INTERFACE TECNOLÓGICA

PREVISÃO DE PRODUÇÃO DE CANA-DE-AÇÚCAR UTILIZANDO FRAMEWORK LOW-CODE PYCARET

SUGARCANE PRODUCTION PREDICTION USING LOW-CODE PYCARET FRAMEWORK

Paulo Sérgio Calor Jr.– paulo.calor@fatec.sp.gov.br
Faculdade de Tecnologia de Bebedouro "Jorge Caram Sabbag" – Bebedouro – SP – Brasil

Renan Guilherme Nespolo – renan.nespolo01@fatec.sp.gov.br
Faculdade de Tecnologia de Bebedouro "Jorge Caram Sabbag" – Bebedouro – SP – Brasil

DOI: 10.31510/infa.v22i2.2371
Data de submissão: 26/09/2025
Data do aceite: 03/12/2025
Data da publicação: 20/12/2025

RESUMO

No presente trabalho é apresentado um estudo sobre o framework *low-code PyCaret*. Esse framework seleciona modelos de Aprendizado de Máquina de forma automatizada (AutoML), englobando pré-processamento, seleção de características e previsão de resultados. No campo do Agronegócio, uma das possíveis aplicações do framework é a previsão de produtividade agrícola, medida em toneladas de cana por hectare (TCH), em usinas brasileiras. Deste modo, o presente estudo concentrou-se em aplicar o framework *low-code* para realizar a previsão do TCH, que realizou a avaliação de vários modelos de regressão. O estudo ainda apresentou uma comparação entre os dois métodos que apresentaram os melhores desempenhos, segundo o framework, a partir das métricas de avaliação adotadas. Os resultados apresentados após a validação do modelo, em cenários agrícolas reais, indicaram a presença de outliers e outros problemas que explicaram a seleção de determinados algoritmos. O estudo contribui para a compreensão do uso de AutoML na agricultura de precisão, apresentando algoritmos, validação de modelos e limitações do uso do framework.

Palavras-chave: AutoML. TCH. Cana-de-açúcar. *PyCaret*. *Framework*. Agricultura de Precisão.

ABSTRACT

This paper presents a study of the low-code PyCaret framework. This framework selects Machine Learning (AutoML) models automatically, encompassing preprocessing, feature selection, and result prediction. In the agribusiness field, one of the framework's potential applications is forecasting agricultural productivity, measured in tons of sugarcane per hectare (TCH), in Brazilian mills. Therefore, this study focused on applying the low-code framework to forecast TCH, evaluating several regression models. The study also compared the two methods that, according to the framework, performed the best performance based on the adopted evaluation metrics. The results presented after model validation in real agricultural scenarios

INTERFACE TECNOLÓGICA

indicated the presence of outliers and other issues that explained the selection of certain algorithms. The study contributes to the understanding of the use of AutoML in precision agriculture by presenting algorithms, model validation, and limitations of the framework's use.

Keywords: *AutoML. TCH. Sugarcane. PyCaret. Framework. Precision Agriculture.*

1 INTRODUÇÃO

A safra 2024/25 de cana-de-açúcar no Brasil apresentou produção estimada em 676,96 milhões de toneladas, consolidando o país como maior produtor mundial desta cultura. O setor sucroenergético brasileiro é responsável por aproximadamente 2% do PIB nacional e gera mais de 800 mil empregos diretos, demonstrando sua relevância econômica e social (Conab, 2025).

Neste contexto, a predição precisa da produtividade, medida por meio da métrica Toneladas de Cana por Hectare (THC), constitui um desafio científico e tecnológico fundamental para otimização dos processos produtivos. O TCH representa o principal indicador de produtividade utilizado no setor sucroenergético, influenciando diretamente decisões estratégicas relacionadas ao planejamento da safra, dimensionamento da capacidade industrial, logística de transporte e viabilidade econômica dos empreendimentos (Pedregosa *et al.*, 2011; Scarpari, 2002).

O Aprendizado de Máquina Automatizado (AutoML) surge como uma solução promissora para analisar o acesso a estas tecnologias avançadas, automatizando tarefas complexas como seleção de algoritmos, otimização de hiper parâmetros e validação de modelos. O *framework* PyCaret, desenvolvida especificamente para facilitar a implementação de workflows de aprendizado de máquina, representa uma ferramenta particularmente adequada para esta finalidade (Ali, 2020; Parekh, 2023).

Portanto o presente estudo buscou desenvolver uma solução de predição de TCH baseado em AutoML, utilizando o *framework* PyCaret para comparação sistemática de algoritmos de regressão e implementação de interface web acessível para aplicação prática no setor sucroenergético. Os objetivos do presente estudo são:

1. Analisar tecnicamente a arquitetura do *framework* PyCaret, identificando os mecanismos internos de seleção automática de algoritmos de aprendizado de máquina.

2. Comparar sistematicamente os 2 melhores algoritmos de regressão de acordo com as métricas estabelecidas no *framework*.
3. Realizar validação dos modelos dos algoritmos utilizando métricas adotadas em problemas de regressão.

O presente estudo está organizado em: 1Introdução; 2 Fundamentação Teórica; 3 Materiais e Métodos; 4 Resultados; 5 Conclusões e Referências.

2 FUNDAMENTAÇÃO TEÓRICA

A presente Seção apresenta toda a fundamentação deste estudo, segmentados em: 2.1 Aprendizado de Máquina Automatizado (AutoML); 2.2 *Framework PyCaret*: Arquitetura e Funcionalidades; 2.3 Validação Cruzada *k-Fold Cross-Validation* e; 2.4 Métricas de avaliação.

2.1 aprendizado de Máquina Automatizado (AutoML)

A existência de uma vasta gama de algoritmos, incluindo os complexos métodos de ensemble discutidos anteriormente, torna a tarefa de selecionar, configurar e otimizar manualmente o melhor modelo para um dado problema um processo árduo, demorado e que exige profundo conhecimento especializado. É para este desafio que emerge o paradigma do Aprendizado de Máquina Automatizado (AutoML) (He; Zhao; Chu, 2021).

O AutoML promete uma mudança de funcionamento na área da ciência de dados, buscando automatizar o processo de ponta a ponta do desenvolvimento de modelos de machine learning. Esta abordagem surge como resposta à crescente complexidade dos workflows e à necessidade de democratizar o acesso a estas tecnologias. O escopo do AutoML, conforme definido pela literatura, abrange a automação de todo o pipeline, incluindo o pré-processamento, a seleção do algoritmo, a otimização de seus hiper parâmetros e a construção de ensembles. O objetivo fundamental é transformar o processo, tradicionalmente artesanal, em uma busca sistemática, eficiente e mais acessível (He; Zhao; Chu, 2021).

2.2 *Framework PyCaret*: Arquitetura e Funcionalidades

Para operacionalizar os conceitos de Aprendizado de Máquina Automatizado (AutoML) em um fluxo prático, sistemático e reprodutível para a predição de TCH, este estudo adotou a biblioteca *open-source PyCaret*. Seu *design* centrado na produtividade e abordagem *low-code* foram decisivos para a condução dos experimentos. O *PyCaret* foi desenvolvido com a filosofia

de democratizar o acesso a técnicas avançadas de *machine learning*, permitindo que o pesquisador foque na análise do problema e na interpretação dos resultados, em vez de lidar com a complexidade da codificação (Parekh, 2023).

A robustez da biblioteca deriva de sua arquitetura, construída sobre pilares consolidados do ecossistema *Python* de ciência de dados, como *scikit-learn*, *pandas*, *numpy*, *matplotlib* e *plotly*. O PyCaret atua como um orquestrador de alto nível, integrando essas bibliotecas para garantir confiabilidade, escalabilidade e desempenho ao fluxo preditivo (Parekh, 2023).

2.3 Validação Cruzada k-Fold cross-validation

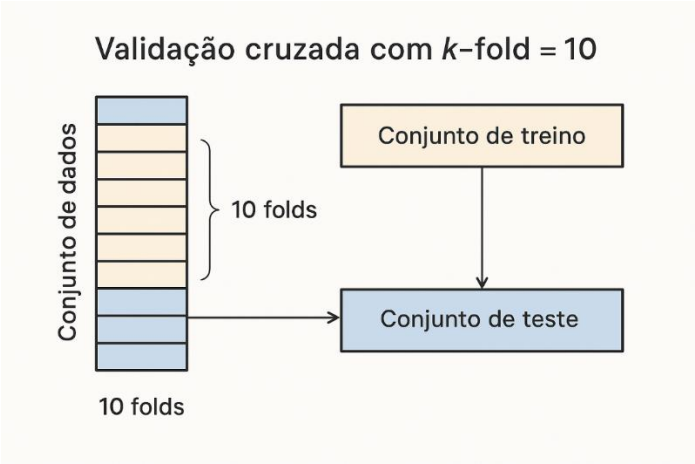
Ao treinar um modelo de *machine learning*, um dos maiores desafios é garantir que sua performance seja generalizável para dados futuros. Uma abordagem simples seria dividir os dados uma única vez em treino e teste, mas essa estratégia apresenta uma fraqueza significativa: o resultado pode variar drasticamente dependendo da divisão aleatória, gerando otimismo irreal sobre o desempenho.

Para superar essa limitação, este trabalho empregou a validação cruzada *k-fold*, considerada padrão-ouro para avaliação de modelos (Hastie, 2009). O processo, implementado através da função *setup()* do PyCaret com *fold*=10, funciona da seguinte forma: O particionamento de dados é realizado a partir do conjunto de dados dado. Este é dividido em $k = 10$ *folds* (partições) de tamanho aproximadamente igual. Então o processo iterativo é iniciado. Para cada *fold*, o modelo é treinado nos outros 9 *folds* e validado no *fold* restante. Quando finalizada esta etapa. A próxima etapa do método de é avaliação do modelo. Para cada tarefa específica de Aprendizado de Máquina, um tipo de métricas de performance deve ser escolhida para realização do cálculo para cada uma das 10 iterações. A performance final do modelo é a média dos 10 resultados individuais (Hastie, 2009).

Esta abordagem garante que cada subconjunto de dados seja utilizado tanto para treino quanto para validação. Para tarefas de classificação as métricas comumente adotadas são: Acurácia, Precisão, *F1-Score* e Revogação. Para a tarefa de Regressão as métricas mais comuns de avaliação são: Média Absoluta de Erro (MAE), Média Quadrada de Erro (MSE), Raiz da média Absoluta de Erro (RMAE) e Raiz Média Quadrada de Erro (RMSE) sendo essa última escolhida para detecção de resultados com bastante variância e anomalias (Filho, 2018). O processo do *k-fold Cross-Validation* é ilustrado esquematicamente na Figura 1, na parte de

robustez estatística.

Figura 1 - Representação esquemática do processo de validação cruzada com *k*=10.



Fonte: Elaborada pelo autor.

2.4 Métricas de avaliação

Para a comparação de métodos de forma quantitativa o desempenho dos algoritmos, é necessário selecionar uma métrica alinhada aos objetivos do problema. Este trabalho adotou duas métricas dadas: o Erro Médio Absoluto (MAE), e a Raiz do Erro Médio Quadrado (RMSE). O MAE é definido na Equação 1 dada:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{1}$$

no qual n é o número total de observações (talhões), $\hat{y}_i$ é o valor de TCH predito pelo modelo para a observação y_i é o valor de TCH real para a observação i.

O RMSE é apresentado como segundo critério na Equação 2:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}, \tag{2}$$

no qual as variáveis utilizadas são as mesmas já apresentadas na Equação 1.

A característica fundamental do RMSE reside no termo $(\hat{y}_i - y_i)^2$. Ao elevar o erro ao quadrado, a métrica não apenas penaliza erros maiores, mas o faz de forma exponencial. Isso significa que um único talhão cuja produtividade foi prevista com um erro de 20 t/ha contribui para o RMSE total mais do que quatro talhões com erros de 10 t/ha cada. No contexto do planejamento agroindustrial, essa sensibilidade é uma vantagem estratégica. A subestimação drástica de um talhão de alta produtividade pode levar à perda de matéria-prima por falta de capacidade logística no pico da colheita. Inversamente, a superestimação pode resultar em ociosidade de frentes de colheita e caminhões, um custo operacional direto.

Ao contrário do Erro Absoluto Médio (MAE), que trata todos os erros de forma linear e igualitária, o RMSE força o algoritmo a dar prioridade máxima à redução desses desvios extremos, que são os mais disruptivos para a operação. Portanto, ao otimizar o modelo para minimizar o RMSE, estamos implicitamente alinhando o objetivo estatístico do modelo ao objetivo de negócio: mitigar os riscos associados às previsões mais imprecisas e aumentar a confiabilidade do planejamento de safra (Chai; Draxler, 2014).

3 MATERIAIS E MÉTODOS

Na presente Seção são apresentados os materiais e métodos do presente estudo, segmentados em: 3.1 Conjunto de dados e; 3.2 Configuração do *framework*.

3.1 Conjunto de Dados

O conjunto de dados utilizado neste estudo é proveniente de registros operacionais e agronômicos de uma usina sucroenergética brasileira, representando um histórico produtivo real em nível de talhão. Os dados foram extraídos de sistemas de gestão agrícola (ERP Agrícola) e submetidos a um processo de seleção de características *(feature selection)*, resultando em um arquivo estruturado (base_modelo.csv).

O recorte final dos dados considerados para compor o *Dataset* foi definido contendo 9.147 registros únicos, no qual cada registro corresponde a um talhão colhido em uma determinada safra, e 19 variáveis (dimensões). O *Dataset* abrange um período de múltiplas safras (2018-2023), englobando algumas condições que para a construção de um modelo, tais como: variabilidade Genética, múltiplas variedades de cana-de-açúcar; variabilidade edáfica, diferentes tipos de solo (Latossolo, Argissolo); variabilidade de manejo, distintos estágios de

corte e sistemas de irrigação e; variabilidade climática, registros de estações meteorológicas locais. Esta heterogeneidade multifatorial assegura a capacidade de generalização dos modelos treinados, condição essencial para aplicabilidade operacional (Benjamin *et al.*, 2023).

3.2 Configurações do *Framework*

Este estudo caracteriza-se como pesquisa aplicada de natureza quantitativa, com abordagem empírico-analítica, voltada ao desenvolvimento e validação de modelos preditivos de produtividade agrícola (TCH) utilizando técnicas de AutoML via PyCaret. Adota-se o paradigma hipotético-dedutivo (Popper, 2005), no qual uma hipótese central é submetida a testes empíricos rigorosos através de experimentação controlada e análise estatística.

Para a configuração do framework foi necessário preencher vários parâmetros para a AutoML realize a identificação de modelos preditivos de TCH com desempenho melhor na métricas de avaliação RMSE, na qual RMSE deve ser inferior 8,5 t/ha, valor estimado em medições anteriores (baseline)

Após a configuração do framework, o motor AutoML realiza a etapa de treinamento, que nesse caso particular é executar a comparação sistemática de vários algoritmos de regressão utilizando a linha de código $compare_models()$. Por fim a avaliação é apresentada para validar os modelos comparados (He; Zhao; Chu, 2021).

Após todos os parâmetros serem definido, eles são inseridos e executados na função *setup()* do *framework* PyCaret. Nessa etapa serão executados todos os algoritmos disponíveis na base do PyCaret para a tarefa de regressão e os melhores serão apresentados com seus respectivos resultados.

O trecho de código abaixo detalha a configuração da função *setup()*.

```
s = setup(
    data=df,
    target='QT_TCHREALIZADO',
    session_id=42,
    normalize=True,
    transformation=True,
    remove_multicollinearity=True,
    multicollinearity_threshold=0.9,
    fold=10,
    verbose=False
)
```

INTERFACE TECNOLÓGICA

Na Tabela 1, são apresentadas de forma detalhada os principais parâmetros de configuração.

Tabela 1 - Parâmetros Completos da Função setup() do PyCaret

Parâmetro	Valor/Config	Justificativa Técnica
data=df	DataFrame principal	**Base de dados** com 9.147 registros para treinamento e validação
target='QT_TCHREALIZADO'	Variável alvo	**Produtividade de TCH** (t/ha) como variável dependente a ser predita
session_id=42	Semente fixa	**Garantia de reprodutibilidade** em processos estocásticos e particionamento
normalize=True	Ativado	**Padronização Z-score** para algoritmos sensíveis à escala (regressões, SVR, KNN)
transformation=True	Ativado	**Transformação Yeo-Johnson** para normalizar distribuições assimétricas
remove_multicollinearity=True	Ativado	**Remoção automática** de variáveis com correlação > 0.9 para estabilidade do modelo
multicollinearity_threshold=0.9	Limiar de 0.9	**Critério de correlação** de Pearson para identificar variáveis redundantes
fold=10	10 partições	**Validação cruzada robusta** (k-fold) para estimativa confiável de generalização
verbose=False	Desativado	**Supressão de logs** detalhados para execução limpa em ambiente de produção

Fonte: Elaborado pelo autor.

A avaliação da performance e a validação da capacidade de generalização dos modelos seguiram um protocolo rigoroso de duas fases, desenhado para garantir uma estimativa realista e não enviesada do desempenho.

Na etapa de Seleção (Via Validação Cruzada): Na etapa de comparação de modelos, a performance de cada algoritmo foi estimada utilizando um procedimento de validação cruzada estratificada com k=10. Este método consiste em dividir os dados de treino em 10 partições, treinando e avaliando cada modelo 10 vezes, de modo que cada partição seja utilizada como conjunto de validação uma única vez. A performance final de cada modelo nesta fase foi a média dos resultados das 10 iterações. O RMSE foi a métrica primária utilizada para ordenar e

selecionar o algoritmo mais promissor, dada sua sensibilidade a erros de grande magnitude. A aplicação da metodologia de validação cruzada *k-fold* e a classificação dos algoritmos com base na métrica RMSE culminaram na identificação de dois modelos com desempenho superior para a predição de TCH: o *Huber Regressor* (Huber, 1992) e o *Light Gradient Boosting Machine* (*LightGBM*) (Ke *et al.*, 2017).

4 RESULTADOS E DISCUSSÃO

No presente capítulo é apresentado os resultados obtidos a partir da aplicação da metodologia de AutoML. O foco da análise é a comparação sistemática dos algoritmos de regressão, a avaliação aprofundada do modelo *Huber Regressor* e a discussão das implicações práticas dos achados para a predição de produtividade da cana-de-açúcar (TCH).

A presença de um modelo linear robusto *(Huber Regressor*) e de um algoritmo de *ensemble* não linear (*LightGBM*) entre os melhores resultados evidencia a importância de se testar diferentes famílias de modelos para problemas complexos como a predição de TCH. Enquanto o *Huber Regressor* se destacou pela capacidade de lidar com *outliers* e fornecer previsões estáveis em cenários com dados ruidosos, o *LightGBM* mostrou-se superior na modelagem de relações não lineares e interações entre múltiplas variáveis agronômicas. Essa diversidade reforça a relevância de ferramentas de AutoML, como o PyCaret, que facilitam a experimentação sistemática e a comparação entre algoritmos.

4.1 Análise Comparativa e Seleção do Modelo

A seleção do modelo foi realizada por meio de um processo competitivo automatizado, utilizando a função *compare_models()* do *PyCaret*. Mais de 15 algoritmos de regressão foram treinados e avaliados sob protocolo de validação cruzada estratificada (*10-fold*). A Tabela 8 apresenta os resultados de desempenho dos dois principais modelos selecionados para este estudo, ordenados pela métrica primária de avaliação, RMSE.

A análise da tabela demonstra que o *Huber Regressor* alcançou o menor RMSE médio (182,67 t/ha), estabelecendo-se como o modelo de melhor desempenho sob o critério primário.

O *LightGBM*, embora seja um ensemble de última geração (Ke *et al.*, 2017), apresentou RMSE superior (192,54 t/ha), como apresentado no Gráfico 1.

Tabela 2 - Desempenho Comparativo dos Modelos de Regressão (Validação Cruzada)

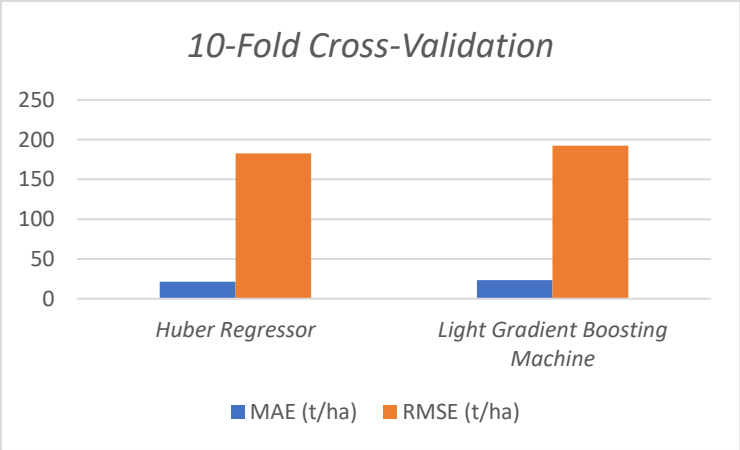
Modelo	MAE (t/ha)	RMSE (t/ha)	Tempo (s)
Huber Regressor	**21.35**	**182.67**	**9.03**
Light Gradient Boosting Machine	23.46	192.54	50.25

Fonte: Elaborado pelo autor.

4.2 Título de subseção/subtítulo da seção

Os resultados deste estudo, em conjunto, contam uma história coesa. A superioridade do *Huber Regressor* sobre *ensembles* mais complexos, como *LightGBM*, sugere que a principal característica para modelar os dados deste problema não foi a capacidade de capturar interações complexas, mas sim a robustez para lidar com a alta variabilidade e os *outliers* inerentes aos dados agrícolas (Huber, 1964).

Gráfico 1 - Comparativo de Erros entre as técnicas *Huber Regressor* e LightGBM



Fonte: Elaborada pelo autor

A escolha do *Huber Regressor foi* justificada por critérios práticos e metodológicos observados ao longo do estudo. Primeiro, ao comparar diretamente os desempenhos, o *Huber*

Regressor apresentou resultados mais estáveis e menos sensíveis a oscilações extremas dos dados, enquanto o *LightGBM*, embora sofisticado e capaz de modelar relações não-lineares, mostrou-se mais suscetível à influência de *outliers* presentes na base. Essa diferença foi perceptível tanto nas métricas de erro quanto na consistência das previsões em conjuntos de validação e teste.

Além disso, a implementação via *framework low-code* PyCaret, destacou a importância de soluções que conciliem facilidade de uso, replicabilidade e estabilidade dos resultados. O *Huber Regressor* demonstrou ser não apenas eficaz, mas também transparente e facilmente interpretável, facilitando a comunicação dos resultados para equipes multidisciplinares e para a tomada de decisão operacional.

5 CONCLUSÃO

Este trabalho teve como objetivo avaliar o potencial do Aprendizado de Máquina Automatizado (AutoML), com ênfase no uso do *framework* PyCaret, para a predição da produtividade agrícola (TCH) em usinas de cana-de-açúcar. A hipótese central era que o AutoML permitiria identificar modelos preditivos robustos, capazes de superar abordagens manuais em cenários de dados agrícolas complexos.

Os resultados demonstraram que, entre vários algoritmos avaliados, o *Huber Regressor* apresentou os menores erros nos critérios de avaliação selecionados, superando outros modelos como o *LightGBM*. O desempenho do *Huber Regressor* manteve-se estável testes realizados, apresentando a presença de *outliers* sobre as previsões do TCH.

Embora o PyCaret ofereça métricas padrão para avaliação de modelos, a customização de métricas específicas ou a implementação de estratégias de avaliação não convencionais pode ser restrita, o que limita a análise em cenários que exigem critérios diferenciados de performance reforçando a necessidade de avaliações críticas ao adotar *frameworks low-code* em projetos científicos, especialmente quando o objetivo é garantir máxima flexibilidade, transparência e controle sobre o desenvolvimento de modelos preditivos.

Como trabalhos futuros testes com abordagens híbridas e redes profundas, tais como: redes neurais e modelos híbridos, são planejados, com o objetivo de estudar o comportamento de relações não lineares e melhorar interações complexas.

REFERÊNCIAS

ALI, M. **PyCaret: An open source, low-code machine learning library in Python.** PyCaret version, [*s. l.*], v. 2, 2020. .

BENJAMIN, S. G.; JAMES, E. P.; SZOKE, E. J.; SCHLATTER, P. T.; BROWN, J. M. The **30 December 2021 Colorado Front Range windstorm and Marshall Fire: Evolution of surface and 3D structure**, NWP guidance, NWS forecasts, and decision support. Weather and Forecasting, [*s. l.*], v. 38, n. 12, p. 2551–2573, 2023. .

CHAI, T.; DRAXLER, R. R. **Root mean square error (RMSE) or mean absolute error (MAE)**. Geoscientific model development discussions, [*s. l.*], v. 7, n. 1, p. 1525–1534, 2014. .

CONAB. **Safra 2024/25 de cana-de-açúcar encerra com produção estimada em 676,96 milhões de toneladas**. 2025. Companhia Nacional de Abastecimento. Disponível em: https://www.gov.br/conab/pt-br/assuntos/noticias/safra-2024-25-de-cana-de-acucar-encerra-com-producao-estimada-em-676-96-milhoes-de-toneladas. Acesso em: 25 set. 2025.

FILHO, M. **As Métricas Mais Populares para Avaliar Modelos de Machine Learning**. 6 maio 2018. Disponível em: https://mariofilho.com/as-metricas-mais-populares-para-avaliar-modelos-de-machine-learning/. Acesso em: 25 set. 2025.

HASTIE, T. **The elements of statistical learning: data mining, inference, and prediction**. [*S. l.*]: Springer, 2009.

HE, X.; ZHAO, K.; CHU, X. **AutoML: A survey of the state-of-the-art**. Knowledge-Based Systems, [*s. l.*], v. 212, p. 106622, 5 jan. 2021. https://doi.org/10.1016/j.knosys.2020.106622.

HUBER, P. J. **Robust Estimation of a Location Parameter**. *In*: KOTZ, S.; JOHNSON, N. L. (orgs.). Breakthroughs in Statistics. Springer Series in Statistics. New York, NY: Springer New York, 1992. p. 492–518. DOI 10.1007/978-1-4612-4380-9_35. Disponível em: http://link.springer.com/10.1007/978-1-4612-4380-9_35. Acesso em: 25 set. 2025.

KE, G.; MENG, Q.; FINLEY, T.; WANG, T.; CHEN, W.; MA, W.; YE, Q.; LIU, T.-Y. **Lightgbm: A highly efficient gradient boosting decision tree**. Advances in neural information processing systems, [*s. l.*], v. 30, 2017. Disponível em: https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html. Acesso em: 25 set. 2025.

PAREKH. **PyCaret 3.0 | Docs.** 19 mar. 2023. Disponível em: https://pycaret.gitbook.io/DOCS. Acesso em: 25 set. 2025.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V. **Scikit-learn: Machine learning in Python**. the Journal of machine Learning research, [*s. l.*], v. 12, p. 2825–2830, 2011. .

POPPER, K. **The logic of scientific discovery**. [*S. l.*]: Routledge, 2005. Disponível em: https://www.taylorfrancis.com/books/mono/10.4324/9780203994627/logic-scientific-discovery-karl-popper-karl-popper. Acesso em: 25 set. 2025.

INTERFACE TECNOLÓGICA

SCARPARI, M. S. **Modelos para a previsão da produtividade da cana-de-açúcar (Saccharum spp.) através de parâmetros climáticos.** 2002. PhD Thesis – Universidade de São Paulo, 2002. Disponível em: https://www.teses.usp.br/teses/disponiveis/11/11136/tde-17122002-165859/publico/maximiliano.pdf. Acesso em: 25 set. 2025.