UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA – CCET
DEPARTAMENTO DE COMPUTAÇÃO – DC
CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

# Análise Comparativa do Desempenho de Modelos de Machine Learning na Previsão de Focos de Incêndio no Cerrado Utilizando Variáveis Climáticas

JULIANO ELENO SILVA PÁDUA

São Carlos - SP
2025

JULIANO ELENO SILVA PÁDUA

# Análise Comparativa do Desempenho de Modelos de Machine Learning na Previsão de Focos de Incêndio no Cerrado Utilizando Variáveis Climáticas

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia de Computação da Universidade Federal de São Carlos, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientação: Prof. Dr. Alexandre Levada

São Carlos - SP
2025

*Dedico este trabalho à minha mãe, Maricéia, guardiã incansável do Parque Estadual do Pau Furado que reúne Mata Atlântica e Cerrado. Seu esforço cotidiano para manter a unidade funcionando, sua atuação firme diante das queimadas que anualmente ameaçam o bioma e o exemplo de quem, além de gerir, enfrenta o fogo com as próprias mãos, são a inspiração e o sentido deste estudo.*

# Agradecimentos

Agradeço à minha família e aos parentes próximos, em especial à minha mãe, Maricéia, ao meu pai, Cláudio, e à minha irmã, Gabriela, cuja formação elevada, compromisso com a cultura e participação ativa na comunidade científica foram exemplo constante e incentivo decisivo ao longo desta jornada. Registro também minha gratidão à minha avó, Maria Tereza, e ao meu avô, Marc Dourojeanni, referências na comunidade científica nas áreas de meio ambiente, preservação e conservação, pelas lições de rigor, ética e serviço público.

Ao meu orientador, Prof. Dr. Alexandre Levada, agradeço pela orientação atenta, pelas discussões técnicas precisas e pela confiança depositada neste trabalho. Estendo o agradecimento aos amigos e às pessoas próximas que ofereceram apoio, paciência e encorajamento nos momentos decisivos do percurso acadêmico.

*“O fogo é o vilão destruidor. Em minutos, ele apaga décadas de regeneração, de florestas, de animais e de esperança.”*

*(Maria Tereza Jorge Pádua, conhecida como a “Mãe dos Parques Nacionais do Brasil”)*

# Resumo

A predição de focos de queimadas no Cerrado a partir de variáveis meteorológicas constitui um problema relevante para monitoramento ambiental e apoio a sistemas de alerta. Neste trabalho, foi realizada uma análise comparativa de modelos de aprendizado de máquina clássico para a predição horária da ocorrência de focos de queimadas, a partir da integração entre dados do INMET e registros do BDQueimadas, do INPE. Foram avaliados os modelos Regressão Logística, Naive Bayes, SVM linear, Random Forest e XGBoost em diferentes cenários de preparação dos dados, incluindo bases originais, bases com variáveis derivadas, imputação por KNN e estratégias de tratamento do desbalanceamento, como *SMOTE* e balanceamento por peso. Os resultados mostraram que os modelos baseados em *ensemble* de árvores foram os mais adequados ao problema, com destaque para o XGBoost, e que a engenharia de atributos e o tratamento explícito do desbalanceamento contribuíram de forma decisiva para o aumento do desempenho, especialmente em métricas mais sensíveis à detecção da classe positiva, como $PR\text{-}AUC$ e *F1-score*.

**Palavras-chave**: queimadas; Cerrado; focos de calor; aprendizado de máquina; INMET; INPE.

# Abstract

Predicting wildfire hotspot occurrence in the Cerrado from meteorological variables is a relevant problem for environmental monitoring and decision support in alert systems. This work presents a comparative analysis of classical machine-learning models for the hourly prediction of wildfire hotspot occurrence, based on the integration of INMET meteorological data and hotspot records from INPE's BDQueimadas system. Logistic Regression, Naive Bayes, linear SVM, Random Forest, and XGBoost were evaluated under different data-preparation scenarios, including original datasets, datasets with derived variables, KNN imputation, and class-imbalance strategies such as *SMOTE* and class weighting. The results showed that tree-ensemble models were the most suitable for the problem, especially XGBoost, and that feature engineering and explicit class-imbalance treatment contributed decisively to performance improvement, particularly in metrics more sensitive to positive-class detection, such as PR-AUC and F1-score.

**Keywords**: wildfires; Cerrado; hotspots; machine learning; INMET; INPE.

# Lista de ilustrações

# Lista de tabelas

# Sumário

# 1 Introdução

## 1.1 Contexto e motivação

As queimadas no Cerrado configuram um fenômeno recorrente e estruturado no tempo e no espaço, com sazonalidade marcada na estação seca e associação a tipos de cobertura da terra e vetores antrópicos (NASCIMENTO; ARAUJO; JUNIOR, 2011). Nesse contexto, abordagens orientadas por dados têm se mostrado adequadas para capturar padrões multivariados e não lineares relacionados à ocorrência de fogo, especialmente em tarefas de classificação e mapeamento de suscetibilidade (ANDRIANARIVONY; AKHLOUFI, 2024; FREITAS et al., 2025).

Para que estimativas preditivas sejam úteis em apoio à decisão, é necessário construir uma base integrada e metodologicamente consistente, combinando registros de focos do BDQueiamdas[1] com variáveis climáticas observadas pelo INMET [2] em resolução horária. Essa integração exige controle explícito de qualidade, padronização de formatos, compatibilização de chaves espaço-temporais e tratamento adequado de valores ausentes.

Em séries climáticas, dados faltantes são frequentes e podem afetar tanto estatísticas descritivas quanto o desempenho de modelos preditivos. Por essa razão, a literatura recomenda tratar a imputação como uma escolha metodológica com impacto potencial sobre os resultados, o que justifica a construção de cenários comparativos com e sem preenchimento de lacunas (ALEJO-SANCHEZ et al., 2025; AFRIFA-YAMOAH et al., 2020; RIBEIRO, 2021). Entre os métodos possíveis, a imputação por $k$-vizinhos mais próximos (KNN) constitui uma alternativa simples, transparente e reprodutível para avaliação de sensibilidade do pipeline de dados.

Nesse quadro, a comparação entre diferentes algoritmos de aprendizado de máquina torna-se relevante, sobretudo porque a previsão de focos de queimadas envolve relações complexas entre variáveis meteorológicas e forte desbalanceamento entre classes. Assim, a avaliação comparativa de modelos sob protocolos consistentes de treino, validação e teste permite identificar quais combinações de base, estratégia de balanceamento e algoritmo produzem melhor desempenho para a tarefa proposta.

## 1.2 Objetivos

O objetivo geral deste trabalho é avaliar, de forma comparativa, modelos de aprendizado de máquina para a previsão horária da ocorrência de focos de queimadas no Cerrado bra-

[1] BDQueimadas: <https://dataserver-coids.inpe.br/queimadas/queimadas/focos/csv/anual/Brasil_sat_ref/>
[2] INMET: <https://portal.inmet.gov.br/dadoshistoricos>

sileiro, a partir da integração entre dados do INMET e do BDQueimadas em um pipeline reprodutível.

Como objetivos específicos, pretende-se:

- construir uma base horária integrada INMET–BDQueimadas, com auditoria de qualidade, padronização de sentinelas e organização por cenários de dados;
- definir cenários sem imputação e com imputação por KNN em variáveis climaticamente relevantes, permitindo análise de sensibilidade quanto ao tratamento de valores ausentes;
- formular o problema como uma tarefa de classificação probabilística binária;
- comparar os modelos Regressão Logística, Random Forest, XGBoost, SVM linear e Naive Bayes (BREIMAN, 2001; CHEN; GUESTRIN, 2016);
- empregar um protocolo de avaliação compatível com a estrutura temporal dos dados, reportando métricas como PR-AUC, revocação, *F1-score*, ROC-AUC e *Brier score*;
- discutir os compromissos entre sensibilidade à classe positiva, capacidade de ranqueamento e custo operacional das diferentes estratégias avaliadas.

## 1.3 Organização

O texto organiza-se como segue. O Capítulo 2 apresenta a fundamentação teórica, abordando Inteligência Artificial, aprendizado de máquina, imputação de dados climáticos, desbalanceamento de classes e os modelos comparados. O Capítulo 3 descreve a metodologia, incluindo o pipeline de integração INMET–BDQueimadas, a auditoria de qualidade dos dados, a definição dos cenários com e sem imputação por KNN e o desenho experimental de comparação. O Capítulo 4 apresenta os resultados obtidos nas bases originais e nas bases com variáveis derivadas, seguido de sua discussão. Por fim, o Capítulo 5 reúne as conclusões do estudo, suas limitações e possibilidades de trabalhos futuros.

# 2 Fundamentação teórica

## 2.1 Inteligência Artificial e o problema proposto

De modo amplo, a Inteligência Artificial (IA) designa o campo científico e tecnológico que investiga como construir sistemas computacionais capazes de realizar tarefas associadas à inteligência, como raciocinar, aprender, resolver problemas e tomar decisões. Nessa perspectiva, a IA não se limita à imitação direta da cognição humana, mas envolve o desenvolvimento de métodos e artefatos capazes de produzir comportamento inteligente por meios computacionais (MCCARTHY, 2007; RUSSELL; NORVIG, 2021).

Entre os principais paradigmas desse campo, destaca-se o Aprendizado de Máquina (*Machine Learning*, ML), que desloca o foco de regras fixas para mecanismos de inferência a partir de dados. Em vez de especificar manualmente todas as relações entre entradas e saídas, busca-se ajustar modelos que aprendam padrões em exemplos históricos e aprimorem seu desempenho com a experiência (MITCHELL, 1997). Essa lógica é especialmente relevante em problemas como a previsão de focos de incêndio, nos quais a interação entre variáveis climáticas e contextuais é complexa, heterogênea e dificilmente capturada por formulações analíticas fechadas (ANDRIANARIVONY; AKHLOUFI, 2024).

A previsão de focos de incêndio no Cerrado a partir de variáveis climáticas se encaixa diretamente nesse enquadramento. Trata-se de um sistema multivariado, com relações não lineares e dependências espaço-temporais, para o qual existem séries históricas abundantes, mas não um conjunto de equações fechado que, sozinho, capte todas as interações relevantes. Assim, a abordagem orientada a dados é adequada: aprende-se, a partir de observações passadas, uma função que mapeia condições meteorológicas e contextuais para risco de ocorrência.

Além disso, há evidência empírica de que o fogo no Cerrado não é aleatório. Sua distribuição apresenta sazonalidade marcada, com concentração na estação seca, dependência do tipo de cobertura e associação com vetores de desmatamento, indicando padrões reprodutíveis que podem ser aprendidos por modelos (NASCIMENTO; ARAUJO; JUNIOR, 2011).

Quando esse problema é formulado em resolução espaço-temporal fina, entretanto, surge uma característica metodológica adicional de grande relevância: a ocorrência efetiva de fogo tende a constituir um evento relativamente raro em comparação ao grande volume de observações sem ocorrência. Em outras palavras, embora existam padrões ambientais e antrópicos associados ao fogo, a maior parte das instâncias observadas em séries históricas corresponde à classe negativa. Na literatura de predição de ocorrência de incêndios, esse comportamento aparece de modo recorrente e tem implicações diretas tanto para o treinamento quanto para a avaliação de classificadores, pois a assimetria entre as classes pode

induzir viés em favor da classe majoritária (PHELPS; WOOLFORD, 2021; PÉREZ-PORRAS et al., 2021).

A literatura sustenta a resolução do problema com aprendizado de máquina para classificação probabilística, em que estima-se $p(\text{FOCO} = 1 \mid \mathbf{x})$ e define-se o limiar de decisão conforme os custos operacionais. Pretende-se, portanto, a comparação de modelos supervisionados clássicos e *ensembles* de árvores, priorizando saídas calibráveis e interpretação alinhada ao domínio. Ao mesmo tempo, a natureza rara do evento positivo exige estratégias metodológicas adicionais para tratar o desbalanceamento de classes e selecionar adequadamente hiperparâmetros, aspectos que são discutidos nas seções posteriores.

## 2.2 Trabalhos relacionados

O conteúdo acadêmico recente aponta um crescimento expressivo do uso de técnicas de aprendizado de máquina e inteligência artificial na previsão de ocorrência e comportamento de incêndios florestais. Os trabalhos variam quanto ao tipo de tarefa (ocorrência versus severidade ou área queimada), à resolução espacial e temporal e ao conjunto de variáveis preditoras utilizadas, como meteorologia, topografia, combustível, fatores antrópicos e índices espectrais (ANDRIANARIVONY; AKHLOUFI, 2024).

Do ponto de vista ecológico e climatológico, estudos clássicos sobre o bioma Cerrado mostram que a distribuição espacial e temporal dos focos de calor não é aleatória. Existe uma forte sazonalidade associada ao regime de chuvas e à fenologia da vegetação, bem como influência de gradientes de uso do solo e de desmatamento (NASCIMENTO; ARAUJO; JUNIOR, 2011). Essa evidência de estruturas determinísticas e multimodais no padrão de fogo fundamenta o uso de modelos de aprendizado de máquina, que são capazes de explorar relações não lineares entre múltiplas variáveis explicativas.

No contexto brasileiro, Silva e White (SILVA; WHITE, 2016) utilizaram a série de focos do Programa Queimadas entre 1999 e 2015 para quantificar a incidência de fogo nos diferentes biomas. O estudo trabalha com agregações anuais e mensais e identifica diferenças marcantes na sazonalidade, com destaque para o Cerrado e o Pantanal. Jesus et al. (JESUS et al., 2020) avançaram nessa análise ao investigar a incidência temporal e espacial em biomas e unidades de conservação entre 2003 e 2017. A metodologia combinou estimativa de densidade Kernel para identificação de *hotspots* com testes de tendência de Mann-Kendall, indicando concentrações persistentes em transições de biomas, como a região Amazônia-Cerrado.

Já Alves et al. (ALVES et al., 2021) integraram focos de calor, reanálises climáticas e índices de grande escala para investigar a relação entre variáveis meteorológicas e a atividade de fogo. Embora esses trabalhos forneçam uma base sólida de caracterização do regime de fogo, eles se concentram em estatística descritiva ou correlações lineares, sem o emprego de algoritmos de aprendizado de máquina supervisionados para predição.

A aplicação direta de ML para estimativa de suscetibilidade foi explorada por Freitas et al. (FREITAS et al., 2025) na Amazônia. O trabalho utilizou modelos Random Forest e XGBoost calibrados com variáveis topográficas, climáticas e de uso da terra, resultando em mapas de suscetibilidade com desempenho elevado. No âmbito algorítmico, Shu et al. (SHU et al., 2021) desenvolveram o método Double Weighted Naive Bayes with Compensation Coefficient (DWCNB) para predição de fogo. O diferencial dessa abordagem é a atribuição de pesos tanto aos atributos quanto aos seus valores, atenuando a suposição de independência condicional. O modelo utiliza um coeficiente de compensação ajustado por testes ortogonais para equilibrar as probabilidades a priori, alcançando ganhos de acurácia significativos em relação ao Naive Bayes convencional.

Além da escolha do algoritmo, parte da literatura recente evidencia que tarefas de predição de ocorrência de incêndios também exigem atenção explícita à distribuição das classes. Em problemas nos quais os dias, áreas ou janelas espaço-temporais com fogo representam apenas uma pequena fração do conjunto total, a classe positiva torna-se minoritária, o que pode enviesar o processo de aprendizado em direção à não ocorrência. Nesse contexto, estratégias de rebalanceamento, geração sintética de exemplos e formulações sensíveis a custo têm sido empregadas como mecanismos para aumentar a capacidade de discriminação sobre a classe rara (PHELPS; WOOLFORD, 2021; PÉREZ-PORRAS et al., 2021; LING; SHENG, 2011).

Globalmente, revisões como a de Andrianarivony e Akhloufi (ANDRIANARIVONY; AKHLOUFI, 2024) indicam que modelos baseados em árvores de decisão e *ensembles* (Random Forest, Gradient Boosting e XGBoost) são os mais recorrentes para tarefas de risco de fogo devido à capacidade de lidar com variáveis heterogêneas. Métodos de boosting constroem sequências de modelos fracos de maneira iterativa para corrigir erros anteriores (FRIEDMAN, 2001). Outros estudos em regiões mediterrâneas e na China reforçam a importância de integrar variáveis climáticas com técnicas de explicabilidade (XAI) e estratégias de validação que respeitem a estrutura espacial e temporal dos dados (CILLI et al., 2022; PANG et al., 2022; AHAJJAM et al., 2025).

## 2.3 Qualidade de dados e imputação em séries climáticas

Em séries temporais climáticas, a presença de valores ausentes constitui um problema recorrente, decorrente de falhas de sensores, interrupções de transmissão, inconsistências de integração entre bases e diferenças na frequência de coleta. Em dados meteorológicos de alta resolução temporal, essas lacunas comprometem não apenas a análise descritiva das variáveis, mas também a construção de atributos derivados e o treinamento de modelos preditivos. Por essa razão, a imputação de dados deve ser compreendida como uma etapa metodológica relevante de reconstrução da base, e não apenas como um procedimento de limpeza (ALEJO-SANCHEZ et al., 2025; RIBEIRO, 2021; AFRIFA-YAMOAH et al., 2020).

A literatura recente indica que a imputação em séries climáticas depende fortemente da variável analisada, da estrutura temporal dos dados, da região de estudo e do mecanismo gerador das ausências. Por conseguinte, não há um método universalmente ótimo para todos os cenários. Em séries de precipitação, por exemplo, trabalhos comparativos mostram que o desempenho relativo dos métodos varia entre regiões semiáridas e úmidas, bem como entre estações meteorológicas distintas, o que reforça a necessidade de justificar a escolha metodológica em função das características do problema estudado (CESPEDES et al., 2023; ALEJO-SANCHEZ et al., 2025). No mesmo sentido, Ribeiro (RIBEIRO, 2021) destaca que a natureza da série temporal e o tipo de *missingness* devem orientar a seleção do método de imputação, uma vez que escolhas inadequadas podem afetar negativamente a etapa posterior de modelagem.

Entre os métodos multivariados de imputação, destaca-se o algoritmo *k*-Nearest Neighbors (KNN), ou *k*-vizinhos mais próximos. Sua lógica consiste em estimar um valor ausente a partir de observações consideradas semelhantes no espaço de atributos. Em vez de substituir a lacuna por uma estatística global, como média ou mediana, o KNN procura exemplos vizinhos à observação incompleta e utiliza os valores desses vizinhos para reconstruir a variável faltante. Essa estratégia preserva melhor a estrutura local dos dados, sendo particularmente útil quando há correlação entre variáveis meteorológicas, como temperatura, umidade relativa, precipitação e vento (TROYANSKAYA et al., 2001; ALEJO-SANCHEZ et al., 2025).

Formalmente, seja $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_p^{(i)})$ a $i$-ésima observação, com valor ausente na variável $j$. Define-se um conjunto de vizinhos $\mathcal{N}_k(i, j)$ formado pelas $k$ observações mais próximas de $\mathbf{x}^{(i)}$ que possuem valor observado em $x_j$. Uma formulação geral para a imputação é dada por:

$$\hat{x}_j^{(i)} = \frac{1}{k} \sum_{m \in \mathcal{N}_k(i,j)} x_j^{(m)}, \tag{2.1}$$

quando se utiliza média simples dos vizinhos. Em uma variante ponderada, pode-se atribuir maior influência aos vizinhos mais próximos:

$$\hat{x}_j^{(i)} = \frac{\sum_{m \in \mathcal{N}_k(i,j)} w_{im}\, x_j^{(m)}}{\sum_{m \in \mathcal{N}_k(i,j)} w_{im}}, \qquad w_{im} = \frac{1}{d(\mathbf{x}^{(i)}, \mathbf{x}^{(m)}) + \varepsilon}, \tag{2.2}$$

em que $d(\mathbf{x}^{(i)}, \mathbf{x}^{(m)})$ representa uma medida de distância entre observações e $\varepsilon > 0$ é uma constante pequena introduzida para evitar divisão por zero. Em aplicações com atributos numéricos, é comum empregar distância euclidiana calculada apenas sobre as coordenadas observadas em comum entre as duas instâncias.

Uma forma conveniente de representar essa distância, quando há valores ausentes, é:

$$d(i, m) = \sqrt{\frac{p}{|O_{im}|} \sum_{\ell \in O_{im}} \left( x_\ell^{(i)} - x_\ell^{(m)} \right)^2}, \tag{2.3}$$

em que $O_{im}$ é o conjunto de atributos observados simultaneamente nas observações $i$ e $m$, $|O_{im}|$ é sua cardinalidade e $p$ é o número total de atributos. Desse modo, a distância é ajustada para considerar apenas a informação efetivamente disponível em comum entre os pares de observações.

Do ponto de vista conceitual, a principal vantagem do KNN é sua flexibilidade: trata-se de um método não paramétrico, capaz de explorar relações locais entre múltiplas variáveis sem impor uma forma funcional rígida para a reconstrução dos dados. Em bases meteorológicas multivariadas, isso é particularmente relevante porque as lacunas em uma variável podem ser informadas, ao menos parcialmente, pelo comportamento conjunto das demais. Além disso, o método é relativamente simples de implementar e tem sido amplamente empregado como referência comparativa em estudos de imputação de séries climáticas (ALEJO-SANCHEZ et al., 2025; TROYANSKAYA et al., 2001).

Entretanto, o método também apresenta limitações. Por ser baseado em distâncias, o KNN é sensível à escala dos atributos, o que torna recomendável a padronização prévia das variáveis numéricas. Além disso, embora possa capturar similaridades multivariadas, o algoritmo não modela explicitamente a dependência temporal da série, operando principalmente a partir da proximidade no espaço de atributos. Assim, seu desempenho depende de fatores como densidade amostral, escolha de $k$, padrão de ausências e grau de correlação entre as variáveis disponíveis (RIBEIRO, 2021; ALEJO-SANCHEZ et al., 2025).

No contexto deste trabalho, a imputação por KNN se mostra teoricamente justificável porque as variáveis climáticas utilizadas apresentam dependência conjunta e são empregadas como insumo para a construção de atributos e para o treinamento dos classificadores supervisionados. Desse modo, a reconstrução local de valores ausentes busca preservar a coerência multivariada da base antes da etapa de modelagem.

## 2.4 Engenharia de atributos orientada ao domínio

Em problemas ambientais, variáveis meteorológicas observadas em alta resolução temporal nem sempre expressam, de forma direta, os mecanismos latentes que controlam a ignição e a manutenção de focos de fogo. Em particular, a propensão à queima depende não apenas do estado instantâneo da atmosfera, mas também do acúmulo de condições antecedentes de secura e do histórico recente de precipitação, que afetam o teor de umidade do combustível fino e, consequentemente, sua inflamabilidade. Assim, é comum adotar engenharia de atributos para condensar, em variáveis de entrada, informação temporal relevante, reduzindo ruído e oferecendo aos modelos uma representação mais aderente ao fenômeno.

Um referencial técnico importante no contexto brasileiro é o método institucional do Programa Queimadas do INPE, no qual o risco de fogo é fundamentado no princípio de que sequências mais longas sem chuva elevam a probabilidade de ocorrência de fogo, mas essa dependência é tratada por meio de uma variável de memória de precipitação. O documento

define o conceito de “Dias de Secura” (PSE), calculado a partir do histórico de precipitação antecedente, atribuindo maior peso às chuvas mais recentes e menor peso às mais antigas, de modo a representar o efeito cumulativo do regime hídrico sobre a secura do combustível ao longo do tempo. Além disso, o método considera ajustes por condições meteorológicas críticas, como temperatura máxima e umidade relativa mínima, e por fatores geofísicos e de cobertura, como vegetação, elevação e latitude, compondo um arcabouço físico-operacional para quantificação do risco (SETZER; SISMANOGLU; SANTOS, 2019).

Do ponto de vista de modelagem, a motivação para a engenharia de atributos é dupla. Primeiro, atributos derivados introduzem estrutura temporal e física que pode não emergir de forma eficiente a partir de variáveis instantâneas, especialmente em cenários com alta variabilidade meteorológica e forte heterogeneidade espacial. Segundo, ao fornecer aos modelos variáveis mais diretamente associadas a mecanismos do domínio, melhora-se a capacidade de discriminar padrões relevantes sem depender exclusivamente do poder de aproximação do algoritmo, o que é particularmente importante em dados ambientais com forte desbalanceamento de classes e com restrições operacionais de generalização espaço-temporal. Nesse sentido, a representação do fenômeno e a estratégia de treinamento não são aspectos dissociados: atributos mais informativos e mecanismos apropriados de tratamento do desbalanceamento atuam de forma complementar na construção de modelos mais sensíveis à ocorrência do evento raro.

## 2.5 Modelos de classificação supervisionada

No contexto deste trabalho, o problema de previsão de ocorrência de fogo é formulado como uma tarefa de classificação supervisionada binária: para cada instância representada por um vetor de atributos $\mathbf{x} \in \mathbb{R}^p$, associado a condições ambientais e antrópicas, busca-se estimar a probabilidade de ocorrência de um evento de interesse $y \in \{0, 1\}$, como, por exemplo, a presença de focos de calor em uma dada região e período. Em termos gerais, um modelo de aprendizado de máquina supervisionado tenta aproximar uma função $f_\theta : \mathbb{R}^p \to [0, 1]$ parametrizada por $\theta$, de modo que $f_\theta(\mathbf{x})$ represente uma estimativa de $P(Y = 1 \mid \mathbf{x})$ (MITCHELL, 1997; RUSSELL; NORVIG, 2016).

Além das variáveis observadas, os modelos recebem atributos derivados construídos para representar memória de precipitação, secura antecedente e regimes meteorológicos críticos, inspirados no racional físico-operacional adotado pelo Programa Queimadas do INPE (SETZER; SISMANOGLU; SANTOS, 2019). Com isso, busca-se alinhar a representação dos dados ao processo gerador do fenômeno antes da etapa de aprendizagem supervisionada. Diversos algoritmos podem ser empregados para construir essa aproximação a partir de um conjunto de treinamento rotulado. Neste trabalho são considerados modelos clássicos de classificação largamente utilizados em aplicações reais: Regressão Logística, Naive Bayes, Máquinas de Vetores de Suporte (SVM), Random Forest e XGBoost. Esta seção apresenta os

fundamentos teóricos de cada um desses métodos, com ênfase em seus princípios de funcionamento, hipóteses subjacentes e implicações para tarefas de previsão em dados ambientais. Para além da escolha do classificador, contudo, o desempenho em problemas de evento raro também depende de estratégias de tratamento do desbalanceamento e de seleção criteriosa de hiperparâmetros, discutidas na seção seguinte.

### 2.5.1 Regressão Logística

A Regressão Logística é um modelo discriminativo que estima diretamente a probabilidade condicional de uma classe binária a partir de uma combinação linear dos preditores. Seja $\mathbf{x} = (x_1, \ldots, x_p)$ o vetor de atributos e $y \in \{0, 1\}$ a variável resposta. O modelo assume que o logit da probabilidade de $Y = 1$ é uma função linear de $\mathbf{x}$:

$$\text{logit}\big(P(Y = 1 \mid \mathbf{x})\big) = \log\left(\frac{P(Y = 1 \mid \mathbf{x})}{1 - P(Y = 1 \mid \mathbf{x})}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j. \tag{2.4}$$

Aplicando-se a função logística, obtém-se a probabilidade modelada:

$$P(Y = 1 \mid \mathbf{x}) = \sigma(\beta_0 + \mathbf{x}^\top \boldsymbol{\beta}) = \frac{1}{1 + e^{-(\beta_0 + \mathbf{x}^\top \boldsymbol{\beta})}}. \tag{2.5}$$

Os parâmetros $\beta_0, \boldsymbol{\beta}$ são estimados geralmente por máxima verossimilhança, equivalente à minimização da perda logarítmica (entropia cruzada). Para um conjunto de dados $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$, a função de custo $\mathcal{L}$ é dada por:

$$\mathcal{L}(\beta_0, \boldsymbol{\beta}) = -\sum_{i=1}^{n} \Big[ y^{(i)} \log \hat{p}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{p}^{(i)}) \Big], \tag{2.6}$$

em que $\hat{p}^{(i)} = P(Y = 1 \mid \mathbf{x}^{(i)})$. A minimização é usualmente realizada por métodos numéricos iterativos, como gradiente descendente, Newton-Raphson ou variantes quasi-Newton.

A Regressão Logística apresenta duas vantagens importantes em aplicações científicas: (i) fornece probabilidades calibradas de ocorrência do evento, o que permite avaliar incertezas e definir limiares de decisão dependentes de custo; e (ii) permite interpretação direta dos coeficientes em termos de razão de chances (*odds ratio*): $e^{\beta_j}$ representa o fator de multiplicação na razão de chances associado a um incremento unitário em $x_j$, mantendo-se os demais atributos constantes. Em problemas ambientais, essa interpretabilidade é útil para quantificar o efeito relativo de variáveis climáticas ou de uso do solo sobre a probabilidade de fogo.

Para evitar sobreajuste, é comum empregar regularização L2 (*ridge*) ou L1 (*lasso*), que introduzem penalizações sobre a magnitude dos coeficientes, favorecendo modelos mais simples e, no caso da regularização L1, promovendo seleção automática de variáveis.

### 2.5.2 Classificadores Naive Bayes

Modelos Naive Bayes são classificadores probabilísticos baseados no Teorema de Bayes. Dado um conjunto de atributos $\mathbf{x}$ e uma variável de classe $Y$, tem-se:

$$P(Y = c \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid Y = c)P(Y = c)}{P(\mathbf{x})}, \tag{2.7}$$

em que $P(\mathbf{x})$ é o mesmo para todas as classes e pode ser tratado como constante no processo de classificação. A ideia central é modelar a distribuição dos atributos condicionada à classe, $P(\mathbf{x} \mid Y = c)$, e a distribuição a priori $P(Y = c)$.

O adjetivo *naive* decorre da hipótese de independência condicional entre os atributos dado o valor da classe:

$$P(\mathbf{x} \mid Y = c) = \prod_{j=1}^{p} P(x_j \mid Y = c). \tag{2.8}$$

Essa suposição é frequentemente irrealista em dados reais, especialmente em domínios ambientais em que variáveis meteorológicas e de vegetação são fortemente correlacionadas. Ainda assim, o modelo apresenta bom desempenho em diversas aplicações, sobretudo quando o número de atributos é grande em relação ao tamanho da amostra e quando as dependências entre variáveis não são extremamente críticas para a discriminação entre as classes.

Na prática, diferentes variantes são utilizadas conforme a natureza dos atributos: Naive Bayes gaussiano (atributos contínuos modelados como gaussianos condicionados à classe), multinomial (contagens) ou Bernoulli (atributos binários). A classificação consiste em escolher a classe que maximiza a probabilidade a posteriori $P(Y = c \mid \mathbf{x})$. Devido à sua formulação fechada, o treinamento é computacionalmente muito eficiente, o que o torna um bom modelo de referência (*baseline*) em experimentos comparativos.

### 2.5.3 Máquinas de Vetores de Suporte (SVM)

Máquinas de Vetores de Suporte são modelos discriminativos que buscam encontrar um hiperplano de separação entre classes com margem máxima (CORTES; VAPNIK, 1995). No caso linearmente separável, considera-se um hiperplano definido por $(\mathbf{w}, b)$ tal que:

$$\mathbf{w}^\top \mathbf{x} + b = 0, \tag{2.9}$$

e impõe-se que, para cada amostra rotulada $(\mathbf{x}^{(i)}, y^{(i)})$, com $y^{(i)} \in \{-1, +1\}$:

$$y^{(i)}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b\right) \geq 1. \tag{2.10}$$

A SVM procura o hiperplano que maximiza a margem, isto é, a distância entre as fronteiras de decisão e os pontos mais próximos (vetores de suporte). Isso equivale a resolver o problema de otimização:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{sujeito a} \quad y^{(i)}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b\right) \geq 1, \ \forall i. \tag{2.11}$$

Em situações não separáveis, introduzem-se variáveis de folga $\xi_i$ e um parâmetro de penalização $C > 0$ que controla o equilíbrio entre largura da margem e erros de classificação. A formulação dual do problema permite incorporar funções de *kernel* $K(\mathbf{x}, \mathbf{x}')$, o que viabiliza fronteiras de decisão não lineares sem necessidade de explicitar a transformação para espaços de alta dimensão ("truque do *kernel*").

SVMs são particularmente úteis quando há fronteira de decisão complexa em um espaço de atributos moderadamente dimensionado e quando se deseja maximizar a margem entre classes, o que tende a melhorar a capacidade de generalização. Em contrapartida, o custo computacional pode se tornar elevado em bases muito grandes, e a interpretação dos modelos é menos direta quando *kernels* não lineares são adotados.

### 2.5.4 Árvores de decisão e Random Forest

Árvores de decisão são modelos fundamentados em partições recursivas do espaço de atributos. O processo consiste em dividir o conjunto de dados em subconjuntos cada vez mais homogêneos em relação à variável alvo $y$. Para um nó $m$, representando um subconjunto dos dados $Q_m$ com $n_m$ observações, a proporção de instâncias da classe $k$ é definida como:

$$p_{mk} = \frac{1}{n_m} \sum_{\mathbf{x}^{(i)} \in Q_m} I(y^{(i)} = k), \tag{2.12}$$

em que $I(\cdot)$ é a função indicadora. A escolha do melhor ponto de corte em cada nó baseia-se em medidas de impureza. A métrica mais utilizada em algoritmos de classificação é o índice de Gini ($G$), que quantifica a probabilidade de uma instância escolhida aleatoriamente ser rotulada incorretamente, sendo expresso por:

$$G(m) = \sum_{k=1}^{K} p_{mk}(1 - p_{mk}) = 1 - \sum_{k=1}^{K} p_{mk}^2. \tag{2.13}$$

O algoritmo busca, para cada nó, a variável $j$ e o limiar $s$ que minimizam a impureza combinada dos nós filhos (esquerdo e direito), resolvendo o seguinte problema de otimização:

$$J(j, s) = \frac{n_{left}}{n_m} G(left) + \frac{n_{right}}{n_m} G(right). \tag{2.14}$$

Embora árvores individuais sejam modelos interpretáveis, elas tendem a apresentar alta variância, pois pequenas variações nos dados de treinamento podem resultar em estruturas

de decisão drasticamente diferentes. O Random Forest (BREIMAN, 2001) atenua essa limitação por meio de um método de comitê (*ensemble*) baseado em *bagging* (*bootstrap aggregating*). O algoritmo constrói $B$ árvores independentes, cada uma treinada em uma amostra *bootstrap* $Z^*$ obtida com reposição do conjunto original.

O diferencial do Random Forest em relação ao *bagging* convencional é a introdução do método de subespaços aleatórios: em cada divisão de cada árvore, o algoritmo seleciona apenas um subconjunto aleatório de $m \approx \sqrt{p}$ atributos para considerar como candidatos ao corte. Essa técnica reduz a correlação entre as árvores da floresta, garantindo que o erro médio do comitê seja menor que o erro médio de suas árvores individuais.

A predição final para uma nova instância $\mathbf{x}$ é obtida por votação majoritária no caso de classificação:

$$\hat{y} = \text{moda}\{\hat{C}_1(\mathbf{x}), \hat{C}_2(\mathbf{x}), \ldots, \hat{C}_B(\mathbf{x})\}, \tag{2.15}$$

em que $\hat{C}_b(\mathbf{x})$ é a classe prevista pela $b$-ésima árvore. Além da robustez preditiva, o modelo permite estimar a importância de cada atributo via *Mean Decrease Gini*, que calcula a redução total na impureza dos nós trazida por uma variável específica ao longo de todas as árvores da floresta, fornecendo interpretação relevante para o domínio do problema (RUSSELL; NORVIG, 2016).

### 2.5.5 Gradient Boosting e XGBoost

Ao passo que o *bagging* reduz a variância ao combinar modelos treinados de forma independente, métodos de *boosting* constroem sequências de modelos fracos de maneira iterativa, cada novo modelo focando em corrigir os erros cometidos pelos anteriores (FRIEDMAN, 2001). No Gradient Boosting, em particular, formula-se a tarefa de aprendizado como um problema de minimização de uma função de perda diferenciável $\mathcal{L}(y, F(\mathbf{x}))$. Parte-se de um modelo inicial $F_0$ e, a cada iteração $m$, ajusta-se um novo modelo fraco $h_m$ aos resíduos negativos do gradiente da perda em relação às previsões correntes, atualizando:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \cdot h_m(\mathbf{x}), \tag{2.16}$$

em que $\nu \in (0, 1]$ é a taxa de aprendizado. Quando os modelos fracos $h_m$ são árvores de decisão de pequena profundidade, obtém-se um *ensemble* capaz de capturar interações complexas entre atributos.

O XGBoost (*Extreme Gradient Boosting*) (CHEN; GUESTRIN, 2016) é uma implementação otimizada de Gradient Boosting com árvores, amplamente utilizada em aplicações práticas pela combinação de alto desempenho preditivo e eficiência computacional. Entre suas principais características estão: (i) regularização explícita da complexidade das árvores na função objetivo, controlando profundidade e número de folhas; (ii) uso de aproximações de segunda ordem (gradiente e hessiano) para acelerar o processo de otimização; (iii) suporte

nativo a paralelismo, amostragem de instâncias e de atributos; e (iv) tratamento eficiente de valores ausentes.

Em tarefas de previsão de incêndios, XGBoost e métodos similares de Gradient Boosting têm se mostrado particularmente competitivos, pois conseguem explorar relações não lineares entre variáveis meteorológicas, de combustíveis e de uso do solo, ao mesmo tempo em que lidam bem com atributos heterogêneos e correlações complexas (ANDRIANARIVONY; AKHLOUFI, 2024).

## 2.6 Desbalanceamento de classes e estratégias de treinamento

### 2.6.1 Desbalanceamento de classes em predição de focos de incêndio

Em problemas de classificação binária, diz-se que há desbalanceamento de classes quando a distribuição da variável resposta é marcadamente assimétrica, isto é, quando uma das classes ocorre com frequência muito superior à outra. Em tarefas de predição de ocorrência de focos de incêndio, essa situação é particularmente plausível porque o evento de interesse, embora relevante do ponto de vista operacional e ambiental, tende a ocorrer em uma fração reduzida das observações quando se considera uma malha espaço-temporal ampla. Como consequência, o conjunto de dados passa a ser dominado por exemplos da classe negativa, enquanto os exemplos positivos permanecem escassos (PHELPS; WOOLFORD, 2021; PÉREZ-PORRAS et al., 2021).

Esse aspecto não é meramente descritivo, mas afeta diretamente o processo de aprendizagem. Em termos intuitivos, se a maioria das observações pertence à classe sem ocorrência, então um classificador pode reduzir sua perda empírica simplesmente favorecendo essa classe majoritária, mesmo que isso implique sensibilidade insuficiente para detectar o evento raro. Em termos estatísticos, a assimetria faz com que a função objetivo seja fortemente influenciada pelos exemplos majoritários, deslocando a fronteira de decisão e degradando o reconhecimento da classe minoritária. Em razão disso, a literatura sobre aprendizado em bases desbalanceadas enfatiza que o problema não deve ser tratado apenas na etapa de avaliação, mas também durante o treinamento do modelo (LING; SHENG, 2011).

No contexto específico da predição de fogo, essa discussão é especialmente importante porque a classe positiva concentra justamente o fenômeno de maior interesse científico e prático. Um modelo que apresente bom desempenho agregado, mas baixa capacidade de identificar condições associadas à ocorrência de fogo, tende a ter utilidade limitada como ferramenta de suporte à decisão.

### 2.6.2 Sobreamostragem sintética com SMOTE

Uma das estratégias mais difundidas para lidar com bases desbalanceadas consiste em aumentar a representatividade da classe minoritária no conjunto de treinamento. Entre essas

abordagens, destaca-se a técnica *Synthetic Minority Over-sampling Technique* (SMOTE), que busca ampliar a presença da classe rara por meio da geração de exemplos sintéticos em vez de simplesmente replicar observações existentes. Em aplicações à predição de incêndios, esse tipo de estratégia tem sido empregado como forma de reduzir o viés em direção à classe majoritária e melhorar a capacidade discriminativa do modelo (PÉREZ-PORRAS et al., 2021).

Do ponto de vista conceitual, o SMOTE opera no espaço de atributos. Para cada instância da classe minoritária, selecionam-se vizinhos próximos pertencentes à mesma classe e geram-se novas amostras sintéticas ao longo do segmento que conecta essas observações. O resultado é uma expansão local da região ocupada pela classe minoritária, de modo a tornar o padrão positivo mais representado durante o ajuste do classificador. Diferentemente da simples duplicação de casos, que preserva exatamente os mesmos pontos e pode favorecer sobreajuste, a sobreamostragem sintética introduz novas combinações plausíveis entre exemplos vizinhos, preservando a estrutura local da minoria.

A motivação para o uso do SMOTE neste trabalho decorre de sua capacidade de atuar diretamente sobre a distribuição de treinamento, oferecendo ao algoritmo mais evidência sobre o evento raro sem modificar o conjunto de teste. Em termos práticos, isso pode tornar o modelo mais sensível às condições associadas à ocorrência de fogo e favorecer a aprendizagem de fronteiras de decisão menos enviesadas em direção à classe negativa. Essa estratégia é particularmente pertinente quando se deseja comparar algoritmos sob uma configuração em que a assimetria de frequência seja mitigada pela própria composição do conjunto de treinamento.

### 2.6.3 Aprendizado sensível a custo e balanceamento por peso

Uma segunda família de abordagens para tratamento do desbalanceamento parte da ideia de que nem todos os erros de classificação devem receber a mesma penalização. Em muitos problemas reais, classificar incorretamente um exemplo da classe minoritária pode ter impacto operacional mais relevante do que cometer o erro oposto. Essa é a lógica do aprendizado sensível a custo (*cost-sensitive learning*), no qual a função de perda do modelo é modificada para refletir custos distintos entre classes ou entre tipos de erro (LING; SHENG, 2011).

No caso particular de classificação binária com forte assimetria entre classes, uma forma comum de implementar essa ideia é o balanceamento por peso (*class weighting*). Nessa formulação, observações da classe minoritária recebem peso maior durante o treinamento, fazendo com que erros sobre essa classe contribuam mais intensamente para a função objetivo. Em modelos probabilísticos baseados em perda logarítmica, por exemplo, a função de custo pode ser escrita de forma ponderada como:

$$\mathcal{L}_w = -\sum_{i=1}^{n} w_{y^{(i)}} \left[ y^{(i)} \log \hat{p}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{p}^{(i)}) \right], \tag{2.17}$$

em que $w_{y^{(i)}}$ representa o peso associado à classe da $i$-ésima observação. Quando a classe positiva recebe peso maior, o algoritmo é induzido a prestar maior atenção aos exemplos raros, reduzindo a tendência de privilegiar a classe majoritária apenas por sua maior frequência.

Essa estratégia apresenta uma diferença importante em relação ao SMOTE. Enquanto a sobreamostragem sintética altera a composição do conjunto de treinamento, o balanceamento por peso preserva a distribuição observada dos exemplos, atuando diretamente sobre o critério de otimização. Em termos metodológicos, essa distinção é relevante porque permite comparar duas lógicas complementares de mitigação do desbalanceamento: uma baseada na reconfiguração da amostra de treinamento e outra baseada na reponderação do erro durante o ajuste do modelo. Por essa razão, o balanceamento por peso é incorporado neste estudo como alternativa teórica e experimentalmente justificável para o tratamento de classes raras (LING; SHENG, 2011).

### 2.6.4 Seleção de hiperparâmetros com validação cruzada e Grid Search

Além do tratamento do desbalanceamento, o desempenho de modelos supervisionados depende da escolha adequada de hiperparâmetros, isto é, configurações definidas antes do treinamento propriamente dito e que controlam aspectos estruturais do algoritmo. Exemplos incluem a intensidade da regularização na Regressão Logística, o parâmetro $C$ em SVMs, a profundidade máxima de árvores, o número de estimadores em *ensembles* e a taxa de aprendizado em métodos de *boosting*. A escolha desses valores influencia a capacidade de generalização do modelo e, por isso, não deve ser feita de forma arbitrária (BERGSTRA; BENGIO, 2012).

Uma estratégia tradicional para essa tarefa é o *Grid Search*, no qual se define previamente uma grade discreta de valores candidatos para cada hiperparâmetro e se avaliam sistematicamente todas as combinações possíveis. Embora nem sempre seja a abordagem mais eficiente do ponto de vista computacional, trata-se de um procedimento simples, reprodutível e amplamente empregado como referência para seleção de hiperparâmetros (BERGSTRA; BENGIO, 2012). Quando associado à validação cruzada, esse processo permite estimar, para cada combinação testada, o desempenho médio do modelo em subconjuntos distintos dos dados de treinamento.

Na validação cruzada, o conjunto de treinamento é particionado em subconjuntos, ou *folds*. Em uma configuração do tipo $k$-*fold*, o modelo é treinado sucessivamente em $k - 1$ partições e validado na partição remanescente, repetindo-se o processo até que todos os *folds* tenham atuado como validação. A média do desempenho obtido ao longo dessas iterações fornece uma estimativa mais robusta da capacidade de generalização associada a determinada configuração de hiperparâmetros. No caso do *GridSearchCV*, esse mecanismo é combinado a uma busca exaustiva sobre a grade de parâmetros especificada, automatizando

a seleção do arranjo mais promissor segundo uma métrica previamente definida (scikit-learn developers, 2026).

Portanto, a seleção de hiperparâmetros por validação cruzada com *Grid Search* se justifica, primeiro, pois reduz a arbitrariedade na comparação entre modelos, uma vez que as configurações são escolhidas por um procedimento sistemático. Segundo, melhora a reprodutibilidade experimental, pois a grade de busca e o critério de seleção podem ser explicitamente documentados. Terceiro, permite que o ajuste de hiperparâmetros seja realizado no interior do processo de treinamento, sem contaminar o conjunto de teste final, que deve permanecer reservado para a estimativa de desempenho fora da amostra. Desse modo, a comparação entre configurações com base original, SMOTE e balanceamento por peso torna-se metodologicamente mais consistente.

## 2.7 Avaliação de modelos de classificação

A avaliação de classificadores binários parte da comparação entre rótulos preditos e rótulos observados, o que permite construir a matriz de confusão e derivar métricas como precisão, revocação, especificidade, acurácia e $F_1$-score. Essas medidas descrevem diferentes aspectos do desempenho do modelo e são particularmente úteis quando interpretadas em conjunto, pois evidenciam o compromisso entre detecção de eventos positivos e incidência de erros de classificação.

No problema estudado, a ocorrência de focos constitui um evento raro, de modo que a acurácia, isoladamente, tem utilidade limitada. Em cenários fortemente desbalanceados, um classificador pode atingir acurácia elevada mesmo sem identificar adequadamente a classe positiva. Por essa razão, tornam-se mais relevantes métricas que enfatizam o desempenho sobre os casos de interesse, especialmente precisão e revocação, bem como medidas derivadas dessas quantidades.

Para classificadores que produzem escores ou probabilidades, duas famílias de medidas são particularmente importantes: a área sob a curva ROC (ROC-AUC) e a área sob a curva Precisão-Revocação (PR-AUC). A ROC-AUC resume a capacidade global de discriminação entre as classes (FAWCETT, 2006). Contudo, em bases muito desbalanceadas, a curva ROC pode sugerir desempenho excessivamente otimista, pois a taxa de falsos positivos pode permanecer numericamente baixa mesmo quando o número absoluto de falsos positivos é elevado. Nesses casos, a curva Precisão-Revocação tende a ser mais informativa, por enfatizar diretamente o comportamento do modelo sobre a classe positiva rara (SAITO; REHMSMEIER, 2015; DAVIS; GOADRICH, 2006).

Assim, neste trabalho, a PR-AUC recebe maior ênfase interpretativa como principal métrica de comparação entre modelos, por ser mais aderente à natureza desbalanceada do problema e à necessidade de avaliar a qualidade da identificação de focos. A ROC-AUC é reportada como medida complementar de separabilidade global, enquanto precisão, revocação,

$F_1$-score, especificidade e matriz de confusão são utilizadas para qualificar o desempenho em termos operacionais, em limiares específicos de decisão.

Como os dados possuem estrutura temporal, a avaliação também deve respeitar essa característica, de modo que as estimativas de desempenho reflitam, de forma mais realista, a capacidade de generalização dos modelos em observações futuras (ANDRIANARIVONY; AKHLOUFI, 2024).

# 3 Metodologia

## 3.1 Formulação do problema e visão geral da metodologia

A metodologia adotada tem como objetivo construir, enriquecer e organizar um conjunto de dados horário que permita a predição de focos de queimada no bioma Cerrado a partir de variáveis climáticas medidas por estações automáticas do Instituto Nacional de Meteorologia (INMET). As principais entradas são os arquivos anuais de focos de calor do sistema BDQueimadas, mantido pelo Instituto Nacional de Pesquisas Espaciais (INPE), e as séries históricas horárias das estações automáticas do INMET. As saídas são bases integradas do tipo `inmet_bdq_{ANO}_cerrado.csv` e suas variações em formato `.parquet`, estruturadas para suportar experimentos de classificação binária e o uso operacional de modelos que recebem apenas dados meteorológicos em tempo quase real.

O problema de interesse consiste em estimar, para cada combinação *município* x *hora*, a ocorrência de pelo menos um foco de queimada nesse conjunto espaço-tempo, tomando como preditores um conjunto de variáveis climáticas observadas na mesma escala temporal. A variável alvo é definida como binária, $y \in \{0, 1\}$, em que $y = 1$ indica a existência de foco de calor no intervalo horário e município correspondentes e $y = 0$ indica ausência de detecção. Os dados de focos de calor são utilizados exclusivamente na fase de construção do conjunto rotulado, enquanto a fase de predição supõe disponibilidade apenas das variáveis do INMET. A distribuição de classes é fortemente desbalanceada, com número de instâncias sem foco muito superior ao de instâncias com foco.

A metodologia organiza-se em quatro macroetapas: (i) obtenção e padronização das bases BDQueimadas e INMET, (ii) processamento e integração espaço temporal com definição da variável alvo, (iii) auditoria de dados ausentes, construção de cenários de modelagem e engenharia de atributos físico inspirada e (iv) treinamento e comparação de algoritmos de classificação supervisionada. As três primeiras etapas são inteiramente automatizadas por scripts em Python disponíveis em repositório público[1] e produzem todos os conjuntos de dados utilizados nos experimentos de aprendizagem de máquina, inclusive versões enriquecidas com variáveis derivadas como chuva ponderada, dias consecutivos sem precipitação, limiares críticos de temperatura e umidade e fatores de propagação do fogo.

O estudo considera cinco algoritmos de classificação supervisionada largamente empregados em problemas de risco: Regressão Logística, Naive Bayes, Máquina de Vetores de Suporte (SVM), Random Forest e XGBoost. A comparação entre esses modelos é orientada por três hipóteses de trabalho: (i) modelos de ensemble baseados em árvores, como Random Forest e XGBoost, tendem a capturar melhor interações não lineares entre variáveis

[1] Disponível em <https://github.com/julianopadua/TCC>.

meteorológicas e a resposta binária, (ii) a inclusão da variável de radiação global e de atributos físico inspirados tem potencial para aumentar o poder preditivo quando o tratamento de dados ausentes é adequado e (iii) estratégias de imputação que preservam instâncias com foco de queimada, como o *K-Nearest Neighbors* (KNN), tendem a produzir ganhos relevantes em métricas sensíveis a eventos raros quando comparadas à simples exclusão de linhas incompletas.

A abordagem está sujeita a algumas restrições estruturais. A integração é realizada em escala municipal, dependente de um dicionário que associa municípios a biomas, e limita-se à interseção temporal entre as fontes BDQueimadas e INMET a partir de 2003. A resolução horária implica que dinâmicas intra-hora de ignição e extinção de focos não são representadas explicitamente na base, o que impõe um limite natural à granularidade temporal das predições.

## 3.2 Obtenção e padronização das bases BDQueimadas e INMET

A composição do conjunto de dados parte da integração de duas fontes principais: (i) o sistema BDQueimadas, do INPE, que fornece registros de focos de calor derivados de sensores orbitais e (ii) as séries históricas das estações automáticas do INMET, que disponibilizam variáveis meteorológicas horárias.

Os dados de focos de calor provêm da coleção `Brasil_sat_ref` do BDQueimadas, acessada por meio do servidor *dataserver* COIDS. O módulo `<src/bdqueimadas_scraper.py>` automatiza o download dos arquivos anuais comprimidos `focos_br_ref_YYYY.zip`, filtrando apenas os nomes compatíveis com o padrão esperado. A Listagem 3.1 ilustra o procedimento de descoberta e filtragem desses arquivos no servidor remoto.

**Listing 3.1 –** Descoberta e filtragem de arquivos anuais no BDQueimadas

```python
def discover_bdq_zip_links(base_url: str = BDQ_BASE_URL) -> List[str]:
    session = get_requests_session()
    links = list_zip_links_from_page(base_url, session=session)
    out = []
    for h in links:
        abs_url = urljoin(base_url, h)
        name = os.path.basename(abs_url.split("?")[0]).lower()
        if name.endswith(".zip") and name.startswith("focos_br_ref_"):
            out.append(abs_url)
    out = sorted(set(out))
    log.info(f"{len(out)} .zip detectados em {base_url}")
    return out
```

Os arquivos listados são armazenados em `<data/raw/ID_BDQUEIMADAS/>` e extraídos para `<data/processed/ID_BDQUEIMADAS/>`, preservando subdiretórios por ano. Paralela-

mente, utiliza-se a exportação manual `exportador_..._ref_YYYY.csv`, obtida via interface do BDQueimadas, que contém variáveis adicionais como o Risco de Fogo, o *Fire Radiative Power* (FRP) e o bioma associado a cada foco. Esses arquivos são armazenados em `data/raw/BDQUEIMADAS/` e utilizados na etapa de consolidação descrita na Seção 3.3.

Como resultado da consolidação e do filtro espacial para o Cerrado, obteve-se o arquivo `bdq_targets_2003_2025_cerrado.csv`, com 1.349.101 linhas, 8 colunas e tamanho em disco de 135,53 MB. Esse consolidado reúne, em nível de registro de foco, a marca temporal do evento, identificadores do BDQueimadas, informações administrativas de localização e atributos auxiliares associados ao foco, como risco de fogo e potência radiativa. A Tabela 1 resume a estrutura desse arquivo.

Tabela 1 – Estrutura do consolidado BDQueimadas utilizado na etapa de integração.

| **Coluna** | **Tipo de dado** | **Descrição** |
|---|---|---|
| `DATAHORA` | data-hora | Marca temporal associada ao registro do foco de calor. |
| `ID_BDQ` | inteiro | Identificador interno do registro no consolidado do BDQueimadas. |
| `FOCO_ID` | texto | Identificador individual do foco de calor. |
| `PAIS` | texto | País ao qual o registro do foco está associado. |
| `ESTADO` | texto | Unidade da federação do registro do foco. |
| `MUNICIPIO` | texto | Município associado ao foco de calor. |
| `RISCO_FOGO` | real | Valor de risco de fogo disponibilizado na exportação enriquecida do BDQueimadas. |
| `FRP` | real | *Fire Radiative Power*, medida associada à intensidade radiativa do foco. |

As variáveis climáticas foram obtidas a partir das estações automáticas do INMET, por meio de rotinas de *scraping* que acessam o portal de dados históricos. O módulo `<inmet_scraper.py>` descobre os arquivos `.zip` disponíveis, faz o download para um diretório bruto configurado e extrai os conteúdos para `data/providers/INMET/raw/csv/`. Em seguida, o procedimento `process_inmet_year` consolida, para cada ano, todos os arquivos das estações em um único arquivo `inmet_YYYY.csv` com metainformação uniforme. A Listagem 3.2 resume os passos principais dessa consolidação.

**Listing 3.2 –** Consolidação anual das séries INMET

```
def process_inmet_year(year: int,
                       drop_cols: Optional[list[str]] = None,
                       overwrite: bool = False) -> Optional[Path]:
    log = get_logger("inmet.load", kind="load", per_run_file=True)
```

```
    year_dir = _inmet_year_dir(year)
    if not year_dir:
        log.warning(f"[WARN] Pasta do ano {year} não encontrada em INMET/csv.")
        return None

    proc_dir = get_path("paths", "providers", "inmet", "processed")
    ensure_dir(proc_dir)

    cfg = loadConfig()
    patt = (cfg.get("filenames", {})
              .get("patterns", {})
              .get("inmet_csv", "inmet_{year}.csv"))
    out_path = Path(proc_dir) / patt.format(year=year)

    files = sorted([*year_dir.glob("*.CSV"), *year_dir.glob("*.csv")])
    dfs = []
    for fp in files:
        header, cidade, lat, lon = _parse_inmet_header(fp)
        df = pd.read_csv(fp, sep=";", skiprows=9,
                         encoding="latin1", engine="python",
                         on_bad_lines="skip")
        df.columns = header
        df["ANO"] = year
        df["CIDADE"] = cidade
        df["LATITUDE"] = lat
        df["LONGITUDE"] = lon
        dfs.append(df)
        log.info(f"[READ] {fp.name}")

    final = pd.concat(dfs, ignore_index=True)
    final.to_csv(out_path, index=False, encoding="utf-8")
    log.info(f"[WRITE] {out_path}")
    return out_path
```

Esse procedimento padroniza o formato das séries horárias, adiciona identificadores de ano e localização e remove colunas estritamente redundantes ou derivadas. O resultado da etapa são arquivos anuais consolidados para o INMET e um conjunto de arquivos anuais do BDQueimadas com conteúdo enriquecido, prontos para a etapa de integração.

Após a consolidação das séries horárias e a filtragem espacial para o bioma Cerrado, o arquivo `inmet_all_years_cerrado.csv` totalizou 49.922.606 linhas, 15 colunas, cobertura temporal de 2000 a 2025 e tamanho em disco de 4,96 GB. A Tabela 2 sintetiza a estrutura final desse consolidado. Entre as variáveis meteorológicas, destaca-se a coluna de radiação global, que apresentou 48,8956% de valores faltantes, constituindo o principal ponto crítico de completude entre os atributos climáticos e motivando sua avaliação explícita em cenários distintos de modelagem.

Tabela 2 – Estrutura do consolidado INMET utilizado na etapa de integração.

| **Coluna** | **Tipo de dado** | **Descrição** |
|---|---|---|
| ANO | inteiro | Ano de referência do registro meteorológico. |
| DATA (YYYY-MM-DD) | data | Data da observação horária. |
| HORA (UTC) | hora | Hora da observação em UTC. |
| CIDADE | texto | Município associado à estação meteorológica. |
| LATITUDE | real | Latitude da estação meteorológica. |
| LONGITUDE | real | Longitude da estação meteorológica. |
| TEMPERATURA DO AR - BULBO SECO, HORARIA (°C) | real | Temperatura horária do ar medida na estação. |
| TEMPERATURA DO PONTO DE ORVALHO (°C) | real | Temperatura horária do ponto de orvalho. |
| UMIDADE RELATIVA DO AR, HORARIA (%) | real | Umidade relativa horária do ar. |
| PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO, HORARIA (mB) | real | Pressão atmosférica horária ao nível da estação. |
| VENTO, DIREÇÃO HORARIA (gr) (° (gr)) | real | Direção horária do vento em graus. |
| VENTO, RAJADA MAXIMA (m/s) | real | Rajada máxima horária do vento. |
| VENTO, VELOCIDADE HORARIA (m/s) | real | Velocidade horária do vento. |
| RADIACAO GLOBAL (KJ/m²) | real | Radiação global horária incidente. |
| PRECIPITAÇÃO TOTAL, HORÁRIO (mm) | real | Precipitação total acumulada no intervalo horário. |

## 3.3 Processamento, integração espaço temporal e definição da variável alvo

O fluxo de *Extract Transform Load* (ETL) foi projetado para correlacionar eventos discretos de fogo com registros climáticos contínuos em escala horária e municipal. A Figura 1 apresenta a visão geral desse pipeline, desde a ingestão das fontes brutas até a produção das

bases integradas de uso em modelagem.

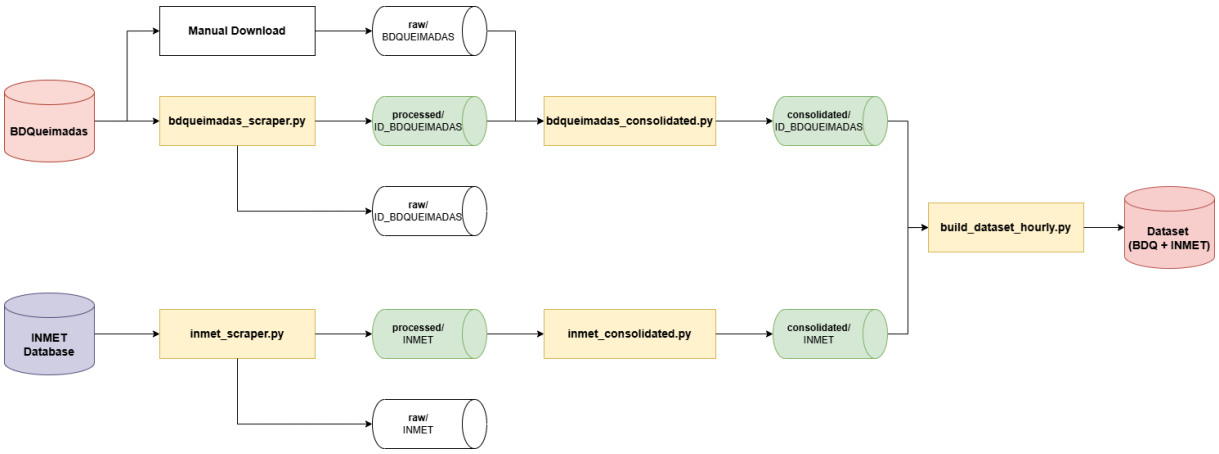



**Figura 1 –** Fluxo geral do pipeline de dados integrando BDQueimadas e INMET.
**Fonte:** elaborado pelo autor.

A consolidação dos focos de calor é realizada no módulo <`src/consolidated_bdqueimadas.py`>. Os arquivos manuais `exportador_..._ref_YYYY.csv` são lidos e normalizados, com conversão de datas para objetos temporais, criação de colunas de saída sem diacríticos e construção de chaves de junção robustas baseadas em data, país, unidade da federação e município. Em paralelo, os arquivos processados `focos_br_ref_YYYY.csv`, extraídos a partir dos `.zip` anuais, são lidos e reduzidos às colunas de interesse (`id_bdq`, `foco_id`, latitude, longitude), com deduplicação por chave. A junção 1:1 é concretizada por meio da chave sintética `__KEY`, mostrada na Listagem 3.3.

**Listing 3.3 –** Chave sintética para junção horário x local no BDQueimadas

```
df["__KEY"] = (
    df["__DT_H"].astype("int64").astype("string") + "|" +
    df["__PAIS_KEY"].astype(str) + "|" +
    df["__UF_KEY"].astype(str) + "|" +
    df["__MUN_KEY"].astype(str)
)
```

Após o *merge*, o módulo organiza, para cada ano, um arquivo `bdq_targets_YYYY_[bioma].csv` contendo, por linha, a combinação hora x município, o Risco de Fogo, o FRP e identificadores `ID_BDQ` e `FOCO_ID`. Um filtro opcional por bioma utiliza chaves normalizadas de município para restringir o conjunto ao Cerrado.

Do lado meteorológico, o módulo <`src/inmet_consolidated.py`> toma os arquivos `inmet_YYYY.csv` processados e gera arquivos consolidados por ano em `data/external/INMET/`, com opção de filtragem por bioma. Um dicionário CSV (`bdq_municipio_bioma.csv`) associa municípios a biomas, permitindo selecionar apenas os registros de estações cuja cidade pertence ao Cerrado. Durante essa consolidação, normaliza-se a representação de datas para

o formato `YYYY-MM-DD` e removem-se linhas em que todas as variáveis de medida assumem valores sentinela (`-999` ou `-9999`), o que reduz o impacto de falhas sistemáticas de sensores.

A etapa de integração espaço temporal é implementada no módulo <`src/build_dataset_hourly.py`>. Para cada ano e bioma, são lidos os arquivos `inmet_YYYY_cerrado.csv` e `bdq_targets_YYYY_cerrado.csv`. No caso do INMET, o código detecta automaticamente o esquema de colunas de data e hora adotado (padrão antigo ou formato pós 2019) e constrói uma marca temporal horária `ts_hour = YYYY-MM-DD HH:00:00`. Para o BDQueimadas, os focos são reamostrados de forma a obter uma única linha por combinação município normalizado x hora, escolhendo o foco com maior FRP quando há múltiplas detecções no mesmo intervalo.

A fusão das fontes é realizada por uma junção entre o `ts_hour` e as chaves de município normalizado em cada base, conforme ilustrado na Listagem 3.4. Esse procedimento define a variável alvo binária `HAS_FOCO`.

**Listing 3.4 –** Integração horária INMET x BDQueimadas e definição de HAS_FOCO

```python
def _fuse_inmet_bdq_year(year: int, biome: str,
                         out_dir: Path,
                         encoding: str = "utf-8") -> Path:
    inmet = _read_inmet_year(year, biome, encoding=encoding)
    bdq = _read_bdq_year_reduced(year, biome, encoding=encoding)

    merged = inmet.merge(
        bdq,
        left_on=["cidade_norm", "ts_hour"],
        right_on=["municipio_norm", "ts_hour"],
        how="left",
        suffixes=("", "_bdq"),
    )

    merged["HAS_FOCO"] = merged["FOCO_ID"].notna().astype("int64")
    merged = merged.drop(columns=["municipio_norm"], errors="ignore")

    date_col = inmet.attrs.get("date_col")
    hour_col = inmet.attrs.get("hour_col")
    if date_col and hour_col and date_col in merged.columns and hour_col in
        merged.columns:
        merged = merged.sort_values([date_col, hour_col, "CIDADE"], kind="stable")

    out_path = out_dir / f"inmet_bdq_{year}_{biome}.csv"
    merged.to_csv(out_path, index=False, encoding=encoding)
    return out_path
```

O módulo `build_hourly_dataset` aplica essa fusão para todos os anos em que há interseção entre as séries consolidadas de INMET e BDQueimadas, restringindo-se a anos

a partir de 2003. Os arquivos anuais `inmet_bdq_YYYY_cerrado.csv` são gravados em `data/dataset/` e, em seguida, concatenados em um arquivo único <`inmet_bdq_all_years_cerrado.csv`>, que é a base integrada utilizada nas etapas de auditoria e modelagem.

Após a concatenação anual, a base integrada `inmet_bdq_all_years_cerrado.csv` reuniu 45.135.924 linhas e 23 colunas, cobrindo o período de 2003 a 2024, com tamanho em disco de 6,18 GB. A variável alvo `HAS_FOCO` apresentou 151.544 observações positivas e 44.984.380 observações negativas, o que corresponde a uma proporção positiva de 0,3358%. Esses valores confirmam o forte desbalanceamento da tarefa de predição proposta. A Tabela 3 resume essas características gerais da base final.

Tabela 3 – Resumo da base integrada final utilizada na modelagem.

| **Característica** | **Valor** |
|---|---|
| Arquivo consolidado final | `inmet_bdq_all_years_cerrado.csv` |
| Período temporal | 2003 a 2024 |
| Número total de linhas | 45.135.924 |
| Número total de colunas | 23 |
| Tamanho em disco | 6,18 GB |
| Observações com `HAS_FOCO = 1` | 151.544 |
| Observações com `HAS_FOCO = 0` | 44.984.380 |
| Proporção da classe positiva | 0,3358% |

## 3.4 Auditoria e construção de cenários de modelagem

A base integrada final `inmet_bdq_all_years_cerrado.csv` apresenta padrões distintos de ausência de dados, que decorrem tanto de falhas de medição quanto da própria lógica de integração entre as fontes. Em particular, por se tratar de uma junção à esquerda entre registros meteorológicos e focos de calor, colunas como `FOCO_ID`, `FRP` e `RISCO_FOGO` permanecem ausentes na maior parte das instâncias sem foco, o que caracteriza ausência estrutural associada ao processo de rotulação e não perda de informação meteorológica. Entre as variáveis climáticas, a radiação global constitui o caso mais crítico de incompletude, com 48,9370% de valores faltantes no consolidado integrado. Para caracterizar de forma sistemática o padrão de completude das variáveis e orientar decisões de pré-processamento, foi desenvolvida uma rotina de auditoria de dados ausentes no módulo <`src/dataset_missing_audit.py`>. A Figura 2 sintetiza o fluxo de auditoria e construção de cenários de modelagem a partir do *dataset* integrado, destacando a interação entre os módulos responsáveis por análise de *missing*, geração das bases A a F e engenharia de atributos.

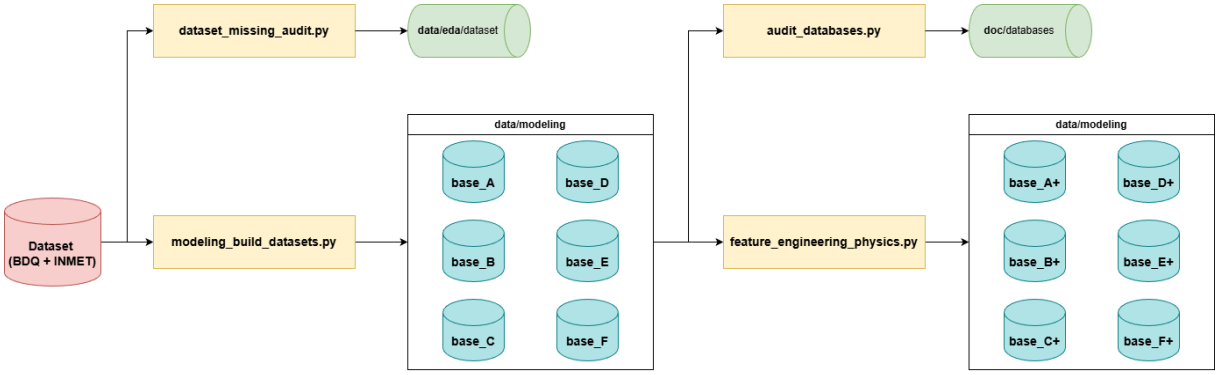



**Figura 2 –** Fluxo de auditoria, construção de cenários de modelagem e engenharia de atributos físico inspirada.
**Fonte:** elaborado pelo autor.

O componente `DatasetMissingAnalyzer` lê, para cada ano, o arquivo consolidado correspondente, harmoniza colunas de radiação global que aparecem com grafias diferentes e constrói uma matriz booleana de missing. Consideram-se como valores faltantes: *NaN* explícito, strings vazias após operação de *strip* e códigos especiais negativos `-999` e `-9999`. A Listagem 3.5 apresenta o núcleo desse procedimento.

**Listing 3.5 –** Construção da matriz de dados faltantes por coluna de feature

```
def build_missing_matrix(self, df: pd.DataFrame) -> pd.DataFrame:
    cols = [c for c in df.columns if c not in self.exclude]
    missing = pd.DataFrame(index=df.index)

    for c in cols:
        s = df[c]
        if pd.api.types.is_bool_dtype(s):
            mask = s.isna()
        elif pd.api.types.is_numeric_dtype(s):
            mask = s.isna() | s.isin(self.missing_codes)
        else:
            s_str = s.astype("string")
            mask = s_str.isna()
            mask |= s_str.str.strip().eq("")
            mask |= s_str.isin(self.missing_codes_str)
        missing[c] = mask.fillna(False)

    return missing
```

Para cada ano analisado, são produzidos dois artefatos em `data/eda/dataset/{ANO}/`: um arquivo `missing_by_column.csv`, contendo contagens e proporções de valores ausentes em cada variável de contexto, e um documento `README_missing.md`, que resume o volume de observações, a proporção de instâncias com foco (`HAS_FOCO = 1`) e as colunas mais críticas em termos de missing. Esses resultados orientam a definição de cenários de modelagem com diferentes estratégias de tratamento de dados faltantes.

A construção das bases primárias de modelagem é conduzida pelo módulo <src/modeling_build_datasets.py>, por meio da classe ModelingDatasetBuilder. A partir dos CSVs anuais integrados, esse componente aplica três etapas principais: (i) harmonização de colunas, (ii) definição de semântica de missing e coerção de tipos e (iii) geração de seis cenários de base, com e sem radiação global, utilizando remoção de linhas ou imputação por KNN. A seleção das colunas de *feature* exclui datas, identificadores, textos e colunas alvo (RISCO_FOGO, FRP, FOCO_ID) e considera apenas variáveis numéricas de contexto.

A semântica de missing é aplicada de forma uniforme sobre todas as bases, substituindo códigos sentinela por valores nulos e convertendo strings vazias em ausência de dado. Em seguida, as colunas de *feature* são convertidas para float32, o que permite o uso eficiente de imputadores numéricos. A imputação por KNN é realizada ano a ano, com possibilidade de segmentação por mês quando grupos de linhas atingem um tamanho mínimo, de forma a respeitar variações sazonais sem comprometer a eficiência computacional. A Listagem 3.6 apresenta a função de alto nível que organiza a imputação.

**Listing 3.6 –** Aplicação da imputação KNN por ano de dados integrados

```
def apply_knn_imputation(self,
                         df: pd.DataFrame,
                         feature_cols: List[str],
                         n_neighbors: int = 5) -> pd.DataFrame:
    df = df.copy()
    num_cols = self.get_numeric_feature_columns(df, feature_cols)
    if not num_cols:
        log.warning("[KNN] Sem colunas numéricas de feature. Imputação ignorada.")
        return df

    if self.imputer_group_by_month and "DATA (YYYY-MM-DD)" in df.columns:
        months = pd.to_numeric(
            df["DATA (YYYY-MM-DD)"].astype("string").str.slice(5, 7),
            errors="coerce"
        ).fillna(-1).astype(int)

        result = df.copy()
        # grupos mensais grandes são imputados separadamente
        # grupos pequenos e meses desconhecidos caem no fallback anual
        ...
        log.info("[KNN] Imputação (agrupada por mês) concluída.")
        return result

    return self._impute_block(df, num_cols, n_neighbors=n_neighbors)
```

Com base nesses componentes, são definidos seis cenários de base, gravados em data/modeling/ como arquivos .parquet particionados por ano:

- **Base F (**base_F_full_original**)**: mantém todas as variáveis, incluindo radiação glo-

bal, substitui códigos sentinela por *NaN*, mas não realiza imputação nem remoção de linhas. Representa a visão mais próxima dos dados observados.

- **Base A (**`base_A_no_rad`**)**: remove a coluna de radiação global, mantém os demais atributos e preserva todas as linhas. Serve como referência para avaliar o impacto da exclusão dessa variável, que frequentemente apresenta elevado nível de missing.
- **Base B (**`base_B_no_rad_knn`**)**: parte da base A e aplica imputação KNN apenas às variáveis numéricas de contexto. Investiga o efeito de recuperar parcialmente a informação perdida sem recorrer à radiação global.
- **Base C (**`base_C_no_rad_drop_rows`**)**: parte da base A e remove todas as linhas que contêm qualquer valor ausente em colunas de *feature*. Implementa uma estratégia conservadora de análise de casos completos sem radiação.
- **Base D (**`base_D_with_rad_drop_rows`**)**: utiliza a base original com radiação global e remove linhas com qualquer valor ausente em *features*. Permite avaliar o comportamento do modelo quando se exige completude total dos atributos, incluindo a variável de radiação.
- **Base E (**`base_E_with_rad_knn`**)**: utiliza a base original, preserva a coluna de radiação global e aplica imputação KNN a todas as *features* numéricas. Representa a estratégia mais agressiva de recuperação de informação, combinando radiação com imputação multivariada.

Após a geração dessas bases de modelagem, o módulo <src/audit_databases.py> executa uma auditoria consolidada em nível de cenário. A classe `ScenarioAuditor` percorre todos os arquivos `.parquet` de cada cenário, acumula estatísticas globais de distribuição da variável alvo `HAS_FOCO`, contabiliza valores ausentes e sentinelas nas variáveis climáticas e sumariza essas informações por ano. Em seguida, a classe `MarkdownReporter` grava, em `doc/databases/`, relatórios em formato `.md` contendo uma visão global da base, uma amostra tabular de registros e tabelas sintéticas de qualidade dos dados por variável e por ano. Esses documentos complementam a auditoria anual inicial e fornecem uma visão agregada da qualidade de cada cenário de modelagem.

Além do tratamento de dados ausentes, foi implementada uma etapa de engenharia de atributos físico no módulo <src/feature_engineering_physics.py>, baseada no método de cálculo do risco de fogo proposto por Setzer et al. (SETZER; SISMANOGLU; SANTOS, 2019). Essa etapa utiliza a Base E, que combina radiação global com imputação KNN, como fonte de referência sem lacunas para o cálculo de variáveis derivadas e, em seguida, distribui essas variáveis para todas as bases de modelagem.

A engenharia de atributos foi, portanto, orientada por um racional físico inspirado no método operacional do Programa Queimadas do INPE, sem buscar reproduzir integralmente seu índice de risco. Em termos metodológicos, optou-se por construir variáveis derivadas que representassem dois eixos centrais do fenômeno: a memória recente de precipitação e secura antecedente, por meio de medidas acumulativas com decaimento temporal, e a ocorrência de

condições meteorológicas críticas associadas à ignição e à propagação do fogo, por meio de indicadores de limiares de temperatura, umidade e vento. O objetivo é fornecer aos algoritmos de classificação atributos mais diretamente relacionados aos mecanismos do domínio, preservando, ao mesmo tempo, simplicidade computacional e reprodutibilidade no pipeline.

A classe `PhysicsFeatureEngineer` percorre os arquivos anuais de `.parquet` da Base E, ordenados por cidade normalizada e carimbo horário, e calcula quatro grupos de atributos: (i) chuva ponderada no tempo, (ii) duração da seca recente, (iii) indicadores binários de limiares críticos de temperatura e umidade e (iv) um fator de propagação que combina vento, temperatura e umidade. A Listagem 3.7 ilustra o cálculo da chuva ponderada (`precip_ewma`) e do contador de dias sem chuva (`dias_sem_chuva`), que preserva memória entre anos consecutivos para cada município.

**Listing 3.7 –** Cálculo de variáveis físico inspiradas de precipitação e seca recente

```
# 1. PRECIPITAÇÃO PONDERADA (EWMA com decaimento 0.5)
df['precip_ewma'] = df.groupby('cidade_norm')[COL_PRECIP].transform(
    lambda x: x.ewm(alpha=0.5, adjust=False).mean()
)

# 2. DIAS SEM CHUVA (contador com memória entre anos)
RAIN_THRESHOLD = 1.0
df['is_rain'] = (df[COL_PRECIP] >= RAIN_THRESHOLD).astype(int)

results = []
for cidade, group in df.groupby('cidade_norm'):
    initial_dry = 0
    if cidade in self.memory_state:
        initial_dry = self.memory_state[cidade]['last_dry']

    vals = group[COL_PRECIP].values
    dry_hours = np.zeros(len(vals), dtype=np.float32)
    current_counter = initial_dry

    for i in range(len(vals)):
        if vals[i] < RAIN_THRESHOLD:
            current_counter += 1
        else:
            current_counter = 0
        dry_hours[i] = current_counter

    self.memory_state[cidade] = {'last_dry': current_counter}
    group_res = pd.DataFrame({'dias_sem_chuva': dry_hours / 24.0}, index=group.index)
    results.append(group_res)

df_dry = pd.concat(results)
df = df.join(df_dry)
```

A partir dessas variáveis, são ainda construídos indicadores binários de risco térmico e hídrico (`risco_temp_max`, `risco_umid_critica`, `risco_umid_alerta`) e um fator de propagação `fator_propagacao`, definido como o produto entre velocidade do vento e temperatura dividido pela umidade relativa acrescida de uma unidade. Essas *features* procuram sintetizar, em escala horária, condições meteorológicas que favorecem a ignição e a propagação do fogo, incorporando informação física de forma compacta ao vetor de *features* numéricas.

Após o cálculo das variáveis derivadas na Base E, o módulo `PhysicsFeatureEngineer` realiza, para cada ano, uma junção por chaves `cidade_norm` e `ts_hour` com todas as bases de modelagem A a F. O resultado é armazenado em novos diretórios, cujos nomes são formados pela concatenação do identificador do cenário original com o sufixo `_calculated`, por exemplo `base_A_no_rad_calculated`, `base_B_no_rad_knn_calculated` e assim por diante. Dessa forma, cada cenário possui uma versão primária, composta apenas pelas variáveis diretamente observadas e imputadas, e uma versão enriquecida, que adiciona as variáveis físico inspiradas calculadas a partir da Base E.

Esses conjuntos, associados aos relatórios consolidados de auditoria por cenário, permitem avaliar de forma sistemática três dimensões metodológicas: (i) o impacto da inclusão ou exclusão da radiação global na presença de missing, (ii) a diferença entre estratégias de remoção de linhas e imputação multivariada via KNN e (iii) os ganhos associados à incorporação de atributos físico inspirados derivados de modelos simplificados de risco de fogo. Em todos os cenários, a variável alvo `HAS_FOCO` permanece inalterada, preservando a mesma definição binária de ocorrência de foco de queimada.

## 3.5 Procedimentos experimentais e comparação de algoritmos

A tarefa de predição é formulada como um problema de classificação binária sob forte desbalanceamento de classes. A variável alvo `HAS_FOCO` indica a ocorrência de focos de queimada e assume valor igual a zero na maioria das instâncias, o que exige cuidados específicos tanto na construção dos conjuntos de treinamento e teste quanto na escolha das métricas de avaliação.

O protocolo experimental é implementado pelo módulo `<src/train_runner.py>`, que oferece uma interface de linha de comando para seleção de cenários de base e algoritmos de classificação. As bases A a F, nas versões primárias e enriquecidas com atributos físico inspirados, são definidas na chave `modeling_scenarios` do arquivo de configuração e podem ser combinadas com os modelos disponíveis em um mesmo lote de execução. A Figura 3 ilustra o fluxo geral quando o usuário seleciona, no *CLI*, todas as bases e todos os algoritmos: os arquivos anuais de `.parquet` de cada cenário são carregados em blocos, particionados em conjuntos de treinamento e teste, submetidos a amostragem e, em seguida, usados para treinar e avaliar cada combinação de modelo e variação de configuração.

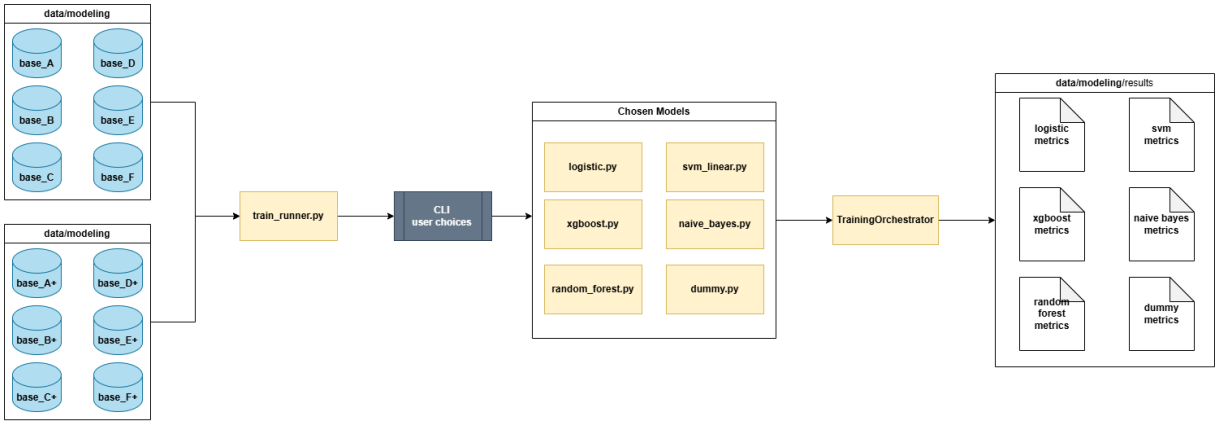



**Figura 3 –** Esquema lógico de seleção de cenários, amostragem e treino de modelos implementado em `src/train_runner.py`. O fluxo representa o caso em que todas as bases e algoritmos são selecionados no *CLI*.

**Fonte:** elaborado pelo autor.

Em cada cenário descrito na Seção 3.4, o módulo `TrainingOrchestrator` carrega os arquivos anuais de `.parquet` de forma incremental, tipicamente um ano por vez, por meio do método `load_split_batched`. Esse método utiliza o ano presente no nome do arquivo para separar cronologicamente os dados em um conjunto de treinamento formado pelos anos mais antigos e um conjunto de teste formado pelos anos mais recentes, garantindo que todas as observações de teste sejam posteriores às de treinamento. Quando a inferência do ano pelo nome não é possível, aplica-se um particionamento alternativo baseado na coluna `ANO`, preservando, em ambos os casos, a ordenação temporal e reduzindo o risco de vazamento de informação entre os conjuntos.

Devido ao desbalanceamento extremo entre instâncias com e sem foco de queimada, o conjunto de treinamento passa por um procedimento de amostragem controlada, implementado na função auxiliar `_downsample_keep_all_pos`. Essa rotina preserva todas as observações positivas (`HAS_FOCO = 1`) e amostra as observações negativas de forma a respeitar simultaneamente um orçamento máximo de linhas por cenário e uma razão limite entre negativos e positivos, com um número mínimo de negativos por bloco anual. A amostragem é realizada de maneira independente em cada arquivo anual, o que reduz o consumo de memória, mantém a representatividade temporal e evita que anos com muitos registros dominem o conjunto de treinamento. O conjunto de teste permanece o mais próximo possível da distribuição original, de forma a avaliar os modelos em uma condição realista de raridade de eventos.

A partir dos conjuntos `X_train`, `y_train`, `X_test` e `y_test`, o *runner* constrói um plano de execução que combina, para cada algoritmo, diferentes variações de configuração. Essas variações são definidas por objetos `VariationOption`, que especificam, entre outros parâmetros, o uso ou não de busca de hiperparâmetros por *GridSearchCV* com validação cruzada temporal, a aplicação de técnicas de balanceamento de classes por pesos ou por amostragem sintética (SMOTE) e a normalização de *features* em modelos sensíveis à escala.

Para cada item do plano, o `TrainingOrchestrator` instancia a classe de *trainer* correspondente (`LogisticTrainer`, `NaiveBayesTrainer`, `SVMTrainer`, `RandomForestTrainer` ou `XGBoostTrainer`), registra metadados da execução e delega o processo de treino, avaliação e salvamento de artefatos ao núcleo comum definido em `<src/ml/core.py>`.

Cada modelo recebe um vetor de *features* numéricas composto pelas variáveis climáticas do INMET e, quando disponível, pelas variáveis físico inspiradas geradas na etapa de engenharia de atributos. Assim, os cenários A a F e suas versões enriquecidas diferem quanto à presença da radiação global, ao tratamento de dados ausentes e ao uso ou não das variáveis derivadas, o que permite quantificar o impacto das decisões de pré-processamento e de engenharia de atributos sobre o desempenho preditivo.

A comparação quantitativa entre os modelos é conduzida com base em métricas adequadas a eventos raros. Além da acurácia e do *F1-Score*, são calculadas a área sob a curva *Precision-Recall* (PR-AUC), a área sob a curva ROC (ROC-AUC), a sensibilidade, a especificidade e o *Brier Score* para calibração probabilística, conforme implementado na classe `TCCMetrics`. Essas métricas são computadas sobre o conjunto de teste mantido separado temporalmente e registrados, junto com os parâmetros de execução, em arquivos `.json` e `.joblib` no diretório de resultados de cada modelo, permitindo avaliar de forma consistente o impacto de cada cenário de modelagem, de cada conjunto de *features* e de cada algoritmo na capacidade de identificar focos de queimada no Cerrado.

# 4 Resultados e Discussão

Após a condução das rodadas experimentais, foram obtidas as métricas de desempenho dos modelos de aprendizado de máquina na tarefa de previsão de focos de queimada. A análise é organizada em três etapas. Primeiro, apresentam-se os classificadores triviais, utilizados como linha de base. Em seguida, discutem-se os resultados dos modelos supervisionados nas bases originais e, posteriormente, nas bases enriquecidas com variáveis derivadas. Ao final, é realizada uma síntese comparativa entre os dois grupos de bases.

## 4.1 Desempenho dos Classificadores Triviais (Dummies)

Como etapa inicial de validação, foram utilizados classificadores do tipo *Dummy*, que servem como linha de base (*baseline*) para a avaliação de ganho preditivo. O uso desses modelos é particularmente importante em problemas com forte desbalanceamento de classes, como a previsão de incêndios, pois evita interpretações indevidas de métricas agregadas, especialmente da acurácia (SAITO; REHMSMEIER, 2015).

A Tabela 4 reúne os resultados das duas estratégias triviais avaliadas. A estratégia de classe majoritária prevê sempre a ausência de focos e, por isso, alcança acurácia muito elevada, porém com revocação e *F1-score* nulos. Já a estratégia estratificada realiza predições aleatórias respeitando a distribuição observada das classes, produzindo valores de $PR\text{-}AUC$ compatíveis com a prevalência da classe positiva na base.

1. **Estratégia de Classe Majoritária**: prevê invariavelmente a ausência de focos ($y = 0$). Devido à raridade do evento de interesse, essa estratégia resulta em uma acurácia próxima a 99%, porém com revocação (*Recall*) e precisão nulas, sendo inútil para um sistema de alerta (ANDRIANARIVONY; AKHLOUFI, 2024). Note-se que, embora a acurácia seja a mais alta entre as baselines, o modelo falha completamente em identificar qualquer foco de incêndio (TP = 0).
2. **Estratégia Estratificada**: realiza predições aleatórias respeitando a distribuição de frequência das classes na base de treinamento. Neste caso, o valor da $PR\text{-}AUC$ converge para a prevalência real da classe positiva na base de dados, permitindo observar o limite inferior de detecção estatística.

### 4.1.1 Análise do Comportamento das Baselines

A Tabela 4 evidencia com clareza a limitação da acurácia em cenários altamente desbalanceados. Embora a estratégia majoritária apresente valores próximos de 99%, ela não recupera nenhum evento positivo. Nesse contexto, a $PR\text{-}AUC$ fornece uma referência mais

Tabela 4 – Desempenho comparativo das baselines Dummy por estratégia e cenário.

| Cenário | Estratégia | PR-AUC | Recall | F1-Score | Acurácia |
|---|---|---|---|---|---|
| Cenário D | Estratificada | 0,0073 | 0,0061 | 0,0067 | 0,9869 |
| | Majoritária | 0,0073 | 0,0000 | 0,0000 | 0,9927 |
| Cenário F | Estratificada | 0,0072 | 0,0070 | 0,0077 | 0,9869 |
| | Majoritária | 0,0072 | 0,0000 | 0,0000 | 0,9928 |
| Cenário C | Estratificada | 0,0044 | 0,0035 | 0,0039 | 0,9923 |
| | Majoritária | 0,0044 | 0,0000 | 0,0000 | 0,9956 |
| Cenário A | Estratificada | 0,0044 | 0,0031 | 0,0035 | 0,9923 |
| | Majoritária | 0,0044 | 0,0000 | 0,0000 | 0,9956 |
| Cenário E | Estratificada | 0,0041 | 0,0035 | 0,0038 | 0,9926 |
| | Majoritária | 0,0041 | 0,0000 | 0,0000 | 0,9959 |
| Cenário B | Estratificada | 0,0041 | 0,0035 | 0,0038 | 0,9926 |
| | Majoritária | 0,0041 | 0,0000 | 0,0000 | 0,9959 |

adequada, pois estabelece o patamar mínimo a ser superado pelos modelos supervisionados para que exista ganho preditivo efetivo (ANDRIANARIVONY; AKHLOUFI, 2024).

## 4.2 Desempenho dos Modelos nas Bases Originais

Nesta seção, analisam-se os resultados obtidos pelos modelos supervisionados nas bases originais do estudo, isto é, antes da incorporação de variáveis derivadas. O objetivo é examinar o comportamento dos classificadores sob três estratégias de treinamento: configuração base, ajuste via *GridSearchCV* com *SMOTE* e ajuste via *GridSearchCV* com balanceamento por peso. Ao final, apresentam-se sínteses por cenário e uma consolidação das melhores combinações desta etapa.

### 4.2.1 Modelos em Configuração Base

Antes da aplicação de técnicas de ajuste de hiperparâmetros ou de tratamento do desbalanceamento, os modelos foram avaliados em sua configuração padrão. Essa etapa permite observar o comportamento inicial dos algoritmos diante da assimetria entre classes e serve como referência para interpretar os ganhos obtidos nas estratégias subsequentes.

A Tabela 5 consolida os resultados segundo $PR\text{-}AUC$, $ROC\text{-}AUC$, *F1-score*, precisão, revocação e *Brier score*. As métricas de acurácia e especificidade foram omitidas da tabela principal por apresentarem valores sistematicamente elevados e pouco informativos neste contexto.

Tabela 5 – Desempenho dos modelos supervisionados na configuração base, sem uso de SMOTE, GridSearchCV ou balanceamento por peso, para todos os cenários avaliados.

| Cenário | Modelo | PR-AUC | ROC-AUC | F1 | Prec. | Rec. | Brier |
|---|---|---|---|---|---|---|---|
| Cenário A (sem radiação) | XGBoost | 0,0446 | 0,9158 | 0,0000 | 0,0000 | 0,0000 | 0,0042 |
| | Random Forest | 0,0281 | 0,8358 | 0,0013 | 0,0427 | 0,0007 | 0,0045 |
| | SVM (Linear) | 0,0369 | 0,9017 | 0,0000 | 0,0000 | 0,0000 | 0,0043 |
| | Regressão Logística | 0,0369 | 0,9006 | 0,0000 | 0,0000 | 0,0000 | 0,0043 |
| | Naive Bayes | 0,0317 | 0,8942 | 0,0629 | 0,0341 | 0,4035 | 0,0363 |
| Cenário B (sem radiação, KNN) | XGBoost | 0,0365 | 0,8726 | 0,0000 | 0,0000 | 0,0000 | 0,0040 |
| | Random Forest | 0,0222 | 0,7877 | 0,0010 | 0,0515 | 0,0005 | 0,0042 |
| | SVM (Linear) | 0,0318 | 0,8631 | 0,0000 | 0,0000 | 0,0000 | 0,0040 |
| | Regressão Logística | 0,0316 | 0,8611 | 0,0000 | 0,0000 | 0,0000 | 0,0040 |
| | Naive Bayes | 0,0252 | 0,8506 | 0,0587 | 0,0321 | 0,3400 | 0,0321 |
| Cenário C (sem radiação, remoção de linhas) | XGBoost | 0,0443 | 0,9160 | 0,0000 | 0,0000 | 0,0000 | 0,0043 |
| | Random Forest | 0,0273 | 0,8375 | 0,0011 | 0,0519 | 0,0006 | 0,0045 |
| | SVM (Linear) | 0,0373 | 0,9016 | 0,0000 | 0,0000 | 0,0000 | 0,0043 |
| | Regressão Logística | 0,0374 | 0,9005 | 0,0000 | 0,0000 | 0,0000 | 0,0043 |
| | Naive Bayes | 0,0320 | 0,8939 | 0,0628 | 0,0341 | 0,4046 | 0,0367 |
| Cenário D (com radiação, remoção de linhas) | XGBoost | 0,0536 | 0,8913 | 0,0000 | 0,0000 | 0,0000 | 0,0070 |
| | Random Forest | 0,0436 | 0,8402 | 0,0003 | 0,0435 | 0,0001 | 0,0072 |
| | SVM (Linear) | 0,0407 | 0,8532 | 0,0000 | 0,0000 | 0,0000 | 0,0071 |
| | Regressão Logística | 0,0405 | 0,8488 | 0,0000 | 0,0000 | 0,0000 | 0,0071 |
| | Naive Bayes | 0,0365 | 0,8441 | 0,0696 | 0,0631 | 0,0778 | 0,0157 |
| Cenário E (com radiação, KNN) | XGBoost | 0,0412 | 0,9002 | 0,0000 | 0,0000 | 0,0000 | 0,0040 |
| | Random Forest | 0,0325 | 0,8304 | 0,0004 | 0,0625 | 0,0002 | 0,0041 |
| | SVM (Linear) | 0,0339 | 0,8749 | 0,0000 | 0,0000 | 0,0000 | 0,0040 |
| | Regressão Logística | 0,0329 | 0,8712 | 0,0000 | 0,0000 | 0,0000 | 0,0040 |
| | Naive Bayes | 0,0261 | 0,8653 | 0,0641 | 0,0363 | 0,2708 | 0,0238 |
| Cenário F (base completa) | XGBoost | 0,0535 | 0,8917 | 0,0005 | 0,5000 | 0,0003 | 0,0070 |
| | Random Forest | 0,0421 | 0,8390 | 0,0003 | 0,0278 | 0,0001 | 0,0072 |
| | SVM (Linear) | 0,0404 | 0,8531 | 0,0000 | 0,0000 | 0,0000 | 0,0071 |
| | Regressão Logística | 0,0401 | 0,8485 | 0,0000 | 0,0000 | 0,0000 | 0,0071 |
| | Naive Bayes | 0,0365 | 0,8442 | 0,0700 | 0,0638 | 0,0776 | 0,0156 |

De modo geral, observa-se que, sem tratamento explícito do desbalanceamento, a maior parte dos modelos apresenta revocação nula ou residual. O *Naive Bayes* constitui a principal exceção, exibindo maior sensibilidade à classe positiva, embora com precisão reduzida e *Brier score* mais elevado. Também se nota que os maiores valores de $PR\text{-}AUC$ já se concentram nos cenários com radiação global, especialmente D e F.

### 4.2.2 Modelos com GridSearchCV e SMOTE

Na sequência, avaliou-se a combinação entre ajuste de hiperparâmetros via *GridSearchCV* e sobreamostragem sintética da classe minoritária por meio do *SMOTE*. Essa estratégia foi empregada com o objetivo de reduzir a insensibilidade dos classificadores à classe positiva e verificar em que medida o reequilíbrio do conjunto de treinamento melhora o desempenho dos modelos.

A Tabela 6 resume os resultados obtidos para todos os cenários sob essa configuração.

Tabela 6 – Desempenho dos modelos supervisionados com ajuste via GridSearchCV e balanceamento por SMOTE

| Cenário | Modelo | PR-AUC | ROC-AUC | F1 | Prec. | Rec. | Brier |
|---|---|---|---|---|---|---|---|
| Cenário A (sem radiação) | Random Forest | 0,0404 | 0,9130 | 0,0862 | 0,0524 | 0,2431 | 0,0204 |
| | XGBoost | 0,0402 | 0,9103 | 0,0799 | 0,0464 | 0,2870 | 0,0248 |
| | SVM (Linear) | 0,0369 | 0,9015 | 0,0735 | 0,0414 | 0,3321 | 0,0289 |
| | Regressão Logística | 0,0368 | 0,9013 | 0,0728 | 0,0409 | 0,3321 | 0,0291 |
| | Naive Bayes | 0,0327 | 0,8943 | 0,0374 | 0,0192 | 0,8092 | 0,1432 |
| Cenário B (sem radiação, KNN) | XGBoost | 0,0344 | 0,8707 | 0,0763 | 0,0446 | 0,2621 | 0,0237 |
| | Random Forest | 0,0324 | 0,8705 | 0,0798 | 0,0501 | 0,1954 | 0,0195 |
| | SVM (Linear) | 0,0318 | 0,8630 | 0,0724 | 0,0423 | 0,2488 | 0,0239 |
| | Regressão Logística | 0,0315 | 0,8623 | 0,0708 | 0,0411 | 0,2578 | 0,0246 |
| | Naive Bayes | 0,0258 | 0,8508 | 0,0332 | 0,0170 | 0,7376 | 0,1356 |
| Cenário C (sem radiação, remoção de linhas) | XGBoost | 0,0404 | 0,9105 | 0,0811 | 0,0469 | 0,3007 | 0,0252 |
| | Random Forest | 0,0388 | 0,9104 | 0,0846 | 0,0511 | 0,2460 | 0,0210 |
| | SVM (Linear) | 0,0373 | 0,9014 | 0,0740 | 0,0416 | 0,3320 | 0,0289 |
| | Regressão Logística | 0,0372 | 0,9012 | 0,0732 | 0,0411 | 0,3327 | 0,0292 |
| | Naive Bayes | 0,0329 | 0,8940 | 0,0375 | 0,0192 | 0,8121 | 0,1448 |
| Cenário D (com radiação, remoção de linhas) | Random Forest | 0,0528 | 0,8894 | 0,1044 | 0,0782 | 0,1570 | 0,0182 |
| | XGBoost | 0,0514 | 0,8857 | 0,1072 | 0,0770 | 0,1768 | 0,0206 |
| | SVM (Linear) | 0,0407 | 0,8531 | 0,0441 | 0,0921 | 0,0290 | 0,0180 |
| | Regressão Logística | 0,0406 | 0,8503 | 0,0911 | 0,0639 | 0,1588 | 0,0251 |
| | Naive Bayes | 0,0373 | 0,8419 | 0,0493 | 0,0256 | 0,6869 | 0,1319 |
| Cenário E (com radiação, KNN) | XGBoost | 0,0392 | 0,8962 | 0,0853 | 0,0502 | 0,2847 | 0,0214 |
| | Random Forest | 0,0373 | 0,8944 | 0,0786 | 0,0469 | 0,2412 | 0,0195 |
| | SVM (Linear) | 0,0349 | 0,8773 | 0,0847 | 0,0578 | 0,1582 | 0,0189 |
| | Regressão Logística | 0,0334 | 0,8742 | 0,0740 | 0,0431 | 0,2622 | 0,0247 |
| | Naive Bayes | 0,0280 | 0,8664 | 0,0349 | 0,0179 | 0,7385 | 0,1259 |
| Cenário F (base completa) | XGBoost | 0,0518 | 0,8864 | 0,1080 | 0,0772 | 0,1795 | 0,0207 |
| | Random Forest | 0,0514 | 0,8884 | 0,1035 | 0,0752 | 0,1658 | 0,0189 |
| | SVM (Linear) | 0,0404 | 0,8530 | 0,0434 | 0,0897 | 0,0286 | 0,0180 |
| | Regressão Logística | 0,0403 | 0,8502 | 0,0906 | 0,0634 | 0,1586 | 0,0251 |
| | Naive Bayes | 0,0373 | 0,8420 | 0,0495 | 0,0257 | 0,6837 | 0,1302 |

Em comparação com a etapa anterior, observa-se aumento generalizado de revocação e de *F1-score*, indicando maior capacidade de recuperação da classe positiva. Os melhores resultados de $PR\text{-}AUC$ concentram-se novamente em *XGBoost* e *Random Forest*, sobretudo nos Cenários D e F. Os modelos lineares passam a detectar uma parcela maior de eventos, mas permanecem abaixo dos melhores *ensembles*. Já o *Naive Bayes* mantém revocação elevada, porém com baixa precisão e pior calibração.

### 4.2.3 Modelos com GridSearchCV e Balanceamento por Peso

Como alternativa ao balanceamento por sobreamostragem sintética, avaliou-se a combinação entre ajuste de hiperparâmetros via *GridSearchCV* e ponderação diferencial das classes no processo de treinamento. Nessa abordagem, erros sobre a classe minoritária recebem maior penalização, deslocando a otimização em favor da detecção de focos de queimada.

A Tabela 7 apresenta os resultados obtidos com essa estratégia.

Tabela 7 – Desempenho dos modelos supervisionados com ajuste via GridSearchCV e balanceamento por peso

| Cenário | Modelo | PR-AUC | ROC-AUC | F1 | Prec. | Rec. | Brier |
|---|---|---|---|---|---|---|---|
| Cenário A (sem radiação) | XGBoost | 0,0434 | 0,9142 | 0,0348 | 0,0177 | 0,9184 | 0,1441 |
| | SVM (Linear) | 0,0367 | 0,9010 | 0,0000 | 0,0000 | 0,0000 | 0,0043 |
| | Regressão Logística | 0,0367 | 0,9011 | 0,0333 | 0,0170 | 0,8917 | 0,1558 |
| | Random Forest | 0,0356 | 0,9021 | 0,0437 | 0,0225 | 0,7399 | 0,0908 |
| | Naive Bayes | 0,0317 | 0,8942 | 0,0629 | 0,0341 | 0,4035 | 0,0363 |
| Cenário B (sem radiação, KNN) | XGBoost | 0,0358 | 0,8726 | 0,0309 | 0,0157 | 0,8159 | 0,1406 |
| | Random Forest | 0,0320 | 0,8602 | 0,0411 | 0,0212 | 0,6889 | 0,0845 |
| | Regressão Logística | 0,0314 | 0,8620 | 0,0277 | 0,0141 | 0,8245 | 0,1616 |
| | SVM (Linear) | 0,0313 | 0,8615 | 0,0000 | 0,0000 | 0,0000 | 0,0040 |
| | Naive Bayes | 0,0252 | 0,8506 | 0,0587 | 0,0321 | 0,3400 | 0,0321 |
| Cenário C (sem radiação, remoção de linhas) | XGBoost | 0,0430 | 0,9149 | 0,0349 | 0,0178 | 0,9229 | 0,1460 |
| | SVM (Linear) | 0,0371 | 0,9008 | 0,0000 | 0,0000 | 0,0000 | 0,0043 |
| | Regressão Logística | 0,0371 | 0,9010 | 0,0335 | 0,0171 | 0,8911 | 0,1559 |
| | Random Forest | 0,0355 | 0,9014 | 0,0437 | 0,0225 | 0,7390 | 0,0912 |
| | Naive Bayes | 0,0320 | 0,8939 | 0,0628 | 0,0341 | 0,4046 | 0,0367 |
| Cenário D (com radiação, remoção de linhas) | XGBoost | 0,0527 | 0,8912 | 0,0557 | 0,0288 | 0,8344 | 0,1334 |
| | Regressão Logística | 0,0446 | 0,8596 | 0,0441 | 0,0227 | 0,8226 | 0,1666 |
| | Random Forest | 0,0436 | 0,8646 | 0,0036 | 0,1008 | 0,0018 | 0,0076 |
| | SVM (Linear) | 0,0413 | 0,8518 | 0,0000 | 0,0000 | 0,0000 | 0,0071 |
| | Naive Bayes | 0,0365 | 0,8441 | 0,0696 | 0,0631 | 0,0778 | 0,0157 |
| Cenário E (com radiação, KNN) | XGBoost | 0,0413 | 0,8995 | 0,0378 | 0,0193 | 0,8075 | 0,1164 |
| | Regressão Logística | 0,0379 | 0,8837 | 0,0299 | 0,0152 | 0,8410 | 0,1483 |
| | Random Forest | 0,0372 | 0,8880 | 0,0493 | 0,0256 | 0,6595 | 0,0690 |
| | SVM (Linear) | 0,0344 | 0,8762 | 0,0000 | 0,0000 | 0,0000 | 0,0040 |

| Cenário | Modelo | PR-AUC | ROC-AUC | F1 | Prec. | Rec. | Brier |
|---|---|---|---|---|---|---|---|
| | Naive Bayes | 0,0261 | 0,8653 | 0,0641 | 0,0363 | 0,2708 | 0,0238 |
| Cenário F (base completa) | XGBoost | 0,0533 | 0,8914 | 0,0559 | 0,0289 | 0,8280 | 0,1313 |
| | Random Forest | 0,0466 | 0,8802 | 0,0681 | 0,0360 | 0,6226 | 0,0811 |
| | Regressão Logística | 0,0443 | 0,8597 | 0,0441 | 0,0226 | 0,8231 | 0,1667 |
| | SVM (Linear) | 0,0410 | 0,8518 | 0,0000 | 0,0000 | 0,0000 | 0,0071 |
| | Naive Bayes | 0,0365 | 0,8442 | 0,0700 | 0,0638 | 0,0776 | 0,0156 |

De modo geral, o balanceamento por peso elevou substancialmente a revocação, sobretudo em *XGBoost* e Regressão Logística. Em contrapartida, esse ganho veio acompanhado, em muitos casos, de queda de precisão e aumento do *Brier score*, sinalizando maior incidência de falsos positivos e pior calibração probabilística. Também nesta configuração, os melhores valores de $PR\text{-}AUC$ permanecem concentrados nos cenários com radiação global, em especial D e F.

### 4.2.4 Melhores Resultados por Cenário nas Bases Originais

Após a análise das três estratégias avaliadas nesta etapa, torna-se possível consolidar, para cada cenário, as combinações de maior destaque. A Tabela 8 reúne os três melhores arranjos por base original, considerando prioritariamente o $PR\text{-}AUC$ e, de forma complementar, revocação, *F1-score* e $ROC\text{-}AUC$.

A consolidação por cenário mostra que os maiores valores de $PR\text{-}AUC$ se concentram nos cenários com radiação global, ao passo que *XGBoost* aparece de forma recorrente entre as melhores combinações. Também se observa a coexistência de dois perfis de solução: combinações com balanceamento por peso, mais orientadas à revocação, e combinações com *SMOTE*, em geral mais equilibradas entre revocação e *F1-score*.

### 4.2.5 Melhores Combinações Gerais nas Bases Originais

Para concluir a análise das bases sem engenharia de atributos, a Tabela 9 apresenta as três melhores combinações globais entre cenário, modelo e estratégia.

A síntese geral confirma a predominância dos Cenários D e F e reforça o papel central dos modelos baseados em árvores, particularmente *XGBoost* e *Random Forest*. Em termos interpretativos, esta etapa sugere que a presença da radiação global e o uso de estratégias explícitas de tratamento do desbalanceamento são os fatores mais associados ao melhor desempenho nas bases originais.

Tabela 8 – Três melhores combinações de modelo e estratégia para cada cenário nas bases originais.

| Cenário | Modelo / Estratégia | PR-AUC | Recall | F1-Score | ROC-AUC |
|---|---|---|---|---|---|
| Cenário A | XGBoost / GridSearchCV + Pesos | 0,0434 | 0,9184 | 0,0348 | 0,9142 |
| | Random Forest / GridSearchCV + SMOTE | 0,0404 | 0,2431 | 0,0862 | 0,9130 |
| | XGBoost / GridSearchCV + SMOTE | 0,0402 | 0,2870 | 0,0799 | 0,9103 |
| Cenário B | XGBoost / GridSearchCV + Pesos | 0,0358 | 0,8159 | 0,0309 | 0,8726 |
| | XGBoost / GridSearchCV + SMOTE | 0,0344 | 0,2621 | 0,0763 | 0,8707 |
| | Random Forest / GridSearchCV + SMOTE | 0,0324 | 0,1954 | 0,0798 | 0,8705 |
| Cenário C | XGBoost / GridSearchCV + Pesos | 0,0430 | 0,9229 | 0,0349 | 0,9149 |
| | XGBoost / GridSearchCV + SMOTE | 0,0404 | 0,3007 | 0,0811 | 0,9105 |
| | Random Forest / GridSearchCV + SMOTE | 0,0388 | 0,2460 | 0,0846 | 0,9104 |
| Cenário D | Random Forest / GridSearchCV + SMOTE | 0,0528 | 0,1570 | 0,1044 | 0,8894 |
| | XGBoost / GridSearchCV + Pesos | 0,0527 | 0,8344 | 0,0557 | 0,8912 |
| | XGBoost / GridSearchCV + SMOTE | 0,0514 | 0,1768 | 0,1072 | 0,8857 |
| Cenário E | XGBoost / GridSearchCV + Pesos | 0,0413 | 0,8075 | 0,0378 | 0,8995 |
| | XGBoost / GridSearchCV + SMOTE | 0,0392 | 0,2847 | 0,0853 | 0,8962 |
| | Random Forest / GridSearchCV + SMOTE | 0,0373 | 0,2412 | 0,0786 | 0,8944 |
| Cenário F | XGBoost / GridSearchCV + Pesos | 0,0533 | 0,8280 | 0,0559 | 0,8914 |
| | XGBoost / GridSearchCV + SMOTE | 0,0518 | 0,1795 | 0,1080 | 0,8864 |
| | Random Forest / GridSearchCV + SMOTE | 0,0514 | 0,1658 | 0,1035 | 0,8884 |

Tabela 9 – Três melhores combinações globais entre cenário, modelo e estratégia nas bases originais.

| Cenário / Modelo / Estratégia | PR-AUC | Recall | F1-Score | ROC-AUC |
|---|---|---|---|---|
| Cenário F / XGBoost / GridSearchCV + Pesos | 0,0533 | 0,8280 | 0,0559 | 0,8914 |
| Cenário D / Random Forest / GridSearchCV + SMOTE | 0,0528 | 0,1570 | 0,1044 | 0,8894 |
| Cenário D / XGBoost / GridSearchCV + Pesos | 0,0527 | 0,8344 | 0,0557 | 0,8912 |

## 4.3 Desempenho dos Modelos nas Bases com Variáveis Derivadas

Nesta seção, analisam-se os resultados obtidos nas bases enriquecidas com variáveis derivadas. Mantém-se a mesma organização adotada para as bases originais, a fim de preservar a comparabilidade entre as etapas do estudo e verificar em que medida a engenharia de atributos altera os padrões observados anteriormente.

### 4.3.1 Modelos em Configuração Base

Como ponto de partida, os modelos foram avaliados em sua configuração padrão, sem técnicas adicionais de balanceamento ou ajuste sistemático de hiperparâmetros.

Tabela 10 – Desempenho dos modelos supervisionados na configuração base para os cenários com variáveis derivadas.

| Cenário | Modelo | PR-AUC | ROC-AUC | F1 | Prec. | Rec. | Brier |
|---|---|---|---|---|---|---|---|
| Cenário A (sem radiação, variáveis derivadas) | XGBoost | 0,0455 | 0,9046 | 0,0000 | 0,0000 | 0,0000 | 0,0059 |
| | SVM (Linear) | 0,0391 | 0,8923 | 0,0039 | 0,1212 | 0,0020 | 0,0060 |
| | Regressão Logística | 0,0381 | 0,8910 | 0,0000 | 0,0000 | 0,0000 | 0,0059 |
| | Random Forest | 0,0353 | 0,8569 | 0,0000 | 0,0000 | 0,0000 | 0,0061 |
| | Naive Bayes | 0,0346 | 0,8857 | 0,0602 | 0,0318 | 0,5649 | 0,0935 |
| Cenário B (sem radiação, KNN, variáveis derivadas) | XGBoost | 0,0544 | 0,8761 | 0,0005 | 0,1739 | 0,0002 | 0,0078 |
| | Regressão Logística | 0,0487 | 0,8704 | 0,0000 | 0,0000 | 0,0000 | 0,0078 |
| | SVM (Linear) | 0,0485 | 0,8719 | 0,0029 | 0,1727 | 0,0015 | 0,0079 |
| | Random Forest | 0,0398 | 0,8230 | 0,0000 | 0,0000 | 0,0000 | 0,0080 |
| | Naive Bayes | 0,0395 | 0,8613 | 0,0757 | 0,0408 | 0,5280 | 0,0911 |
| Cenário C (sem radiação, remoção de linhas, variáveis derivadas) | XGBoost | 0,0456 | 0,9056 | 0,0000 | 0,0000 | 0,0000 | 0,0058 |
| | SVM (Linear) | 0,0392 | 0,8924 | 0,0041 | 0,1633 | 0,0021 | 0,0058 |
| | Regressão Logística | 0,0375 | 0,8911 | 0,0000 | 0,0000 | 0,0000 | 0,0058 |
| | Random Forest | 0,0337 | 0,8567 | 0,0000 | 0,0000 | 0,0000 | 0,0060 |
| | Naive Bayes | 0,0337 | 0,8855 | 0,0587 | 0,0309 | 0,5776 | 0,0953 |
| Cenário D (com radiação, remoção de linhas, variáveis derivadas) | XGBoost | 0,1040 | 0,8911 | 0,0003 | 0,3333 | 0,0001 | 0,0134 |
| | Regressão Logística | 0,0942 | 0,8536 | 0,0000 | 0,0000 | 0,0000 | 0,0136 |
| | Random Forest | 0,0928 | 0,8550 | 0,0003 | 0,0769 | 0,0001 | 0,0135 |
| | SVM (Linear) | 0,0926 | 0,8556 | 0,0058 | 0,3717 | 0,0029 | 0,0137 |
| | Naive Bayes | 0,0816 | 0,8493 | 0,1130 | 0,0637 | 0,4966 | 0,0994 |
| Cenário E (com radiação, KNN, variáveis derivadas) | XGBoost | 0,0827 | 0,9015 | 0,0042 | 0,1932 | 0,0021 | 0,0077 |
| | Random Forest | 0,0643 | 0,8575 | 0,0000 | 0,0000 | 0,0000 | 0,0078 |
| | Regressão Logística | 0,0499 | 0,8705 | 0,0000 | 0,0000 | 0,0000 | 0,0078 |
| | SVM (Linear) | 0,0484 | 0,8690 | 0,0025 | 0,1681 | 0,0012 | 0,0079 |
| | Naive Bayes | 0,0390 | 0,8608 | 0,0747 | 0,0405 | 0,4832 | 0,0851 |
| Cenário F (base completa, variáveis derivadas) | XGBoost | 0,1064 | 0,8907 | 0,0016 | 0,2500 | 0,0008 | 0,0136 |
| | Regressão Logística | 0,0947 | 0,8533 | 0,0000 | 0,0000 | 0,0000 | 0,0138 |
| | SVM (Linear) | 0,0930 | 0,8554 | 0,0057 | 0,3294 | 0,0029 | 0,0139 |
| | Random Forest | 0,0861 | 0,8532 | 0,0016 | 0,4138 | 0,0008 | 0,0138 |
| | Naive Bayes | 0,0832 | 0,8491 | 0,1152 | 0,0652 | 0,4931 | 0,0983 |

De modo geral, a Tabela 10 mostra que a incorporação de variáveis derivadas elevou os valores de $PR\text{-}AUC$ em vários cenários, com destaque para D e F. Ainda assim, a maior parte dos modelos continua apresentando revocação nula ou muito baixa, o que indica que o enriquecimento do espaço de atributos, isoladamente, não elimina os efeitos do desbalanceamento. O *Naive Bayes* permanece como o modelo mais sensível à classe positiva nessa configuração, enquanto o *XGBoost* preserva os maiores valores de $PR\text{-}AUC$ nos cenários de melhor desempenho.

### 4.3.2 Modelos com GridSearchCV e SMOTE

Na sequência, avaliou-se a combinação entre ajuste de hiperparâmetros via *GridSearchCV* e sobreamostragem sintética por *SMOTE*, agora aplicada às bases com variáveis derivadas. O objetivo é verificar se o enriquecimento dos atributos amplia os ganhos obtidos com o reequilíbrio amostral.

A Tabela 11 apresenta os resultados para todos os cenários.

Tabela 11 – Desempenho dos modelos supervisionados com ajuste via GridSearchCV e balanceamento por SMOTE

| Cenário | Modelo | PR-AUC | ROC-AUC | F1 | Prec. | Rec. | Brier |
|---|---|---|---|---|---|---|---|
| Cenário A (sem radiação, variáveis derivadas) | XGBoost | 0,0443 | 0,9003 | 0,0883 | 0,0525 | 0,2791 | 0,0291 |
| | Random Forest | 0,0439 | 0,8968 | 0,0910 | 0,0687 | 0,1348 | 0,0168 |
| | SVM (Linear) | 0,0390 | 0,8927 | 0,0794 | 0,0466 | 0,2664 | 0,0302 |
| | Regressão Logística | 0,0375 | 0,8906 | 0,0756 | 0,0430 | 0,3105 | 0,0337 |
| | Naive Bayes | 0,0345 | 0,8858 | 0,0515 | 0,0266 | 0,7657 | 0,1481 |
| Cenário B (sem radiação, KNN, variáveis derivadas) | XGBoost | 0,0544 | 0,8762 | 0,1088 | 0,0690 | 0,2564 | 0,0282 |
| | SVM (Linear) | 0,0487 | 0,8729 | 0,1007 | 0,0655 | 0,2180 | 0,0276 |
| | Regressão Logística | 0,0478 | 0,8700 | 0,0975 | 0,0597 | 0,2656 | 0,0311 |
| | Random Forest | 0,0433 | 0,8601 | 0,0849 | 0,0672 | 0,1151 | 0,0206 |
| | Naive Bayes | 0,0395 | 0,8613 | 0,0653 | 0,0342 | 0,7176 | 0,1445 |
| Cenário C (sem radiação, remoção de linhas, variáveis derivadas) | XGBoost | 0,0454 | 0,9042 | 0,0908 | 0,0541 | 0,2806 | 0,0273 |
| | Random Forest | 0,0431 | 0,8971 | 0,0916 | 0,0665 | 0,1471 | 0,0169 |
| | SVM (Linear) | 0,0388 | 0,8929 | 0,0776 | 0,0454 | 0,2652 | 0,0300 |
| | Regressão Logística | 0,0369 | 0,8908 | 0,0739 | 0,0420 | 0,3096 | 0,0334 |
| | Naive Bayes | 0,0337 | 0,8855 | 0,0497 | 0,0257 | 0,7746 | 0,1513 |
| Cenário D (com radiação, remoção de linhas, variáveis derivadas) | XGBoost | 0,1054 | 0,8911 | 0,1779 | 0,1425 | 0,2365 | 0,0260 |
| | Random Forest | 0,1031 | 0,8902 | 0,1593 | 0,1526 | 0,1666 | 0,0232 |
| | SVM (Linear) | 0,0893 | 0,8537 | 0,1304 | 0,1699 | 0,1058 | 0,0231 |
| | Regressão Logística | 0,0838 | 0,8509 | 0,1472 | 0,1358 | 0,1608 | 0,0294 |
| | Naive Bayes | 0,0818 | 0,8491 | 0,0926 | 0,0497 | 0,6855 | 0,1632 |
| Cenário E (com radiação, KNN, variáveis derivadas) | XGBoost | 0,0831 | 0,9009 | 0,1437 | 0,0903 | 0,3516 | 0,0261 |
| | Random Forest | 0,0695 | 0,8971 | 0,1463 | 0,1136 | 0,2056 | 0,0175 |
| | SVM (Linear) | 0,0487 | 0,8698 | 0,0969 | 0,0728 | 0,1448 | 0,0233 |
| | Regressão Logística | 0,0482 | 0,8692 | 0,0994 | 0,0625 | 0,2436 | 0,0294 |
| | Naive Bayes | 0,0392 | 0,8608 | 0,0650 | 0,0341 | 0,7022 | 0,1411 |
| Cenário F (base completa, variáveis derivadas) | XGBoost | 0,1057 | 0,8908 | 0,1782 | 0,1427 | 0,2370 | 0,0265 |
| | SVM (Linear) | 0,0893 | 0,8533 | 0,1301 | 0,1689 | 0,1058 | 0,0234 |
| | Random Forest | 0,0884 | 0,8804 | 0,1311 | 0,1352 | 0,1273 | 0,0228 |
| | Regressão Logística | 0,0836 | 0,8505 | 0,1470 | 0,1358 | 0,1603 | 0,0298 |
| | Naive Bayes | 0,0835 | 0,8490 | 0,0943 | 0,0507 | 0,6803 | 0,1616 |

Em relação à configuração base, observa-se aumento expressivo de revocação e *F1-*

*score*, sem perda relevante da capacidade de ranqueamento. Os melhores resultados concentram-se novamente em *XGBoost* e *Random Forest*, mas agora em patamar absoluto superior ao das bases originais, sobretudo nos Cenários D e F. Também se nota que o Cenário E passa a se tornar mais competitivo após a incorporação das variáveis derivadas.

### 4.3.3 Modelos com GridSearchCV e Balanceamento por Peso

Como alternativa ao *SMOTE*, avaliou-se a combinação entre *GridSearchCV* e balanceamento por peso das classes nas bases com variáveis derivadas. Nessa abordagem, a maior penalização dos erros sobre a classe positiva busca ampliar a sensibilidade dos classificadores sem alterar a distribuição amostral do treinamento.

A Tabela 12 reúne os resultados correspondentes.

Tabela 12 – Desempenho dos modelos supervisionados com ajuste via GridSearchCV e balanceamento por peso.

| Cenário | Modelo | PR-AUC | ROC-AUC | F1 | Prec. | Rec. | Brier |
|---|---|---|---|---|---|---|---|
| Cenário A (sem radiação, variáveis derivadas) | Random Forest | 0,0444 | 0,8976 | 0,0611 | 0,0320 | 0,6840 | 0,0822 |
| | XGBoost | 0,0444 | 0,9006 | 0,0447 | 0,0229 | 0,9120 | 0,1561 |
| | Regressão Logística | 0,0364 | 0,8904 | 0,0430 | 0,0220 | 0,8999 | 0,1701 |
| | SVM (Linear) | 0,0356 | 0,8902 | 0,0000 | 0,0000 | 0,0000 | 0,0060 |
| | Naive Bayes | 0,0346 | 0,8857 | 0,0602 | 0,0318 | 0,5649 | 0,0935 |
| Cenário B (sem radiação, KNN, variáveis derivadas) | Random Forest | 0,0557 | 0,8699 | 0,0828 | 0,0445 | 0,6013 | 0,0711 |
| | XGBoost | 0,0550 | 0,8756 | 0,0578 | 0,0299 | 0,8318 | 0,1456 |
| | Regressão Logística | 0,0458 | 0,8686 | 0,0572 | 0,0296 | 0,8332 | 0,1584 |
| | SVM (Linear) | 0,0449 | 0,8696 | 0,0000 | 0,0000 | 0,0000 | 0,0078 |
| | Naive Bayes | 0,0395 | 0,8613 | 0,0757 | 0,0408 | 0,5280 | 0,0911 |
| Cenário C (sem radiação, remoção de linhas, variáveis derivadas) | XGBoost | 0,0422 | 0,9004 | 0,0438 | 0,0224 | 0,9111 | 0,1544 |
| | Random Forest | 0,0413 | 0,8909 | 0,0623 | 0,0329 | 0,5993 | 0,0679 |
| | Regressão Logística | 0,0358 | 0,8907 | 0,0422 | 0,0216 | 0,8975 | 0,1674 |
| | SVM (Linear) | 0,0350 | 0,8904 | 0,0000 | 0,0000 | 0,0000 | 0,0058 |
| | Naive Bayes | 0,0337 | 0,8855 | 0,0587 | 0,0309 | 0,5776 | 0,0953 |
| Cenário D (com radiação, remoção de linhas, variáveis derivadas) | XGBoost | 0,1088 | 0,8925 | 0,1041 | 0,0555 | 0,8395 | 0,1343 |
| | Random Forest | 0,0958 | 0,8855 | 0,1331 | 0,0747 | 0,6112 | 0,0751 |
| | Regressão Logística | 0,0857 | 0,8536 | 0,0793 | 0,0417 | 0,8205 | 0,1758 |
| | SVM (Linear) | 0,0843 | 0,8538 | 0,0000 | 0,0000 | 0,0000 | 0,0137 |
| | Naive Bayes | 0,0816 | 0,8493 | 0,1130 | 0,0637 | 0,4966 | 0,0994 |
| Cenário E (com radiação, KNN, variáveis derivadas) | XGBoost | 0,0921 | 0,9026 | 0,0734 | 0,0384 | 0,8285 | 0,1150 |
| | Random Forest | 0,0806 | 0,8942 | 0,1130 | 0,0629 | 0,5517 | 0,0485 |
| | Regressão Logística | 0,0471 | 0,8690 | 0,0569 | 0,0295 | 0,8283 | 0,1531 |
| | SVM (Linear) | 0,0470 | 0,8698 | 0,0000 | 0,0000 | 0,0000 | 0,0078 |
| | Naive Bayes | 0,0390 | 0,8608 | 0,0747 | 0,0405 | 0,4832 | 0,0851 |
| Cenário F (base completa, variáveis derivadas) | XGBoost | 0,1105 | 0,8925 | 0,1054 | 0,0562 | 0,8422 | 0,1353 |

| Cenário | Modelo | PR-AUC | ROC-AUC | F1 | Prec. | Rec. | Brier |
|---|---|---|---|---|---|---|---|
| | Random Forest | 0,0872 | 0,8772 | 0,1280 | 0,0720 | 0,5753 | 0,0751 |
| | Regressão Logística | 0,0862 | 0,8531 | 0,0804 | 0,0423 | 0,8223 | 0,1768 |
| | SVM (Linear) | 0,0841 | 0,8532 | 0,0000 | 0,0000 | 0,0000 | 0,0140 |
| | Naive Bayes | 0,0832 | 0,8491 | 0,1152 | 0,0652 | 0,4931 | 0,0983 |

O padrão observado é semelhante ao verificado nas bases originais, porém em um nível mais alto de $PR\text{-}AUC$. O *XGBoost* volta a concentrar os melhores resultados, especialmente nos Cenários D e F, com revocação elevada e forte capacidade de ranqueamento. O *Random Forest* também apresenta desempenho competitivo, sobretudo em *F1-score* e *Brier score*. Já a Regressão Logística exibe alta revocação, mas com precisão reduzida, enquanto o SVM permanece pouco adaptativo nessa configuração.

### 4.3.4 Melhores Resultados por Cenário nas Bases com Variáveis Derivadas

Com a incorporação das variáveis derivadas, torna-se possível consolidar, para cada cenário, as combinações de modelo e estratégia de maior destaque. A Tabela 13 reúne os três melhores arranjos por base, considerando o $PR\text{-}AUC$ como critério principal e utilizando revocação, *F1-score* e $ROC\text{-}AUC$ como medidas complementares de interpretação.

Tabela 13 – Três melhores combinações de modelo e estratégia para cada cenário nas bases com variáveis derivadas.

| Cenário | Modelo / Estratégia | PR-AUC | Recall | F1-Score | ROC-AUC |
|---|---|---|---|---|---|
| Cenário A | Random Forest / GridSearchCV + Pesos | 0,0444 | 0,6840 | 0,0611 | 0,8976 |
| | XGBoost / GridSearchCV + Pesos | 0,0444 | 0,9120 | 0,0447 | 0,9006 |
| | XGBoost / GridSearchCV + SMOTE | 0,0443 | 0,2791 | 0,0883 | 0,9003 |
| Cenário B | Random Forest / GridSearchCV + Pesos | 0,0557 | 0,6013 | 0,0828 | 0,8699 |
| | XGBoost / GridSearchCV + Pesos | 0,0550 | 0,8318 | 0,0578 | 0,8756 |
| | XGBoost / GridSearchCV + SMOTE | 0,0544 | 0,2564 | 0,1088 | 0,8762 |
| Cenário C | XGBoost / GridSearchCV + SMOTE | 0,0454 | 0,2806 | 0,0908 | 0,9042 |
| | XGBoost / GridSearchCV + Pesos | 0,0422 | 0,9111 | 0,0438 | 0,9004 |
| | Random Forest / GridSearchCV + SMOTE | 0,0431 | 0,1471 | 0,0916 | 0,8971 |
| Cenário D | XGBoost / GridSearchCV + Pesos | 0,1088 | 0,8395 | 0,1041 | 0,8925 |
| | XGBoost / GridSearchCV + SMOTE | 0,1054 | 0,2365 | 0,1779 | 0,8911 |
| | Random Forest / GridSearchCV + SMOTE | 0,1031 | 0,1666 | 0,1593 | 0,8902 |
| Cenário E | XGBoost / GridSearchCV + Pesos | 0,0921 | 0,8285 | 0,0734 | 0,9026 |
| | XGBoost / GridSearchCV + SMOTE | 0,0831 | 0,3516 | 0,1437 | 0,9009 |
| | Random Forest / GridSearchCV + Pesos | 0,0806 | 0,5517 | 0,1130 | 0,8942 |
| Cenário F | XGBoost / GridSearchCV + Pesos | 0,1105 | 0,8422 | 0,1054 | 0,8925 |
| | XGBoost / GridSearchCV + SMOTE | 0,1057 | 0,2370 | 0,1782 | 0,8908 |
| | SVM (Linear) / GridSearchCV + SMOTE | 0,0893 | 0,1058 | 0,1301 | 0,8533 |

A leitura da Tabela 13 mostra que os melhores resultados passam a depender quase integralmente de estratégias com tratamento explícito do desbalanceamento. Além disso, *XGBoost* e *Random Forest* continuam predominando entre as combinações selecionadas, com vantagem mais clara do *XGBoost* nos cenários com radiação global. Também se acentua a distinção entre dois perfis: soluções com balanceamento por peso, mais orientadas à revocação, e soluções com *SMOTE*, em geral mais equilibradas entre revocação e *F1-score*.

### 4.3.5 Melhores Combinações Gerais nas Bases com Variáveis Derivadas

Para concluir a análise das bases enriquecidas, a Tabela 14 apresenta as três melhores combinações globais entre cenário, modelo e estratégia.

Tabela 14 – Três melhores combinações entre cenário, modelo e estratégia nas bases com variáveis derivadas.

| **Cenário / Modelo / Estratégia** | **PR-AUC** | **Recall** | **F1-Score** | **ROC-AUC** |
|---|---|---|---|---|
| Cenário F / XGBoost / GridSearchCV + Pesos | 0,1105 | 0,8422 | 0,1054 | 0,8925 |
| Cenário D / XGBoost / GridSearchCV + Pesos | 0,1088 | 0,8395 | 0,1041 | 0,8925 |
| Cenário F / XGBoost / GridSearchCV + SMOTE | 0,1057 | 0,2370 | 0,1782 | 0,8908 |

A síntese geral torna mais nítido o padrão já observado nas subseções anteriores. As melhores combinações concentram-se exclusivamente nos Cenários D e F e são todas obtidas com *XGBoost*. Em termos substantivos, isso indica que a engenharia de atributos fortaleceu ainda mais a competitividade desse algoritmo e ampliou o ganho associado aos cenários que preservam a variável de radiação global.

### 4.3.6 Síntese Comparativa entre Bases Originais e Bases com Variáveis Derivadas

Com base nas duas etapas de avaliação, a Tabela 15 reúne as seis combinações de maior destaque do estudo, correspondentes aos três melhores resultados das bases originais e aos três melhores resultados das bases com variáveis derivadas.

Tabela 15 – Três melhores combinações de cada grupo, com e sem variáveis derivadas.

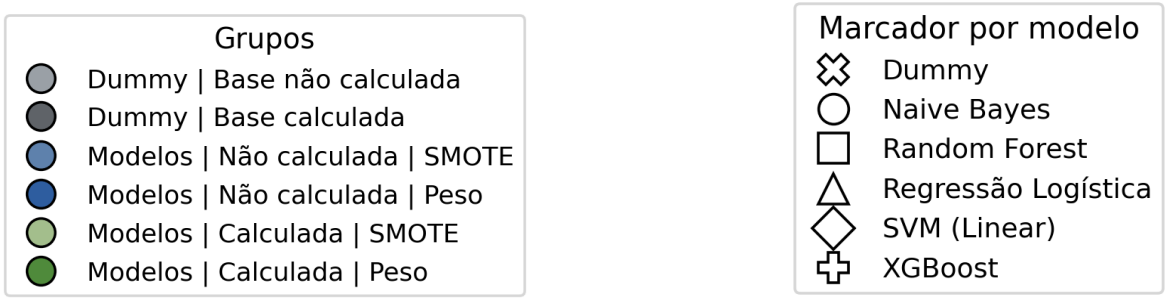
| **Cenário / Modelo / Estratégia** | **PR-AUC** | **Recall** | **F1-Score** | **ROC-AUC** |
|---|---|---|---|---|
| Cenário F Derivado / XGBoost / GridSearchCV + Pesos | 0,1105 | 0,8422 | 0,1054 | 0,8925 |
| Cenário D Derivado / XGBoost / GridSearchCV + Pesos | 0,1088 | 0,8395 | 0,1041 | 0,8925 |
| Cenário F Derivado / XGBoost / GridSearchCV + SMOTE | 0,1057 | 0,2370 | 0,1782 | 0,8908 |
| Cenário F / XGBoost / GridSearchCV + Pesos | 0,0533 | 0,8280 | 0,0559 | 0,8914 |
| Cenário D / Random Forest / GridSearchCV + SMOTE | 0,0528 | 0,1570 | 0,1044 | 0,8894 |
| Cenário D / XGBoost / GridSearchCV + Pesos | 0,0527 | 0,8344 | 0,0557 | 0,8912 |

A comparação direta entre os dois grupos mostra que a incorporação de variáveis derivadas elevou de forma expressiva o desempenho dos melhores modelos em termos de $PR\text{-}AUC$. Enquanto, nas bases originais, o melhor resultado foi 0,0533, nas bases com variáveis derivadas o melhor valor alcançou 0,1105, indicando ganho substancial na capacidade de separação no espaço precisão-revocação.

No conjunto da análise, quatro regularidades se destacam. A primeira é a predominância do *XGBoost*, que passa a liderar de forma inequívoca os melhores resultados globais após a engenharia de atributos. A segunda é a relevância do balanceamento por peso, responsável pelas combinações de maior revocação e pelos maiores valores de $PR\text{-}AUC$ entre os melhores modelos. A terceira é a importância persistente da variável de radiação global, já que os melhores resultados continuam concentrados nos Cenários D e F. A quarta é o papel metodológico do KNN: embora os cenários com imputação não liderem as bases originais, essa etapa foi estrutural para a construção das bases derivadas e, portanto, participou indiretamente da obtenção dos melhores resultados finais.

Além da comparação tabular, a Figura 4 apresenta a codificação visual adotada nas visualizações de evolução por estágio metodológico. Nessas figuras, as cores representam os grupos experimentais, os marcadores identificam os modelos e as letras no interior dos pontos indicam o cenário de base correspondente.

**Figura 4 –** Legendas utilizadas nas visualizações de evolução por estágio metodológico.




**(a)** Grupos experimentais.

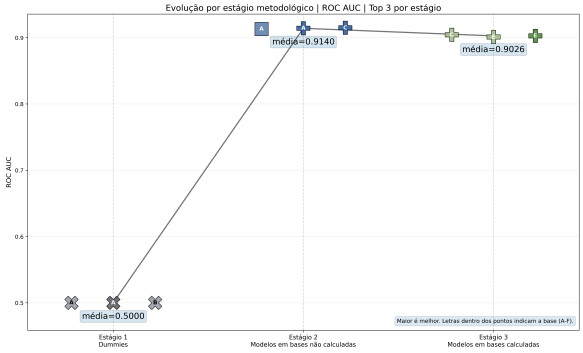
**(b)** Marcadores por modelo.

**Fonte:** elaborado pelo autor.

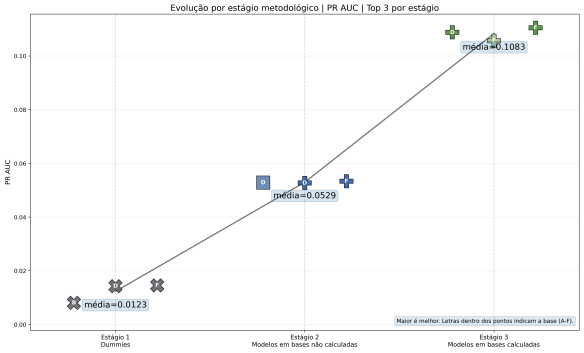
Com base nessa legenda, a Figura 5 sintetiza a evolução dos três melhores resultados em cada estágio metodológico para as métricas de discriminação, isto é, $ROC\text{-}AUC$ e $PR\text{-}AUC$.

Essas visualizações reforçam um padrão já observado nas tabelas. De um lado, o $PR\text{-}AUC$ apresenta crescimento expressivo da Etapa 1 para a Etapa 2 e novo avanço na Etapa 3, indicando ganho consistente na capacidade de separação no espaço precisão-revocação à medida que o pipeline incorpora modelos supervisionados e, depois, variáveis derivadas. De outro, o $ROC\text{-}AUC$ já atinge valores elevados entre os melhores arranjos da Etapa 2 e permanece em patamar semelhante na Etapa 3, com variações relativamente pequenas. Isso sugere que a principal contribuição da engenharia de atributos não foi ampliar drasticamente

**Figura 5 –** Evolução dos três melhores resultados por estágio metodológico nas métricas de discriminação.




**(a)** $ROC\text{-}AUC$.



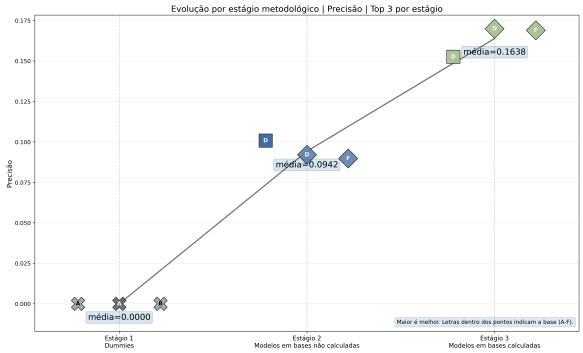

**(b)** $PR\text{-}AUC$.

**Fonte:** elaborado pelo autor.

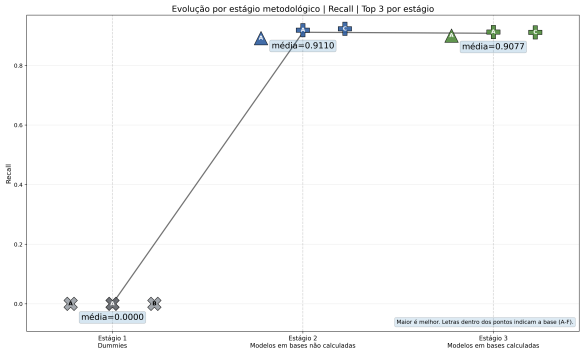
a separabilidade global, mas refinar o desempenho nas métricas mais sensíveis à qualidade das predições positivas.

Seguindo a mesma codificação visual, a Figura 6 apresenta a evolução das métricas dependentes do limiar de decisão, isto é, precisão, revocação e *F1-score*.

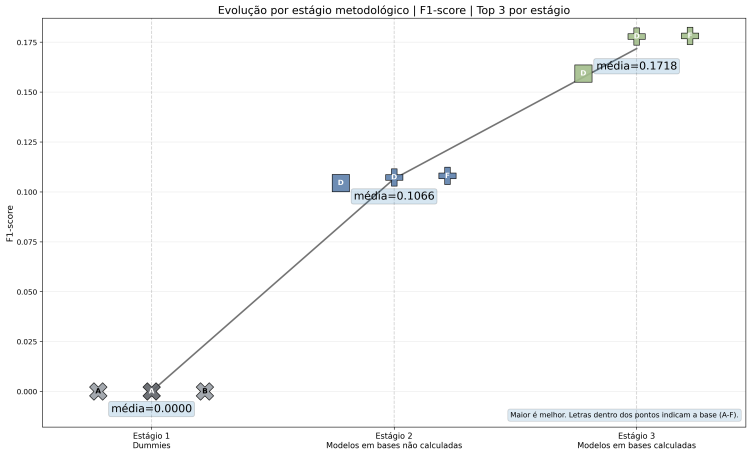
**Figura 6 –** Evolução dos três melhores resultados por estágio metodológico nas métricas dependentes do limiar de decisão.




**(a)** Precisão.




**(b)** Revocação.

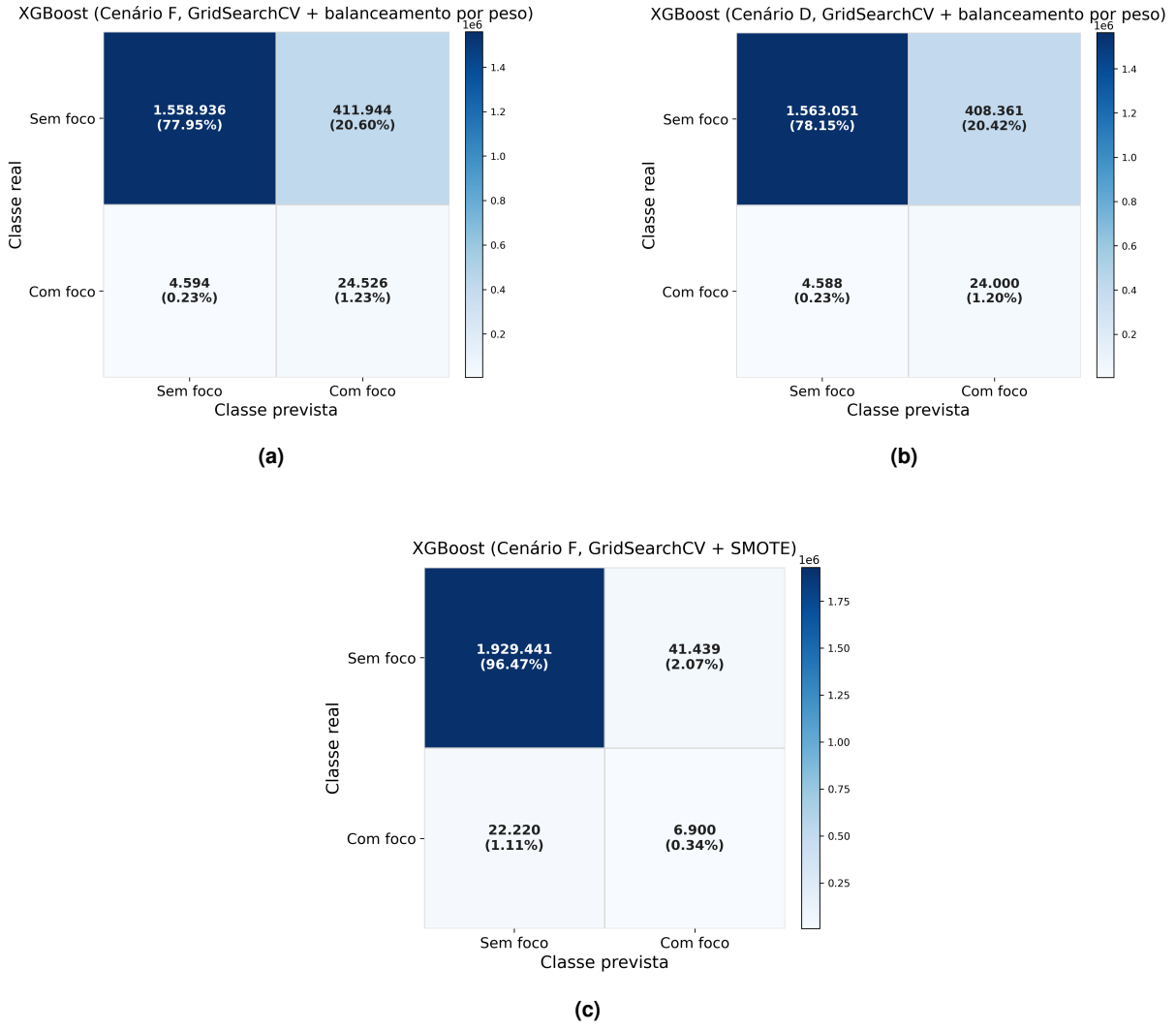



**(c)** *F1-score*.

Nessas métricas, o avanço também se torna claro. A precisão e o *F1-score* aumentam de forma consistente ao longo dos estágios, o que indica melhora na qualidade das predições positivas e no equilíbrio entre precisão e revocação. Já a revocação, embora apresente salto acentuado em relação aos *Dummies*, mantém-se em nível semelhante entre as Etapas 2 e 3 para os melhores arranjos. Em conjunto, essas figuras mostram que a passagem para as bases calculadas trouxe ganhos mais evidentes nas métricas associadas ao compromisso entre acerto positivo e estabilidade da decisão do que em sensibilidade bruta, que já se encontrava elevada entre os melhores modelos da etapa anterior.

A leitura agregada das métricas é complementada, por fim, pelas matrizes de confusão da Figura 7, referentes aos três melhores modelos obtidos nas bases com variáveis derivadas.

**Figura 7 –** Matrizes de confusão dos três melhores modelos obtidos nas bases com variáveis derivadas: (a) XGBoost com *GridSearchCV* e balanceamento por peso no Cenário F; (b) XGBoost com *GridSearchCV* e balanceamento por peso no Cenário D; e (c) XGBoost com *GridSearchCV* e *SMOTE* no Cenário F.




**Fonte:** elaborado pelo autor.

As matrizes tornam mais concreta a diferença entre os perfis das estratégias de balanceamento. Nos dois casos de *XGBoost* com balanceamento por peso, correspondentes aos

Cenários F e D, observa-se maior número de verdadeiros positivos, coerente com os altos valores de revocação anteriormente discutidos, mas também aumento expressivo dos falsos positivos. Já o *XGBoost* com *SMOTE* no Cenário F apresenta perfil relativamente mais conservador, com menor número de verdadeiros positivos e maior presença de falsos negativos, porém com redução dos falsos positivos. Esse contraste confirma a leitura já extraída das tabelas e dos gráficos de evolução: o balanceamento por peso favorece soluções mais agressivas na recuperação da classe minoritária, enquanto o *SMOTE* tende a produzir um compromisso mais equilibrado entre precisão e revocação, refletido também no *F1-score*.

Em síntese, os resultados indicam que o melhor desempenho do estudo foi obtido quando se combinaram três elementos: presença da variável de radiação global, engenharia de atributos e tratamento explícito do desbalanceamento, sobretudo por meio de balanceamento por peso. Nesse arranjo, o *XGBoost* consolidou-se como o modelo mais promissor para a tarefa de predição de focos de incêndio no Cerrado brasileiro.

# 5 Conclusão

Neste trabalho, foi realizada uma análise comparativa de modelos de aprendizado de máquina supervisionado para a predição horária da ocorrência de focos de queimadas no bioma Cerrado, a partir da integração entre dados meteorológicos do INMET e registros de focos do BDQueimadas. A avaliação contemplou bases originais, bases com variáveis derivadas e diferentes estratégias de tratamento do desbalanceamento.

Os resultados mostraram que os modelos baseados em *ensemble* de árvores foram os mais adequados ao problema estudado. O *XGBoost* apresentou o melhor desempenho global, sobretudo nos cenários com presença da variável de radiação global, uso de variáveis derivadas e balanceamento por peso. De modo geral, a engenharia de atributos elevou de forma expressiva o desempenho dos melhores modelos, especialmente em métricas mais sensíveis à detecção da classe positiva, como $PR\text{-}AUC$ e *F1-score*. Sem o tratamento do desbalanceamento, a maior parte dos modelos apresentou revocação nula ou residual, ainda que com valores elevados de acurácia e, em alguns casos, de $ROC\text{-}AUC$. Nesse contexto, o balanceamento por peso favoreceu soluções mais sensíveis à classe minoritária, enquanto o *SMOTE* frequentemente produziu combinações mais equilibradas entre precisão e revocação. Já imputação por KNN, embora não tenha conduzido isoladamente aos melhores resultados absolutos, mostrou-se útil no pipeline do estudo para a construção de bases mais completas e aptas à geração de variáveis derivadas.

Entre as limitações deste trabalho, destaca-se o custo computacional elevado da metodologia adotada. A execução dos experimentos foi realizada em máquina pessoal, com processador, memória RAM e armazenamento de uso cotidiano, o que impôs restrições práticas importantes ao volume de testes, ao refinamento da busca de hiperparâmetros e ao tempo total de processamento.

Como continuidade, recomenda-se aprofundar o estudo da imputação em séries climáticas, com especial atenção ao KNN e a métodos alternativos de preenchimento, bem como ampliar a investigação sobre modelos de *ensemble*, dado o forte alinhamento dessas abordagens com a natureza do problema. Também merece estudo mais aprofundado o comportamento do Naive Bayes, cujos resultados sugerem potencial para variantes probabilísticas mais elaboradas. Além disso, o trabalho pode ser ampliado por meio de estratégias de fusão de dados, integrando variáveis meteorológicas observadas, produtos de sensoriamento remoto, informações de uso e cobertura da terra e, eventualmente, dados coletados em tempo quase real por sensores distribuídos e dispositivos de *Internet of Things* (IoT). Outra frente promissora consiste em incorporar de forma mais explícita a estrutura espaço-temporal do fenômeno, seja pela construção de atributos espaciais e temporais mais ricos, seja pelo uso de abordagens de modelagem temporal e espaço-temporal. No eixo temporal, isso inclui desde

modelos clássicos aplicados a séries agregadas, como ARIMA, até formulações mais diretamente voltadas à previsão supervisionada com dependência temporal. Por fim, trabalhos futuros podem buscar a transformação da metodologia em uma aplicação de monitoramento e apoio à decisão em tempo real, além de executá-la em infraestrutura computacional mais robusta.

# Referências

AFRIFA-YAMOAH, E. et al. Missing data imputation of high-resolution temporal climate time series data. *Meteorological Applications*, v. 27, n. 1, p. e1873, 2020.

AHAJJAM, A. et al. Enhancing prediction of wildfire occurrence and behavior in alaska using spatio-temporal clustering and ensemble machine learning. *Ecological Informatics*, v. 77, p. 102963, 2025.

ALEJO-SANCHEZ, L. E. et al. Missing data imputation of climate time series: A review. *MethodsX*, v. 15, p. 103455, 2025.

ALVES, J. M. B. et al. Um estudo de focos de calor no bioma caatinga e suas relações com variáveis meteorológicas. *Revista Brasileira de Meteorologia*, v. 36, n. 3 (Suplemento), p. 513–527, 2021. Disponível em: <http://dx.doi.org/10.1590/0102-77863630015>.

ANDRIANARIVONY, H. S.; AKHLOUFI, M. A. Machine learning and deep learning for wildfire spread prediction: A review. *Fire*, v. 7, n. 12, p. 482, 2024. Revisão sistemática comparando modelos físicos vs. data-driven.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, v. 13, p. 281–305, 2012.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, 2001.

CESPEDES, J. M. N. et al. A comparison of missing value imputation methods applied to daily precipitation in a semi-arid and a humid region of mexico. *Atmosfera*, v. 37, p. 33–52, 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.

CILLI, R. et al. Explainable artificial intelligence (xai) detects wildfire occurrence in the mediterranean countries of southern europe. *Scientific Reports*, v. 12, p. 16832, 2022.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, p. 273–297, 1995.

DAVIS, J.; GOADRICH, M. The relationship between precision-recall and roc curves. In: *Proceedings of the 23rd International Conference on Machine Learning*. Pittsburgh, PA, USA: [s.n.], 2006. Demonstra a relação entre os espaços ROC e PR e discute por que PR pode ser mais informativa em bases desbalanceadas.

FAWCETT, T. An introduction to roc analysis. *Pattern Recognition Letters*, Elsevier, v. 27, n. 8, p. 861–874, 2006.

FREITAS, K. M. et al. Prediction of forest fire susceptibility using machine learning tools in the triunfo do xingu environmental protection area, amazon, brazil. *Journal of South American Earth Sciences*, v. 153, p. 105366, 2025.

FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, Institute of Mathematical Statistics, v. 29, n. 5, p. 1189–1232, 2001.

JESUS, J. B. d. et al. Análise da incidência temporal, espacial e de tendência de fogo nos biomas e unidades de conservação do brasil. *Ciência Florestal*, v. 30, n. 1, p. 176–191, 2020. ISSN 1980-5098. Disponível em: <https://doi.org/10.5902/1980509837696>.

LING, C. X.; SHENG, V. S. Cost-sensitive learning and the class imbalance problem. In: SAMMUT, C.; WEBB, G. I. (Ed.). *Encyclopedia of Machine Learning*. Boston, MA: Springer, 2011. p. 231–235.

MCCARTHY, J. What is artificial intelligence? *Documento online, Stanford University*, 2007. Definição clássica pelo criador do termo. Disponível em: <http://jmc.stanford.edu/articles/whatisai/whatisai.pdf>.

MITCHELL, T. M. *Machine learning*. [S.l.]: McGraw-hill, 1997.

NASCIMENTO, D. T. F.; ARAUJO, F. M. d.; JUNIOR, L. G. F. Análise dos padrões de distribuição espacial e temporal dos focos de calor no bioma cerrado. *Revista Brasileira de Cartografia*, v. 63, n. 4, p. 461–475, 2011. ISSN 1808-0936. Fundamental para caracterizar o fogo no Cerrado como fenômeno determinístico e sazonal.

PANG, Y. et al. Forest fire occurrence prediction in china based on machine learning methods. *Remote Sensing*, v. 14, n. 21, p. 5546, 2022.

PÉREZ-PORRAS, F.-J. et al. Machine learning methods and synthetic data generation to predict large wildfires. *Sensors*, v. 21, n. 11, p. 3694, 2021.

PHELPS, N.; WOOLFORD, D. G. Guidelines for effective evaluation and comparison of wildland fire occurrence prediction models. *International Journal of Wildland Fire*, v. 30, n. 4, p. 225–240, 2021.

RIBEIRO, S. M. *Imputation by decomposition and by time series nature: novel imputation methods for missing data in time series*. Dissertação (Dissertacao de Mestrado em Engenharia Eletrica) — Universidade Federal de Minas Gerais, Belo Horizonte, MG, 2021. Disponível em: <http://hdl.handle.net/1843/46099>.

RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. [S.l.]: Pearson Education Limited, 2016.

RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 4th. ed. Hoboken, NJ: Pearson, 2021. (Pearson Series in Artificial Intelligence). ISBN 978-0-13-461099-3. Disponível em: <http://aima.cs.berkeley.edu>.

SAITO, T.; REHMSMEIER, M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, v. 10, n. 3, p. e0118432, 2015. Mostra que curvas Precision-Recall são mais informativas que ROC em cenários com forte desbalanceamento.

scikit-learn developers. *GridSearchCV*. [S.l.], 2026. Documentation for scikit-learn. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html>.

SETZER, A. W.; SISMANOGLU, R. A.; SANTOS, J. G. M. d. *Método do Cálculo do Risco de Fogo do Programa do INPE - Versão 11, Junho/2019*. São José dos Campos, SP, Brasil, 2019. Substitui a versão 20130910_RF_V9.docx e anteriores. Disponível em: <http://urlib.net/8JMKD3MGP3W34R/3UEDKUB>.

SHU, L. et al. Towards fire prediction accuracy enhancements by leveraging an improved naïve bayes algorithm. *Symmetry*, v. 13, n. 4, 2021. Propõe o método DWCNB (Double Weighted Naive Bayes with Compensation Coefficient). Disponível em: <https://www.mdpi.com/2073-8994/13/4/530>.

SILVA, M. F. A.; WHITE, B. L. A. Detecção de focos de calor através de satélites nos distintos biomas brasileiros de 1999 a 2015. In: UNIVERSIDADE FEDERAL DE SERGIPE. *Anais do 4º Simpósio sobre as geotecnologias e geoinformação no Estado de Alagoas (GeoAlagoas)*. Alagoas, Brasil, 2016.

TROYANSKAYA, O. et al. Missing value estimation methods for DNA microarrays. *Bioinformatics*, v. 17, n. 6, p. 520–525, 2001.