrition*, v. 74, p. 472–480, 2020.

ARTIGO 2 - Influência de variáveis climáticas na temperatura de Florianópolis – SC: uma comparação entre modelos de regressão distribucional e outros algoritmos de aprendizado de máquina

Influência de variáveis climáticas na temperatura de Florianópolis – SC: uma comparação entre modelos de regressão distribucional e outros algoritmos de aprendizado de máquina

Viviane C. Silva[1], Luiz R. Nakamura[2], Geraldo M.C. Pereira[2],
Thiago G. Ramires[3], Dimitrios M. Stasinopoulos[4]

Resumo

A temperatura média de uma cidade é um indicador importante do clima local, afetando a saúde humana, a agricultura, o turismo e outros setores da economia. Portanto, é importante entender os fatores que influenciam a temperatura média de uma cidade. Diferentes metodologias podem ser aplicadas para a análise deste tipo de dado meteorológico, como, por exemplo, os algoritmos de *machine learning* amplamente utilizados na literatura: *random forest*, *support vector regression*, *extreme gradient boosting* e *prophet*. Outra alternativa é a utilização de modelos de regressão distribucional, ou modelos aditivos generalizados para locação, escala e forma (GAMLSS), uma importante e flexível classe de modelos de regressão univariados. Assim, a ideia deste trabalho é comparar a performance preditiva dos GAMLSS com os quatro outros algoritmos de *machine learning* citados. Para tal, foram utilizados dados provenientes de uma estação meteorológica automática na cidade de Florianópolis – SC, coletados no período de 30 de março de 2013 a 28 de março de 2023. Os GAMLSS baseados na distribuição Box-Cox t apresentaram resultados mais satisfatórios na maioria das métricas utilizadas para a comparação dos modelos ajustados, provando ser uma interessante alternativa para o ajuste e predição de dados meteorológicos.

Palavras-chave: *Extreme gradient boosting*, *Prophet*, *Random forest*, *Support vector regression*

1 Introdução

Os climas globais são um dos problemas ambientais mais estudados e comentados, pois a temperatura é um fator determinante na nossa sensação de conforto e bem-estar, especialmente nas cidades. Para compreender os elementos climáticos de uma localidade, é necessário considerar dois fatores principais: as características locais e a circulação geral da atmosfera (MUNIZ; CARACRISTI, 2021). Neste sentido, a temperatura é um dos aspectos que expressa uma relação entre sociedade e economia, pois mudanças ligadas a essa váriável fora dos padrões, atingem diretamente a vida da população, por isso a importância em estudar a temperatura de determinada localidade.

Os algoritmos de *machine learning* pertencem a um campo da ciência da computação que lida com o desenvolvimento de algoritmos que podem aprender e se adaptar a partir

[1]Programa de Pós-Graduação em Estatística e Experimentação Agropecuária, Universidade Federal de Lavras, Lavras, Brasil.
[2]Departamento de Estatística, Universidade Federal de Lavras, Lavras, Brasil.
[3]Departamento Acadêmico de Matemática, Universidade Tecnológica Federal do Paraná, Apucarana, Brasil.
[4]School of Computing & Mathematical Science, University of Greenwich, Reino Unido.

de dados (SAID; PINHEIRO, 2024). Os algoritmos de aprendizado de máquina estão revolucionando a ciência, pois fornecem métodos poderosos para analisar dados complexos, extrair relações não lineares em conjuntos de dados massivos e construir modelos preditivos precisos (KASHINATH et al., 2021). Trabalhos como o de Bochenek e Ustrnul (2022) e Jones (2017) usando *machine learning* para a análise climática, têm o potencial de revolucionar a maneira como entendemos e prevemos o clima. Dentre os algoritmos mais utilizados na literatura, destacam-se o *random forest* (BREIMAN, 2001), *support vector regression* Vapnik (1999), *extreme gradient boosting* (CHEN; GUESTRIN, 2016) e *prophet* (TAYLOR; LETHAM, 2018).

Por outro lado, na literatura estatística, os modelos aditivos generalizados para locação, escala e forma (GAMLSS) (RIGBY; STASINOPOULOS, 2005), também conhecidos como modelos de regressão distribucional, vêm ganhando notória visibilidade por conta de sua grande flexibilidade. Os GAMLSS permitem analisar a relação entre uma variável resposta (por exemplo, a temperatura média de uma cidade), e várias variáveis explicativas, como os elementos climáticos e outros fatores relevantes. A grande vantagem dos GAMLSS é a capacidade de modelar diferentes aspectos da distribuição da variável resposta, incluindo sua locação (média ou mediana, por exemplo), dispersão (desvio padrão, por exemplo) e forma (assimetria e curtose). Trabalhos com elementos climáticos utilizando esta abordagem podem ser vistos em Zhang et al. (2015), Villarini et al. (2010) e Costa et al. (2023), por exemplo.

Assim, o objetivo principal deste trabalho é estudar a relação entre a temperatura média da cidade de Florianópolis – SC, e variáveis climáticas específicas, por meio das metodologias supracitadas, comparando o poder preditivo de cada um dos modelos.

2 Material e métodos

2.1 Conjunto de dados

Os dados utilizados neste trabalho foram obtidos do Instituto Nacional de Meteorologia (INMET) e compreendem 3.651 observações coletadas diariamente da estação meteorológica automática A806, situada em Florianópolis, Santa Catarina. A estação está localizada nas coordenadas de latitude -27,602530 e longitude -48,620096, a uma altitude de 4,87 metros, e as medições foram realizadas no período de 30 de março de 2013 a 28 de março de 2023. O conjunto de dados apresenta 168 valores ausentes, que foram substituídos pela média das variáveis correspondentes.

A variável de interesse neste trabalho é a temperatura média (em °C) na cidade. Com o intuito de explicar o comportamento desta série temporal, foram consideradas as seguintes covariáveis candidatas: tempo (em dias), temperatura do ponto de orvalho (em °C), precipitação total (em mm), pressão atmosférica (em mB), umidade (em %) e velocidade do vento (em $m.s^{-1}$).

2.2 Modelos preditivos candidatos

Conforme mencionado na Seção 1, neste trabalho, nosso objetivo principal é avaliar e comparar o desempenho preditivo dos modelos de regressão distribucional, ou GAMLSS, em contraste com outros algoritmos de aprendizado de máquina, a saber: *random forest* (RF), *support vector regression* (SVM), *extreme gradient boosting* (XGBoosting) e *prophet*. O banco de dados é dividido em treinamento e teste, sendo o conjunto de treinamento,

utilizado para ajuste dos modelos e, a segunda parte, chamada de conjunto de teste, utilizada para a avaliação da qualidade de ajuste dos modelos. No conjunto teste serão calculadas as métricas de desempenho apresentadas na Tabela 1.

Tabela 1: Métricas de avaliação de modelos preditivos

Métrica	**Fórmula**
Erro médio absoluto (MAE)	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert Y_i - \hat{Y}_i\rvert$
Raiz do erro quadrático médio (RMSE)	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}$
Erro percentual absoluto médio (MAPE)	$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{Y_i - \hat{Y}_i}{Y_i}\right\rvert \times 100\%$
Erro escalonado médio absoluto (MASE)	$MASE = \frac{1}{n}\sum_{i=1}^{n}\frac{\lvert Y_i - \hat{Y}_i\rvert}{\frac{1}{n-1}\sum_{i=2}^{n}\lvert Y_i - Y_{i-1}\rvert}$
Erro percentual absoluto médio simétrico (SMAPE)	$SMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{2\times\lvert Y_i - \hat{Y}_i\rvert}{\lvert Y_i\rvert + \lvert\hat{Y}_i\rvert} \times 100\%$
Coeficiente de determinação (R^2)	$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$

Apresentaremos agora, de maneira bastante breve, os algoritmos utilizados neste artigo para o ajuste da temperatura média da cidade de Florianópolis – SC.

Modelos de regressão distribucional

Matematicamente, vamos considerar que uma variável resposta Y pode ser descrita por uma distribuição $\mathcal{D}(\theta_k)$ qualquer, em que θ_k, $k = 1, \ldots, p$, é o vetor de parâmetros associados a esta distribuição. Assim, definimos os GAMLSS como

$$g_k(\theta_k) = \boldsymbol{X}_k\boldsymbol{\beta}_k + \sum_{j=1}^{J_k} s_{jk}(\boldsymbol{x}_{jk}), \tag{1}$$

em que $g_k(\cdot)$ é uma função de ligação vinculada ao parâmetro k, usualmente escolhida de acordo com o espaço paramétrico (DE BASTIANI et al., 2018), $\boldsymbol{X}_k$ é uma matriz de delineamento, $\boldsymbol{\beta}_k = (\beta_{1k}, \ldots, \beta_{J_k k})^\top$ é um vetor de parâmetros e $s_{jk}(\cdot)$ é uma função de suavização, como o p-spline (EILERS; MARX, 1996), que explica o relacionamento da covariável $\boldsymbol{x}_{jk}$ com o parâmetro θ_k.

No que tange à seleção da distribuição $\mathcal{D}(\theta_k)$, conforme Stasinopoulos et al. (2017) uma das estratégias iniciais mais utilizadas é o ajuste marginal da resposta a partir de diferentes opções, comparadas e selecionadas a partir de algum critério de ajuste, como, por exemplo, o critério de informação de Akaike (AIC) (AKAIKE, 1974). Baseado nesta abordagem inicial, quatro distribuições foram selecionadas com o intuito de explicar a temperatura média da cidade de Florianópolis: Box-Cox t (BCT) e Box-Cox exponencial potência (BCPE) – ambas com quatro parâmetros: mediana, coeficiente de variação, assimetria e curtose –, Box-Cox Cole-Green (BCCG) – com três parâmetros: mediana, coeficiente de variação e assimetria – e Weibull, com dois parâmetros, relacionados à média e à dispersão. Mais informações sobre cada uma dessas distribuições podem ser encontradas em Rigby et al. (2019). Por fim, para a seleção das covariáveis em cada uma das estruturas de regressão dos parâmetros dessas quatro distribuições foi utilizada a denominada Estratégia A (RAMIRES et al., 2021).

Uma vez que estamos lidando com uma série temporal, para corrigir a autocorrelação entre as observações, a estratégia adotada neste trabalho foi calcular os resíduos parciais

relacionados ao parâmetro de locação das distribuições utilizadas – no caso das distribuições da família Box-Cox, a mediana, e no caso da Weibull, a média –, ajustar um modelo auto-regressivo integrado de médias móveis (ARIMA) a estes resíduos e utilizar os valores obtidos neste ajuste, como um *offset* no modelo original.

Random forest

O *random forest* (RF) são representações simples do conhecimento e um meio eficiente de construir classificadores que predizem classes baseadas nos valores de atributos de um conjunto de dados (BREIMAN, 2001). O objetivo principal no algoritmo é construir várias árvores de decisão durante o treinamento e, em seguida, combinar suas previsões para obter uma resposta mais estável e geral (GUARNIZO et al., 2021) para realizar previsões. Matematicamente, emprega-se a formulação

$$\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x),$$

em que, $\hat{f}_{rf}^{B}(x)$ é a predição do algoritmo *Random Forest* para um novo dado x, após ter sido treinado em B árvores de decisão, B é o número de árvores, $T_b(x)$ predição da b-ésima árvore de decisão para o dado x.

Support vector machine

O *support vector regression* (SVR), proposto por Vapnik (1999), é um método de regressão não linear que utiliza máquinas de suporte vetorial para prever valores contínuos, sendo robusto a ruídos e podendo ser empregado em uma ampla variedade de problemas de regressão complexos. Funciona encontrando um hiperplano que minimize a distância entre os pontos de dados e o hiperplano (KOR; ALTUN, 2020). Os pontos de dados que estão mais próximos do hiperplano são chamados de vetores de suporte. O SVR é formulado pelo seguinte problema de otimização

$$\text{minimizar:} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*)$$

$$\text{sujeito a:} \quad \begin{cases} y_i - \langle w, x_i\rangle - b \leq \epsilon + \xi_i \\ \langle w, x_i\rangle + b - y \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

em que, w é um vetor de pesos, representando os coeficientes da função de regressão linear, C uma constante positiva de regularização. ξ_i e ξ_i^* são variáveis de folga, m o número de pontos de treinamento no conjunto de dados, ε a margem de tolerância, $\langle w, x_i\rangle$ o produto interno entre o vetor de pesos w e o vetor de características x_i, representando a predição da função de regressão para o ponto x_i e y_i o valor alvo real associado ao ponto de treinamento x_i.

Extreme gradient boosting

O *extreme gradient boosting* (XGBoost), proposto por Chen e Guestrin (2016), é um algoritmo de aprendizado supervisionado que implementa um processo denominado *boosting* (impulsionar) para produzir modelos precisos, tendo sido projetado para superar,

como uma generalização, as limitações do *gradient boosting*, como sensibilidade a *overfitting* e seu baixo desempenho em conjuntos de dados grandes, através de técnicas como regularização e manipulação eficiente de árvores de decisão. Refere-se à técnica de aprendizado de conjunto que envolve a construção sequencial de diversos modelos, onde cada novo modelo é projetado para corrigir as deficiências identificadas no modelo anterior (MITCHELL; FRANK, 2017). Matematicamente, é escrito da seguinte forma:

$$y = \sum_{\ell=1}^{L} f_\ell(x_i), f_\ell \in F$$

em que L é o número de árvores, f é uma função no espaço F e o espaço F são todos os valores das árvores de decisão (MARINHO et al., 2021).

Prophet

O *Prophet* é um algoritmo de previsão de séries temporais desenvolvido pelo Facebook (TAYLOR; LETHAM, 2018), tendo sido projetado para lidar com séries complexas que apresentam tendências, sazonalidades e eventos importantes. O algoritmo é capaz de aprender os padrões sazonais de uma série de dados sem a necessidade de ajustes personalizados (YAN et al., 2019). Matematicamente, podemos representá-lo por

$$Y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

em que $g(t)$ representa a tendência, $s(t)$ representa a sazonalidade, $h(t)$ representa o efeito do feriado sobre o comportamento dos dados. O valor de ϵ_t é um erro ou alteração nos dados que não estão contidos no modelo (BASHIR et al., 2022).

3 Resultados e discussões

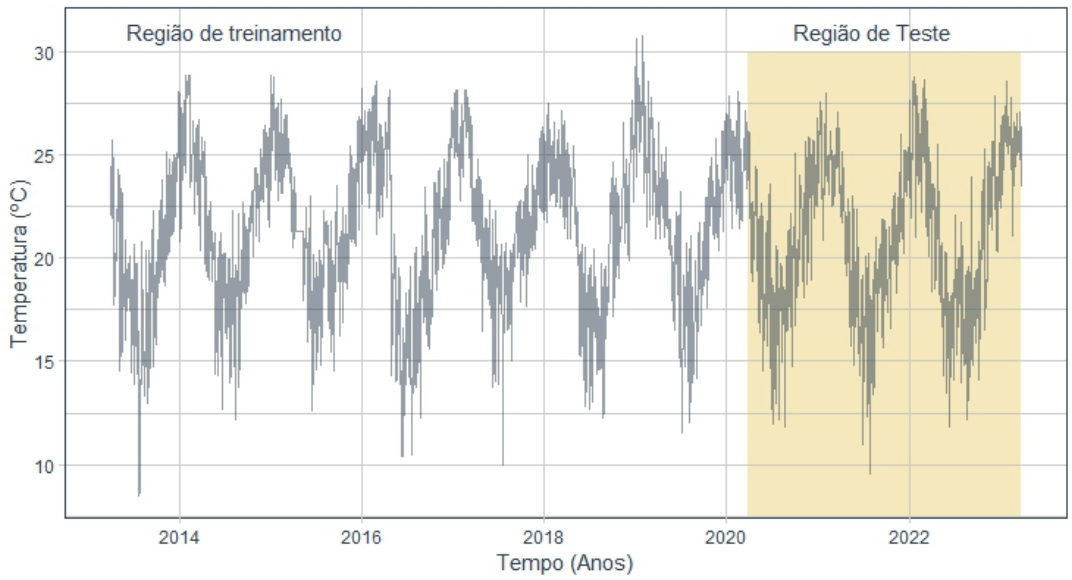
A Tabela 2 apresenta algumas medidas descritivas da variável resposta temperatura média ao longo de todo o período em estudo, onde observamos que a resposta média e mediana durante o período de estudo foram de 21,2 ºC e 21,4 ºC, respectivamente, com desvio padrão de 3,65 ºC. A série em estudo pode ser observada na Figura 1(a), dividida, conforme citado na Seção 2.2, em 70% das observações para o banco de treinamento e 30% para o banco de teste, onde notamos um comportamento sazonal similar em todos os anos, onde nos meses de janeiro são observadas as temperaturas mais elevadas na cidade, decorrente do verão, como descrito no trabalho (COSTA et al., 2023).

Tabela 2: Medidas descritivas acerca da variável temperatura média

Média	Mediana	Desvio padrão	Assimetria	Curtose
21,28	21,42	3,65	-0,27	-0,42

Conforme podemos observar no Painel (b), a distribuição marginal da resposta apresenta um comportamento ligeiramente assimétrico negativo (coeficiente de assimetria equivalente a -0,27) e um grau de curtose negativo, igual a -0,42, isto é, há poucas observações nas caudas da distribuição, sendo assim considerada uma distribuição platicúrtica. Ainda, há a presença de alguns valores discrepantes como apresentado no gráfico de caixa (Figura 1(c)).

(a)



(b) (c)

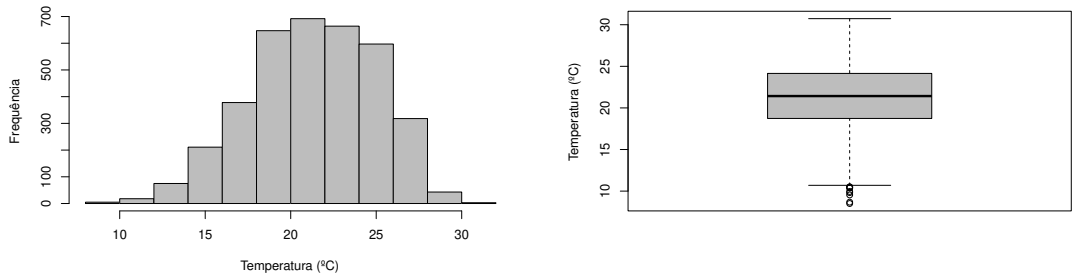
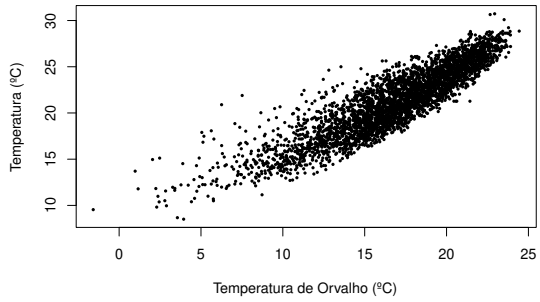


Figura 1: Temperatura média mensal da cidade de Floriánópolis no período de 03/2013 a 03/2023: (a) série temporal com as regiões demilitadas de treinamento e teste; (b) histograma; (c) gráfico de caixa.

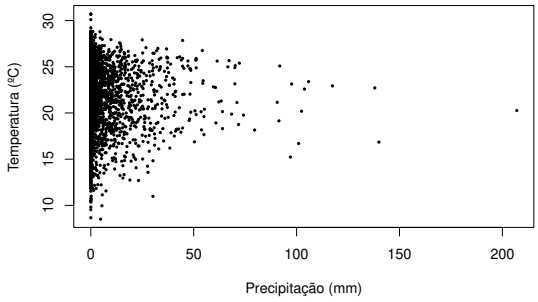
Na Figura 2 observam-se os relacionamentos, dois a dois, entre a resposta e cada uma das variáveis explicativas candidatas, com exceção da variável tempo já apresentada na Figura 1(a). No Painel (a) observamos um relacionamento aparentemente linear positivo entre as variáveis temperatura média e temperatura de orvalho. Ainda, há uma dispersão um pouco mais na resposta quando a temperatura de orvalho é baixa, ao passo que conforme esta temperatura de orvalho aumenta, a dispersão diminui.

Conforme esperado (HUFF; JR, 1973), observamos que há uma quantidade significativa de precipitação registrada como zero, no Painel (b). No Painel (c) notamos que o relacionamento entre temperatura e pressão atmoférica é negativo, indicando que quanto maior a pressão menor a temperatura, isso ocorre porque, a densidade do ar diminui com o aumento da temperatura. O ar frio é mais denso que o ar quente, por isso ele exerce uma pressão maior sobre o ar que está abaixo dele., elevando, assim, a pressão

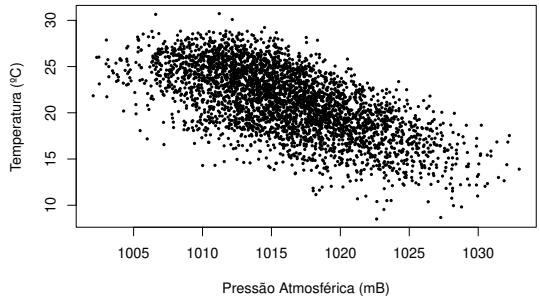
(a)



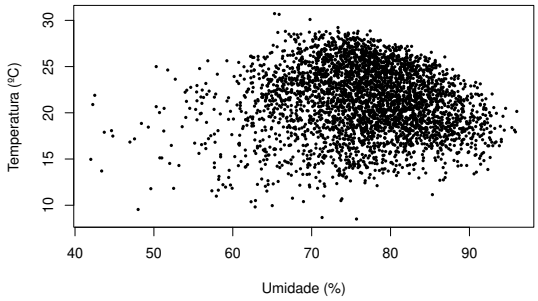
(b)



(c)



(d)



(e)

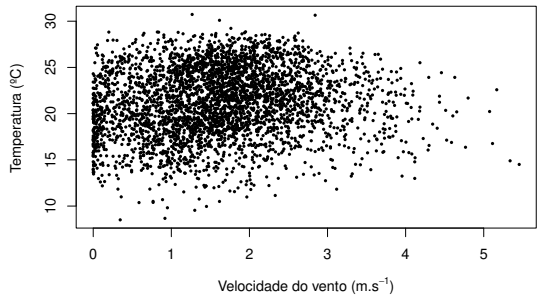


Figura 2: Relacionamento entre a resposta temperatura média e variáveis explicativas: (a) temperatura em ponto de orvalho; (b) precipitação total; (c) pressão atmosférica; (d) umidade; e (e) velocidade do vento

atmosférica (MEDEIROS; VIEIRA, 1997). A relação de temperatura e umidade é observada no Painel (d),onde a umidade do ar nesses anos ultrapassa, na maioria dos dias, 60% da umidade relativa do ar. Segundo Oliveira et al. (2020) isso se dá por conta da cidade de Florianópolis, em sua maioria, ser situada em uma ilha, e banhada pelo oceano Atlântico, o que contribui para a alta umidade relativa do ar ao longo do ano. Do Painel (e), o relacionamento entre a velocidade do vento e a temperatura média não é muito claro, entretanto é possível observar a diminuição da variabilidade da resposta conforme a velocidade aumenta.

Uma vez conduzida e finalizada a análise descritiva e exploratória dos dados, foram realizados todos os treinamentos dos modelos descritos na Seção 2.2. Visto que os GAMLSS passam por um processo de seleção (Estratégia A) de covariáveis nas diferentes estruturas de regressão, apresentamos as diferentes variáveis explicativas incluídas a partir de uma função de suavização no modelo final para cada uma das distribuições consideradas na Tabela 3. Não obstante, conforme mencionado na Seção 2.2, reiteramos que houve inclusão de um *offset* no modelo a partir do ajuste de um ARIMA nos resíduos parciais do parâmetro referente à locação. As variáveis temperatura, precipitação, pressão atmosférica e velocidade do vento são as mais importantes para explicar a variabilidade dos parâmetros das distribuições em todas as variáveis meteorológicas consideradas.

Tabela 3: Variáveis selecionadas em cada um dos parâmetros das distribuições consideradas nos GAMLSS

Distribuição	Parâmetro	Temperatura de orvalho	Precipitação	Pressão atmosférica	Umidade	Velocidade do vento
BCT	Mediana	×	×	×	×	×
	Coeficiente de variação	×		×		×
	Assimetria	×		×		
	Curtose	×	×			
BCPE	Mediana	×	×	×	×	×
	Coeficiente de variação	×	×	×	×	×
	Assimetria					×
	Curtose		×	×		×
BCCG	Mediana	×	×	×		×
	Coeficiente de variação	×	×	×		×
	Assimetria					×
Weibull	Média	×	×	×	×	×
	Dispersão	×		×	×	×

As métricas para comparação dos ajustes e predições estão disponíveis na Tabela 4. Os modelos que se destacaram com os melhores desempenhos, de acordo com essas métricas no conjunto de treinamento, foram o GAMLSS baseado na distribuição BCT com SMAPE(0,007), MAPE(0,070), RMSE(0,288), MAE(0,148), MASE(0,130), R^2(0,993) e o XGBoost com um SMAPE(0,090), MAPE(0,090), RMSE(0,028), MAE(0,018), MASE(0,016) e R^2(1,00).

Após a seleção dos dois melhores modelos no conjunto de treinamento, eles foram aplicados e comparados no conjunto de teste para verificar se também apresentaram um bom desempenho.

Aplicando as variâncias de importância ao modelo XGBoost, observamos na Figura 3 que apenas temperatura de orvalho foi significativa, com valores de importância superiores a 0,8. A umidade, a pressão atmosférica, a precipitação e a velocidade do vento apresentaram importâncias mínimas, com valores inferiores a 0,2. E a data não teve nenhuma significância no modelo. Notamos que a variável temperatura de orvalho foi muito importante para o modelo XGBoost como também para o modelo GAMLSS, ela

Tabela 4: Desempenho dos modelos ajustados no conjunto de treinamento

Modelo	SMAPE	MAPE	RMSE	MAE	MASE	R^2
GAMLSS (BCT)	0,007	0,770	0,288	0,148	0,130	0,993
GAMLSS (BCPE)	0,008	0,008	0,304	0,159	0,140	0,993
GAMLSS (BCCG)	0,044	4,346	1,199	0,902	0,794	0,891
GAMLSS (Weibull)	0,015	1,544	0.504	0.272	0,239	0.980
XGBoost	0,090	0,090	0,028	0,018	0,016	1,000
Random Forest	0,549	0,550	0,197	0,107	0,094	0,997
Prophet	0,646	0,646	0,249	0,130	0,115	0,995
SVR	0,787	0,791	0,275	0,160	0,141	0,995

influenciou todos os parâmetros do modelo baseado na distribuição BCT.

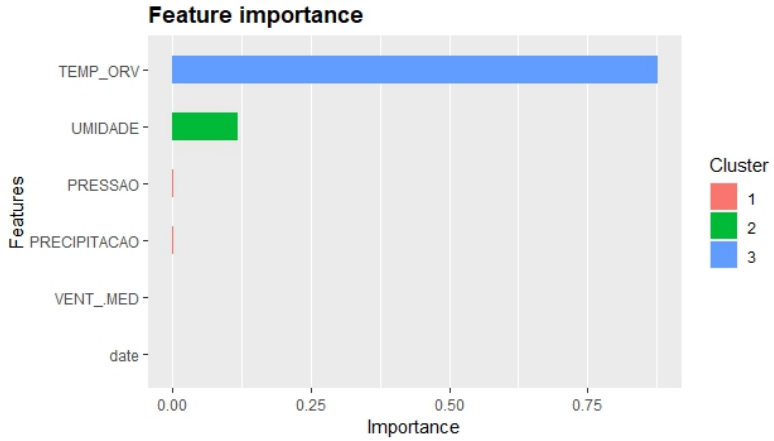


Figura 3: Variáveis de importância do modelo XGBoost no conjunto de teste

Na Tabela 5, são apresentadas as métricas para os dois modelos finais, no conjunto de banco de dados teste, observamos que o modelo de regressão distribucional baseado na distribuição BCT obteve um desempenho superior ao modelo XGBoost no conjunto de teste, com valores de MAE (0,127), MAPE (0,671), MASE (0,041) e SMAPE (0,671) menores, bem como um R^2 (0,997) ligeiramente maior. A única exceção foi o RMSE, em que o modelo XGBoost obteve um resultado superior, com um valor de 0,153, contra 0,195 dos GAMLSS.

Tabela 5: Modelos finais GAMLSS e Algoritmos de Machine Learning no banco de dados teste

Modelo	**SMAPE**	**MAPE**	**RMSE**	**MAE**	**MASE**	$\mathbf{R^2}$
GAMLSS (BCT)	0,671	0,671	0,195	0,127	0,041	0,997
XGBoost	0,795	0,800	0,153	0,153	0,130	0,995

Do ponto de vista estatístico, podemos afirmar que as inferências, tomadas de decisão e predições realizadas a partir dos GAMLSS são confiáveis uma vez que a análise de resíduos é satisfatória. A Figura 4 apresenta os gráficos referentes à função de autocorrelação (ACF), à função de autocorrelação parcial (PACF), densidade e o Q-QPlot dos resíduos obtidos a partir do modelo final baseado na distribuição BCT. Os gráficos referentes a ACF

e PACF mostram que não há autocorrelação significativa entre os resíduos, indicando que a estratégia empregada em relação à inclusão de um *offset* com base em um modelo ARIMA nos resíduos parciais do parâmetro de locação da distribuição BCT foi realizada com êxito. O gráfico de densidade mostra que os resíduos se aproximam de uma distribuição normal, qualidade necessária nos resíduos quantílicos normalizados (DUNN; SMYTH, 1996) utilizados nesta análise. Isso é confirmado pelo gráfico Q-QPlot, que mostra que mais de 95% dos pontos estão sobre a linha.

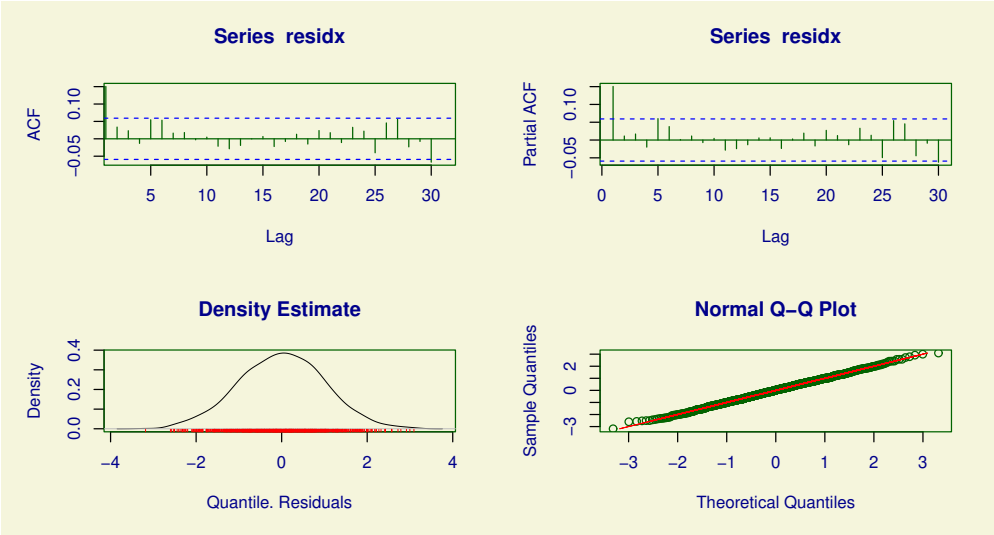


Figura 4: Análise de resíduos do modelo BCTo

Considerações finais

Neste trabalho, foi realizada uma comparação entre os modelos aditivos generalizados para locação, escala e forma (GAMLSS), ou modelos de regressão distribucional, e outros algoritmos de *machine learning* bastante utilizados na literatura para modelar os dados da temperatura média da cidade de Florianópolis. Os resultados obtidos indicam que os GAMLSS são uma interessante alternativa para modelar dados de temperatura, pois são capazes de capturar a variabilidade da distribuição de forma flexível e robusta. Os algoritmos comumente utilizados de apredizado de máquina também são uma opção viável, mas podem apresentar resultados menos precisos, dependendo do algoritmo e das configurações utilizadas. Em particular, os GAMLSS baseado na distribuição BCT obtiveram um resultado superior na maioria das métricas em comparação com o algoritmo XGBoost, que apresentou as melhores métricas dentre os usualmente utilizados.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Referências

AKAIKE, H. A new look at the statistical model identification. **IEEE transactions on automatic control**, Ieee, v. 19, n. 6, p. 716–723, 1974.

BASHIR, T.; HAOYONG, C.; TAHIR, M. F.; LIQIANG, Z. Short term electricity load forecasting using hybrid prophet-lstm model optimized by bpnn. **Energy reports**, Elsevier, v. 8, p. 1678–1686, 2022.

BOCHENEK, B.; USTRNUL, Z. Machine learning in weather prediction and climate analyses—applications and perspectives. **Atmosphere**, MDPI, v. 13, n. 2, p. 180, 2022.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, p. 5–32, 2001.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. [S.l.: s.n.], 2016. p. 785–794.

COSTA, V.; NAKAMURA, L. R.; RAMIRES, T. G.; PEREIRA, G. M. Análise da temperatura de Florianópolis (SC) utilizando uma abordagem GAMLSS. **Sigmae**, v. 12, n. 1, p. 129–138, 2023.

DE BASTIANI, F.; RIGBY, R. A.; STASINOPOULOS, D. M.; CYSNEIROS, A. H. M. A.; URIBE-OPAZO, M. A. Gaussian markov random field spatial models in GAMLSS. **Journal of Applied Statistics**, Taylor & Francis, v. 45, n. 1, p. 168–186, 2018.

DUNN, P. K.; SMYTH, G. K. Randomized quantile residuals. **Journal of Computational and graphical statistics**, Taylor & Francis, v. 5, n. 3, p. 236–244, 1996.

EILERS, P. H.; MARX, B. D. Flexible smoothing with b-splines and penalties. **Statistical science**, Institute of Mathematical Statistics, v. 11, n. 2, p. 89–121, 1996.

GUARNIZO, J. A. Y. et al. Métodos supervisionados de machine learning aplicados à produtividade agrícola de cana-de-açúcar. [sn], 2021.

HUFF, F.; JR, S. C. Precipitation modification by major urban areas. **Bulletin of the American Meteorological Society**, American Meteorological Society, v. 54, n. 12, p. 1220–1233, 1973.

JONES, N. How machine learning could help to improve climate forecasts. **Nature**, Nature Publishing Group, v. 548, n. 7668, 2017.

KASHINATH, K. et al. Physics-informed machine learning: case studies for weather and climate modelling. **Philosophical Transactions of the Royal Society A**, The Royal Society Publishing, v. 379, n. 2194, p. 20200093, 2021.

KOR, K.; ALTUN, G. Is support vector regression method suitable for predicting rate of penetration? **Journal of Petroleum Science and Engineering**, Elsevier, v. 194, p. 107542, 2020.

MARINHO, T. L. et al. Otimização de hiperparâmetros do xgboost utilizando meta-aprendizagem. Universidade Federal de Alagoas, 2021.

MEDEIROS, L. F. D.; VIEIRA, D. H. Bioclimatologia animal. **Instituto de Zootecnia, Universidade Federal Rural do Rio de Janeiro, Seropédica, RJ. 126p**, 1997.

MITCHELL, R.; FRANK, E. Accelerating the xgboost algorithm using gpu computing. **PeerJ Computer Science**, PeerJ Inc., v. 3, p. e127, 2017.

MUNIZ, F. G. L.; CARACRISTI, I. Análise da variação da temperatura e umidade no período de pré-estação chuvosa na cidade de sobral/ce. **Research, Society and Development**, v. 10, n. 17, p. e214101724780–e214101724780, 2021.

OLIVEIRA, C. C. d.; RUPP, R. F.; GHISI, E. Influência da umidade do ar no conforto térmico de usuários de edificações de escritórios em florianópolis/sc. **Ambiente Construído**, SciELO Brasil, v. 20, p. 7–21, 2020.

RAMIRES, T. G. et al. Validation of stepwise-based procedure in GAMLSS. **Journal of Data Science**, v. 19, n. 1, p. 96–110, 2021.

RIGBY, R. A.; STASINOPOULOS, D. M. Generalized additive models for location, scale and shape. **Journal of the Royal Statistical Society: Series C (Applied Statistics)**, Wiley Online Library, v. 54, n. 3, p. 507–554, 2005.

RIGBY, R. A.; STASINOPOULOS, M. D.; HELLER, G. Z.; BASTIANI, F. D. **Distributions for modeling location, scale, and shape: Using GAMLSS in R**. [S.l.]: CRC press, 2019.

SAID, R. A.; PINHEIRO, J. M. d. S. A inteligência artificial à serviço da administração. UBM-Centro Universitário de Barra Mansa, 2024.

STASINOPOULOS, M. D.; RIGBY, R. A.; HELLER, G. Z.; VOUDOURIS, V.; BASTIANI, F. D. **Flexible regression and smoothing: using GAMLSS in R**. [S.l.]: CRC Press, 2017.

TAYLOR, S. J.; LETHAM, B. Forecasting at scale. **The American Statistician**, Taylor & Francis, v. 72, n. 1, p. 37–45, 2018.

VAPNIK, V. **The nature of statistical learning theory**. [S.l.]: Springer science & business media, 1999.

VILLARINI, G.; SMITH, J. A.; NAPOLITANO, F. Nonstationary modeling of a long record of rainfall and temperature over rome. **Advances in Water Resources**, Elsevier, v. 33, n. 10, p. 1256–1267, 2010.

YAN, J. et al. A time-series classification approach based on change detection for rapid land cover mapping. **ISPRS Journal of Photogrammetry and Remote Sensing**, Elsevier, v. 158, p. 249–262, 2019.

ZHANG, D.-d.; YAN, D.-h.; WANG, Y.-C.; LU, F.; LIU, S.-h. Gamlss-based nonstationary modeling of extreme precipitation in beijing–tianjin–hebei region of china. **Natural Hazards**, Springer, v. 77, p. 1037–1053, 2015.

3 CONCLUSÃO

Os resultados dos dois artigos apresentados nesta dissertação indicam que os modelos de regressão distribucional são uma abordagem promissora para o ajuste e predição de dados meteorológicos. Os GAMLSS são capazes de capturar a complexidade da distribuição da resposta (temperatura média), o que pode levar a um melhor ajuste e predição do que modelos de regressão convencionais ou outros algoritmos de aprendizado de máquina.

No primeiro artigo, foi demonstrado que o GAMLSS com distribuição Box-Cox exponencial potência (BCPE) foi capaz de modelar a temperatura média diária em Florianópolis, SC. Modelos de regressão menos complexos não foram capazes de explicar completamente a resposta, devido às diferentes estruturas de regressão construídas na sua distribuição.

No segundo artigo, foi comparada a performance preditiva dos GAMLSS baseado na distribuição Box-Cox *t* (BCT), Box-Cox exponencial potência (BCPE), Box-Cox Cole-Green (BCCG) e Weibull com quatro outros algoritmos de aprendizado de máquina, sendo eles *random forest*, *support vector regression*, *extreme gradient boosting* e *prophet*. Os GAMLSS baseados na distribuição Box-Cox *t* apresentaram resultados mais satisfatórios na maioria das métricas utilizadas para a comparação dos modelos ajustados. Este resultado é um ponto chave deste trabalho, uma vez que usualmente se comparam algoritmos extremamente sofisticados de *machine learning* apenas com o modelo de regressão clássico de Gauss-Markov, que, frequentemente, não é o mais recomendado em estudos de dados mais complexos.

Os resultados desta dissertação sugerem que os GAMLSS podem ser uma alternativa interessante para pesquisadores e profissionais que trabalham com dados meteorológicos. Os GAMLSS são uma abordagem flexível e robusta que pode ser utilizada para ajustar e prever uma ampla gama de variáveis meteorológicas.

REFERÊNCIAS

AGHELPOUR, P.; MOHAMMADI, B.; BIAZAR, S. M. Long-term monthly average temperature forecasting in some climate types of iran, using the models sarima, svr, and svr-fa. **Theoretical and Applied Climatology**, Springer, v. 138, n. 3-4, p. 1471–1480, 2019.

AKAIKE, H. Information measures and model selection. **Int Stat Inst**, v. 44, p. 277–291, 1974.

ALMEIDA, T. A.; YAMAKAMI, A. Redução de dimensionalidade aplicada na classificação de spams usando filtros bayesianos. **Revista Brasileira de Computação Aplicada**, v. 3, n. 1, p. 16–29, 2011.

BASAK, A.; RAHMAN, A. S.; DAS, J.; HOSONO, T.; KISI, O. Drought forecasting using the prophet model in a semi-arid climate region of western india. **Hydrological Sciences Journal**, Taylor & Francis, v. 67, n. 9, p. 1397–1417, 2022.

BASHIR, T.; HAOYONG, C.; TAHIR, M. F.; LIQIANG, Z. Short term electricity load forecasting using hybrid prophet-lstm model optimized by bpnn. **Energy reports**, Elsevier, v. 8, p. 1678–1686, 2022.

BATURYNSKA, I.; MARTINSEN, K. Prediction of geometry deviations in additive manufactured parts: comparison of linear regression with machine learning algorithms. **Journal of Intelligent Manufacturing**, Springer, v. 32, p. 179–200, 2021.

BOX, G. E.; JENKINS, G. M.; REINSEL, G. C.; LJUNG, G. M. **Time series analysis: forecasting and control**. [S.l.]: John Wiley & Sons, 2015.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, p. 5–32, 2001.

BRIX, K. V.; TEAR, L.; DEFOREST, D. K.; ADAMS, W. J. Development of multiple linear regression models forpredicting chronic iron toxicity to aquatic organisms. **Environmental Toxicology and Chemistry**, Wiley Periodicals LLC, v. 42, n. 6, p. 1386–1400, 2023.

BUDHOLIYA, K.; SHRIVASTAVA, S. K.; SHARMA, V. An optimized xgboost based diagnostic system for effective prediction of heart disease. **Journal of King Saud University-Computer and Information Sciences**, Elsevier, v. 34, n. 7, p. 4514–4523, 2022.

BUUREN, S. V.; FREDRIKS, M. Worm plot: a simple diagnostic device for modelling growth reference curves. **Statistics in Medicine**, Wiley Online Library, v. 20, n. 8, p. 1259–1277, 2001.

CALLENS, A.; MORICHON, D.; ABADIE, S.; DELPEY, M.; LIQUET, B. Using random forest and gradient boosting trees to improve wave forecast at a specific location. **Applied Ocean Research**, Elsevier, v. 104, p. 102339, 2020.

CAZEIRO, V. V.; OLIVEIRA, A. L. de. Análise de dados de vendas utilizando séries temporais, algoritmo apriori e prophet. **Revista Ibero-Americana de Humanidades, Ciências e Educação**, v. 9, n. 3, p. 2053–2072, 2023.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. [S.l.: s.n.], 2016. p. 785–794.

COLE, T. J.; GREEN, P. J. Smoothing reference centile curves: the lms method and penalized likelihood. **Statistics in medicine**, Wiley Online Library, v. 11, n. 10, p. 1305–1319, 1992.

COSTA, C. H. Uma análise dos impactos de hiperparâmetros aplicados ao algoritmo kde para a geração de mapas de hotspots criminais. 2022.

COSTA, E. F. S.; TEIXEIRA, G. M. T.; FREIRE, F. A. M.; DIAS, J. F.; FRANSOZO, A. Effects of biological and environmental factors on the variability of paralonchurus brasiliensis (sciaenidae) density: An gamlss application. **Journal of Sea Research**, Elsevier, v. 183, p. 102203, 2022.

DASARI, S. K.; CHEDDAD, A.; ANDERSSON, P. Random forest surrogate models to support design space exploration in aerospace use-case. In: SPRINGER. **Artificial Intelligence Applications and Innovations: 15th IFIP WG 12.5 International Conference, AIAI 2019, Hersonissos, Crete, Greece, May 24–26, 2019, Proceedings 15**. [S.l.], 2019. p. 532–544.

DE BASTIANI, F.; RIGBY, R. A.; STASINOPOULOS, D. M.; CYSNEIROS, A. H. M. A.; URIBE-OPAZO, M. A. Gaussian markov random field spatial models in gamlss. **Journal of Applied Statistics**, Taylor & Francis, v. 45, n. 1, p. 168–186, 2018.

DENG, A. Time series cross validation: A theoretical result and finite sample performance. **Economics Letters**, Elsevier, p. 111369, 2023.

DIMITRIADOU, S.; NIKOLAKOPOULOS, K. G. Multiple linear regression models with limited data for the prediction of reference evapotranspiration of the peloponnese, greece. **Hydrology**, Multidisciplinary Digital Publishing Institute, v. 2022, n. 9, p. 124, 2022.

DUNN, P. K.; SMYTH, G. K. Randomized quantile residuals. **Journal of Computational and graphical statistics**, Taylor & Francis, v. 5, n. 3, p. 236–244, 1996.

EHLERS, R. S. Análise de séries temporais. **Laboratório de Estatística e Geoinformação. Universidade Federal do Paraná**, v. 1, p. 1–118, 2007.

FERRAZ, R. S.; CRUZ, F. d. C.; FERRAZ, R. S.; CORREIA, A. F.; SIMAS, E. F. de. Previsão multi-passos da velocidade do vento através de redes neurais artificiais. In: **Artigo cientifico. 12º Congresso Latino Americano sobre geração e transmissão de eletricidade CLAGTEE**. [S.l.: s.n.], 2017.

FIGUEIRA, C. V. Modelos de regressão logística. 2006.

FOX, D. G. Judging air quality model performance: a summary of the ams workshop on dispersion model performance, woods hole, mass., 8–11 september 1980. **Bulletin of the American Meteorological Society**, American Meteorological Society, v. 62, n. 5, p. 599–609, 1981.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). **The annals of statistics**, Institute of Mathematical Statistics, v. 28, n. 2, p. 337–407, 2000.

HASTIE, T.; TIBSHIRANI, R. **Generalized additive models**. London: Chapman and Hall, 1990.

HASTIE, T. et al. Random forests. **The elements of statistical learning: Data mining, inference, and prediction**, Springer, p. 587–604, 2009.

HAYKIN, S. Redes neurais: princípios e prática.[sl]: Bookman editora, 2007. **Citado**, v. 3, p. 28–30.

HE, C.; CHEN, F.; LONG, A.; LUO, C.; QIAO, C. Frequency analysis of snowmelt flood based on gamlss model in manas river basin, china. **Water**, Multidisciplinary Digital Publishing Institute, v. 2021, n. 13, p. 2007, 2021.

HE, Y.; CHEN, C.; LI, B.; ZHANG, Z. Prediction of near-surface air temperature in glacier regions using era5 data and the random forest regression method. **Remote Sensing Applications: Society and Environment**, Elsevier, v. 28, p. 100824, 2022.

HELLER, G. Z.; ROBLEDO, K. P.; MARSCHNER, I. C. Distributional regression in clinical trials: treatment effects on parameters other than the mean. **BMC Medical Research Methodology**, v. 22, p. 56, 2022.

HYNDMAN, R. J.; KOEHLER, A. B. Another look at measures of forecast accuracy. **International journal of forecasting**, Elsevier, v. 22, n. 4, p. 679–688, 2006.

JABBAR, H.; KHAN, R. Z. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). **Computer Science, Communication and Instrumentation Devices**, v. 70, p. 163–172, 2015.

JAIN, A.; NANDAKUMAR, K.; ROSS, A. Score normalization in multimodal biometric systems. **Pattern recognition**, Elsevier, v. 38, n. 12, p. 2270–2285, 2005.

LI, J. et al. Feature selection: A data perspective. **ACM computing surveys (CSUR)**, ACM New York, NY, USA, v. 50, n. 6, p. 1–45, 2017.

LIN, K.; LIN, Q.; ZHOU, C.; YAO, J. Time series prediction based on linear regression and svr. In: IEEE. **Third International Conference on Natural Computation (ICNC 2007)**. [S.l.], 2007. v. 1, p. 688–691.

LÓPEZ, O. A. M.; LÓPEZ, A. M.; CROSSA, J. Overfitting, model tuning, and evaluation of prediction performance. In: **Multivariate statistical machine learning methods for genomic prediction**. [S.l.]: Springer, 2022. p. 109–139.

LORENA, A. C.; CARVALHO, A. C. P. d. L. F. Introdução às máquinas de vetores suporte (support vector machines). 2003.

MA, X.; FANG, C.; JI, J. Prediction of outdoor air temperature and humidity using xgboost. In: IOP PUBLISHING. **IOP conference series: earth and environmental science**. [S.l.], 2020. v. 427, n. 1, p. 012013.

MAISELI, B. J. Optimum design of chamfer masks using symmetric mean absolute percentage error. **EURASIP Journal on Image and Video Processing**, SpringerOpen, v. 2019, n. 1, p. 1–15, 2019.

MARINHO, T. L. et al. Otimização de hiperparâmetros do xgboost utilizando meta-aprendizagem. Universidade Federal de Alagoas, 2021.

MARKOVICS, D.; MAYER, M. J. Comparison of machine learning methods for photovoltaic power forecasting based on numerical weather prediction. **Renewable and Sustainable Energy Reviews**, Elsevier, v. 161, p. 112364, 2022.

MITCHELL, T. M. Does machine learning really work? **AI magazine**, v. 18, n. 3, p. 11–11, 1997.

MORETTIN, P. A. **Econometria financeira: um curso em séries temporais financeiras**. [S.l.]: Editora Blucher, 2017.

MORETTIN, P. A.; TOLOI, C. M. d. C. Análise de séries temporais. 2022.

NAKAMURA, L. R. et al. Cattle reference growth curves based on centile estimation: A gamlss approach. **Computers and Electronics in Agriculture**, v. 192, p. 106572, 2022.

NAKAMURA, L. R. et al. Modelling location, scale and shape parameters of the Birnbaum-Saunders generalized t distribution. **Journal of Data Science**, v. 15, p. 221–238, 2017.

NELDER, J. A.; WEDDERBURN, R. W. Generalized linear models. **Journal of the Royal Statistical Society: Series A (General)**, Wiley Online Library, v. 135, n. 3, p. 370–384, 1972.

NOBRE, J.; NEVES, R. F. Combining principal component analysis, discrete wavelet transform and xgboost to trade in the financial markets. **Expert Systems with Applications**, Elsevier, v. 125, p. 181–194, 2019.

OLIVEIRA, T. A. et al. An application of generalized additive models of location, scale, and shape (GAMLSS) to estimate the eucalyptus height. **Ciência e Natura**, v. 42, p. 1–10, 2019.

PANG, B.; YUE, J.; ZHAO, G.; XU, Z. et al. Statistical downscaling of temperature with the random forest model. **Advances in Meteorology**, Hindawi, v. 2017, 2017.

PAULA, G. A. **Modelos de regressão: com apoio computacional**. [S.l.]: IME-USP São Paulo, 2004.

PRIYADARSHINI, I.; COTTON, C. A novel lstm–cnn–grid search-based deep neural network for sentiment analysis. **The Journal of Supercomputing**, Springer, v. 77, n. 12, p. 13911–13932, 2021.

PROBST, P.; WRIGHT, M. N.; BOULESTEIX, A.-L. Hyperparameters and tuning strategies for random forest. **Wiley Interdisciplinary Reviews: data mining and knowledge discovery**, Wiley Online Library, v. 9, n. 3, p. e1301, 2019.

QIN, C. et al. Xgboost optimized by adaptive particle swarm optimization for credit scoring. **Mathematical Problems in Engineering**, Hindawi Limited, v. 2021, p. 1–18, 2021.

R Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2023. Disponível em: <https://www.R-project.org/>.

RAMIRES, T. G. et al. Validation of stepwise-based procedure in gamlss. **Journal of Data Science**, v. 19, p. 96–110, 2021.

RAMIRES, T. G. et al. Predicting survival function and identifying associated factors in patients with renal insufficiency in the metropolitan area of maringá, paraná state, brazil. **Cadernos de Saúde Pública**, v. 34, n. 1, p. e00075517, 2018.

RAMIRES, T. G. et al. A new semiparametric weibull cure rate model: fitting different behaviors within gamlss. **Journal of Applied Statistics**, Taylor & Francis, v. 46, n. 15, p. 2744–2760, 2019.

RATH, S.; TRIPATHY, A.; TRIPATHY, A. R. Prediction of new active cases of coronavirus disease (covid-19) pandemic using multiple linear regression model. **Diabetes & Metabolic Syndrome: Clinical Research & Reviews**, Elsevier, v. 14, n. 5, p. 1467–1474, 2020.

RIGBY, R. A.; STASINOPOULOS, D. A semi-parametric additive model for variance heterogeneity. **Statistics and Computing**, Springer, v. 6, p. 57–65, 1996.

RIGBY, R. A.; STASINOPOULOS, D. M. Generalized additive models for location, scale and shape. **Journal of the Royal Statistical Society: Series C (Applied Statistics)**, Wiley Online Library, v. 54, n. 3, p. 507–554, 2005.

RIGBY, R. A.; STASINOPOULOS, M. D.; HELLER, G. Z.; BASTIANI, F. D. **Distributions for modeling location, scale, and shape: Using GAMLSS in R**. [S.l.]: CRC press, 2019.

RIGHETTO, A. J. et al. Predicting weed invasion in a sugarcane cultivar using multispectral image. **Journal of Applied Statistics**, Taylor & Francis, v. 46, n. 1, p. 1–12, 2019.

RODGERS, J. L.; NICEWANDER, W. A. Thirteen ways to look at the correlation coefficient. **The American Statistician**, Taylor & Francis, v. 42, n. 1, p. 59–66, 1988.

ROQUIM, F. V. et al. Building flexible regression models: including the birnbaum-saunders distribution in the gamlss package. **Semina: Exact and Technological Sciences**, v. 42, n. 2, p. 163–168, 2021.

RUEZZENE, C. B.; MIRANDA, R. B. de; TECH, A. R. B.; MAUAD, F. F. Preenchimento de falhas em dados de precipitação através de métodos tradionais e por inteligência artificial. **Revista Brasileira de Climatologia**, v. 29, p. 177–204, 2021.

SÁFADI, T. Uso de séries temporais na análise de vazão de água na represa de furnas. **Ciência e Agrotecnologia**, SciELO Brasil, v. 28, p. 142–148, 2004.

SANTOS, H. G. d. et al. Machine learning para análises preditivas em saúde: exemplo de aplicação para predizer óbito em idosos de são paulo, brasil. **Cadernos de Saúde Pública**, SciELO Public Health, v. 35, p. e00050818, 2019.

SILVA, F. E. M. d.; OLIVEIRA, L. M. d.; ANTUNES, F. L. M.; JUNIOR, E. M. S. Previsão de geração de energia elétrica renovável em curto prazo no estado do ceará utilizando modelo de regressão prophet. Research, Society and Development, 2022.

SILVA, R.; NETO, D. R. d. S. Inteligência artificial e previsão de óbito por covid-19 no brasil: uma análise comparativa entre os algoritmos logistic regression, decision tree e random forest. **Saúde em Debate**, SciELO Public Health, v. 46, p. 118–129, 2023.

SILVA, V. C.; NAKAMURA, L. R.; RAMIRES, T. G.; PEREIRA, G. M. C. Análise da temperatura de Florianópolis (SC) utilizando uma abordagem GAMLSS. **Sigmae**, v. 12, n. 1, p. 129–138, 2023.

SONG, X.; LIU, X.; LIU, F.; WANG, C. Comparison of machine learning and logistic regression models in predicting acute kidney injury: A systematic review and meta-analysis. **International Journal of Medical Informatics**, Elsevier, v. 151, p. 104484, 2021.

SOUSA, G. C. de; ALVES, J. M. S. A regressão linear de galton: atividades históricas para função afim e estatística básica usando planilhas eletrônicas. **Conexões-Ciência e Tecnologia**, v. 9, n. 4, p. 26–36, 2016.

STASINOPOULOS, M.; RIGBY, R.; VOUDOURIS, V.; HELLER, G.; BASTIANI, F. D. Flexible regression and smoothing: The gamlss packages in r. **GAMLSS for Statistical Modelling. GAMLSS for Statistical Modeling**, 2015.

STASINOPOULOS, M. D.; RIGBY, R. A.; HELLER, G. Z.; VOUDOURIS, V.; BASTIANI, F. D. **Flexible regression and smoothing: using GAMLSS in R**. [S.l.]: CRC Press, 2017.

SUN, Y.; DING, S.; ZHANG, Z.; JIA, W. An improved grid search algorithm to optimize svr for prediction. **Soft Computing**, Springer, v. 25, p. 5633–5644, 2021.

TAYLOR, S. J.; LETHAM, B. Forecasting at scale. **The American Statistician**, Taylor & Francis, v. 72, n. 1, p. 37–45, 2018.

THOTTAKKARA, P. et al. Application of machine learning techniques to high-dimensional clinical data to forecast postoperative complications. **PLoS ONE**, v. 11, n. 5, p. e0155705, 2016.

TIMMERMAN, M. E.; VONCKEN, L.; ALBERS, C. J. A tutorial on regression-based norming of psychological tests with gamlss. **Psychological Methods**, v. 26, n. 3, p. 357–373, 2021.

TURKMAN, M. A. A.; SILVA, G. L. Modelos lineares generalizados-da teoria à prática. **Sociedade Portuguesa de Estatística, Lisboa**, 2000.

TURNER, C. R.; FUGGETTA, A.; LAVAZZA, L.; WOLF, A. L. A conceptual basis for feature engineering. **Journal of Systems and Software**, Elsevier, v. 49, n. 1, p. 3–15, 1999.

UÇAR, M. K.; NOUR, M.; SINDI, H.; POLAT, K. et al. The effect of training and testing process on machine learning in biomedical datasets. **Mathematical Problems in Engineering**, Hindawi, v. 2020, 2020.

VAPNIK, V. **The nature of statistical learning theory**. [S.l.]: Springer science & business media, 1999.

VIEIRA, L. A. **Redução de uso de agrotóxicos por meio de inteligência artificial**. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2021. Disponível em: <https://repositorio.utfpr.edu.br/jspui/bitstream/1/26112/1/reducaoagrotoxicosinteligenciaartificial.pdf>.

WOOD, S. N. **Generalized Additive Models: an introduction with R**. Boca Raton: CRC Press, 2017.

XU, H.; WANG, H.; LIANG, J. et al. Support vector machine regress algorithm and its application. **J Beijing Inst Petrochem Technol**, v. 1, p. 66–70, 2010.

ZHANG, J.; SONG, W.; JIANG, B.; LI, M. Measurement of lumber moisture content based on pca and gs-svm. **Journal of forestry research**, Springer, v. 29, n. 2, p. 557–564, 2018.

APÊNDICE A - Códigos

Estão apresentados neste apêndice os códigos em R utilizados neste trabalho, também disponíveis no repositório <https://github.com/Viviane-Costas/GAMLSS-e-Machine-Learning-.git>.

Artigo 1

```
# ARTIGO 1
#-----------------------------------------------------

## Pacotes necessários
library(gamlss)
library(e1071)

## Diretório do R
setwd("<PATH>")

##  Banco de Dados
dados <- read.csv("Dados - Flori.csv", sep = ";" , dec = "," ,header =
 ↪  TRUE)
names(dados)
head(dados)

## Descritiva dos dados
summary(dados)
skewness(Temperatura)
kurtosis(Temperatura)
var(Temperatura)
sd(Temperatura)

## Gráficos descritivos

DDD <- par(mfrow=c(2,2))
plot(Temperatura~Data.Medicao, data = dados, col= 4,pch = 17, cex = 0.5)
plot(Temperatura~Precipitacao.total,
     data = dados, col= 4,pch = 17, cex = 0.5)
plot(Temperatura~Pressao.Atmosferica, data = dados, col=4,pch = 17, cex =
 ↪  0.5)
plot(Temperatura~Temperatura.Orvalho, data = dados, col=4,pch = 17, cex =
 ↪  0.5)
plot(Temperatura~Umidade, data = dados, col=4,pch = 17, cex = 0.5)
plot(Temperatura~Velocidade.Vento, data = dados, col=4,pch = 17, cex =
 ↪  0.5)
par(DDD)
par(mfrow=c(1,2))
hist(dados$Temperatura,main="Temperatura Florianópolis - 06/2021 -
 ↪  06/2022",
```

```
xlab="Temperatura",prob=TRUE,ylim=c(0,0.12), col = 4 );
lines(density(dados$Temperatura,na.rm=TRUE),col=1)
boxplot(dados$Temperatura,horizontal=FALSE, col = 4, ylab =
 ↪  "Temperatura")
plot(Temperatura, col = 4 )


## _____________________________________________________

## Análise do modelo Gamlss

# _________________
# Com suavização
# _________________

Model_BCPEo <- gamlss(Temperatura~1,data = dados,
method = mixed(20,100), family = BCPEo) # modelo nulo
Model_BCPEo <- stepGAICAll.A(Model_BCPEo, scope = list(lower=~1,
upper = ~ pb(Data.Medicao) + pb(Precipitacao.total) +
 ↪  pb(Pressao.Atmosferica) + pb(Temperatura.Orvalho) + pb(Umidade) +
pb(Velocidade.Vento)))
summary(Model_BCPEo)
term.plot(Model_BCPEo,pages = 1, ask = FALSE, ylim = "free")



r2 <- gamlss( Temperatura ~ pb(Temperatura.Orvalho) +
                pb(Data.Medicao) + pb(Precipitacao.total) +
                 ↪  pb(Pressao.Atmosferica),
              sigma.formula = ~pb(Temperatura.Orvalho) +
               ↪  pb(Velocidade.Vento),
              nu.formula = ~pb(Precipitacao.total) + pb(Data.Medicao),
              family = BCPEo, data = dados, method = mixed(20,
                                                           100), trace =
                                                            ↪  FALSE,
                                                            ↪  tau.formula
                                                            ↪  = ~1) #
                                                            ↪  familia
                                                            ↪  BCPEo
summary(r2)

# Comportamento da função de suavização para mu
term.plot(r2, pages=1, ask = FALSE, ylim = "free")

# Comportamento da função de suavização para sigma
term.plot(r2, what="sigma", pages = 1, ask = FALSE, ylim = "free")

# Comportamento da função de suavização para nu
```

```
term.plot(r2, what='nu',pages = 1, ask = FALSE, ylim = "free")

# Comportamento da função de suavização para tau
term.plot(r2, what='tau',pages = 1, ask = FALSE, ylim = "free")

# Análise de Residuo
plot(r2)


# Análise de residuo - worm plot
wp(r2, ylim.all = 1.5) ; title("Worm plot - BCPEo")

#_____

# Retirando a suavização das variáveis que apresentaram
 ↪  comportamento linear


r22 <- gamlss(Temperatura ~ Temperatura.Orvalho +
                pb(Data.Medicao) + pb(Precipitacao.total) +
                 ↪  pb(Pressao.Atmosferica),
              sigma.formula = ~Temperatura.Orvalho + Velocidade.Vento,
              nu.formula = ~Precipitacao.total + Data.Medicao,
              family = BCPEo, data = dados, method = mixed(20,
                                                           100), trace =
                                                            ↪  FALSE,
                                                            ↪  tau.formula
                                                            ↪  = ~1) #
                                                            ↪  familia
                                                            ↪  BCPEo
summary(r22)

# Comportamento da função de suavização para mu
term.plot(r22, pages=1, ask = FALSE, ylim = "free")

# Comportamento da função de suavização para sigma
term.plot(r22, what="sigma", pages = 1, ask = FALSE, ylim = "free")

# Comportamento da função de suavização para nu
term.plot(r22, what='nu',pages = 1, ask = FALSE, ylim = "free")

# Comportamento da função de suavização para tau
term.plot(r22, what='tau',pages = 1, ask = FALSE, ylim = "free")


# Analise de Residuo
plot(r22)
```

```
# Análise de resíduos - worm plot
wp(r22, ylim.all = 1.5) ; title("Worm plot - BCPEo")


# Fim.
```

Artigo 2

```
# ARTIGO 2
#-------------------------------------------------
## Análise GAMLSS
## Florianopólis 10 anos

## Pacotes necessários
library(gamlss)
library(e1071)
library(gamlss.foreach)
library(zoo)
library(tictoc)
library(forecast)

## Diretório do R
setwd("<PATH>")

## Banco de Dados
dados <- readxl::read_xlsx("Atualizada.xlsx")
names(dados)
head(dados)
summary(dados)
colnames(dados)<-c("Data_Medicao", "PRECIPITACAO", "PRESSAO",
 ↪  "TEMP_ORV",     "TEMP_MEDIA",   "UMIDADE",      "VENT_MED")
dados$Data_Medicao <- as.Date(dados$Data_Medicao)

## Calculando estatísticas
skewness(dados$TEMP_MEDIA)
kurtosis(dados$TEMP_MEDIA)
var(dados$TEMP_MEDIA)
sd(dados$TEMP_MEDIA)


## Relação das covariáveis com a resposta

plot(TEMP_MEDIA ~ Data_Medicao, data = dados, type = "l", xlab = "Tempo
 ↪  (Anos)", ylab = "Temperatura (°C)", col = 1, lty = 1)
plot(TEMP_MEDIA~PRECIPITACAO, data = dados, xlab = "Precipitação (mm)",
 ↪  ylab = "Temperatura (°C)", col= 1,pch = 16, cex = 0.5)
```

```
plot(TEMP_MEDIA~PRESSAO, data = dados,  xlab = "Pressão Atmosférica (mB)",
 ↪  ylab = "Temperatura (°C)", col=1,pch = 16, cex = 0.5)
plot(TEMP_MEDIA~TEMP_ORV, data = dados, xlab = "Temperatura de Orvalho
 ↪  (°C)", ylab = "Temperatura (°C)", col=1,pch = 16, cex = 0.5)
plot(TEMP_MEDIA~UMIDADE, data = dados, xlab = "Umidade (%)", ylab =
 ↪  "Temperatura (°C)", col=1,pch = 16, cex = 0.5)
plot(TEMP_MEDIA ~ VENT_MED, data = dados, xlab = expression("Velocidade do
 ↪  vento (m.s"^{-1} * ")"), ylab = "Temperatura (°C)", col = 1, pch = 16,
 ↪  cex = 0.5)

## Histograma e BloxPlot

hist(dados$TEMP_MEDIA, main="", xlab="Temperatura (°C)",
 ↪  ylab="Frequência", col="gray")
lines(density(dados$TEMP_MEDIA,na.rm=TRUE),col=1)
boxplot(dados$TEMP_MEDIA,horizontal=FALSE, col = "gray", ylab =
 ↪  "Temperatura (°C)")


# Calcular estatísticas do boxplot
boxplot_stats <- boxplot.stats(dados$TEMP_MEDIA)
boxplot_stats

# Obtendo os limites superior e inferior
lower_limit <- boxplot_stats$out[1]
upper_limit <- boxplot_stats$out[length(boxplot_stats$out)]

# Identificando outliers
outliers <- dados$TEMP_MEDIA[dados$TEMP_MEDIA < upper_limit ]
outliers

##
 ↪  ____________________________________________________________________________

## Divisão dos dados - Treinamento e Teste

seed(123)
fim<-round(length(dados$Data_Medicao)*0.7) # 70% dos dados
training <- dados[1:fim,]
dim(training)
test <- dados[(fim+1):length(dados$Data_Medicao),]
dim(test)

y2<-dados$Data_Medicao[length(dados$Data_Medicao)]
y1<-dados$Data_Medicao[fim+1]

# Posição das legendas
x2<-dados$Data_Medicao[fim+500]
```

```
x1<-dados$Data_Medicao[1+500]


f1 <- fitDist(training$TEMP_MEDIA, type = 'realplus')
f1$fits[1:10]

## Análise dos modelos Gamlss - (BCTo, BCPEo, BCCGo e WEI3)

# _____________________________
# Com funções de  suavização
# _____________________________

Model_BCTo <- gamlss(TEMP_MEDIA~1,data = training, method = mixed(20,100),
 ↪  family = BCTo) # modelo nulo
Model_BCTo <- stepGAICAll.A(Model_BCTo, scope = list(lower=~1,
                                                     upper = ~
                                                      ↪  pb(Data_Medicao)
                                                      ↪  +
                                                      ↪  pb(PRECIPITACAO)
                                                      ↪  + pb(PRESSAO) +
                                                       pb(TEMP_ORV) +
                                                        ↪  pb(UMIDADE) +
                                                        ↪  pb(VENT_MED)))
summary(Model_BCTo)
term.plot(Model_BCTo,pages =1, ask=FALSE, ylim = "free")


Model_BCPEo <- gamlss(TEMP_MEDIA~1,data = training, method =
 ↪  mixed(20,100), family = BCPEo) # modelo nulo
Model_BCPEo <- stepGAICAll.A(Model_BCPEo, scope = list(lower=~1,
                                                       upper = ~
                                                        ↪  pb(Data_Medicao)
                                                        ↪  +
                                                        ↪  pb(PRECIPITACAO)
                                                        ↪  + pb(PRESSAO) +
                                                         pb(TEMP_ORV) +
                                                          ↪  pb(UMIDADE) +
                                                          ↪  pb(VENT_MED)))
summary(Model_BCPEo)
term.plot(Model_BCPEo,pages = 1, ask = FALSE, ylim = "free")

Model_BCCGo <- gamlss(TEMP_MEDIA~1,data = training, method =
 ↪  mixed(20,100), family = BCCGo) # modelo nulo
Model_BCCGo <- stepGAICAll.A(Model_BCCGo, scope = list(lower=~1,
```

```
                                                    upper = ~
                                                     ↪  pb(Data_Medicao)
                                                     ↪  +
                                                     ↪  pb(PRECIPITACAO)
                                                     ↪  + pb(PRESSAO) +
                                                      pb(TEMP_ORV) +
                                                       ↪  pb(UMIDADE) +
                                                       ↪  pb(VENT_MED)))
summary(Model_BCCGo)
term.plot(Model_BCCGo,pages = 1,ask = FALSE, ylim = "free")


Model_WEI3 <- gamlss(TEMP_MEDIA~1,data =training, method = mixed(20,100),
 ↪  family = WEI3) # modelo nulo
Model_WEI3 <- stepGAICAll.A(Model_WEI3, scope = list(lower=~1,
                                                     upper = ~
                                                      ↪  pb(Data_Medicao) +
                                                      ↪  pb(PRECIPITACAO) +
                                                      ↪  pb(PRESSAO) +
                                                       pb(TEMP_ORV) +
                                                        ↪  pb(UMIDADE) +
                                                        ↪  pb(VENT_MED)))
summary(Model_WEI3)
term.plot(Model_WEI3,pages = 1,ask = FALSE, ylim = "free")


r1 <- gamlss(TEMP_MEDIA ~ pb(TEMP_ORV) + pb(UMIDADE) + pb(VENT_MED) +
               pb(PRECIPITACAO) + pb(PRESSAO), sigma.formula =
                ↪  ~pb(TEMP_ORV) +
               pb(PRESSAO) + pb(VENT_MED), nu.formula = ~pb(TEMP_ORV) +
               pb(PRESSAO), tau.formula = ~pb(TEMP_ORV) +
                ↪  pb(PRECIPITACAO),
             family = BCTo, data = training, method = mixed(20, 100)) #
              ↪  familia BCTo
summary(r1)

r2 <- gamlss(TEMP_MEDIA ~ pb(TEMP_ORV) + pb(UMIDADE) + pb(VENT_MED) +
               pb(PRECIPITACAO) + pb(PRESSAO), sigma.formula =
                ↪  ~pb(TEMP_ORV) +
               pb(PRESSAO) + pb(UMIDADE) + pb(VENT_MED) +
                ↪  pb(PRECIPITACAO),
             nu.formula = ~pb(VENT_MED), tau.formula = ~pb(VENT_MED) +
               pb(PRESSAO) + pb(PRECIPITACAO), family = BCPEo, data =
                ↪  training,
             method = mixed(20, 100)) # familia BCPEo
summary(r2)

r3 <- gamlss(TEMP_MEDIA ~ pb(TEMP_ORV) + pb(PRECIPITACAO) +
```

```
                pb(VENT_MED) + pb(PRESSAO), sigma.formula = ~pb(TEMP_ORV) +
                pb(PRESSAO) + pb(VENT_MED) + pb(PRECIPITACAO), nu.formula =
                 ↪  ~pb(VENT_MED),
              family = BCCGo, data = training, method = mixed(20, 100)) #
               ↪  familia BCCGo

r4 <- gamlss(TEMP_MEDIA ~ pb(TEMP_ORV) + pb(UMIDADE) + pb(PRESSAO) +
                pb(VENT_MED) + pb(PRECIPITACAO), sigma.formula =
                 ↪  ~pb(UMIDADE) +
                pb(TEMP_ORV) + pb(PRESSAO) + pb(VENT_MED), family = WEI3,
              data = training, method = mixed(20, 100)) # familia WEI3

summary(r4)



# Comparação entre os modelos ajustados (Critério de Informaçao
 ↪  de Akaike)

AIC(r1,r2,r3,r4)


# Comportamento das funções de suavização para mu

term.plot(r1, pages =1, ask = FALSE, ylim = "free")
term.plot(r2, pages=1, ask = FALSE, ylim = "free")
term.plot(r3, pages=1, ask = FALSE, ylim = "free")
term.plot(r4, pages=1, ask = FALSE, ylim = "free")

# Comportamento das funções de suavização para sigma

term.plot(r1, what="sigma",pages = 1, ask = FALSE, ylim = "free")
term.plot(r2, what="sigma", pages = 1, ask = FALSE, ylim = "free")
term.plot(r3, what="sigma", pages = 1, ask = FALSE, ylim = "free")
term.plot(r4, what="sigma", pages = 1, ask = FALSE, ylim = "free")


# Comportamento das funções de suavização para v

term.plot(r1, what='nu',pages = 1, ask = FALSE, ylim = "free")
term.plot(r2, what='nu',pages = 1, ask = FALSE, ylim = "free")
term.plot(r3, what='nu',pages = 1, ask = FALSE, ylim = "free")


# Comportamento das funções de suavização para t

term.plot(r1, what='tau',pages = 1, ask = FALSE, ylim = "free")
term.plot(r2, what='tau',pages = 1, ask = FALSE, ylim = "free")
```

```
# Análise de Resíduos

plot(r1,ts = TRUE)
plot(r2,ts = TRUE)
plot(r3,ts = TRUE)
plot(r4,ts = TRUE)


# Análise de Resísuos - worm plot

wp(r1, ylim.all = 1.5) ; title("Worm plot - BCTo")
wp(r2, ylim.all = 1.5) ; title("Worm plot - BCPEo")
wp(r3, ylim.all = 1.5) ; title("Worm plot - BCCGo")
wp(r4, ylim.all = 1.5) ; title("Worm plot - WEI3")

## SERIES TEMPORAIS - GAMLSS ##
## 5 Ad-hoc estimation: The spreads again

r1$mu.df
r1$sigma.df
r1$nu.df
r1$tau.df

r2$mu.df
r2$sigma.df
r2$nu.df
r2$tau.df

r3$mu.df
r3$sigma.df
r3$nu.df
r3$tau.df

r4$mu.df
r4$sigma.df
r4$nu.df
r4$tau.df

## Obtendo resíduos de mu

mures_r1 <-residuals(r1, what="mu")
mures_r2 <-residuals(r2, what="mu")
mures_r3 <-residuals(r3, what="mu")
mures_r4 <-residuals(r4, what="mu")
```

```
## Ajustando um modelo ARIMA para mu

a1 <- auto.arima(mures_r1)
a1

plot(ts(residuals(r1, what="mu")))
lines(fitted(a1), col="red")

a2 <- auto.arima(mures_r2)
a2

plot(ts(residuals(r2, what="mu")))
lines(fitted(a2), col="red")

a3 <- auto.arima(mures_r3)
a3

plot(ts(residuals(r3, what="mu")))
lines(fitted(a3), col="red")

a4 <- auto.arima(mures_r4)
a4

plot(ts(residuals(r4, what="mu")))
lines(fitted(a4), col="red")

## pegue os valores ajustados deste modelo e desloque-os no
 ↪  modelo

r1_1 <-  gamlss(TEMP_MEDIA ~ offset(fitted(a1)) + pb(TEMP_ORV) +
 ↪  pb(UMIDADE) + pb(VENT_MED) +
                  pb(PRECIPITACAO) + pb(PRESSAO), sigma.formula =
                   ↪  ~pb(TEMP_ORV) +
                  pb(PRESSAO) + pb(VENT_MED), nu.formula = ~pb(TEMP_ORV) +
                  pb(PRESSAO), tau.formula = ~pb(TEMP_ORV) +
                   ↪  pb(PRECIPITACAO),
                family = BCTo, data = training, method = mixed(10, 100)) #
                 ↪  familia BCTo

plot(r1_1,ts = TRUE)

r22_1 <-  gamlss(TEMP_MEDIA ~ offset(fitted(a2)) + pb(TEMP_ORV) +
 ↪  pb(UMIDADE) + pb(VENT_MED) +
                   pb(PRECIPITACAO) + pb(PRESSAO), sigma.formula =
                    ↪  ~pb(TEMP_ORV) +
                   pb(PRESSAO) + pb(UMIDADE) + pb(VENT_MED) +
                    ↪  pb(PRECIPITACAO),
                 nu.formula = ~pb(VENT_MED), tau.formula = ~pb(VENT_MED) +
```

```
                    pb(PRESSAO) + pb(PRECIPITACAO), family = BCPEo, data =
                     ↪  training,
                   method = mixed(20, 100)) # familia BCPEo

plot(r22_1,ts = TRUE)

r33_1 <-  gamlss(TEMP_MEDIA ~ offset(fitted(a3)) + TEMP_ORV +
 ↪  pb(PRECIPITACAO) +
                    pb(VENT_MED) + pb(PRESSAO), sigma.formula =
                     ↪  ~pb(TEMP_ORV) +
                    pb(PRESSAO) + pb(VENT_MED) + pb(PRECIPITACAO),
                     ↪  nu.formula = ~pb(VENT_MED),
                   family = BCCGo, data = training, method = mixed(10, 200))
                    ↪  # familia BCCGo

plot(r33_1,ts = TRUE)


r44_1 <-  gamlss(TEMP_MEDIA ~ offset(fitted(a4)) + TEMP_ORV + UMIDADE +
 ↪  pb(PRESSAO) +
                    pb(VENT_MED) + pb(PRECIPITACAO), sigma.formula =
                     ↪  ~pb(UMIDADE) +
                    TEMP_ORV + pb(PRESSAO) + pb(VENT_MED), family = WEI3,
                   data = training, method = mixed(20, 100)) # familia
                    ↪  WEI3


plot(r44_1,ts = TRUE)

# Função para calcular métricas dos modelos

calculate_metrics <- function(predictions, actual_values) {
  rmse <- sqrt(mean((predictions - actual_values)^2))
  mae <- mean(abs(predictions - actual_values))
  mase <- mean(abs(predictions - actual_values) /
   ↪  mean(abs(diff(training$TEMP_MEDIA))))
  smape <- 2 * mean(abs(predictions - actual_values) / (abs(predictions) +
   ↪  abs(actual_values)))
  srq <- 1 - sum((predictions - actual_values)^2) / sum((actual_values -
   ↪  mean(actual_values))^2)
  mape <- mean(abs((actual_values - predictions) / actual_values)) * 100

  return(c(RMSE = rmse, MAE = mae, MASE = mase, SMAPE = smape, SRQ = srq,
   ↪  MAPE = mape))
}

# Modelos
models <- list(r1_1, r22_1, r33_1, r44_1)
```

```
# Loop através dos modelos
for (i in 1:length(models)) {
  model <- models[[i]]

  # Fazer previsões no conjunto de treinamento
  train_predictions <- predict(model, type = "response")

  # Valores reais no conjunto de treinamento
  train_actual_values <- training$TEMP_MEDIA

  # Calcular e imprimir as métricas
  metrics <- calculate_metrics(train_predictions, train_actual_values)
  cat(paste("Métricas para o Modelo ", i, ":\n"))
  print(metrics)
  cat("\n")
}


## Aplicando o modelo que obteve as melhores métricas no
 ↪  conjunto de treinamento
## no conjunto teste (BCTo)

R1_teste <-  gamlss(TEMP_MEDIA ~  pb(TEMP_ORV) + pb(UMIDADE) +
 ↪  pb(VENT_MED) +
                  pb(PRECIPITACAO) + pb(PRESSAO), sigma.formula =
                   ↪  ~pb(TEMP_ORV) +
                  pb(PRESSAO) + pb(VENT_MED), nu.formula = ~pb(TEMP_ORV) +
                  pb(PRESSAO), tau.formula = ~pb(TEMP_ORV) +
                   ↪  pb(PRECIPITACAO),
                family = BCTo, data = test) # familia BCTo

summary(R1_teste)
plot(R1_teste ,ts = TRUE)

## Obtendo resíduos de mu
mures_R1_teste <-residuals(R1_teste, what="mu")

a1_teste <- auto.arima(mures_R1_teste)
a1_teste

plot(ts(residuals(R1_teste, what="mu")))
lines(fitted(a1_teste), col="red")


R1_teste_1 <-  gamlss(TEMP_MEDIA ~ offset(fitted(a1_teste)) + pb(TEMP_ORV)
 ↪  + pb(UMIDADE) + pb(VENT_MED) +
```

```
                    pb(PRECIPITACAO) + pb(PRESSAO), sigma.formula =
                     ↪  ~pb(TEMP_ORV) +
                    pb(PRESSAO) + pb(VENT_MED), nu.formula =
                     ↪  ~pb(TEMP_ORV) +
                    pb(PRESSAO), tau.formula = ~pb(TEMP_ORV) +
                     ↪  pb(PRECIPITACAO),
                  family = BCTo, data = test) # familia BCTo

summary(R1_teste_1)
plot(R1_teste_1 ,ts = TRUE)

# Prevendo valores com o modelo R1_teste
predictions_R1_teste_1 <- fitted(R1_teste_1)

# Calculando RMSE, MAE, MAPE, MSE, R², MASE e SMAPE
rmse_R1_teste_1 <- sqrt(mean((test$TEMP_MEDIA -
 ↪  predictions_R1_teste_1)^2))
mae_R1_teste_1 <- mean(abs(test$TEMP_MEDIA - predictions_R1_teste_1))
mape_R1_teste_1 <- mean(abs((test$TEMP_MEDIA - predictions_R1_teste_1) /
 ↪  test$TEMP_MEDIA)) * 100
mse_R1_teste_1 <- mean((test$TEMP_MEDIA - predictions_R1_teste_1)^2)

# Calculando MASE
mae_mean <- mean(abs(test$TEMP_MEDIA - mean(test$TEMP_MEDIA)))
mase_R1_teste_1 <- mae_R1_teste_1 / mae_mean

# Calculando SMAPE
smape_R1_teste_1 <- mean(2 * abs(predictions_R1_teste_1 - test$TEMP_MEDIA)
 ↪  / (abs(predictions_R1_teste_1) + abs(test$TEMP_MEDIA))) * 100

# Calculando R²
SSE_R1_teste_1 <- sum((test$TEMP_MEDIA - predictions_R1_teste_1)^2)
SST_R1_teste_1 <- sum((test$TEMP_MEDIA - mean(test$TEMP_MEDIA))^2)
r_squared_R1_teste_1 <- 1 - SSE_R1_teste_1 / SST_R1_teste_1

# Criando um data frame com as métricas
metrics_R1_teste_1 <- data.frame(
  Model = "R1_teste",
  RMSE = rmse_R1_teste_1,
  MAE = mae_R1_teste_1,
  MAPE = mape_R1_teste_1,
  MSE = mse_R1_teste_1,
  MASE = mase_R1_teste_1,
  SMAPE = smape_R1_teste_1,
  R_squared = r_squared_R1_teste_1
)

metrics_R1_teste_1
```

```
# Comportamento das funções de suavização  para mu

term.plot(R1_teste_1, what="mu",pages = 1, ask = FALSE, ylim = "free")
term.plot(R1_teste_1,what="mu", terms =1, ask = FALSE, ylim = "free",
 ↪  ylabs = expression(log(mu)), xlabs = "Temperatura de Orvalho (°C)")
term.plot(R1_teste_1,what="mu", terms =2, ask = FALSE, ylim = "free",
 ↪  ylabs = expression(log(mu)), xlabs = "Umidade (%)")
term.plot(R1_teste_1,what="mu", terms =3, ask = FALSE, ylim = "free",
 ↪  ylabs = expression(log(mu)), xlabs = expression("Velocidade do vento
 ↪  (m.s"^{-1} * ")"))
term.plot(R1_teste_1,what="mu", terms =4, ask = FALSE, ylim = "free",
 ↪  ylabs = expression(log(mu)), xlabs = "Precipitação (mm)")
term.plot(R1_teste_1,what="mu", terms =5, ask = FALSE, ylim = "free",
 ↪  ylabs = expression(log(mu)), xlabs = "Pressão Atmosférica (mB)")

# Comportamento das funções de suavização para sigma

term.plot(R1_teste_1, what="sigma",pages = 1, ask = FALSE, ylim = "free")
term.plot(R1_teste_1,what="sigma", terms =1, ask = FALSE, ylim = "free",
 ↪  ylabs = expression(log(sigma)), xlabs = "Temperatura de Orvalho (°C)")
term.plot(R1_teste_1,what="sigma", terms =2, ask = FALSE, ylim = "free",
 ↪  ylabs = expression(log(sigma)), xlabs = "Pressão Atmosférica (mB)")
term.plot(R1_teste_1,what="sigma", terms =3, ask = FALSE, ylim = "free",
 ↪  ylabs = expression(log(sigma)), xlabs = expression("Velocidade do
 ↪  vento (m.s"^{-1} * ")"))


# Comportamento das funções de suavização para v

term.plot(R1_teste_1, what='nu',pages = 1, ask = FALSE, ylim = "free")
term.plot(R1_teste_1,what="nu", terms =1, ask = FALSE, ylim = "free",
 ↪  ylabs = expression(nu), xlabs = "Temperatura de Orvalho (°C)")
term.plot(R1_teste_1,what="nu", terms =2, ask = FALSE, ylim = "free",
 ↪  ylabs = expression(nu), xlabs = "Pressão Atmosférica (mB)")

# Comportamento das funções de suavização para t

term.plot(R1_teste_1, what='tau',pages = 1, ask = FALSE, ylim = "free")
term.plot(R1_teste_1,what="tau", terms =1, ask = FALSE, ylim = "free",
 ↪  ylabs = expression(log(tau)), xlabs = "Temperatura de Orvalho (°C)")
term.plot(R1_teste_1,what="tau", terms =2, ask = FALSE, ylim = "free",
 ↪  ylabs = expression(log(tau)), xlabs = "Precipitação (mm)")


# Análise de Residuo
```

```
plot(R1_teste_1,ts = TRUE)

# Análise de resíduos - worm plot

wp(R1_teste_1, ylim.all = 1.5) ; title("Worm plot - BCTo")


# Fim.
```

Artigo 2

```
# ARTIGO 2
#-------------------------------------------------
## Análise Machine Learning
## Florianopólis 10 anos

## Pacotes necessários
library(dplyr)
library(ranger)
library(readxl)
library(tidyverse)
library(timetk)
library(tsibble)
library(tsibbledata)
library(fastDummies)
library(tidymodels)
library(skimr)
library(xgboost)
library(parsnip)
library(kernlab)
library(lightgbm)
library(kableExtra)
library(modeltime)
library("bonsai")


## Entrada dos dados
setwd("<PATH>")
dados <- read_xlsx("Atualizada.xlsx")
colnames(dados)<-c("Data.Medicao", "PRECIPITACAO", "PRESSAO","TEMP_ORV",
                   "TEMP_MEDIA","UMIDADE","VENT_.MED")
glimpse(dados)


# Convertendo objeto para o tipo tibble e
# Transforma data de medição para o tipo date
dados1 <- dados %>%
  tk_tbl() %>%
  mutate(Data.Medicao = as.Date(Data.Medicao))
dados1

# Plot da série temporal
dados1 %>%
  plot_time_series(Data.Medicao, TEMP_MEDIA,
                   .title = NULL,
                   .smooth  = FALSE,
```

```
                    .interactive = TRUE)

# Criando os conjuntos de Treino e teste
particao <- dados1 %>%
  time_series_split(Data.Medicao,
                    assess = 1095,
                    cumulative = TRUE)

# Quantidade de elementos por conjuntos Treino/Teste/Total
particao

particao %>%
  tk_time_series_cv_plan() %>% glimpse()

# Plot da serie nos conjuntos de Treinio e Teste
particao %>%
  tk_time_series_cv_plan() %>%
  plot_time_series_cv_plan(.date_var = Data.Medicao,
                           .value = TEMP_MEDIA,
                           .title = NULL)

# Preparando dados para análise
recipe <- recipe(TEMP_MEDIA ~ ., data = training(particao)) %>%
  step_rm(matches("(.xts$)|(.iso$)|(hour)|(minute)|(second)|(day)|
  (week)|(am.pm)")) %>%
  step_dummy(all_nominal(), one_hot = TRUE)

recipe

# Ajuste dos métodos de ML aos dados de treino (sem seleção dos
hiperparametros)
# Random Forest
fit_rf <- workflow() %>%
  add_model(
    spec = rand_forest(
      mode = "regression"
    ) %>%
      set_engine("ranger")
  ) %>%
  add_recipe(recipe %>%
               update_role(Data.Medicao, new_role = "indicator")) %>%
  fit(training(particao))

fit_rf
fit_rf$pre


# XGBoost
```

```
fit_xgboost <- workflow() %>%
  add_model(
    spec = boost_tree(
      mode = "regression"
    ) %>%
      set_engine("xgboost")
  ) %>%
  add_recipe(recipe %>%
               update_role(Data.Medicao, new_role = "indicator")) %>%
  fit(training(particao))

fit_xgboost
fit_xgboost$pre


# Prophet
fit_prophet <- workflow() %>%
  add_model(
    spec = prophet_reg(
      seasonality_daily  = FALSE,
      seasonality_weekly = FALSE,
      seasonality_yearly = TRUE
    ) %>%
      set_engine("prophet")
  ) %>%
  add_recipe(recipe) %>%
  fit(training(particao))

fit_prophet
fit_prophet$pre


# Prophet Boost
fit_prophet_boost <- workflow() %>%
  add_model(
    spec = prophet_boost(
      seasonality_daily  = FALSE,
      seasonality_weekly = FALSE,
      seasonality_yearly = TRUE
    ) %>%
      set_engine("prophet_xgboost")
  ) %>%
  add_recipe(recipe) %>%
  fit(training(particao))

fit_prophet_boost
fit_prophet_boost$pre
```

```
# SVR
fit_svr <- workflow() %>%
  add_model(
    spec = svm_linear(
      mode = "regression"
    ) %>%
      set_engine("kernlab")
  ) %>%
  add_recipe(recipe %>%
               update_role(Data.Medicao, new_role = "indicator")) %>%
  fit(training(particao))

fit_svr
fit_svr$pre


# Avaliacao Modelos
Avaliacao_Modelos <- modeltime_table(
  fit_rf,
  fit_xgboost,
  fit_prophet,
  fit_prophet_boost,
  fit_svr
)

Avaliacao_Modelos

# Modelos Ajustados _ Conjunto Treinamento
modelos_ajustados_treinamento <- Avaliacao_Modelos %>%
  modeltime_calibrate(new_data = training(particao))

modelos_ajustados_treinamento

# Avaliacao Modelos_Treinamento
modelos_ajustados_treinamento %>%
  modeltime_accuracy(training(particao)) %>%
  arrange(rmse)


# Modelos Ajustados _ Conjunto Teste
modelos_ajustados <- Avaliacao_Modelos %>%
  modeltime_calibrate(new_data = testing(particao))

modelos_ajustados


# Avaliacao Modelos
```

```
modelos_ajustados %>%
  modeltime_accuracy(testing(particao)) %>%
  arrange(rmse)


# Plot dos valores preditos
modelos_ajustados %>%
  modeltime_forecast(
    new_data    = testing(particao),
    actual_data = dados1,
    keep_data   = TRUE
  ) %>%
  plot_modeltime_forecast(
    .conf_interval_show = FALSE,
    .interactive        = TRUE
  )


# Ajuste dos hiperparametros - Usando
# Rolling Origin Forecast Resampling


# Número maximo de Slices k = 5
set.seed(123)
r_origin<- time_series_cv(data = training(particao),
                          date_var = Data.Medicao,
                          initial     = "4 years",  # Treino
                          assess      = "1 years",  # Teste
                          skip        = "6 months", # Translado da serie
                          cumulative  = FALSE,
                          slice_limit = 5)

r_origin %>%
  tk_time_series_cv_plan()%>%
  plot_time_series_cv_plan(Data.Medicao, TEMP_MEDIA, .interactive = FALSE)


# Random Forest
rf_tune <- rand_forest(
  mode = "regression",
  mtry = tune(),
  trees = tune(),
  min_n = tune(),
) %>%
  set_engine("ranger")

wflw_rf_tune <- workflow() %>%
  add_model(rf_tune) %>%
```

```
  add_recipe(recipe)
wflw_rf_tune


# Random Forest
recipe %>%
  update_role(Data.Medicao, new_role = "indicator") %>%
  prep() %>%
  summary() %>%
  group_by(role) %>%
  summarise(n=n())

# Busca aleatória de Hiperparametros
set.seed(123)
grid_hiper<- grid_latin_hypercube(
  extract_parameter_set_dials(rf_tune) %>%
    update(mtry = mtry(range = c(1, 6)),
           trees = trees(range = c(630, 1000)),
           min_n = min_n(range=c(12, 25))),
  size = 35     # Numero máximo de combinações dos hiperparametros
)

resultados_rf<- wflw_rf_tune %>%
  tune_grid(
    resamples  = r_origin,
    grid = grid_hiper,
    control = control_grid(verbose = TRUE,
                           allow_par = TRUE)
  )


resultados_rf%>%
  show_best("rmse", n = 2)

# Ajuste do melhor modelo pelo RMSE
set.seed(123)
fit_rf_tuned <- wflw_rf_tune %>%
  finalize_workflow(
    select_best(resultados_rf, "rmse", n=1)) %>%
  fit(training(particao))

modeltime_table(fit_rf_tuned) %>%
  modeltime_calibrate(testing(particao)) %>%
  modeltime_accuracy()


# XGBoost
xgboost_tune <- boost_tree(
```

```
  mode = "regression",
  mtry = tune(),
  trees = tune(),
  min_n = tune(),
  #  tree_depth = tune(),
  learn_rate = tune(),
  #  loss_reduction = tune(),
  #  sample_size = tune(),
) %>%
  set_engine("xgboost")

wflw_xgboost_tune <- workflow() %>%
  add_model(xgboost_tune) %>%
  add_recipe(recipe %>%
               update_role(Data.Medicao, new_role = "indicator"))
wflw_xgboost_tune


# Busca aleatória de Hiperparametros
set.seed(123)
grid_hiper<- grid_latin_hypercube(
  extract_parameter_set_dials(xgboost_tune) %>%
    update(mtry = mtry(range = c(1, 6)),
           learn_rate = learn_rate(range = c(-1.7, -0.58)),
           trees = trees(range = c(800, 1050))),
  size = 35
)


resultados_xgboost<- wflw_xgboost_tune %>%
  tune_grid(
    resamples  = r_origin,
    grid = grid_hiper,
    control = control_grid(verbose = TRUE,
                           allow_par = TRUE)
  )


resultados_xgboost%>%
  show_best("rmse", n = 2)


# Ajuste do melhor modelo pelo RMSE
set.seed(123)
fit_xgboost_tuned <- wflw_xgboost_tune %>%
  finalize_workflow(
    select_best(resultados_xgboost, "rmse", n=1)) %>%
  fit(training(particao))
```

```
modeltime_table(fit_xgboost_tuned) %>%
  modeltime_calibrate(testing(particao)) %>%
  modeltime_accuracy()


# Prophet
prophet_tune <- prophet_reg(
  mode = "regression",
  seasonality_yearly = TRUE,
  seasonality_weekly = TRUE,
  seasonality_daily = TRUE,
) %>%
  set_engine("prophet")

wflw_prophet_tune <- workflow() %>%
  add_model(prophet_tune) %>%
  add_recipe(recipe)
wflw_prophet_tune

recipe %>%
  update_role(Data.Medicao, new_role = "indicator") %>%
  prep() %>%
  summary() %>%
  group_by(role) %>%
  summarise(n=n())


# Ajuste do melhor modelo pelo RMSE
set.seed(123)
fit_prophet_tuned <- wflw_prophet_tune %>%
  fit(training(particao))

modeltime_table(fit_prophet_tuned) %>%
  modeltime_calibrate(testing(particao)) %>%
  modeltime_accuracy()


# Prophet Boost
prophet_boost_tune <- prophet_boost(
  mode = "regression",
  changepoint_num = tune(),
  seasonality_yearly = TRUE,
  seasonality_weekly = TRUE,
  seasonality_daily = TRUE,
  mtry = tune(),
  trees = tune(),
  min_n = tune(),
```

```
  tree_depth = tune(),
  learn_rate = tune(),
  loss_reduction = tune(),
) %>%
  set_engine("prophet_xgboost")

wflw_prophet_boost_tune <- workflow() %>%
  add_model(prophet_boost_tune) %>%
  add_recipe(recipe)
wflw_prophet_boost_tune

recipe %>%
  update_role(Data.Medicao, new_role = "indicator") %>%
  prep() %>%
  summary() %>%
  group_by(role) %>%
  summarise(n=n())


# Busca aleatória de Hiperparametros
set.seed(123)
grid_hiper<- grid_latin_hypercube(
  extract_parameter_set_dials(prophet_boost_tune) %>%
    update(mtry = mtry(range = c(1, 5)),
           learn_rate = learn_rate(range = c(-1.7, -0.58)),
           trees = trees(range = c(600, 1500))),
  size = 5
)


resultados_prophet_boost <- wflw_prophet_boost_tune %>%
  tune_grid(
    resamples  = r_origin,
    grid = grid_hiper,
    control = control_grid(verbose = TRUE,
                           allow_par = TRUE)
  )

resultados_prophet_boost %>%
  show_best("rmse", n = 2)



# Ajuste do melhor modelo pelo RMSE
set.seed(123)
fit_prophet_boost_tuned <- wflw_prophet_boost_tune %>%
  finalize_workflow(
    select_best(resultados_prophet_boost, "rmse", n=1)) %>%
```

```
  fit(training(particao))

modeltime_table(fit_prophet_boost_tuned) %>%
  modeltime_calibrate(testing(particao)) %>%
  modeltime_accuracy()



# SVR
svr_tune <- svm_linear(
  mode = "regression",
  cost = tune(),
  #  margin = tune(),
) %>%
  set_engine("kernlab")

model_svr_tune <- workflow() %>%
  add_model(svr_tune) %>%
  add_recipe(recipe)
model_svr_tune


recipe %>%
  update_role(Data.Medicao, new_role = "indicator") %>%
  prep() %>%
  summary() %>%
  group_by(role) %>%
  summarise(n=n())

# Busca aleatória de Hiperparametros
set.seed(123)
grid_hiper <- grid_latin_hypercube(
  extract_parameter_set_dials(svr_tune) %>%
    update(cost = cost(range = c(0.1, 1))),
  size = 5
)

grid_hiper


# SVR - Tune Grid
resultados_svr <- model_svr_tune %>%
  tune_grid(
    resamples  = r_origin,
    grid = grid_hiper,
    control = control_grid(verbose = TRUE,
                           allow_par = TRUE)
  )
```

```
resultados_svr %>%
  show_best("rmse", n = Inf)



# Ajuste do melhor modelo pelo RMSE
set.seed(123)
fit_svr_tuned <- model_svr_tune %>%
  finalize_workflow(
    select_best(resultados_svr, "rmse", n=1)) %>%
  fit(training(particao))

modeltime_table(fit_svr_tuned) %>%
  modeltime_calibrate(testing(particao)) %>%
  modeltime_accuracy()

recipe %>%
  update_role(Data.Medicao, new_role = "indicator") %>%
  prep() %>%
  summary() %>%
  group_by(role) %>%
  summarise(n=n())

# Resutados finais

# Modelos ajustados sem escolha dos hiperparametros
Avaliacao_Modelos <- modeltime_table(
  fit_rf,
  fit_xgboost,
  fit_prophet,
  fit_prophet_boost,
  fit_svr
)

# Modelos ajustados com escolha dos hiperparametros
modelos_c_selecao <- modeltime_table(
  fit_rf_tuned,
  fit_xgboost_tuned,
  fit_prophet_tuned,
  fit_prophet_boost_tuned,
  fit_svr_tuned,
  fit_ligthgbm_tuned

) %>%
  update_model_description(1, "RANGER - Tuned") %>%
  update_model_description(2, "XGBOOST - Tuned") %>%
```

```
  update_model_description(3, "PROPHET W/ REGRESSORS - Tuned") %>%
  update_model_description(4, "PROPHET W/ XGBOOST ERRORS - Tuned") %>%
  update_model_description(5, "SVR - Tuned") %>%
  combine_modeltime_tables(Avaliacao_Modelos)

modelos_c_selecao


ajuste_todos_treinamento <- modelos_c_selecao %>%
  modeltime_calibrate(training(particao))

ajuste_todos_treinamento %>%
  modeltime_accuracy() %>%
  arrange(rmse)

ajuste_todos <- modelos_c_selecao %>%
  modeltime_calibrate(testing(particao))

ajuste_todos %>%
  modeltime_accuracy() %>%
  arrange(rmse)

#Plot dos valores preditos
ajuste_todos %>%
  modeltime_forecast(
    new_data    = testing(particao),
    actual_data = artifacts$data$dados1,
    keep_data   = TRUE
  ) %>%
  plot_modeltime_forecast(
    #.facet_ncol          = 4,
    .conf_interval_show = FALSE,
    .interactive        = TRUE
  )


# Fim.
```