PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

A INTELIGÊNCIA ARTIFICIAL APLICADA NA EXPLORAÇÃO ESPACIAL MODERNA

ALAN ARAÚJO FERNANDES

GOIÂNIA
2025

ALAN ARAÚJO FERNANDES

A INTELIGÊNCIA ARTIFICIAL APLICADA NA EXPLORAÇÃO ESPACIAL MODERNA

Trabalho de Conclusão de Curso apresentado à Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título em Ciência da Computação.

Orientador:

Prof. Me. Gustavo Siqueira Vinhal

Banca examinadora:
Prof.: Dr. Sibelius Lellis Vieira
Prof.: Me. Fernando Gonçalves Abadia

GOIÂNIA
2025

ALAN ARAÚJO FERNANDES

A INTELIGÊNCIA ARTIFICIAL APLICADA NA EXPLORAÇÃO ESPACIAL MODERNA

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel em Ciências da Computação, e aprovado em sua forma final pela Escola de Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, em ____/_____/_____.

_____

Orientador(a): Prof. Gustavo Siqueira Vinhal

_____

Prof. Dr. Sibelius Lellis Vieira

_____

Prof. Me. Fernando Gonçalves Abadia

GOIÂNIA
2025

AGRADECIMENTOS

Agradeço ao professor Gustavo Siqueira Vinhal, meu orientador acadêmico, pelo apoio e confiança depositados neste trabalho, agradeço também e tão quanto importante a minha família, minha mãe Maria José de Oliveira Fernandes Luz e meu pai Agnaldo Araújo Moura Luz, agradeço também aos meus amigos que me apresentaram essa temática da astronomia que me fascina até hoje, e é por eles que me dediquei a fazer este trabalho afim de eternizar toda uma filosofia e união entre eu e quem me incentivou, obrigado a todos.

RESUMO

O presente trabalho investiga a aplicação do algoritmo *Extreme Gradient Boosting* (XGBoost) na classificação binária de exoplanetas, um desafio crucial imposto pelo grande volume de dados gerados pelas missões espaciais. O objetivo central foi implementar e avaliar um modelo de *Machine Learning* com ênfase na otimização de hiperparâmetros e na comprovação da capacidade de generalização entre os conjuntos de dados distintos das missões Kepler e TESS. A metodologia envolveu a coleta e o pré-processamento de dados de ambas as missões, a seleção de 10 características (*features*) relevantes e o treinamento do XGBoost, com o ajuste fino de hiperparâmetros por meio da técnica *RandomizedSearchCV*. Os resultados demonstraram o sucesso da otimização: o modelo otimizado alcançou uma Acurácia de Teste de 84,36% no *dataset* Kepler, superando modelos que apresentaram subajuste (*underfitting*) e sobreajuste (*overfitting*). A generalização foi então comprovada com a aplicação do mesmo modelo no *dataset* da missão TESS, mantendo um alto desempenho com Acurácia de Teste de 79,75%. Conclui-se que o modelo desenvolvido é uma ferramenta eficiente, robusta e versátil, capaz de modernizar e acelerar o processo de identificação de candidatos a exoplanetas na exploração espacial moderna.

Palavras - chave: Inteligência Artificial. XGBoost. Exoplanetas. *Machine Learning*. Kepler. TESS.

ABSTRACT

This undergraduate thesis investigates the application of the Extreme Gradient Boosting (XGBoost) algorithm in the binary classification of exoplanets, a crucial challenge posed by the large volume of data generated by space missions. The central objective was to implement and evaluate a Machine Learning model with an emphasis on hyperparameter optimization and verification of its generalization capacity between the distinct datasets of the Kepler and TESS missions. The methodology involves the collection and preprocessing of data from both missions, the selection of 10 relevant features, and the training of XGBoost, with fine-tuning of hyperparameters performed using the *RandomizedSearchCV* technique. The results obtained demonstrate the success of the optimization: the optimized model achieved a Test Accuracy of 84.36% on the Kepler dataset, outperforming models that exhibited underfitting and overfitting. The generalization was then proven by applying the same model to the TESS mission dataset, maintaining high performance with a Test Accuracy of 79.75%. It is concluded that the developed model is an efficient, robust, and versatile tool, capable of modernizing and accelerating the process of identifying exoplanet candidates in modern space exploration.

Key - words: Artificial Intelligence. XGBoost. Exoplanets. Machine Learning. Kepler. TESS.

LISTA DE ABREVIATURAS E SIGLAS

CSV	*Comma-Separated Values*
FP	*False Positive*
IA	Inteligência Artificial
KOIs	Objetos de Interesse Kepler
ML	*Machine Learning*
NaN	*Not a Number*
NASA	*National Aeronautics and Space Administration*
PC	*Planet Candidate*
SVMs	*Support Vector Machines*
TESS	*Transiting Exoplanet Survey Satellite*
TOIs	Objetos de Interesse TESS
XGBoost	*Extreme Gradient Boosting*

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Nas últimas décadas, o avanço tecnológico tem impulsionado de forma significativa a área da astronomia, culminando em missões espaciais de grande escala focadas na descoberta de exoplanetas, definidos como planetas que orbitam outras estrelas além do Sol. Dessas missões, destacam-se o Telescópio Espacial Kepler e o Satélite de Pesquisa de Exoplanetas em Trânsito (TESS), os quais revolucionaram a ciência de exoplanetas ao acumularem vastas quantidades de dados essenciais para a busca de novos planetas (JENKINS, 2022).

Essas missões espaciais geraram dados principalmente através do método de trânsito, que consiste em monitorar o brilho das estrelas ao longo do tempo e perceber pequenas variações ou quedas em seu brilho, o que pode indicar a passagem de um planeta. O resultado são milhões de “curvas de luz”, ou seja, grandes conjuntos de dados de séries temporais (BORUCKI, 2016). Nisso a verificação manual de cada sinal suspeito por especialistas é inviável devido ao volume de dados, criando assim a necessidade de soluções automatizadas e inteligentes (ARMSTRONG et al., 2019). Assim encontramos um desafio computacional onde deve-se analisar automaticamente essa grande quantidade de dados para identificar esses padrões sutis que correspondam a um trânsito planetário real, onde esses padrões são eventos raros e precisam ser distinguidos de diversas fontes de ruído, como variações naturais da estrela ou falhas instrumentais (CUÉLLAR et al., 2022).

Diante do desafio imposto pelo volume e pela complexidade dos dados das curvas de luz, a Inteligência Artificial (IA), e mais especificamente o *Machine Learning* (Aprendizado de máquina - ML), oferecem ferramentas computacionais poderosas para a análise automatizada (MITCHELL, 1997). A principal força do ML está na sua capacidade de aprender padrões complexos diretamente a partir dos dados, superando as abordagens de programação tradicionais em tarefas de reconhecimento, como a identificação de trânsitos planetários. Esse estudo aplica a Aprendizagem Supervisionada, que é uma técnica de ML no qual o algoritmo é treinado com exemplos previamente rotulados, neste caso, curvas de luz identificadas como “Planeta Candidato” ou “Falso Positivo”, para aprender a fazer classificações precisas em novos dados. Embora o ML englobe diversas metodologias, incluindo o *Deep Learning* (Aprendizagem Profunda*)* (LECUN et al., 2015), algoritmos baseados em árvores de decisão e métodos de *ensemble* (métodos de conjunto), como o *Gradient Boosting*, têm se mostrado particularmente eficazes no tratamento de dados

tabulares estruturados, formato comum aos catálogos de candidatos a exoplanetas (SHWARTZ-ZIV et al., 2021).

Com base nisso, o algoritmo escolhido para este trabalho foi o *Extreme Gradient Boosting* (XGBoost), uma implementação de *Gradient Boosting* de alta performance conhecida por sua precisão e eficiência (CHEN et al., 2016). O desempenho de modelos como o XGBoost, no entanto, depende diretamente da correta calibração de seus hiperparâmetros. Uma configuração mal ajustada pode levar ao *overfitting* (sobreajuste) quando o modelo "decora" os dados de treino e não consegue generalizar para novos dados ou ao *underfitting* (subajuste) quando o modelo é simples demais para capturar os padrões relevantes.

A partir disso, formula-se o problema de pesquisa deste estudo: de que forma o processo de otimização de hiperparâmetros pode garantir o melhor desempenho de um modelo XGBoost para classificar os candidatos a exoplanetas das missões da NASA?

Este estudo tem como objetivo geral desenvolver e avaliar um modelo XGBoost otimizado para essa tarefa de classificação. Para isso, os objetivos específicos são:

a) Realizar o pré-processamento e a engenharia de *features* (características) nos dados da missão Kepler base_kepler.csv.
b) Aplicar os hiperparâmetros otimizados ao modelo XGBoost, definidos a partir de uma etapa prévia de otimização utilizando *RandomizedSearchCV*.
c) Treinar e avaliar o modelo otimizado nos dados do Kepler, analisando sua acurácia final e outras métricas.
d) Comparar o desempenho do modelo otimizado (Kepler) com configurações alternativas para demonstrar na prática os efeitos do *overfitting* e *underfitting.*
e) Testar a generalização do modelo, aplicando a mesma metodologia de análise aos dados da missão TESS base_tess.csv e avaliando seu desempenho.

A relevância deste trabalho, portanto, vai além do desafio computacional e se firma na própria justificativa de exploração espacial. Conforme argumentado pelo astrônomo Carl Sagan, o qual serviu como principal inspiração para esta pesquisa, a sobrevivência a longo prazo da humanidade depende de sua capacidade de se tornar uma espécie viajante espacial. Em sua obra, Sagan defende que confinar-se a um único planeta, por mais vasto que pareça, expõe a civilização a um risco existencial,

e que a expansão pelo espaço é, em última análise, um “instinto de sobrevivência” (SAGAN, 2019).

Nesse contexto, a busca por exoplanetas, mundos que possam, um dia, abrigar a vida ou mesmo a própria humanidade, é o primeiro passo prático e fundamental nessa jornada. Como demonstrado por missões como Kepler e TESS, essa busca configura atualmente um desafio estruturado de ciência de dados. Portanto, desenvolver e otimizar métodos de IA, como os propostos neste trabalho, não é apenas um exercício técnico; é a criação das ferramentas essenciais que permite, de forma prática e escalável, analisar o “dilúvio” de dados e identificar os “pálidos pontos azuis” (*pale blue dots*, SAGAN, 2019) que podem garantir esse futuro.

Este trabalho está estruturado em cinco capítulos, com o objetivo de guiar o leitor de forma lógica desde o contexto do problema até a sua solução e conclusão.

O Capítulo 1, introdução, apresenta a contextualização do tema, detalhando o desafio computacional na detecção de exoplanetas, o problema de pesquisa focado na otimização de modelos, os objetivos gerais e específicos, e a justificativa que conecta a pesquisa à exploração espacial.

O Capítulo 2 revisa a literatura pertinente, abordando os métodos de detecção de exoplanetas, com foco no método de trânsito (missões Kepler e TESS), e os conceitos fundamentais de IA, ML, Aprendizagem Supervisionada e algoritmos de *ensemble*, detalhando o funcionamento do *Gradient Boosting* e do XGBoost.

O Capítulo 3 descreve toda sequência prática desenvolvida. Isso inclui a origem e seleção dos dados das missões Kepler e TESS, as etapas de limpeza e pré-processamento, a engenharia de *features* aplicada, e a configuração detalhada dos experimentos, incluindo a definição dos hiperparâmetros para os modelos (otimizado, *overfitting* e *underfitting*) e as métricas de avaliação.

O Capítulo 4 apresenta e analisa os dados obtidos nos experimentos. Este capítulo detalha a performance do modelo otimizado nos dados do Kepler, compara seus resultados com os modelos de *overfitting* e *underfitting* para validar a otimização, e discute o desempenho do modelo nos dados do TESS como prova de generalização.

O Capítulo 5 sintetiza os resultados do trabalho, retorna o problema de pesquisa e os objetivos para verificar se foram alcançados, discute as limitações encontradas durante o estudo e propõe direções para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos fundamentais necessários para a compreensão do trabalho. Inicialmente, aborda-se a área da astronomia foca na busca por exoplanetas, detalhando os principais métodos de detecção e as missões espaciais relevantes. Em seguida, explora-se o campo da IA, com ênfase em ML Aprendizagem Supervisionada e especificamente, o algoritmo XGBoost utilizado neste estudo, bem como os desafios associados à otimização de seus hiperparâmetros.

2.1 Fundamentos da detecção de exoplanetas

A busca por planetas fora do sistema solar, conhecidos como exoplanetas, representa uma das fronteiras mais ativas e fascinantes da astronomia moderna. A confirmação do primeiro exoplaneta orbitando uma estrela semelhante ao Sol (MAYOR et al., 1995) marcou o início de uma nova era na exploração cósmica, motivando o desenvolvimento de técnicas e missões espaciais cada vez mais sofisticadas.

2.1.1 Métodos de detecção de exoplanetas

A detecção de exoplanetas é um desafio técnico considerável, dada a imensa distância e o contraste de brilho entre as estrelas e seus planetas. Diversos métodos foram desenvolvidos para superar essas dificuldades, sendo os principais (PERRYMAN, 2018):

a) **Velocidade Radial**: Mede a pequena oscilação na posição de uma estrela causada pela atração gravitacional de um planeta em órbita. Essa oscilação provoca um desvio Doppler no espectro de luz da estrela, permitindo inferir a presença e a massa mínima do planeta. (MAYOR et al., 1995).
b) **Trânsito Planetário**: Detecta a diminuição na queda no brilho de uma estrela que ocorre quando um planeta passa em frente a ela, bloqueando parte de sua luz. Este método fornece informações sobre o tamanho do planeta e seu período orbital.
c) **Microlente Gravitacional**: Baseia-se no efeito de lente gravitacional. Quando um objeto massivo (como uma estrela com um planeta) passa na frente de uma estrela mais distante, sua gravidade curva a luz da estrela do fundo, amplificando-a temporariamente.

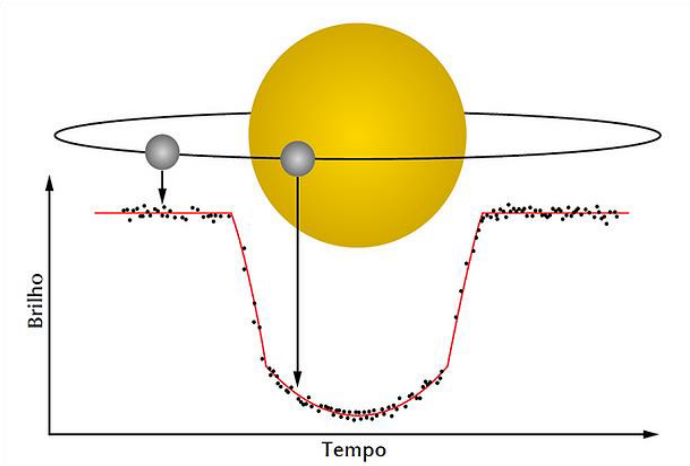
d) **Imageamento Direto**: Consiste em obter imagens diretas do exoplaneta. É extremamente desafiador devido ao brilho intenso da estrela hospedeira, mas avanços em óptica adaptativa têm possibilitado a detecção direta de planetas grandes e distantes de suas estrelas.
e) **Astrometria**: Mede o minúsculo movimento lateral de uma estrela no céu, causado pela órbita de um planeta ao seu redor.

2.1.2 O Método de trânsito planetário

O método de trânsito planetário, utilizado pelas missões Kepler e TESS, tornou-se uma das técnicas mais produtivas na descoberta de exoplanetas (PERRYMAN, 2018). Ele se baseia na observação contínua do brilho de um grande número de estrelas. Quando um planeta, em sua órbita, passa entre a estrela e o observador (Terra), ele bloqueia uma pequena fração da luz, estelar, causando uma diminuição periódica e característica no brilho observado.

A análise dessas diminuições, registradas em gráficos conhecidos como “curvas de luz” (séries temporais de brilho), permite determinar o período orbital do planeta (o tempo entre trânsitos sucessivos) e estimar seu raio (relacionado à profundidade da queda de brilho), conforme ilustrado na **Figura 1**.

Figura 1 – Diagrama conceitual do método de trânsito planetário



Fonte: NASA/JPL-Caltech (s.d.)

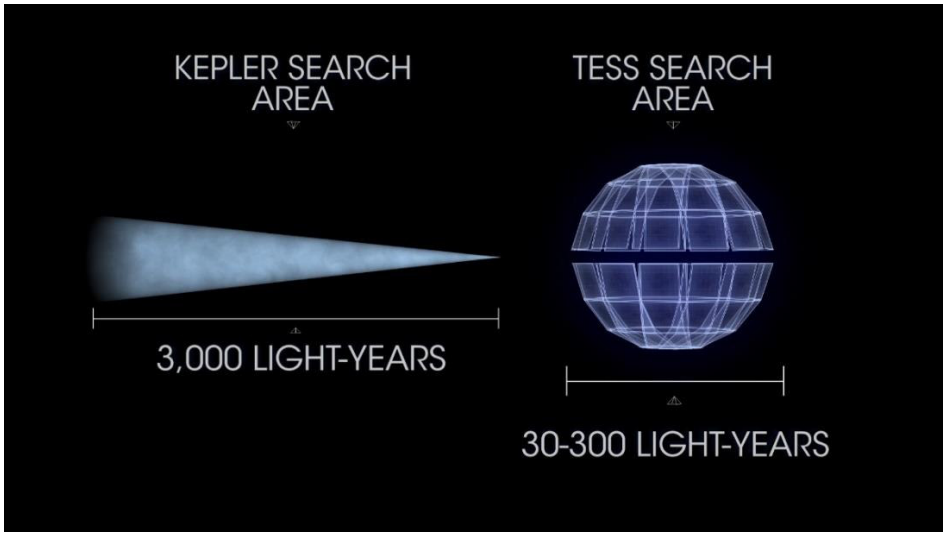
A principal vantagem do método de trânsito é a capacidade de detectar planetas relativamente pequenos, incluindo aqueles de tamanho similar à Terra, e a possibilidade de caracterizar suas atmosferas através de espectroscopia de transmissão (analisando a luz da estrela que passa pela atmosfera do planeta durante

o trânsito). No entanto, o método requer um alinhamento geométrico preciso entre a estrela, o planeta e o observador, fazendo com que apenas uma pequena fração dos sistemas planetários existentes seja detectável dessa forma. Além disso, sinais semelhantes a trânsitos podem ser gerados por outros fenômenos astrofísicos (como estrelas binárias eclipsantes) ou ruídos instrumentais, resultando em um grande número de “falsos positivos” que precisa, ser cuidadosamente filtrados e verificados (BORUCKI, 2016).

2.1.3 Missões kepler e tess

Embora o método de trânsito seja conceitualmente simples, sua aplicação prática para encontrar planetas do tamanho da Terra exigiu o desenvolvimento de observatórios espaciais dedicados, capazes de monitorar continuamente o brilho de centenas de milhares de estrelas com precisão fotométrica extrema (PERRYMAN, 2018). As duas missões mais impactantes nesse esforço, e que fornecem os dados para este trabalho, são o Telescópio Espacial Kepler e o TESS. A estratégia de observação das duas missões é fundamentalmente diferente, conforme ilustrado na **Figura 2**.

Figura 2 – Comparação da estratégia de observação do Kepler e do TESS



Fonte: NASA/Goddard Space Flight Center (s.d.)

O Telescópio Espacial Kepler, lançado em 2009, foi revolucionário. Sua estratégia consistiu em observar de forma contínua e fixa um único e pequeno campo de visão (cerca de 115 graus quadrados) na constelação de Cisne (BORUCKI, 2016). A missão principal do Kepler gerou um volume de dados brutos (as curvas de luz) na

ordem de centenas de gigabytes, tornando a análise manual impraticável (SILVA et al., 2023). Esses dados de séries temporais foram processados pela NASA para identificar todos os sinais periódicos de trânsito, que formam o catálogo de "Objetos de Interesse Kepler" (KOIs). Para cada KOIs, foram extraídas diversas *features* que resumem as propriedades do trânsito (como período, profundidade, duração etc.), resultando em um catálogo de dados tabulares estruturados, como o base_kepler.csv utilizado neste estudo, que é a base para a análise via ML.

O TESS, lançado em 2018, opera com uma estratégia fundamentalmente diferente. Em vez de um campo fixo e profundo, o TESS realiza uma varredura de quase todo o céu, observando diferentes setores por aproximadamente 27 dias cada (RICKER et al., 2015). Essa abordagem prioriza a descoberta de exoplanetas que transitam em estrelas mais próximas e brilhantes, facilitando estudos de acompanhamento. De forma similar ao Kepler, o TESS gera seu próprio catálogo de "Objetos de Interesse TESS" (TOIs) em formato tabular, como o base_tess.csv aqui analisado.

Ambas as missões, embora bem-sucedidas, enfrentam um desafio central idêntico: nem todo sinal de trânsito detectado é um planeta real. Como visto na Seção 2.1.2, uma variedade de fenômenos, como sistemas de estrelas binárias eclipsantes ou ruído instrumental, pode imitar um trânsito planetário, gerando um "falso positivo". Estima-se que uma fração significativa dos candidatos iniciais – em alguns catálogos, mais de 30% – não sejam planetas reais (CUNHA et al., 2021).

É exatamente neste ponto de chave – um volume massivo de catálogos de candidatos e a necessidade crítica de diferenciar "planetas candidatos" de "falsos positivos" de forma eficiente – que as técnicas de ML se tornam ferramentas indispensáveis (FARIA et al., 2022).

2.2 Uso de machine learning para classificação

Após a contextualização do problema de domínio – a necessidade de classificar eficientemente os catálogos de candidatos a exoplanetas (Seção 2.1) – esta seção aprofunda os fundamentos teóricos da solução computacional adotada: o ML. O foco será justificar a escolha de Aprendizagem Supervisionada e, especificamente, do algoritmo XGBoost, detalhando o desafio de otimização que é o ponto principal desta pesquisa.

2.2.1 Inteligência artificial e machine learning

A IA é um campo amplo da ciência da computação focado na criação de sistemas que podem simular capacidades humanas, como raciocinar, aprender e perceber (RUSSELL et al., 2013). Dentro da IA, encontra-se a subárea do ML, que se distingue por sua abordagem: em vez de serem explicitamente programados com regras, os sistemas de ML utilizam algoritmos para analisar grandes volumes de dados, aprender padrões neles contidos e, com base nisso, fazer predições ou tomar decisões (MITCHELL, 1997).

2.2.2 Aprendizagem supervisionada e classificação

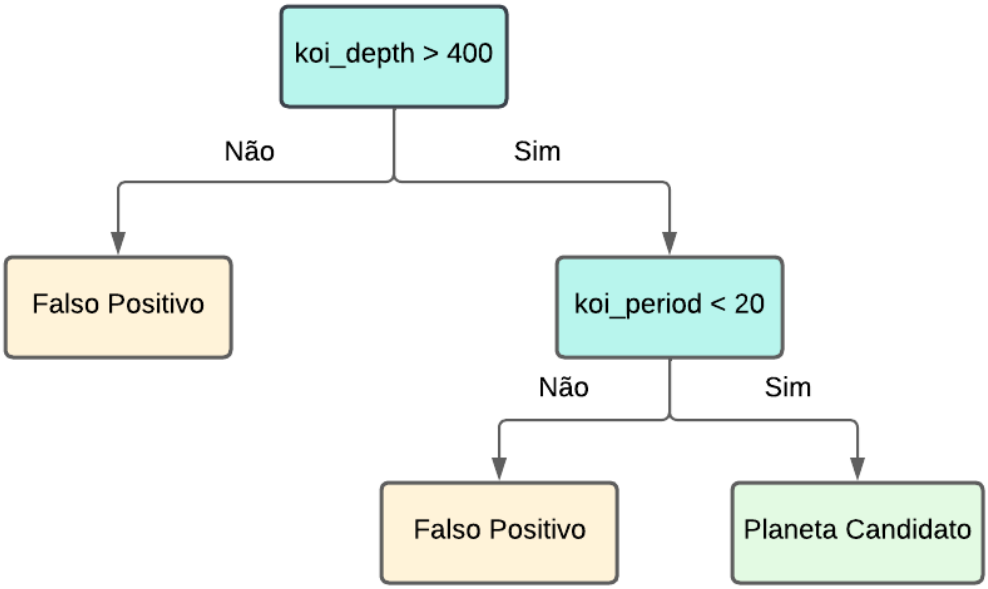
Os algoritmos de ML são frequentemente divididos em paradigmas, sendo o mais comum a Aprendizagem Supervisionada. Esta técnica é aplicada quando se possui um conjunto de dados onde os exemplos já contêm as "respostas corretas" (os rótulos, ou *labels*). O objetivo do algoritmo é aprender uma "função" que mapeie corretamente os dados de entrada *features* para os rótulos de saída (MITCHELL, 1997).

O problema central deste trabalho, classificar os KOIs e TOIs, é um exemplo de Aprendizagem Supervisionada, especificamente uma tarefa de classificação binária. Os dados de entrada são as 10 *features* (ex: koi_period, koi_depth) e os rótulos de saída são as duas classes possíveis: “Planeta Candidato” ou “Falso Positivo”. O modelo de ML, portanto, deve aprender os padrões complexos nas *features* que melhor separam essas duas classes.

2.2.3 Árvores de decisão

Dentre os diversos algoritmos de classificação supervisionada, a Árvore de Decisão é um dos mais intuitivos e fundamentais. Uma Árvore de Decisão funciona dividindo o conjunto de dados recursivamente através de uma série de perguntas simples ex: o koi_depth é > 400?; koi_period é < 20?. Essas divisões criam "ramos" que levam a "folhas", que representam a decisão final (a classe prevista), conforme ilustrado no diagrama conceitual da **Figura 3**.

Figura 3 – Exemplo conceitual de uma Árvore de Decisão



Fonte: Elaborado pelo autor (2025).

Embora simples e interpretáveis, árvores de decisão individuais são propensas a "decorar" os dados de treino, um problema conhecido como *overfitting* (HASTIE et al., 2009).

As Árvores de Decisão são modelos supervisionados que operam por meio da partição recursiva do espaço de atributos, criando divisões sucessivas que aproximam a função de decisão desejada. Esse processo ocorre a partir do nó raiz, que contém todo o conjunto de dados, e se estende pelos nós internos, onde são aplicadas regras lógicas baseadas nos atributos por exemplo, depth > 200, até alcançar as folhas, que representam as classes finais atribuídas às instâncias.

O algoritmo avalia diferentes cortes possíveis para cada atributo e seleciona aquele que produz a divisão mais “pura”, ou seja, aquela em que os subconjuntos resultantes tendem a conter exemplos predominantemente de uma única classe. Para isso, emprega métricas como *Gini Impurity* ou *Information Gain*, que medem a qualidade de cada divisão. A seleção dessas divisões ocorre repetidamente, construindo a árvore de forma recursiva até que critérios de parada sejam satisfeitos, como profundidade máxima, quantidade mínima de amostras por nó ou ganho de informação reduzido.

Após treinada, a árvore realiza previsões conduzindo cada nova instância pelos nós internos, realizando as comparações necessárias até que ela alcance uma folha, que contém a classe mais frequente naquele nó durante o treinamento. Embora intuitivas e interpretáveis, Árvores de Decisão podem sofrer com *overfitting*, especialmente quando se tornam excessivamente profundas, motivo pelo qual técnicas posteriores, como *ensemble methods*, foram desenvolvidas para superar essas limitações.

2.2.4 Métodos de ensemble

Para superar as limitações de modelos individuais (como as Árvores de Decisão), foram desenvolvidos os métodos de *ensemble*. A premissa central é que, ao combinar as predições de múltiplos modelos mais simples (chamados de "aprendizes fracos"), o modelo final ("aprendiz forte") será muito mais preciso e consistente à variância dos dados. Existem duas estratégias principais de *ensemble* (HASTIE et al., 2009):

a) ***Bagging* (ex: *Random Forest*):** Vários modelos (ex: árvores) são treinados de forma paralela e independente, cada um em uma subamostra aleatória dos dados. A predição final é decidida por uma "votação" da maioria.
b) ***Boosting* (ex: *Gradient Boosting*):** Vários modelos (ex: árvores) são treinados de forma sequencial. A primeira árvore tenta resolver o problema; a segunda árvore é treinada para corrigir os erros da primeira; a terceira é treinada para corrigir os erros da segunda, e assim por diante. O modelo final é uma soma ponderada de todos os aprendizes.

2.2.5 Gradient boosting e o algoritmo xgboost

O *Gradient Boosting* é da família de algoritmos de *boosting* mais eficaz para dados tabulares estruturados, como os catálogos de exoplanetas (SHWARTZ-ZIV et al., 2021). O XGBoost, escolhido para este trabalho, é uma implementação específica de *Gradient Boosting* que se tornou o padrão da indústria e de competições acadêmicas por sua performance e eficiência (CHEN et al., 2016). O XGBoost otimiza o processo de *boosting* com técnicas como regularização (para prevenir *overfitting*), paralelização eficiente e tratamento otimizado de valores ausentes, resultando em alta acurácia preditiva.

O funcionamento do XGBoost baseia-se na construção sequencial de várias Árvores de Decisão do tipo *Decision Tree Regressor*. A cada iteração, o modelo ajusta uma nova árvore para corrigir os erros residuais da soma das anteriores. Esse processo segue a lógica do *gradient boosting*, no qual o modelo aprende gradualmente o gradiente da função de perda, reduzindo iterativamente o erro.

Cada árvore gerada é de baixa profundidade, mas a combinação de centenas delas forma um modelo altamente preditivo. O XGBoost utiliza ainda uma função objetivo composta por dois termos: o erro de treinamento e um termo de regularização, que penaliza árvores muito profundas ou modelos excessivamente complexos. Essa regularização controla o *overfitting* e melhora a capacidade de generalização.

Outro aspecto central é o tratamento eficiente dos dados. O algoritmo implementa divisão de nós baseada em ganho de informação calculado de forma otimizada, manipula valores ausentes ao aprender automaticamente para qual ramo enviar o *NaN*, emprega paralelização nas buscas dos melhores cortes e utiliza taxa de aprendizagem para estabilizar o processo de *boosting*. Em conjunto, essas características tornam o XGBoost especialmente adequado para dados tabulares estruturados, como os catálogos Kepler e TESS utilizados neste trabalho.

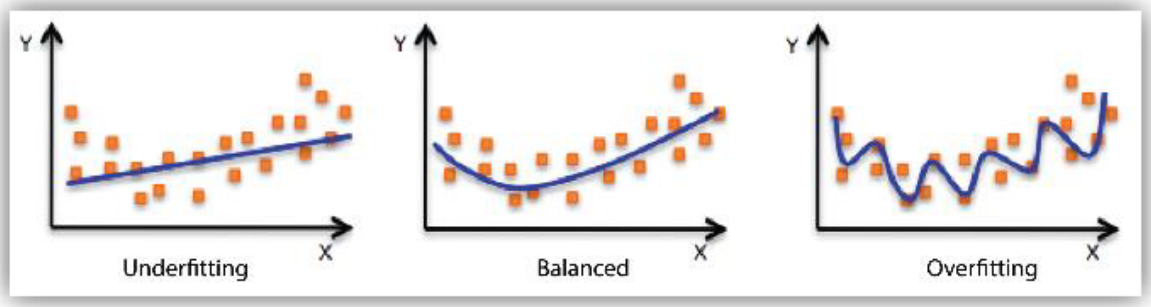
2.2.6 Otimização de hiperparâmetros

A eficácia de um modelo complexo como o XGBoost não é automática, ela depende diretamente de suas finas "configurações" internas, conhecidas como hiperparâmetros. Hiperparâmetros são os "botões" do algoritmo que o cientista de dados ajusta antes do treinamento (ex: max_depth - a profundidade máxima das árvores; n_estimators - o número de árvores; learning_rate - a taxa de aprendizado; *subsample* - a fração de amostras de treino; gamma - o parâmetro de regularização; e colsample_bytree - a fração de *features* por árvore).

O ajuste dos hiperparâmetros é o desafio central trabalho, pois busca encontrar o equilíbrio perfeito entre dois problemas comuns (HASTIE et al., 2009), conforme ilustrado na **Figura 4**:

a) ***Overfitting*:** Ocorre quando o modelo é complexo demais (ex: árvores muito profundas). Ele "decora" perfeitamente os dados de treino, mas falha miseravelmente em generalizar seu aprendizado para novos dados (os dados de teste).
b) ***Underfitting*:** Ocorre quando o modelo é simples demais (ex: poucas árvores). Ele falha em capturar os padrões complexos nos dados e apresenta um desempenho ruim tanto no treino quanto no teste.

Figura 4 – Relação entre complexidade do modelo, *Overfitting* e *Underfitting*



Fonte: Amazon Web Services (s.d.).

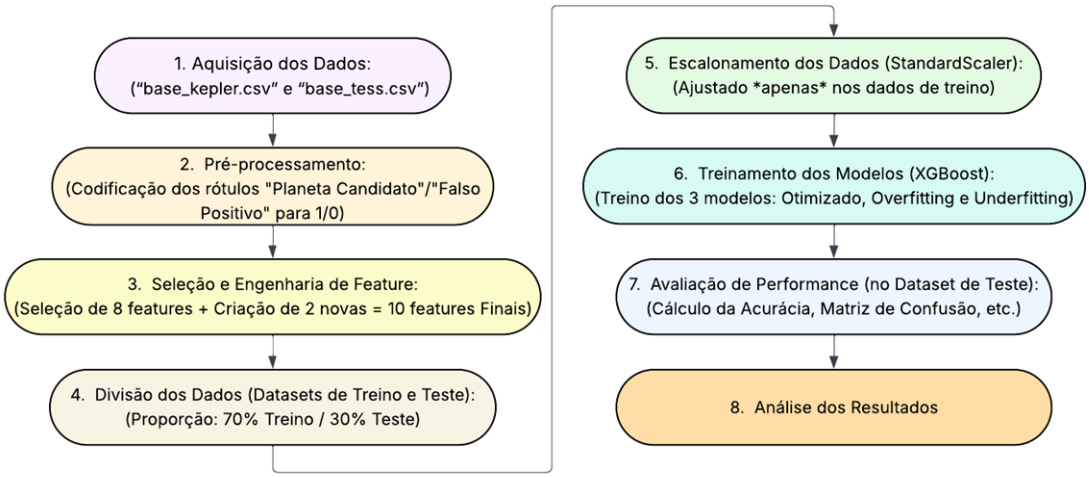
Encontrar o ponto ideal, o ponto de menor erro de teste, ou (*Balanced*), entre esses dois extremos é o objetivo da otimização de hiperparâmetros. Para este trabalho, os parâmetros otimizados foram determinados em uma etapa prévia de testes utilizando a técnica *RandomizedSearchCV*. A técnica de validação cruzada é

usada durante esse processo para simular como novos dados, permitindo uma escolha robusta dos hiperparâmetros que melhor generalizam. O experimento de comparação, mencionado no objetivo 'd' da Introdução (Capítulo 1), foi feito para mostrar esse fenômeno na prática, usando os dados reais do trabalho.

3 PROCEDIMENTOS METODOLÓGICOS

Este capítulo detalha a metodologia adotada, classificada (GIL, 2017) como bibliográfica e experimental. A pesquisa bibliográfica foi consolidada no Capítulo 2, enquanto o experimento prático, tem natureza quantitativa e objetivos explicativos. O experimento consistiu em uma sequência de passos para comparar um modelo otimizado, demonstrando a relação de causa e efeito dos hiperparâmetros. O fluxograma completo desses passos está ilustrado na **Figura 5**.

Figura 5 – Fluxograma da sequência metodológico



Fonte: Elaborado pelo autor (2025)

O objetivo é fornecer uma descrição clara e sistemática que garanta a reprodutibilidade dos experimentos, validando os resultados que serão discutidos no capítulo seguinte.

3.1 Obtenção e preparação dos dados

O primeiro passo da sequência metodológica consistiu na obtenção dos *datasets* (conjuntos de dados) que serviram como base para todo o treinamento e validação dos modelos.

Ambos os conjuntos de dados foram extraídos do repositório oficial do NASA *Exoplanet Archive*, um serviço de dados astronômicos mantido pela NASA e pelo Instituto de Tecnologia da Califórnia (AKESON et al., 2013). Os dados foram baixados em formato CSV (*Comma-Separated Values*). Para fins de organização e clareza do trabalho, os arquivos originais foram renomeados: o catálogo de KOIs foi nomeado base_kepler.csv, servindo como conjunto de dados principal para treinamento e otimização; e o catálogo de TOIs foi nomeado base_tess.csv, utilizado como conjunto secundário para o experimento de validação e generalização (objetivo 'e').

Após a obtenção, foi realizado um pré-processamento inicial em ambos os arquivos, utilizando a biblioteca Pandas (MCKINNEY, 2010). Esta etapa de preparação, que reflete o início dos *scripts* desenvolvidos ml_xgboost_kepler.ipynb e ml_xgboost_tess.ipynb, incluiu dois passos principais:

a) **Tratamento de Valores Ausentes:** Foi verificado que os catálogos selecionados já estavam curados e não continham valores ausentes *null* (Nulo) *ou NaN* (Not a Number – Não é um número) nas colunas de interesse, não exigindo etapas de imputação de dados.
b) **Codificação dos Rótulos:** O objetivo do modelo é uma classificação binária, mas os rótulos de alvo nos arquivos originais estavam em formato de texto. Portanto, eles foram convertidos para um formato numérico que o algoritmo XGBoost pudesse entender. No arquivo do Kepler, os rótulos “Planeta Candidato” (ou “PC” em dados do TESS) foram mapeados para o valor 1 (a classe positiva), e os rótulos “Falso Positivo” (ou “FP”) foram mapeados para o valor 0 (a classe negativa).

Com os dados carregados e os rótulos codificados, o próximo passo foi a análise e preparação das *features* de entrada, detalhada na seção seguinte.

3.2 Seleção e engenharia de features

Após a obtenção e o pré-processamento inicial dos dados das missões Kepler e TESS (detalhados na seção anterior), o conjunto de dados ainda possui um número muito elevado de colunas, ou *features*. No contexto de ML, nem todas as *features* são úteis; algumas podem ser redundantes, irrelevantes ou até mesmo introduzir "ruído" que prejudica a capacidade de aprendizado do modelo.

Esta etapa, portanto, é dividida em duas subatividades: a seleção de *features* e a engenharia de *features*.

A *Feature Selection* (Seleção de *Características)* é o processo de identificar e reter apenas o subconjunto de atributos que possui a maior relevância e poder preditivo para o problema. Conforme destaca AGGARWAL (2015), este processo é fundamental para mitigar o fenômeno conhecido como *curse of dimensionality* (maldição da dimensionalidade). Para este trabalho, foi realizada uma análise baseada em estudos correlatos da área de detecção de exoplanetas e na própria documentação da NASA sobre os catálogos (KOI e TOI). Esta análise permitiu isolar 8 *features* principais que descrevem as características físicas mais impactantes de um trânsito planetário, conforme detalhado no *script* ml_xgboost_kepler.ipynb.

Em complemento, foi aplicada a *Feature Engineering* (Engenharia de *Características)*. Esta técnica não se limita a selecionar dados existentes, mas sim em criar *features* a partir de combinações ou transformações matemáticas das *features* originais (TAN et al., 2006). Para este estudo, 2 novas *features* foram criadas. O objetivo desta engenharia é explicitar relações físicas que são fortes indicadores de falsos positivos, mas que não são óbvias para o modelo quando as *features* são analisadas isoladamente.

Ao final desta etapa, o conjunto de dados de entrada (X), que é efetivamente utilizado pelo modelo, foi consolidado em um conjunto otimizado de 10 *features* finais (as 8 originais selecionadas e as 2 criadas pela engenharia). Este conjunto de dados mais enxuto e denso em informação é a base que será utilizada nas etapas subsequentes de divisão, escalonamento e treinamento do modelo.

Tabela 1 – Features Finais Utilizadas no Modelo

Nome da Feature	Descrição (Justificativa e Cálculo)	Origem
period	**Nome Original:** koi_period Período orbital do candidato em dias.	Original
duration	**Nome Original:** koi_duration Duração do trânsito (em horas).	Original
depth	**Nome Original:** koi_depth Profundidade do trânsito (em ppm).	Original
radius	**Nome Original:** koi_prad Raio do planeta (em raios terrestres).	Original
insol	**Nome Original:** koi_insol Fluxo de insolação (radiação) recebido pelo planeta.	Original
stellar_temp	**Nome Original:** koi_steff Temperatura efetiva da estrela (em Kelvin).	Original
stellar_radius	**Nome Original:** koi_srad Raio da estrela (em raios solares).	Original
slogg	**Nome Original:** koi_slogg Gravidade de superfície da estrela (log(g)).	Original
depth_per_radius_sq	**Cálculo:** depth / (radius**2) Normaliza a profundidade do trânsito pelo quadrado do raio do planeta. Ajuda a distinguir trânsitos planetários de falsos positivos	Engenharia
duration_per_period	**Cálculo:** duration / period Cria uma razão entre a duração do trânsito e o período orbital, capturando a geometria do trânsito para identificar anomalias.	Engenharia

Fonte: Elaborado pelo autor (2025)

3.3 Preparação de Dados para o Modelo

Antes que o modelo de ML seja alimentado com os dados, é necessário realizar duas etapas críticas de preparação: dividir os dados e normalizá-los. A ordem e a maneira como isso é feito é fundamental para construir um modelo confiável.

3.3.1 Divisão de Treino e Teste

O objetivo principal de um modelo preditivo é fazer previsões precisas sobre dados que ele nunca viu antes. Para simular esse cenário e avaliar realisticamente o desempenho do modelo, não é viável treiná-lo e testá-lo usando o mesmo conjunto de dados.

Para isso, foi dividido o conjunto de dados total (com 9564 amostras) em dois subconjuntos:

a) **Conjunto de Treino (Dados de Treino):** composto por 70% dos dados originais (6695 amostras). Este é o conjunto que o modelo usará para "aprender" os padrões e é aqui que o modelo ajustará seus parâmetros internos;
b) **Conjunto de Teste (Dados de Teste):** composto pelos 30% restantes dos dados (2869 amostras). Este conjunto é "guardado" e mantido separado, o modelo nunca o verá durante o treinamento e ele será usado apenas no final, para avaliar o desempenho do modelo em dados "novos", fornecendo uma métrica imparcial de sua eficácia no mundo real.

Para garantir que a proporção de 'Planeta Candidato' e 'Falso Positivo' fosse a mesma em ambos os conjuntos, foi utilizada a técnica de estratificação (representada por stratify=y no código). Isso evita que, por azar, um conjunto tenha muito mais amostras de uma classe do que o outro.

3.3.2 A necessidade de normalização

Modelos de ML, especialmente algoritmos como Regressão Logística e SVMs (*Support Vector Machines*), podem ser sensíveis a *features* em escalas diferentes. No conjunto de dados Kepler do presente estudo, existem colunas com grandezas físicas muito distintas.

Por exemplo, a feature koi_period (Período Orbital) pode ter valores que variam de menos de 1 a mais de 1000 dias, enquanto a feature koi_impact (Parâmetro de Impacto) é um valor que, por definição, varia tipicamente entre 0 e 1.

Se não houver um tratamento disso, o modelo pode dar um peso indevidamente maior à coluna koi_period simplesmente porque seus números são maiores, e não porque ela é mais importante para a predição. O algoritmo não entende dias ou parâmetros, ele apenas vê números grandes e pequenos.

Para resolver isso, foi utilizado o *StandardScaler*. Essa técnica transforma os dados para que cada feature tenha uma média igual a 0 e um desvio padrão igual a 1. Isso coloca todas a *features* na mesma escala, garantindo que todas contribuam de forma justa para o resultado da previsão.

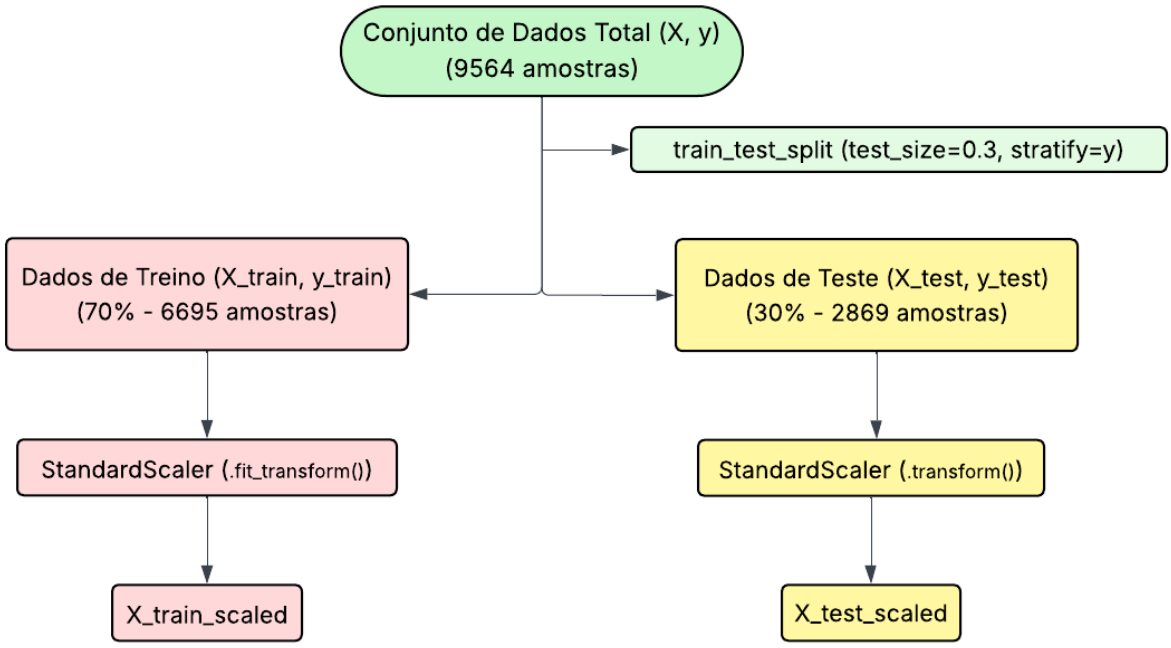
3.3.3 O processo correto e evitando "vazamento de dados"

Esta é a etapa mais crítica do pré-processamento. Aplicar a normalização na ordem errada pode levar a um problema grave chamado *data leakage (*vazamento de dados).

O vazamento de dados ocorre quando informações do conjunto de teste "vazam" para o processo de treinamento. Isso faz com que o modelo pareça ter um desempenho excelente nos testes, mas falhe miseravelmente quando exposto a dados verdadeiramente novos.

Para evitar isso, foi seguido o fluxo de trabalho rigoroso ilustrado no diagrama da **Figura 6**:

Figura 6 - Fluxograma do pré-processamento de dados.



Fonte: Elaborado pelo autor (2025)

O fluxo do processo foi executado em três etapas principais. Primeiro, na etapa de Divisão, os dados foram divididos em X_train (treino) e X_test (teste). Na sequência, para aprender com o treino, o *StandardScaler* foi instanciado e o método .fit_transform() foi aplicado apenas no conjunto de treino X_train; neste passo, o .fit() aprende a média e o desvio padrão somente dos dados de treino, e o .transform() aplica essa transformação, gerando o X_train_scaled. Por fim, para transformar o teste, foi usado o mesmo *scaler* já treinado com os dados de treino e foi aplicado apenas o método .transform() no conjunto de teste X_test. Este procedimento aplica a mesma escala (a média e o desvio padrão aprendidos no treino) aos dados de teste e simula perfeitamente como o modelo receberá dados no futuro: usando as regras que aprendeu no passado (treino) para transformar dados novos (teste). Ao final desse processo, temos os dados X_train_scaled e X_test_scaled prontos para serem usados na próxima etapa de treinamento e avaliação do modelo.

3.4 Modelo de machine learning selecionado

Para o problema de classificação dos dados da missão Kepler, este trabalho propõe o uso de um algoritmo de *ensemble* de alta performance o XGBoost.

3.4.1 Justificativa da escolha

Conforme detalhado na Fundamentação Teórica (Seção 2.2.5), este algoritmo é uma implementação otimizada de *Gradient Boosting* (CHEN et al., 2016). A sua seleção como modelo principal para este projeto se justifica por três motivos principais:

a) **alto Desempenho Preditivo:** sua reconhecida performance em competições de dados e problemas com dados tabulares (CHEN et al., 2016);
b) **eficiência Computacional:** sua capacidade de lidar com grandes volumes de dados de forma rápida e escalável (CHEN et al., 2016);

3.4.2 Metodologia de funcionamento

O funcionamento do XGBoost, que se baseia na construção sequencial de árvores de decisão na qual cada nova árvore corrige os erros (resíduos) da anterior (NIELSEN, 2016), foi extensamente abordado na (Seção 2.2.5). Na metodologia, este algoritmo foi alimentado com os dados de X_train_scaled para o seu treinamento.

3.5 Métricas de avaliação

Para avaliar o desempenho do modelo XGBoost de forma robusta, não basta apenas analisar a acurácia geral. Em problemas de classificação, especialmente com dados do mundo real que podem ser desbalanceados (como o do Kepler, onde " Falso Positivos" são numerosos), é essencial analisar o desempenho do modelo em mais detalhes. Neste trabalho, a avaliação do modelo será baseada em duas ferramentas principais da biblioteca Scikit-learn, que foram utilizadas no desenvolvimento do modelo a Matriz de Confusão e o Relatório de Classificação.

3.5.1 Matriz de Confusão

A Matriz de Confusão é uma tabela que visualiza o desempenho do algoritmo. Ela compara os valores reais contidos em y_test com as previsões feitas pelo modelo contidas em y_pred_xgb e tabula os resultados. Isso permite visualizar quatro cenários distintos para cada classe:

a) **Verdadeiro Positivo (VP):** o modelo previu uma classe (ex: "Planeta Candidato") e o valor real era "Planeta Candidato";
b) **Falso Positivo (FP):** o modelo previu uma classe (ex: " Planeta Candidato"), mas o valor real era outro (ex: "Falso Positivo"). Este é conhecido como Erro Tipo I;
c) **Verdadeiro Negativo (VN):** o modelo previu uma classe negativa (ex: "Falso Positivo") e o valor real era "Falso Positivo";
d) **Falso Negativo (FN):** o modelo previu uma classe negativa (ex: "Falso Positivo"), mas o valor real era positivo (ex: "Planeta Candidato"). Este é conhecido como Erro Tipo II.

3.5.2 Relatório de Classificação

O relatório de classificação é uma métrica textual que fornece um resumo das principais métricas de avaliação, calculadas para cada classe ("Planeta Candidato" e "Falso Positivo") com base na Matriz de Confusão:

a) Acurácia: é a métrica mais simples. Representa a porcentagem total de previsões corretas. Embora útil, pode ser enganosa em dados desbalanceados. É calculada pela Equação (1):

$$Acurácia = \frac{VP+VN}{Total} \quad (1)$$

b) Precisão: de todas as vezes que o modelo previu uma classe (ex: " Planeta Candidato"), quantas ele acertou? Uma alta precisão é importante para minimizar os Falsos Positivos. É calculada pela Equação (2):

$$Precisão = \frac{VP}{VP+FP} \quad (2)$$

c) Revocação: de todas as amostras que realmente eram de uma classe (ex: "Planeta Candidato"), quantas o modelo conseguiu encontrar? Uma alta revocação é crucial para minimizar os Falsos Negativos (ou seja, não deixar passar nenhum planeta). É calculada pela Equação (3):

$$Revocação = \frac{VP}{VP+FN} \quad (3)$$

d) *F1-Score*: é a média harmônica entre Precisão e Revocação. É uma métrica excelente para encontrar um equilíbrio entre as duas, especialmente quando as classes são desbalanceadas. É calculada pela Equação (4):

$$F1 - Score = 2\ x\ \frac{Precisão\ x\ Revocação}{Precisão + Revocação} \quad (4)$$

3.6 Treinamento e avaliação do modelo

A etapa final da metodologia consiste na execução do treinamento do modelo XGBoost e sua subsequente avaliação, conforme o processo definido no código-fonte deste projeto.

O treinamento foi realizado utilizando a função fit() do modelo. Neste processo, o algoritmo foi alimentado com os dados de treino X_train_scaled e seus respectivos rótulos y_train. O modelo, então, construiu iterativamente as árvores de decisão para aprender os padrões que separam as classes "Planeta Candidato" e "Falso Positivo").

Após o término do treinamento, o modelo foi submetido à avaliação. Para isso, foi utilizada a função predict() para gerar previsões para o conjunto de teste X_test_scaled, que o modelo nunca havia visto.

Os resultados dessas previsões y_pred_xgb foram, então, comparados com os rótulos reais y_test utilizando as métricas de avaliação definidas na seção anterior (Matriz de Confusão e Relatório de Classificação). Os resultados detalhados e a análise desse desempenho são apresentados no capítulo seguinte.

3.7 Definição dos modelos experimentais

Para validar de maneira prática os efeitos da otimização de hiperparâmetros, conforme descrito na Seção 2.2.6, foram definidos três perfis de modelo distintos para a etapa de experimentação.

A **Tabela 2** detalha a configuração exata de hiperparâmetros utilizada para cada um dos três modelos. Para garantir um teste de generalização justo e direto (objetivo 'e'), estes mesmos parâmetros foram aplicados de forma idêntica nos experimentos realizados com os datasets Kepler e TESS.

Tabela 2 – Configuração de Hiperparâmetros (Experimentos Kepler e TESS)

Hiperparâmetro	*Underfitting*	*Balanced*	*Overfitting*
n_estimators	1.0	500	50000
max_depth	1.0	5.0	500
learning_rate	0.001	0.02	0.6
gamma	1000.0	0.2	0.0
subsample	0.1	0.8	0.2
colsample_bytree	0.1	0.8	0.2

Fonte: Elaborado pelo autor (2025)

4 RESULTADOS E DISCUSSÃO

Este capítulo apresenta os resultados práticos obtidos pela execução da metodologia experimental descrita no Capítulo 3. Os dados foram extraídos das execuções dos scripts ml_xgboost_kepler.ipynb e ml_xgboost_tess.ipynb, que avaliaram os modelos no conjunto de dados de teste (30% do total) que não foi utilizado durante o treinamento.

O capítulo está dividido em três seções: a primeira detalha o desempenho do modelo principal (otimizado) no *dataset* Kepler; a segunda compara este modelo com as versões de *overfitting* e *underfitting* para validar a otimização; e a terceira avalia a capacidade de generalização do modelo no *dataset* do TESS.

4.1 Resultados do Modelo Otimizado (Kepler)

O Modelo Otimizado *Balanced*, que utilizou os hiperparâmetros ajustados detalhados na **Tabela 2**, foi avaliado no conjunto de teste do Kepler, composto por 2.870 amostras. O primeiro indicador de desempenho foi a acurácia geral, que representa o percentual total de acertos do modelo, que foi alcançada uma (Acurácia de 84,36%). Para uma análise mais detalhada do desempenho em cada classe, foi gerado o relatório de classificação, apresentado na **Tabela 3**.

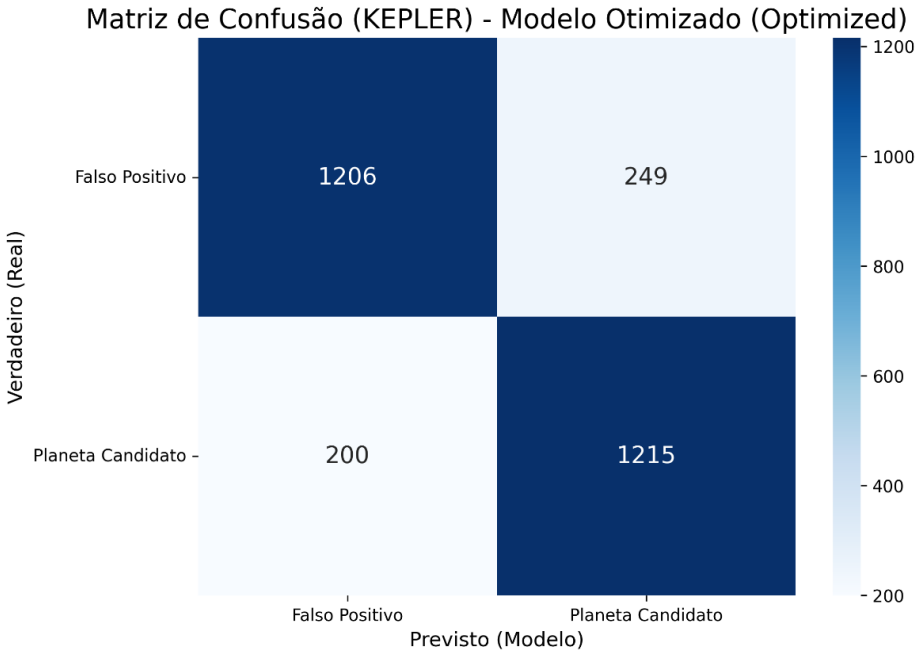
Tabela 3 – Relatório de Classificação (Kepler – Otimizado)

Classe	Precisão	Revocação	F1-Score	Amostras
Falso Positivo	0.86	0.83	0.84	1455
Planeta Candidato	0.83	0.86	0.84	1415
Média Ponderada	0.84	0.84	0.84	2870

Fonte: Elaborado pelo autor (2025)

Enquanto a **Tabela 3** fornece as métricas percentuais, a **Figura 7** apresenta a contagem absoluta de acertos e erros do modelo através da Matriz de Confusão.

Figura 7 – Matriz de Confusão (Kepler – Otimizado)



Fonte: Elaborado pelo autor (2025)

4.1.1 Análise dos resultados (Kepler – Otimizado)

A análise conjunta dos resultados demonstra um desempenho robusto do “Modelo Otimizado”. A Acurácia de 84,36% indica um alto índice de acertos gerais.

O Relatório de Classificação **Tabela 4** mostra um equilíbrio excelente entre as métricas. O *F1-Score*, que é a média harmônica entre precisão e revocação, foi de 0.84 para ambas as classes ("Falso Positivo" e "Planeta Candidato"). Isso indica que o modelo não demonstrou um viés significativo, sendo igualmente competente para identificar as duas classes.

A Matriz de Confusão **Figura 7** permite uma análise detalhada dos erros cometidos pelo modelo. A matriz mostra que:

- **Verdadeiros Negativos (VN):** 1.206 amostras que eram "Falso Positivo" foram corretamente classificadas.
- **Verdadeiros Positivos (VP):** 1.215 amostras que eram "Planeta Candidato" foram corretamente classificadas.
- **Erros Tipo I (Falsos Positivos):** O modelo errou 249 vezes, classificando um "Falso Positivo" como se fosse um "Planeta Candidato".
- **Erros Tipo II (Falsos Negativos):** O modelo errou 200 vezes, classificando um "Planeta Candidato" real como se fosse um "Falso Positivo".

Este desempenho é considerado muito positivo. O modelo demonstrou alta capacidade tanto em identificar planetas reais (Revocação de 0.86 para "Planeta Candidato") quanto em não classificar ruído como planeta (Precisão de 0.83 para "Planeta Candidato").

4.2 Análise comparativa dos modelos (Kepler)

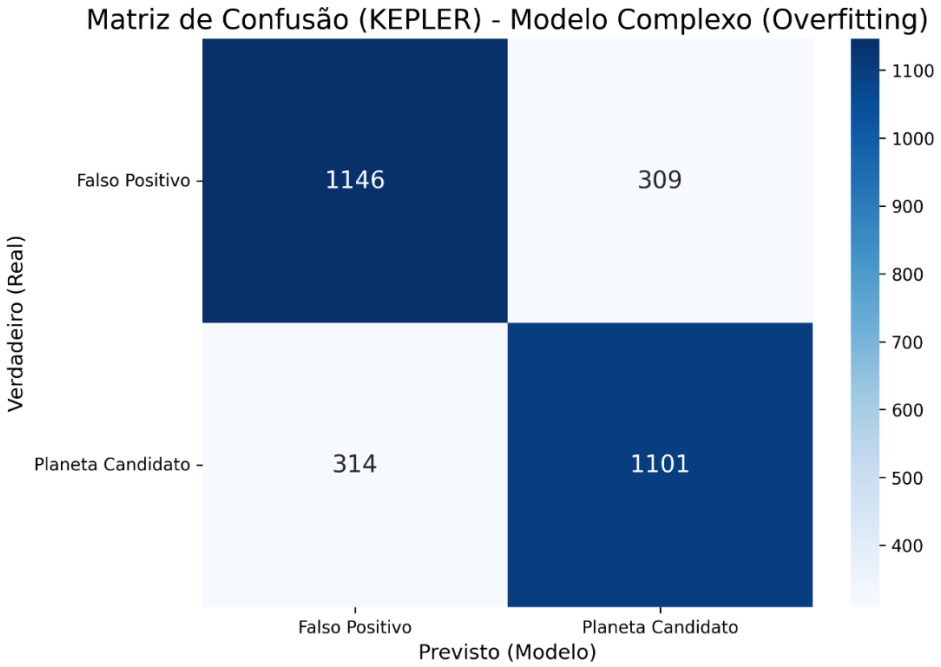
Conforme definido no objetivo 'd' da introdução, esta seção compara o modelo otimizado com as duas versões alternativas (supercomplexa e simples demais) para demonstrar empiricamente os efeitos do *overfitting* e *underfitting*. A **Tabela 5** resume as principais métricas de desempenho dos três modelos, enquanto as **Figuras 8 e 9** detalham as matrizes de confusão dos modelos não otimizados.

Tabela 4 – Comparativo de Desempenho dos Modelos (Kepler)

Métrica	Modelo Simples	Modelo Complexo	Modelo Otimizado
Acurácia (TREINO)	50,67%	99,99%	88,50%
Acurácia (TESTE)	50,70%	78,29%	84,36%
Gap (Treino - Teste)	+0,03%	-21,70%	-4,14%
F1-Score (Planeta Cand.)	0.00	0.78	0.84

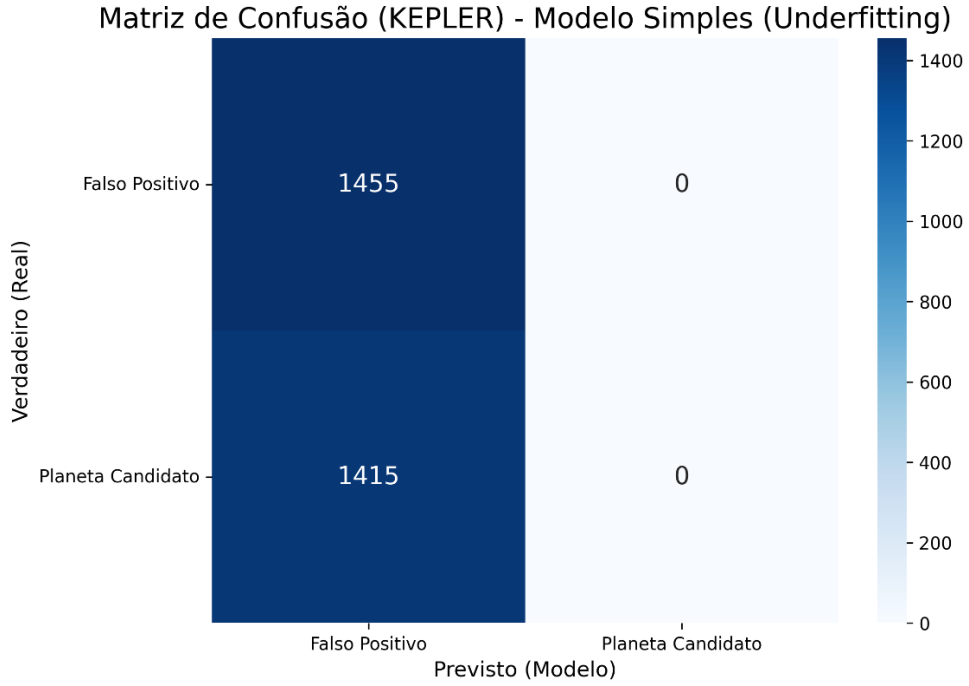
Fonte: Elaborado pelo autor (2025)

Figura 8 – Matriz de Confusão (Kepler – Complexo)



Fonte: Elaborado pelo autor (2025)

Figura 9 – Matriz de Confusão (Kepler – Simples)



Fonte: Elaborado pelo autor (2025)

4.2.1 Análise Comparativa

A análise comparativa **Tabela 4** valida experimentalmente a importância da otimização de hiperparâmetros. O “Modelo Otimizado” alcançou a maior Acurácia de Teste (84,36%), superando as demais configurações.

A eficácia da otimização é evidenciada pela análise do *gap* (diferença) entre as acurácias de treino e teste. O "Modelo Otimizado" apresentou uma diferença de apenas -4,14%, indicando generalização robusta. Em contrapartida, o "Modelo Complexo", apesar de atingir 99,99% de acurácia no treino, sofreu uma queda para 78,29% no teste, resultando em um *gap* de -21,70% que caracterizando uma sobre-aderência (*overfitting*).

O "Modelo Simples" demonstrou incapacidade de aprendizado. Com hiperparâmetros mínimos (ex: n_estimators=1), sua acurácia de teste (50,70%) é estatisticamente próxima a de um classificador aleatório para um problema binário. A **Figura 9** e o *F1-Score* de 0.00 para "Planeta Candidato" confirmam que o modelo previu apenas a classe ("Falso Positivo").

Conclui-se que a otimização de hiperparâmetros foi fundamental para modular a complexidade do modelo, evitando os extremos do *underfitting* e do *overfitting*, maximizando assim o desempenho preditivo em dados não vistos.

4.3 Resultados de generalização (TESS)

A etapa final do experimento consistiu em testar a capacidade de generalização do modelo, conforme o objetivo 'e'. Para isso, os mesmos três conjuntos de hiperparâmetros (Otimizado, Complexo, Simples) definidos na **Tabela 2** foram aplicados ao *dataset* da missão TESS, um conjunto de dados possui características de ruído e observação diferentes do Kepler.

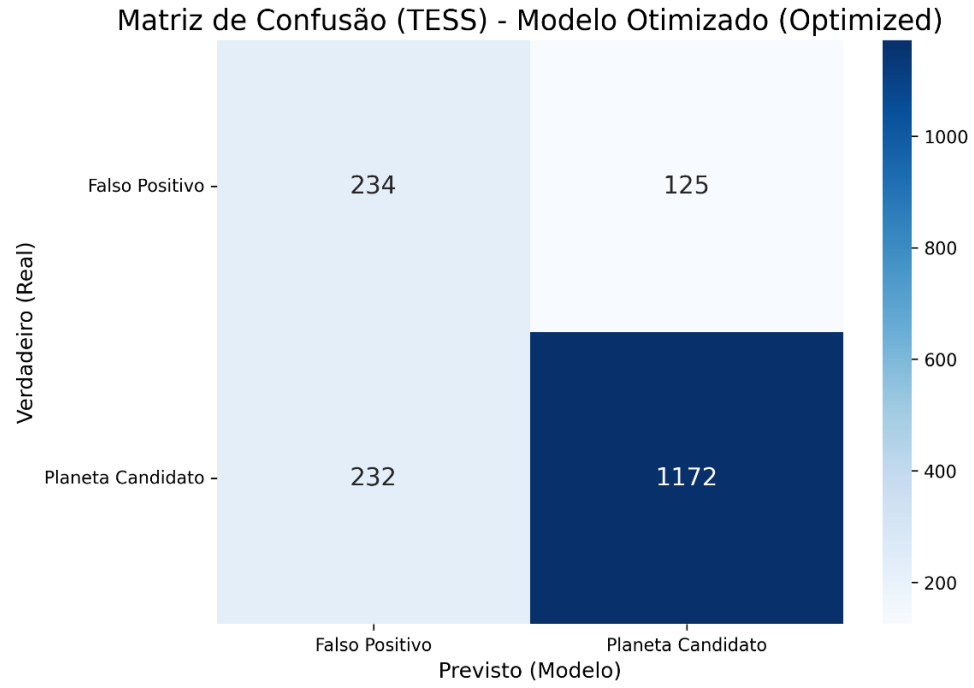
Tabela 5 – Comparativo de Desempenho dos Modelos (TESS)

Métrica	Modelo Simples	Modelo Complexo	Modelo Otimizado
Acurácia (TREINO)	20,37%	100,00%	87,60%
Acurácia (TESTE)	20,36%	76,52%	79,75%
Gap (Treino - Teste)	-0,01%	-23,48%	-7,85%
F1-Score (Planeta Cand.)	0.00	0.85	0.87

Fonte: Elaborado pelo autor (2025)

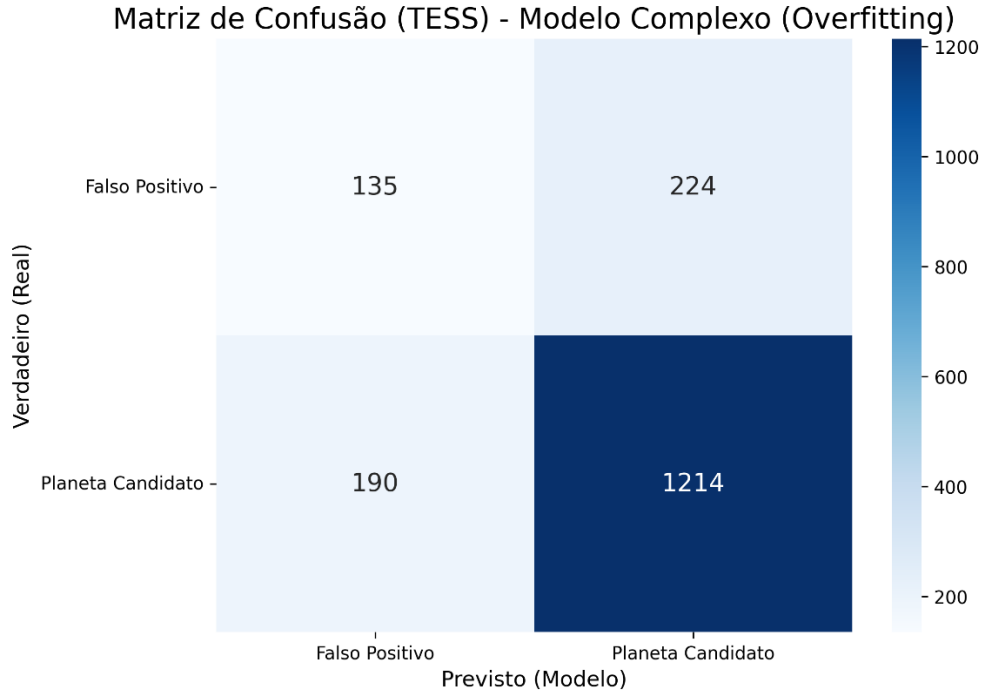
As **Figura 10, Figura 11** e **Figura 12** detalham as Matrizes de Confusão para cada um dos três cenários no *dataset* TESS.

Figura 10 – Matriz de Confusão (TESS – Otimizado)



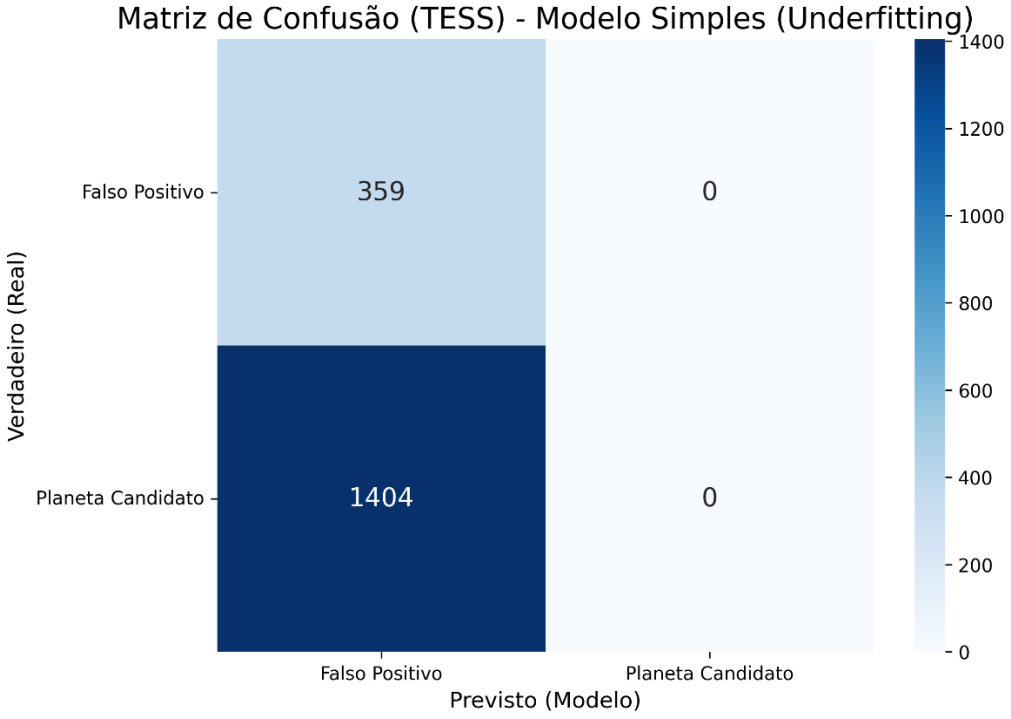
Fonte: Elaborado pelo autor (2025)

Figura 11 – Matriz de Confusão (TESS – Complexo)



Fonte: Elaborado pelo autor (2025)

Figura 12 – Matriz de Confusão (TESS – Simples)



Fonte: Elaborado pelo autor (2025)

4.3.1 Análise da Generalização (TESS)

Os resultados do experimento TESS reforçam as conclusões obtidas com o Kepler. O "Modelo Otimizado" novamente alcançou a maior Acurácia de Teste (79,75%), superando o modelo complexo (76,52%) e o de *underfitting* (20,36%). Isso demonstra que os hiperparâmetros otimizados no Kepler são robustos e generalizam bem para um *dataset* completamente diferente.

O fenômeno do *underfitting* foi ainda mais extremo no TESS, com o modelo simples **Figura 12** falhando em identificar um único "Planeta Candidato" (Precision 0.00). O *overfitting* **Figura 11**, por sua vez, demonstrou o maior *gap* de todos os experimentos (-23,48%), provando ser o pior método para generalização.

É notável que o "Modelo Otimizado" sofreu uma queda de performance ao ser aplicado no TESS (Acurácia de 84,36% no Kepler vs. 79,75% no TESS). Essa queda era esperada e é um resultado realista. Conforme discutido na Seção 2.1.3, as missões Kepler e TESS possuem estratégias de observação distintas, o que gera dados com perfis de ruído e características diferentes. A performance ainda alta (79,75%) em um *dataset* invisível e de outra missão valida o modelo como uma ferramenta de generalização eficaz, cumprindo o objetivo final da pesquisa.

5 CONCLUSÃO

Este trabalho teve como objetivo investigar a capacidade do algoritmo XGBoost na classificação de exoplanetas, utilizando dados obtidos das missões Kepler e TESS. Ao longo do desenvolvimento deste estudo, a metodologia aplicada e os resultados obtidos validaram completamente os objetivos propostos. O objetivo geral, que consistia em implementar e avaliar o desempenho do XGBoost na identificação de possíveis exoplanetas, com foco tanto na otimização de hiperparâmetros quanto na capacidade de generalização do modelo entre diferentes bases de dados astronômicos, foi alcançado. O modelo otimizado demonstrou desempenho confiável no conjunto de teste do Kepler, atingindo 84,36% de acurácia, e manteve uma performance elevada no *dataset* TESS, com acurácia de 79,75%, confirmando assim a hipótese central deste trabalho.

Os objetivos específicos também foram comprovados, desde a coleta e o pré-processamento dos dados, passando pela análise exploratória, seleção de *features* relevantes e aplicação da técnica de RandomizedSearchCV para a determinação dos hiperparâmetros ideais, até a validação da eficácia do modelo otimizado em relação a cenários de *overfitting* e *underfitting*. Além disso, foi verificado a capacidade de generalização do modelo otimizado quando aplicado a um conjunto de dados não utilizado na etapa de ajuste, o que reforçou a aplicabilidade em diferentes missões espaciais.

As contribuições deste estudo demonstram que o XGBoost se apresenta como uma alternativa altamente eficiente e de alto desempenho para a classificação de exoplanetas em dados tabulares. Além disso, a possibilidade de transferir os parâmetros otimizados entre diferentes bases astronômicas, como Kepler e TESS, evidencia um grande potencial de escalabilidade dos modelos de Inteligência Artificial na análise de dados provenientes de futuras missões de observação espacial, contribuindo significativamente para o avanço científico na astrofísica moderna.

Entretanto, este trabalho também reconhece limitações metodológicas, principalmente relacionadas à dependência de dados previamente processados e organizados em formato tabular. A utilização de séries temporais completas, representadas por curvas de luz originais das estrelas, poderia proporcionar ao modelo um aprendizado mais profundo e refinado das características que indicam potenciais exoplanetas. Nesse sentido, estudos futuros podem se concentrar no desenvolvimento de técnicas avançadas de engenharia de *features* a partir do sinal bruto, bem como na inclusão de algoritmos de Deep Learning, como redes

convolucionais, para estabelecer comparações diretas de desempenho e ampliar o escopo das análises. Outra possibilidade para continuidade deste trabalho consiste na implementação de um protótipo de software capaz de integrar o modelo otimizado a uma sequência de análise em tempo real, permitindo a classificação imediata de novos objetos de interesse conforme são identificados pela missão TESS.

Em suma, os resultados alcançados não apenas respondem à questão de pesquisa proposta, como também mostra o grande potencial da Inteligência Artificial como uma ferramenta capaz de modernizar, acelerar e ampliar a precisão da análise de grandes volumes de dados no campo da astrofísica, contribuindo para o contínuo avanço na descoberta e compreensão de exoplanetas fora do Sistema Solar.

REFERÊNCIAS BIBLIOGRÁFICAS

ARMSTRONG, David J. et al. **Machine-learning approaches to exoplanet transit detection and candidate validation in wide-field ground-based surveys**. Monthly Notices of the Royal Astronomical Society, v. 483, n. 4, p. 5534-5547, mar. 2019. Disponível em: https://academic.oup.com/mnras/article/483/4/5534/5199219. Acesso em: 10 jul. 2025.

BORUCKI, William J. **KEPLER Mission: development and overview**. Reports on Progress in Physics, v. 79, n. 3, p. 036901, fev. 2016. Disponível em: https://iopscience.iop.org/article/10.1088/0034-4885/79/3/036901. Acesso em: 12 set. 2025.

CHEN, Tianqi; GUESTRIN, Carlos. **XGBoost: A Scalable Tree Boosting System**. *In*: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 22., 2016, San Francisco. Proceedings [...]. New York: ACM, 2016. p. 785–794. Disponível em: https://arxiv.org/pdf/1603.02754. Acesso em: 11 ago. 2025.

CUÉLLAR, S. et al. **Deep learning exoplanets detection by combining real and synthetic data**. PLoS ONE, v. 17, n. 6, p. e0268310, jun. 2022. Disponível em: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0268310. Acesso em: 20 ago. 2025.

CUNHA, Júlia K. et al. **Validação de planetas e o problema dos falsos positivos**. *In*: CONGRESSO DE INICIAÇÃO CIENTÍFICA DA UNIVERSIDADE DE BRASÍLIA, 27., 2021, Brasília: UnB, 2021. p. 45-46. Disponível em: https://proic.unb.br/images/proic/27_pibic/ANAIS_PIBIC_2021.pdf. Acesso em: 23 out. 2025.

FARIA, Deivid R.; SILVA, Alan F.; LEÃO, Diogo S. **Processamento automático de curvas de luz para a identificação de exoplanetas por meio de uso de algoritmos de aprendizado de máquina**. Revista Brasileira de Iniciação Científica, v. 9, p. 1-20, 2022. Disponível em: https://periodicoscientificos.itp.ifsp.edu.br/index.php/rbic/article/view/1403. Acesso em: 23 out. 2025.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. **The elements of statistical learning**: data mining, inference, and prediction. 2. ed. New York: Springer, 2009. Disponível em: https://www.sas.upenn.edu/~fdiebold/NoHesitations/BookAdvanced.pdf. Acesso em: 25 out. 2025.

JENKINS, Jon M. *Chasing* ***Shadows in the Night: How NASA's Kepler and TESS Missions Are Revolutionizing Exoplanet Science***. NASA NTRS - Technical Reports Server, 2022. Disponível em: https://ntrs.nasa.gov/citations/20220017489. Acesso em: 19 ago. 2025.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. **Deep learning**. Nature, v. 521, n. 7553, p. 436–444, maio 2015. Disponível em:

https://www.researchgate.net/publication/277411157_Deep_Learning. Acesso em: 21 out. 2025.

MAYOR, Michel; QUELOZ, Didier. **A Jupiter-mass companion to a solar-type star**. Nature, v. 378, n. 6555, p. 355-359, nov. 1995. Disponível em: https://wwelsh.sdsu.edu/CLASSES/ASTR510/PAPERS/Mayor-Queloz_51Peg.pdf. Acesso em: 25 out. 2025.

MITCHELL, Tom M. **Machine Learning**. New York: McGraw-Hill, 1997. Disponível em: https://www.cs.cmu.edu/~tom/files/MachineLearningTomMitchell.pdf. Acesso em: 25 out. 2025.

PERRYMAN, Michael. **The exoplanet handbook**. 2. ed. Cambridge: Cambridge University Press, 2018. Disponível em: https://www.astro.amu.edu.pl/~voyager/EXO/vdoc.pub_the-exoplanet-handbook-2nd-edition.pdf. Acesso em: 25 out. 2025.

RICKER, George R. et al. Transiting Exoplanet Survey Satellite (TESS). **Journal of Astronomical Telescopes, Instruments, and Systems**, v. 1, n. 1, p. 014003, jan. 2015. Disponível em: https://www.spiedigitallibrary.org/journals/journal-of-astronomical-telescopes-instruments-and-systems/volume-1/issue-1/014003/Transiting-Exoplanet-Survey-Satellite-TESS/10.1117/1.JATIS.1.1.014003.full. Acesso em: 23 out. 2025.

RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência Artificial**. 3. ed. Rio de Janeiro: Elsevier, 2013. Disponível em: https://www.kufunda.net/publicdocs/Inteligência%20Artificial%20(Peter%20Norvig,%20Stuart%20Russell).pdf. Acesso em: 25 out. 2025.

SAGAN, Carl. **Pálido ponto azul**: uma visão do futuro da humanidade no espaço. Nova edição. São Paulo: Companhia das Letras, 2019.

SHWARTZ-ZIV, Ravid; ARMON, Amitai. ***Tabular Data: Deep Learning is Not All You Need***. arXiv preprint arXiv:2106.03253, 2021. Disponível em: https://arxiv.org/abs/2106.03253. Acesso em: 21 out. 2025.

SILVA, João P.; PAZ-CHINCHÓN, Fernando; LEÃO, Irapuan. **Detecção de exoplanetas em dados da missão Kepler usando deep learning**. *In*: SEMINÁRIO DE INICIAÇÃO CIENTÍFICA E INOVAÇÃO TECNOLÓGICA (SIIC), 13., 2023, Florianópolis: UTFPR, 2023. p. 1-8. Disponível em: https://anais.utfpr.edu.br/siic/2023/artigo/coens-10. Acesso em: 25 out. 2025.

AKESON, R. L. et al. **The NASA Exoplanet Archive: Data and Tools for Exoplanet Research**. Publications of the Astronomical Society of the Pacific, v. 125, n. 930, p. 989, 2013. Disponível em: https://exoplanetarchive.ipac.caltech.edu/docs/PASP_FINAL_The.NASA.Exoplanet.Archive.Data.and.Tools.for.Exoplanet.Research_July2013.pdf. Acesso em: 27 out. 2025.

MCKINNEY, Wes. **Data Structures for Statistical Computing in Python**. In: PROCEEDINGS OF THE 9TH PYTHON IN SCIENCE CONFERENCE (SCIPY 2010), 2010, Austin. p. 56-61. Disponível em: https://pdfs.semanticscholar.org/ef4e/f7f38bb907e5d7b4df3e6ff1db269d4970f5.pdf . Acesso em: 27 out. 2025.

AGGARWAL, Charu C. **Data mining:** the textbook. Cham: Springer, 2015. Disponível em: https://pzs.dstu.dp.ua/DataMining/bibl/Data%20Mining%20The%20Textbook.pdf. Acesso em: 30 out. 2025.

TAN, Pang-Ning; STEINBACH, Michael; KUMAR, Vipin. **Introduction to data mining**. Boston: Pearson Addison-Wesley, 2006. Disponível em: https://www.ceom.ou.edu/media/docs/upload/Pang-Ning_Tan_Michael_Steinbach_Vipin_Kumar_-_Introduction_to_Data_Mining-Pe_NRDK4fi.pdf. Acesso em: 30 out. 2025.

BROWNLEE, Jason**. XGBoost With Python. Machine Learning Mastery, 2016.** Disponível em: https://machinelearningmastery.com/xgboost-with-python
Acesso: 25 out. 2025.

NIELSEN, Michael. ***Neural Networks and Deep Learning*. 2016.** Disponível em: http://neuralnetworksanddeeplearning.com
Acesso: 27 out. 2025.

GIL, Antônio Carlos. **Métodos e técnicas de pesquisa social**. 7. ed. São Paulo: Atlas, 2017.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR

Av. Universitária, 1069 ● Setor Universitário
Caixa Postal 86 ● CEP 74605-010
Goiânia ● Goiás ● Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br ● reitoria@pucgoias.edu.br

RESOLUÇÃO n° 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O estudante ALAN ARAÚJO FERNANDES do Curso de CIÊNCIA DA COMPUTAÇÃO matrícula 2020.1.0028.0054-3, telefone: 62985603060 e-mail 2020.1.0028.0054-3@pucgo.edu.br, na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do Autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado Inteligência artificial aplicada na exploração espacial moderna, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto(PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 29 de setembro de 2025.

gov.br Documento assinado digitalmente
ALAN ARAUJO FERNANDES
Data: 29/09/2025 12:49:58-0300
Verifique em https://validar.iti.gov.br

Assinatura do autor: _____
Nome completo do autor: ALAN ARAÚJO FERNANDES

gov.br Documento assinado digitalmente
GUSTAVO SIQUEIRA VINHAL
Data: 29/09/2025 12:26:28-0300
Verifique em https://validar.iti.gov.br

Assinatura do professor–orientador: _____
Nome completo do professor-orientador: GUSTAVO SIQUEIRA VINHAL