UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA MECÂNICA
CURSO DE ENGENHARIA MECÂNICA

DESENVOLVIMENTO DE UM MODELO DE MACHINE LEARNING DE CLASSIFICAÇÃO PARA PREVISÃO DE FALHAS EM EQUIPAMENTOS

NOME DA ALUNA:
ISABELA MEDEIROS CHAVES

ORIENTADOR:
PROF. FRANCISCO FERNANDO R. PEREIRA

RECIFE
2025

ISABELA MEDEIROS CHAVES

DESENVOLVIMENTO DE UM MODELO DE MACHINE LEARNING DE CLASSIFICAÇÃO PARA PREVISÃO DE FALHAS EM EQUIPAMENTOS

Trabalho de Conclusão de Curso de Graduação em Engenharia Mecânica do Centro de Tecnologia e Geociências da Universidade Federal Pernambuco, como requisito para obtenção do título de Engenheira Mecânica.
Orientador: Prof. Francisco Fernando R. Pereira.

RECIFE

2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Chaves, Isabela Medeiros.
Desenvolvimento de um modelo de machine learning de classificação para previsão de falhas em equipamentos / Isabela Medeiros Chaves. - Recife, 2025.
53 p. : il., tab.

Orientador(a): Francisco Fernando R. Pereira
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia Mecânica - Bacharelado, 2025.

1. manutenção preditiva. 2. machine learning. 3. classificação de falhas. I. Pereira, Francisco Fernando R.. (Orientação). II. Título.

620 CDD (22.ed.)

Universidade Federal de Pernambuco
Departamento de Engenharia Mecânica Centro de Tecnologia e Geociências- CTG/EEP

DEMEC

ATA DE SESSÃO DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO – TCC2

Ao 26° dia do mês de março do ano de dois mil e vinte e cinco, às 15:00 horas, reuniu-se a banca examinadora para a sessão pública de defesa do Trabalho de Conclusão de Curso em Engenharia Mecânica da Universidade Federal de Pernambuco, intitulado **Desenvolvimento de um modelo de machine learning para previsão de falhas em equipamentos**, elaborado pela aluna **Isabela Medeiros Chaves**, matrícula 20170009580, sob a orientação do Prof. **Francisco Fernando Roberto Pereira**. A banca foi composta pelos avaliadores: Prof. **Francisco Fernando Roberto Pereira** (orientador), Profa. **Janaina Moreira de Meneses** (avaliadora), e Prof. **Marcelo José Alves de Santana** (avaliador). Após a exposição oral do trabalho, a candidata foi arguida pelos componentes da banca que em seguida reuniram-se e deliberaram pela sua aprovação, atribuindo-lhe a média 9,5, julgando-a apta( ) / inapta( ) à conclusão do curso de Engenharia Mecânica. Para constar, redigi a presente ata aprovada por todos os presentes, que vai assinada pelos membros da banca.

Orientador: Prof. Francisco Fernando Roberto Pereira Nota: 9,5

Assinatura gov.br Documento assinado digitalmente FRANCISCO FERNANDO ROBERTO PEREIRA Data: 26/03/2025 17:47:43-0300 Verifique em https://validar.iti.gov.br

Avaliadora Interna: Profa. Janaina Moreira de Meneses Nota: 9,5

Assinatura gov.br Documento assinado digitalmente JANAINA MOREIRA DE MENESES Data: 07/04/2025 00:20:20-0300 Verifique em https://validar.iti.gov.br

Avaliador Externo: Prof. Marcelo José Alves de Santana Nota: 9,5

Assinatura gov.br Documento assinado digitalmente MARCELO JOSE ALVES DE SANTANA Data: 08/04/2025 10:45:31-0300 Verifique em https://validar.iti.gov.br

Recife, 26 de março de 2025.

Prof. Marcus Costa de Araújo
Coordenador de Trabalho de Conclusão de curso - TCC
Curso de Graduação em Engenharia Mecânica – CTG/EEP-UFPE

Ao meu avô, Olímpio Florentino de Medeiros.

Obrigada por me influenciar a ser uma engenheira quando eu tinha 14 anos.

Nada disso seria possível sem aquele primeiro pontapé.

Te amo, voinho! Onde quer que o senhor esteja.

AGRADECIMENTOS

À minha mãe, Andréa, que cuida de mim há vinte e sete anos com uma força implacável. Às minhas duas irmãs Gabriela e Daniela, sem o companheirismo de vocês a vida seria completamente sem graça.

Às minhas tias Rita de Kássia e Ana Maria, pelas conversas, conselhos e risadas que deixaram toda essa caminhada muito mais leve.

Às minhas avós, Maria José e Eny, pelo carinho, cuidado e paciência. Aos meus avôs, Luciano e Olímpio, por sempre me ensinarem a buscar mais conhecimento.

Aos meus amigos, Rebeca, Soraia, Suelen, Sofia, Carol, Iasmim, Yumi, Mariana, Juliana, Júlia, Eryca, Emanuely, Eduardo, Jonathan, Jonas e colegas de curso pelo apoio em todos os momentos.

Ao meu orientador, Francisco Fernando, cujo apoio incansável foi imprescindível para a realização deste trabalho.

A Marina, que faz todas as minhas conquistas fazerem sentido.

RESUMO

A manutenção preditiva desempenha um papel essencial na Indústria 4.0, possibilitando a antecipação de falhas em equipamentos por meio de análise de dados. Neste contexto, este estudo teve como objetivo desenvolver um modelo de *Machine Learning* para a classificação da ocorrência de falhas em ativos industriais. Para isso, foi utilizado um conjunto de dados sintético, ao qual foram aplicadas técnicas de pré-processamento, modelagem e avaliação de desempenho de seis algoritmos de classificação: Regressão Logística, *Random Forest*, SVM, KNN, Árvore de Decisão e *Gradient Boosting*. Os resultados indicaram que o modelo de Árvore de Decisão ofereceu um melhor equilíbrio entre acurácia e interpretabilidade, mostrando-se o mais adequado para aplicações de manutenção preditiva, apesar de maior precisão dos modelos de *SVM*, *Random Forest* e *Gradient Boosting*. Como continuidade deste estudo, sugere-se a validação do modelo com dados reais e a investigação por técnicas mais avançadas, como *Deep Learning*. A aplicação de *Machine Learning* na manutenção preditiva tem o potencial de aprimorar significativamente a detecção antecipada de falhas, reduzindo custos operacionais e aumentando a confiabilidade dos equipamentos. Assim, este trabalho contribui para a adoção de soluções inteligentes na manutenção industrial, reforçando a importância da digitalização e do uso da inteligência artificial na otimização dos processos produtivos.

Palavras-chave: manutenção preditiva; machine learning; classificação de falhas.

ABSTRACT

Predictive maintenance plays a crucial role in Industry 4.0, enabling the anticipation of equipment failures through data analysis. This study aimed to develop a Machine Learning model for classifying the occurrence of failures in industrial assets. A synthetic dataset was used, upon which preprocessing techniques, modeling, and performance evaluation of six classification algorithms were applied: Logistic Regression, Random Forest, SVM, KNN, Decision Tree, and Gradient Boosting. The results indicated that the Decision Tree model offered the best balance between accuracy and interpretability, making it the most suitable choice for predictive maintenance applications, despite the higher accuracy achieved by the SVM, Random Forest and Gradient Boosting models. As a continuation of this work, it is recommended to validate the model with real-world data and explore more advanced techniques, such as Deep Learning. The application of machine learning in predictive maintenance has the potential to significantly enhance early failure detection, reducing operational costs and increasing equipment reliability. In this way, this study contributes to the adoption of intelligent solutions in industrial maintenance, emphasizing the importance of digitalization and the use of Artificial Intelligence in optimizing production processes.

Keywords: predictive maintenance; machine learning; failure classification.

SUMÁRIO

1 INTRODUÇÃO

A Primeira Revolução Industrial, iniciada no final do século XVIII, marcou a transição dos métodos de produção artesanais para processos mecanizados, dando origem à Indústria 1.0. Atualmente, com a Indústria 4.0, observa-se a integração de tecnologias avançadas, como Inteligência Artificial, Internet das Coisas e *Big Data*, que permitem a otimização de processos e a tomada de decisões baseada em dados (Passos, 2021). Nesse contexto, a evolução da indústria também impulsionou avanços significativos na manutenção industrial, resultando na Manutenção 4.0. Essa fase caracteriza-se pela aplicação de tecnologias digitais para o monitoramento de equipamentos em tempo real e análises preditivas, gerando maior eficiência operacional, reduzindo custos e minimizando falhas inesperadas (Farias; Quelhas, 2021).

A manutenção preditiva destaca-se nesse cenário como uma abordagem estratégica para a gestão de equipamentos industriais. Baseada na capacidade de prever falhas antes que ocorram, essa metodologia utiliza sensores para coletar dados em tempo real, monitorando variáveis como temperatura, vibração, pressão e desempenho. A identificação precoce de anomalias possibilita intervenções precisas no momento ideal, reduzindo o tempo de inatividade dos ativos, os custos com reparos emergenciais e os riscos de falhas catastróficas (Achouch *et al.*, 2022).

Com o avanço da Manutenção 4.0, a manutenção preditiva consolidou-se como uma tendência global em diversos setores industriais. Empresas que adotam essa abordagem não apenas aumentam a eficiência operacional, mas também promovem a sustentabilidade ao reduzir o desperdício de recursos e otimizar o consumo energético. Dessa forma, a implementação da manutenção preditiva tornou-se um diferencial competitivo essencial, permitindo que as organizações enfrentem desafios como alta concorrência, demandas crescentes por qualidade e maior pressão social pela adoção de práticas sustentáveis, ao mesmo tempo em que reduzem custos operacionais e garantem maior confiabilidade em suas operações (Borlido, 2023).

Nesse cenário, este trabalho tem como objetivo o desenvolvimento de um modelo de *Machine Learning* para a previsão de falhas em equipamentos industriais. Para isso, foi utilizado um conjunto de dados sintético, no qual foram realizadas etapas de análise exploratória, pré-processamento e modelagem utilizando seis algoritmos de classificação: Regressão Logística, *Random Forest*, *Support Vector Machine (SVM)*,

K-Nearest Neighbors (KNN), Árvore de Decisão e *Gradient Boosting*. A partir disso, os modelos foram avaliados de acordo com as seguintes métricas: acurácia, precisão, *recall*, *F1-Score*, Curva ROC e matriz de confusão.

Nesta análise, o modelo de classificação que obteve o melhor *F1-score* foi o *Gradient Boosting*. No entanto, a escolha do modelo ideal para a aplicação depende de um compromisso entre acurácia e interpretabilidade. Logo, o modelo de classificação Árvore de Decisão, o qual obteve um desempenho sutilmente menor na métrica *F1-score* que o *Gradient Boosting*, é considerado uma escolha mais acertada para essa aplicação, visto que, por ser um modelo mais simples, possui também uma maior interpretabilidade e oferece uma maior transparência nas previsões.

Dessa forma, este estudo contribui tanto para o aprimoramento acadêmico daqueles interessados na aplicação de *Machine Learning* na manutenção preditiva, quanto para o avanço de soluções voltadas à confiabilidade operacional em ambientes industriais. Ao apresentar uma abordagem estruturada para a predição de falhas em equipamentos, a pesquisa reforça a importância do uso de técnicas de *Machine Learning* na otimização da manutenção, reduzindo custos e aumentando a eficiência dos processos. Além disso, os resultados obtidos podem servir como referência para futuras investigações, incentivando a aplicação dessas metodologias em diferentes setores industriais e fomentando o desenvolvimento de estratégias mais inteligentes e sustentáveis para a gestão de ativos.

2 OBJETIVOS

Com o uso crescente de tecnologias digitais, tornou-se possível aplicar métodos de *Machine Learning* para prever falhas em equipamentos. Essa abordagem contribui para melhorar a eficiência e reduzir custos com manutenção. Este trabalho busca desenvolver um modelo de *Machine Learning* com esse objetivo. Abaixo, são apresentados o objetivo geral e os objetivos específicos do projeto.

2.1 OBJETIVO GERAL E ESPECÍFICOS

O objetivo deste trabalho é desenvolver um modelo de *Machine Learning* de classificação para previsão de falhas em equipamentos. Para alcançar esse propósito, foram definidos os seguintes objetivos específicos:

a) Realizar uma análise exploratória dos dados;

b) Limpar os dados utilizando técnicas estatísticas;

c) Pré-processar os dados;

d) Desenvolver o modelo;

e) Treinar o modelo;

f) Definir a precisão por meio de testes com novos dados.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão abordados temas relacionados à predição de falhas, ao conceito de *Machine Learning* e aos principais modelos de classificação utilizados nesse tipo de problema.

3.1 PREDIÇÃO DE FALHA

A manutenção é um conjunto de ações destinadas a garantir o bom funcionamento de máquinas e equipamentos, prevenindo falhas e assegurando a conservação dos componentes. Sua importância cresceu com a Revolução Industrial e a intensificação da concorrência entre as empresas, que a tornaram uma estratégia financeira fundamental. De acordo com a Associação Brasileira de Manutenção e Gestão de Ativos (ABRAMAN), cerca de 5% do faturamento anual bruto das empresas brasileiras é destinado ao custo total anual de manutenção (ABRAMAN, 2021).

De acordo com Fogliatto e Ribeiro (2011), é possível definir três tipos principais de manutenção:

1. Corretiva: realizada após a falha do equipamento, gerando custos mais altos e maior tempo de intervenção;

2. Preventiva: realizada em intervalos regulares, mesmo que o equipamento esteja funcionando bem, o que pode levar a substituições desnecessárias de componentes;

3. Preditiva: monitoramento contínuo das condições dos equipamentos, permitindo a substituição dos componentes no momento ideal e evitando falhas.

A Manutenção Preditiva, que surgiu nos anos 70 como uma estratégia eficiente para melhorar a produtividade, o lucro e a qualidade, é a mais eficaz, pois permite acompanhar a evolução das falhas e garantir a operação contínua e eficiente dos equipamentos. Ela permite avaliar a vida útil dos componentes e otimizar seu aproveitamento. Seus principais objetivos são: identificar a necessidade de manutenção, evitar desmontagens desnecessárias, aumentar a disponibilidade dos equipamentos, reduzir intervenções corretivas, prevenir danos maiores, maximizar a vida útil do equipamento, aumentar sua confiabilidade e auxiliar na priorização das manutenções durante paradas de produção (Moubray, 1999).

A técnica utiliza instrumentos para monitorar sinais de possíveis falhas, como vibrações, temperaturas elevadas, ruídos e desgastes. A coleta de dados é realizada periodicamente por técnicos treinados, seguindo um plano de manutenção que leva em consideração a importância do equipamento e o número de pontos a serem monitorados. Quando uma irregularidade é detectada, as informações são enviadas para planejar a intervenção no momento adequado, levando em conta a gravidade do defeito e o seu impacto na produção (Silva; Nascimento, 2020).

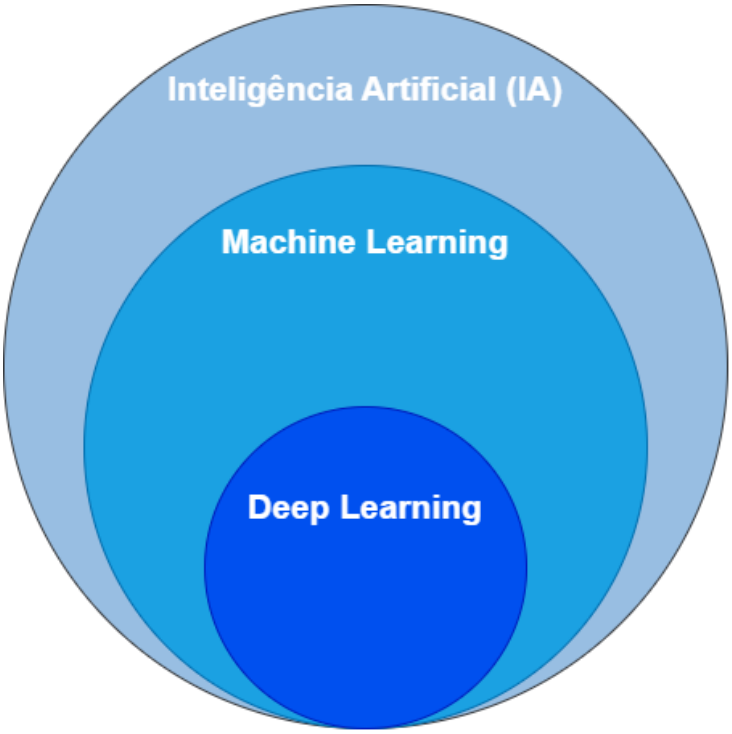
A influência da Indústria 4.0 na Manutenção Preditiva está relacionada ao uso de tecnologias como *Big Data*, que possibilita o armazenamento de grandes volumes de dados e informações digitais, além de oferecer alta velocidade e acesso rápido a partir de diversos locais. Outra tecnologia que trouxe benefícios significativos para a manutenção preditiva é a Inteligência Artificial, pois permite que *softwares* realizem diagnósticos autônomos e auxiliem os técnicos nas análises, direcionando de maneira mais rápida e precisa a identificação de falhas nos equipamentos (Souza *et al.*, 2022).

3.2 FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL

A Inteligência Artificial (IA) busca simular o comportamento humano por meio de máquinas (Escovedo; Koshiyama, 2020). No contexto da IA, o *Machine Learning* identifica padrões e constrói modelos matemáticos para automatizar tarefas complexas, solucionando problemas de predição como regressão e classificação (Oscar, 2023).

O *Deep Learning* aprofunda essa abordagem utilizando redes neurais com múltiplas camadas, permitindo aplicações como reconhecimento de fala e classificação de imagens (Howard; Gugger, 2020). Já a *Data Science* envolve a coleta e a análise sistematizada de grandes volumes de dados, frequentemente empregando *Machine Learning* para previsões (Escovedo; Koshiyama, 2020). A figura 1 ilustra a relação entre IA, *Machine Learning* e *Deep Learning*.

Figura 1 – Estrutura da Inteligência Artificial



Fonte: Adaptado de Escovedo e Koshiyama (2020)

O *Machine Learning* se divide em aprendizado supervisionado, que utiliza dados rotulados para prever categorias ou valores, e aprendizado não supervisionado, que identifica padrões em dados não rotulados. Modelos como o *ChatGPT* são treinados dessa forma, aprendendo contextos linguísticos sem rótulos explícitos (Izbicki; Santos, 2022).

Os principais problemas abordados em *Data Science* incluem Classificação, Regressão, Agrupamento e Associação. Enquanto Classificação e Regressão são preditivos, diferenciando-se pelo tipo de variável-alvo, Agrupamento e Associação analisam relações e padrões nos dados (Escovedo; Koshiyama, 2020).

3.3 ALGORITMOS DE CLASSIFICAÇÃO

O problema de classificação é um dos mais frequentemente enfrentados pelos cientistas de dados, e pode ser considerada a forma mais importante de previsão. Essa forma de predição tem como principal objetivo a classificação de dados, seja de forma binária ou múltipla. Na classificação binária, é avaliado se um elemento pertence ou não a uma única classe, enquanto na classificação múltipla considera-se a probabilidade

de pertencimento a diversas classes. (Bruce; Bruce; Gedeck, 2020).

Dado que este trabalho tem como foco a previsão de falhas em equipamentos, um problema típico de classificação, é importante destacar que, durante o treinamento dos modelos, devem ser evitados problemas como o *overfitting* e o *underfitting*, que podem comprometer a capacidade de generalização do modelo. O *overfitting* ocorre quando o modelo se ajusta excessivamente aos dados de treino, capturando ruídos e detalhes irrelevantes, o que prejudica seu desempenho em novos dados. Por outro lado, o *underfitting* acontece quando o modelo é muito simples e não consegue capturar padrões essenciais dos dados, resultando em baixo desempenho tanto no treino quanto na validação (James *et al.*, 2023). O controle desses problemas é crítico para o sucesso do modelo e eles devem ser monitorados constantemente, especialmente em contextos de previsão de falhas, como no caso deste trabalho.

3.3.1 Regressão Logística

A regressão logística é um algoritmo de aprendizado supervisionado amplamente utilizado para resolver problemas de classificação, como o abordado neste trabalho. Esse método calcula a probabilidade de um evento pertencer a uma das classes previamente definidas, baseando-se na função logística, ou sigmoide, que transforma um valor real em um intervalo entre 0 e 1. A equação 1 define essa função (James *et al.*, 2023).

$$p(x) = \frac{e^{(\beta_0+\beta_1\times x_1+\beta_2\times x_2+...+\beta_n\times x_n)}}{1+e^{(\beta_0+\beta_1\times x_1+\beta_2\times x_2+...+\beta_n\times x_n)}} \tag{1}$$

Os coeficientes $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ são estimados com base nos dados de treinamento do modelo. A interpretação desses coeficientes é intuitiva, permitindo observar o impacto de cada variável preditora na probabilidade de ocorrência de um evento.

Outra forma de interpretar a regressão logística é por meio da função *odds*, que expressa a probabilidade relativa de um evento ocorrer em comparação com a probabilidade de ele não ocorrer, conforme a equação 2 (James *et al.*, 2023).

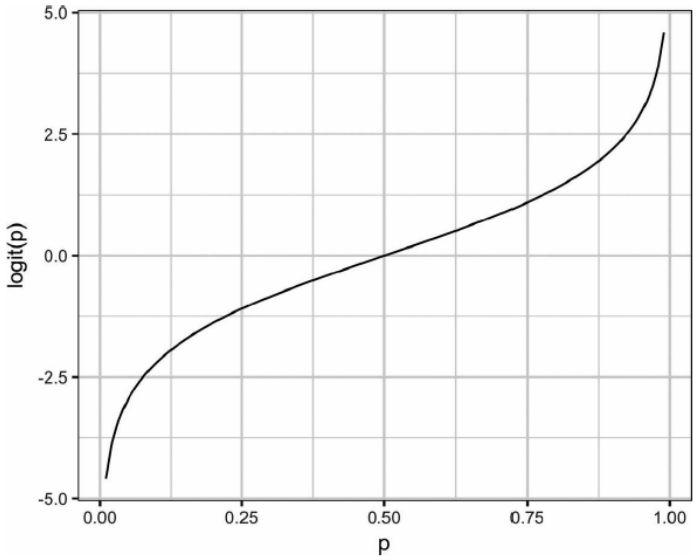
$$\textit{Odds} = \frac{p(x)}{1-p(x)} \tag{2}$$

Essa função pode ser reescrita em sua forma logarítmica, conhecida como logito (*log-odds*), conforme a equação 3 (James *et al.*, 2023).

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \ldots + \beta_n \cdot x_n \quad (3)$$

Essa transformação permite utilizar um modelo linear para prever probabilidades, classificando registros a partir de um limiar definido. Esse comportamento é ilustrado na figura 2, que demonstra a relação entre a função logito e a probabilidade de um evento ocorrer.

Figura 2 – Gráfico da função logito (*logit*) versus a probabilidade



Fonte: Bruce, Bruce e Gedeck (2020)

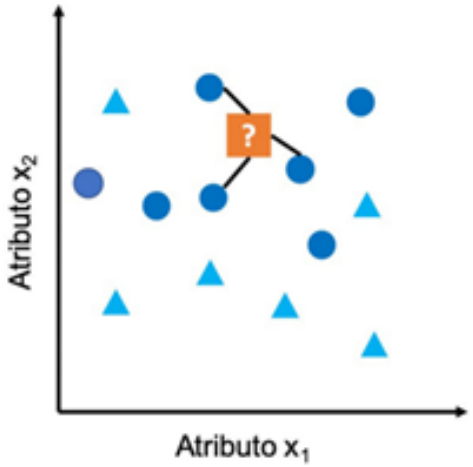
A Regressão Logística é um algoritmo de fácil interpretação, sendo amplamente utilizado em classificação binária. No entanto, apresenta limitações quando as relações entre variáveis independentes e dependentes são altamente não lineares, quando o conjunto de dados possui classes desbalanceadas ou quando o tamanho da amostra é pequeno (Fernando, 2024).

3.3.2 k-Nearest Neighbours (KNN)

O algoritmo *k-Nearest Neighbours (KNN)*, ou k-Vizinhos Mais Próximos, é um método simples e eficaz para classificação e regressão, baseado na premissa de que registros semelhantes encontram-se próximos no espaço de características. O objetivo é classificar um novo registro considerando os k vizinhos mais próximos no conjunto de treinamento, atribuindo-lhe a classe predominante entre eles (Escovedo;

Koshiyama, 2020). A figura 3 ilustra o funcionamento do algoritmo, onde os registros são representados como pontos em um espaço dimensional.

Figura 3 – Exemplificação de um modelo KNN



Fonte: Escovedo e Koshiyama (2020)

O funcionamento do KNN depende do cálculo de distâncias entre registros. A mais utilizada é a distância Euclidiana, conforme a equação 4, pois mede a distância entre dois vetores no espaço (James *et al.*, 2023).

$$d(x_i, y_i) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{4}$$

Para obter bons resultados, é essencial que os dados sejam previamente padronizados, evitando que variáveis de maior escala dominem o cálculo da distância (Bruce; Bruce; Gedeck, 2020). Valores de distância muito baixos podem tornar o modelo sensível a ruídos (*overfitting*), enquanto valores muito altos podem dificultar a captura de padrões locais (*underfitting*) (Shmueli *et al.*, 2023).

O KNN não requer uma fase explícita de treinamento, tornando-se flexível e de fácil implementação. No entanto, apresenta desvantagens como a necessidade de calcular distâncias para cada nova predição, tornando-o computacionalmente custoso em grandes volumes de dados (Fernando, 2024).

3.3.3 Support Vector Machine (SVM)

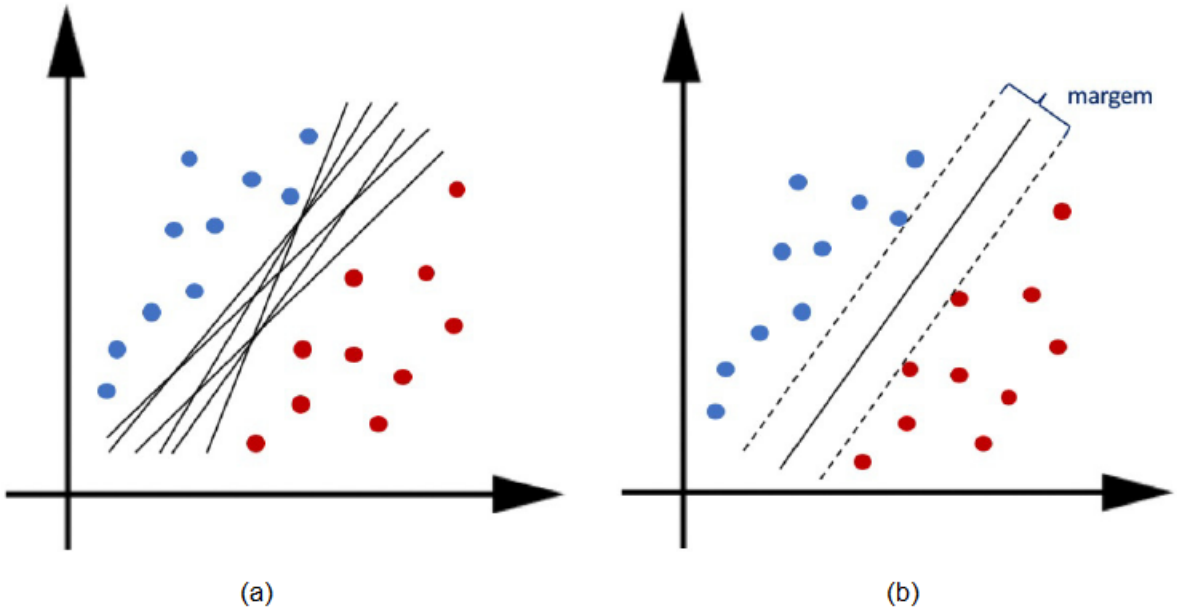
O algoritmo *Support Vector Machine (SVM)* é um método eficiente para classificação, baseado na busca de um hiperplano ótimo que melhor separa as classes de um conjunto de dados. A principal ideia do SVM é maximizar a margem entre os pontos de classes diferentes mais próximos do hiperplano, chamados de *vetores de suporte*. Quanto maior a margem, maior a capacidade de generalização do modelo (James *et al.*, 2023).

A equação de um hiperplano em um espaço de p dimensões é dada pela equação 5.

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p = 0 \tag{5}$$

A figura 4 ilustra os possíveis hiperplanos separadores e a margem entre as duas classes. A escolha da margem ótima é crucial para o bom desempenho do SVM.

Figura 4 – (a) Possíveis hiperplanos separadores (b) Margem entre duas classes



Fonte: Escovedo e Koshiyama (2020)

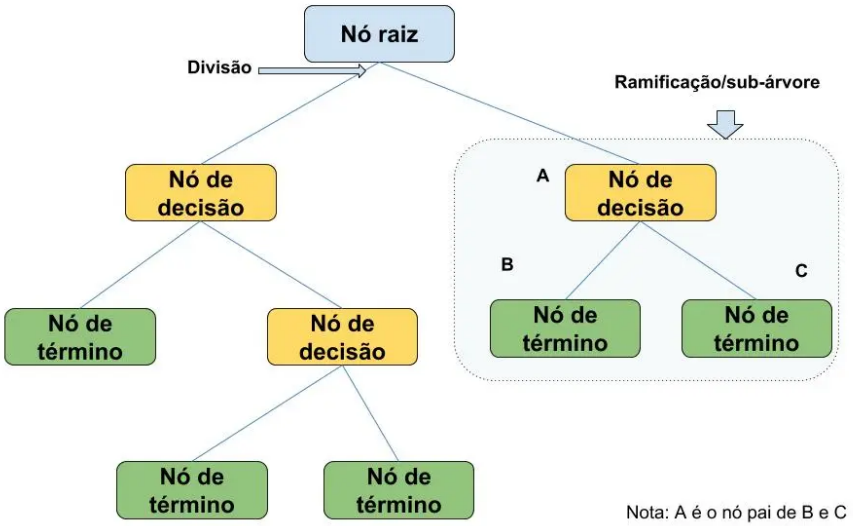
No entanto, em muitos casos, os dados não são linearmente separáveis. Para resolver esse problema, o SVM utiliza o conceito de *kernels*, que transforma os dados em um espaço de maior dimensão, onde a separação se torna possível. Esse método é conhecido como *kernel trick* (Escovedo; Koshiyama, 2020).

O SVM apresenta bons resultados em problemas de alta dimensionalidade, sendo robusto contra *overfitting*. No entanto, pode ser computacionalmente custoso para grandes bases de dados, e a escolha do *kernel* adequado é crucial para um bom desempenho (Fernando, 2024).

3.3.4 Árvore de Decisão

O algoritmo de Árvore de Decisão para classificação organiza a tomada de decisão de forma hierárquica. O nó raiz representa o ponto inicial da árvore e contém todo o conjunto de dados. A partir dele, ocorre a divisão, onde o conjunto é separado com base em um critério que minimiza a impureza dos dados, como o Índice de Gini. Cada nó de decisão realiza um teste sobre um atributo específico, direcionando os dados para diferentes caminhos. Esse processo gera a ramificação, onde cada ramo representa um possível resultado do teste. O processo continua até que os nós se tornem folhas, as quais representam as classes finais previstas pelo modelo (Demirović *et al.*, 2022). A figura 5 mostra o esquema visual do funcionamento desse algoritmo.

Figura 5 – Esquematização de uma Árvore de Decisão



Fonte: Picouto (2023)

O Índice de Gini mede a impureza de um conjunto de dados, baseado na probabilidade de que uma amostra ser classificada incorretamente se for rotulada aleatoriamente de acordo com a distribuição das classes. Sua fórmula matemática é definida pela equação 6 (Shmueli *et al.*, 2023),

$$Gini(S) = 1 - \sum_{i=1}^{C} p_i^2 \tag{6}$$

onde p_i representa a proporção de elementos da classe i dentro do conjunto S e C é o número total de classes. O Índice de Gini varia de 0 a 0,5 para problemas binários, sendo que quanto maior o valor, maior a impureza do nó.

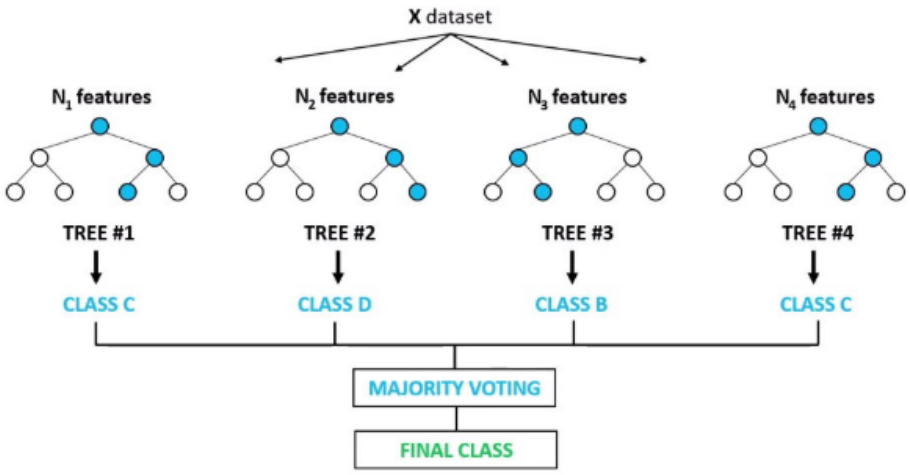
Como o Índice de Gini é utilizado para determinar o melhor critério de divisão dos dados em um nó, o algoritmo avalia todas as possíveis divisões e escolhe aquela que resulta na maior redução de impureza, conhecida como ganho de Gini.

Um dos principais desafios para o algoritmo Árvore de Decisão é o *overfitting*, que ocorre quando a árvore cresce excessivamente, aprendendo padrões específicos dos dados de treinamento que não generalizam bem para novos dados. Para evitar esse problema, pode-se aplicar técnicas como a poda (*pruning*), que remove divisões irrelevantes ou ajusta a profundidade da árvore para melhorar sua capacidade preditiva (Bruce; Bruce; Gedeck, 2020).

3.3.5 Random Forest

A *Random Forest* (Floresta Aleatória) expande o conceito de Árvores de Decisão ao combinar múltiplas árvores para formar um modelo mais robusto. Ele constrói diversas árvores de forma independente, utilizando subconjuntos aleatórios dos dados e dos atributos disponíveis, o que é exemplificado na figura 6. Cada árvore gera uma predição, e a decisão final é tomada por meio de um mecanismo de votação majoritária.

Figura 6 – Visualização do Random Forest



Fonte: David (2020)

Embora a *Random Forest* ofereça um desempenho sólido e seja menos sensível a variações de hiperparâmetros que a Árvore de Decisão, ela pode ser computacionalmente custosa quando o número de árvores é muito grande. Além disso, por ser uma combinação de múltiplos modelos, perde-se a interpretabilidade característica das Árvores de Decisão individuais (James *et al.*, 2023).

3.3.6 Gradient Boosting

O *Gradient Boosting* é um método que ajusta sequencialmente modelos simples, como Árvores de Decisão de pequena profundidade, para minimizar o erro preditivo. Em cada iteração, um novo modelo é treinado para reduzir o erro residual do anterior (Géron, 2022).

De acordo com James *et al.* (2023), o algoritmo pode ser descrito formalmente através do seguinte passo a passo: inicialmente, define-se o modelo como zero e os resíduos como os valores originais das saídas. A equação 7 define o conjunto de treinamento,

$$\hat{f}(x) = 0, \quad r_i = y_i \tag{7}$$

onde:

a) $\hat{f}(x)$ representa a função preditora do modelo;

b) r_i são os resíduos iniciais;

c) y_i são os valores reais das variáveis alvo.

Para cada iteração $b = 1, 2, \ldots, B$, realizam-se os seguintes passos:

1. Ajuste da árvore: ajusta-se uma árvore de decisão $\hat{f}_b(x)$ com d divisões (resultando em $d + 1$ nós terminais) aos dados de treinamento (X, r), em que $\hat{f}_b(x)$ é a nova função preditora baseada na árvore de decisão ajustada;

2. Atualização do modelo: o modelo é atualizado adicionando-se uma versão reduzida da nova árvore,

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}_b(x) \tag{8}$$

onde:

a) λ é a taxa de aprendizado que controla a contribuição da nova árvore ao modelo;

b) $\hat{f}_b(x)$ é a nova árvore ajustada.

3. Atualização dos resíduos: os resíduos são atualizados, subtraindo-se a nova predição:

$$r_i \leftarrow r_i - \lambda \hat{f}_b(x_i) \tag{9}$$

onde:

a) r_i representa os resíduos atualizados;

b) $\lambda \hat{f}_b(x_i)$ é a contribuição da nova árvore na predição do modelo.

Após B iterações, o modelo final é definido como:

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}_b(x) \tag{10}$$

onde $\sum_{b=1}^{B} \lambda \hat{f}_b(x)$ representa a soma ponderada das B árvores ajustadas.

Essa abordagem permite que o modelo aprenda iterativamente, reduzindo os resíduos a cada passo e melhorando a precisão das previsões.

O *Gradient Boosting* é amplamente utilizado em aplicações como modelagem financeira, previsão de rotatividade de clientes e tarefas de classificação, devido à sua alta precisão e flexibilidade. Embora seja altamente eficaz, ele pode ser computacionalmente dispendioso, especialmente ao lidar com grandes conjuntos de dados, e pode ser propenso a *overfitting* se não for devidamente regularizado (Ali *et al.*, 2025).

3.4 MÉTRICAS DE AVALIAÇÃO

A avaliação precisa de modelos de classificação é essencial para garantir a eficácia de sistemas de *Machine Learning*. Este capítulo aborda as principais métricas utilizadas para avaliar o desempenho de classificadores: matriz de confusão, acurácia, precisão, *recall*, *F1-score* e a curva ROC.

3.4.1 Matriz de Confusão

A Matriz de Confusão é uma ferramenta que resume o desempenho de um classificador, comparando as previsões do modelo com os valores reais. Para um problema de classificação binária, a estrutura da matriz é demonstrada na figura 7 (Escovedo; Koshiyama, 2020).

Figura 7 – Matriz de Confusão: problemas binários

Classes	Predita C1	Predita C2
Verdadeira C1	Verdadeiros Positivos	Falsos Negativos
Verdadeira C2	Falsos Positivos	Verdadeiros Negativos

Fonte: Escovedo e Koshiyama (2020)

Em que:

a) Verdadeiro Positivo (VP): número de casos positivos corretamente previstos;

b) Falso Positivo (FP): número de casos negativos incorretamente previstos como positivos;

c) Falso Negativo (FN): número de casos positivos incorretamente previstos como negativos;

d) Verdadeiro Negativo (VN): número de casos negativos corretamente previstos.

Dessa forma, é possível visualizar o número de acertos e erros cometidos pelo modelo, separando os diferentes tipos de erros.

3.4.2 Tabela de Métricas de Avaliação

A tabela 1 mostra as principais métricas comumente utilizadas na avaliação de classificadores.

Tabela 1 – Métricas de Avaliação de Modelos

Métrica	Fórmula	O que a métrica avalia
Acurácia	$\frac{VP+VN}{VP+FP+FN+VN}$	A proporção de previsões corretas em relação ao total de casos.
Precisão	$\frac{VP}{VP+FP}$	A confiabilidade das previsões positivas.
Recall	$\frac{VP}{VP+FN}$	A capacidade do modelo de identificar corretamente os casos positivos.
F1-Score	$2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}$	O equilíbrio entre precisão e *recall*, útil para classes desbalanceadas.

Fonte: Autoria própria

Essas métricas fornecem uma visão abrangente do desempenho do modelo, permitindo a escolha da abordagem mais adequada de acordo com as características do problema.

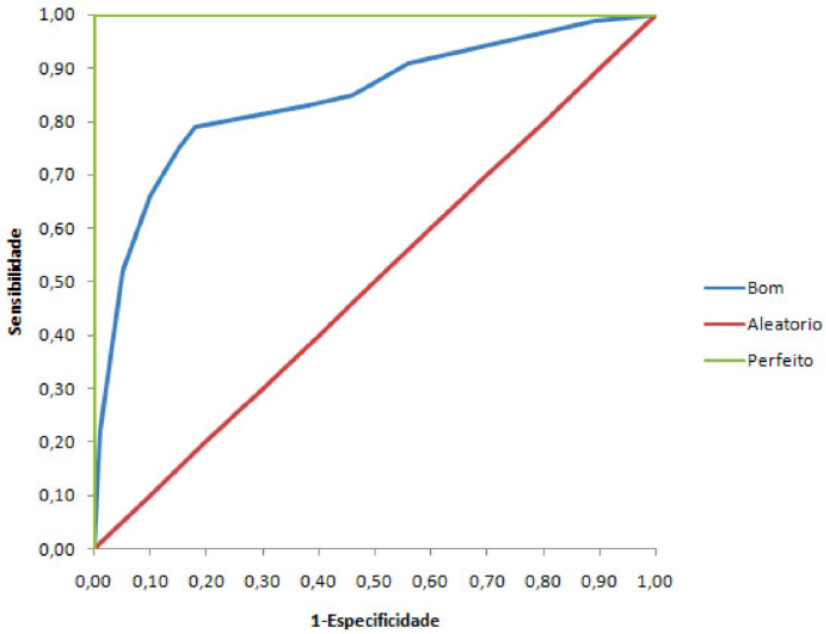
3.4.3 Curva ROC e AUC

A Curva ROC (*Receiver Operating Characteristic*) é um gráfico que representa a relação entre o *recall* e a Taxa de Falsos Positivos (TFP), que é calculada pela equação 11, para diferentes limiares de decisão (James *et al.*, 2023).

$$\text{TFP} = \frac{FP}{FP + TN} \tag{11}$$

A área sob a Curva ROC, conhecida como AUC (*Area Under the Curve*), quantifica a capacidade do classificador de distinguir entre as classes. Um AUC de 0,5 indica desempenho aleatório, enquanto um AUC de 1,0 representa uma classificação perfeita. A imagem 8 mostra exemplos de Curva ROC (Bruce; Bruce; Gedeck, 2020).

Figura 8 – Exemplos de Curvas ROC



Fonte: Escovedo e Koshiyama (2020)

A Curva ROC é muito importante quando se quer entender a relação entre detectar mais falhas, falsos negativos, e evitar o aumento de falsos positivos.

3.4.4 Correção de Underfitting e Overfitting

O *underfitting* ocorre quando um modelo é muito simples para capturar os padrões dos dados, resultando em um desempenho ruim tanto no conjunto de treinamento quanto no de teste. O *overfitting*, por outro lado, acontece quando o modelo se ajusta excessivamente aos dados de treinamento e tem baixo desempenho em novos dados. A forma mais robusta de corrigir *underfitting* e *overfitting* é utilizando a validação cruzada (*cross-validation*). Esse método divide os dados em k subconjuntos, treinando e validando o modelo k vezes, alternando o subconjunto de validação a cada iteração. Isso permite uma avaliação mais confiável do desempenho do algoritmo, já que cada ponto de dados é utilizado tanto para treinamento quanto para validação. Com isso, pode-se verificar o comportamento do modelo em diferentes subconjuntos de dados e prevenir *underfitting* e *overfitting* (Bishop, 2021).

A ocorrência de *underfitting* é indicada quando o desempenho do modelo é consistentemente baixo em todas as iterações da validação cruzada, tanto nos dados de treinamento quanto nos de validação. Já o *overfitting* é identificado quando o modelo

apresenta um bom desempenho nos dados de treinamento, mas seu desempenho nos dados de validação é significativamente inferior. Em ambos os casos, a validação cruzada fornece uma análise mais detalhada e confiável, permitindo ajustes adequados no modelo para melhorar sua generalização.

4 METODOLOGIA

Esse trabalho trata-se de uma pesquisa classificada como aplicada quanto à finalidade, descritiva quanto aos objetivos, qualitativa e quantitativa quanto à sua abordagem e, quanto aos procedimentos, pesquisa bibliográfica e estudo de caso. O método científico utilizado será o hipotético-dedutivo. As próximas seções abordarão as atividades e os procedimentos para a geração, a coleta e a organização dos dados da pesquisa.

4.1 PROCEDIMENTOS EXPERIMENTAIS

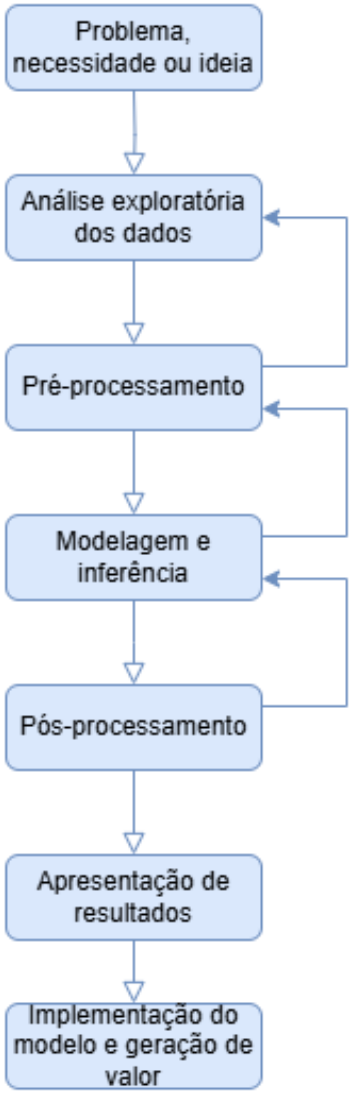
Um projeto de *Data Science* surge da necessidade de resolver um problema específico, sendo essencial que seus objetivos estejam claramente definidos.

O primeiro passo é a análise exploratória dos dados, que ajuda a entender suas características e identificar padrões ou problemas. Em seguida, realiza-se o pré-processamento, que envolve a preparação dos dados, incluindo a remoção de dados inconsistentes e o tratamento de valores ausentes.

Após isso, é feita a seleção e treinamento do modelo de *Machine Learning* mais adequado para prever falhas. Por fim, a avaliação do modelo é realizada, comparando seu desempenho com outras abordagens para validação e ajustes.

A seguir, a figura 9 ilustra a esquematização do fluxo de um projeto de *Data Science*, conforme abordado neste trabalho.

Figura 9 – Fluxograma de um projeto de Data Science



Fonte: Adaptado de Escovedo e Koshiyama (2020)

O fluxograma apresenta as principais etapas de um projeto de *Data Science*, evidenciando a necessidade de um processo estruturado. A análise exploratória fornece uma visão inicial sobre os dados, enquanto o pré-processamento garante a qualidade da informação utilizada. A modelagem e o pós-processamento permitem validar a abordagem adotada, garantindo que o modelo seja eficiente e adequado ao problema em questão.

4.1.1 Descrição dos Dados

O conjunto de dados utilizado neste trabalho é composto por informações relacionadas à manutenção preditiva de máquinas. No contexto deste trabalho, foi criado um

banco de dados, visto que não foi possível obter acesso a dados reais de indústrias devido ao sigilo das informações. Além disso, os bancos de dados disponíveis publicamente sobre o tema também consistiam em *datasets* artificiais. Para tornar o banco de dados criado mais próximo da realidade, foram adicionados ruídos durante a sua construção, simulando variações e imperfeições comumente encontradas em dados reais. As variáveis presentes no *dataset* são descritas na tabela 2, que apresenta uma visão geral das características e seus respectivos tipos de dados.

Tabela 2 – Descrição das Variáveis do Dataset

Nome da Variável	Tipo de Dados	Descrição
ID único	Inteiro	Identificador único para cada instância
ID do Produto	Categórico	Identificador do produto
Temperatura do Ar [℃]	Float	Temperatura do ar em graus Celsius
Temperatura do Processo [℃]	Float	Temperatura do processo em graus Celsius
Velocidade de Rotação [rpm]	Float	Velocidade de rotação em rotações por minuto (rpm)
Torque [Nm]	Float	Torque aplicado em Newton-metro (Nm)
Desgaste da Ferramenta [min]	Float	Tempo de desgaste da ferramenta em minutos
Falha da Máquina	Binário	Indica se houve falha na máquina (1) ou não (0)

Fonte: Autoria própria.

As variáveis listadas representam aspectos fundamentais para a análise preditiva de falhas em máquinas. Grandezas físicas como temperatura, torque e velocidade de rotação influenciam diretamente o desgaste da ferramenta e a ocorrência de falhas. A variável de saída, "Falha da Máquina", permite a formulação do problema como uma tarefa de classificação binária, sendo crucial para a modelagem e avaliação dos algoritmos de aprendizado de máquina.

4.1.2 Ferramentas Computacionais

Para a realização deste trabalho, foram utilizadas diversas ferramentas computacionais e bibliotecas da linguagem *Python*. A linguagem *Python* foi escolhida devido à sua vasta gama de bibliotecas especializadas em *Data Science* e *Machine Learning*. As principais bibliotecas utilizadas foram:

1. *Pandas*: para manipulação e análise de dados;
2. *NumPy*: para operações numéricas e manipulação de *arrays*;

3. *Seaborn* e *Matplotlib*: para visualização de dados, incluindo a geração de gráficos como histogramas, matrizes de correlação e distribuições dos dados;

4. *Scikit-learn* (*sklearn*): para implementação de modelos de classificação, pré-processamento de dados, validação cruzada e avaliação de modelos. Foram utilizados modelos como Regressão Logística, *Random Forest*, SVM, KNN, Árvore de Decisão e *Gradient Boosting*;

5. *Joblib*: Para salvar e carregar modelos treinados.

A geração de gráficos foi essencial para a análise exploratória dos dados. Foram criadas matrizes de correlação para entender as relações entre as variáveis, histogramas para visualizar a distribuição das variáveis numéricas e gráficos de distribuição para identificar possíveis *outliers*. Essas visualizações permitiram uma compreensão mais aprofundada do comportamento dos dados e auxiliaram na identificação de padrões e anomalias.

4.1.3 Pré-processamento

O pré-processamento dos dados foi uma etapa crucial para garantir a qualidade dos modelos de classificação. As seguintes etapas foram realizadas:

1. Limpeza de dados: foram identificados e tratados valores negativos nas colunas "Desgaste da Ferramenta [min]" e "Velocidade de Rotação [rpm]", substituindo-os pela média das respectivas colunas. Isso foi necessário porque valores negativos não fazem sentido nesse contexto;

2. Padronização: as variáveis numéricas foram padronizadas utilizando o *StandardScaler* do *Scikit-learn*, que transforma os dados para ter média zero e desvio padrão unitário. Isso é importante para modelos que são sensíveis à escala dos dados, como SVM e KNN;

3. Tratamento de valores nulos: valores nulos nas colunas numéricas foram preenchidos com a média, enquanto nas colunas categóricas, foram preenchidos com a moda (valor mais frequente);

4. Visualização de dados: foram gerados histogramas e matrizes de correlação para entender a distribuição e a relação entre as variáveis. A matriz de correlação foi útil para identificar possíveis multicolinearidades entre as variáveis preditoras.

4.1.4 Modelagem e Inferência

A modelagem foi realizada utilizando seis diferentes algoritmos de classificação: Regressão Logística, *Random Forest*, SVM, KNN, Árvore de Decisão e *Gradient Boosting*. Para cada modelo, foi realizada uma busca de hiperparâmetros utilizando *Grid Search* com validação cruzada de 5 *folds*. A validação cruzada foi aplicada apenas no conjunto de treino para garantir que o modelo generalize bem para dados não vistos, evitando *overfitting*.

O *dataset* foi inicialmente dividido em conjuntos de treino (80%) e teste (20%). Essa divisão foi feita de forma estratificada, garantindo que a proporção das classes fosse mantida em ambos os conjuntos. Hiperparâmetros são configurações dos modelos que não são aprendidas diretamente dos dados, mas que precisam ser definidas antes do treinamento. Eles controlam o comportamento do algoritmo e têm um impacto significativo no desempenho do modelo. Foram explorados diferentes hiperparâmetros para cada modelo, como é possível observar na tabela 3. O *Grid Search* foi utilizado para encontrar a combinação ótima de hiperparâmetros que maximiza o *F1-Score*.

Tabela 3 – Hiperparâmetros explorados para cada modelo

Modelo	Hiperparâmetros explorados
Regressão Logística	- C: valores testados: 0.1, 1, 10 (controla a regularização do modelo)
KNN	- Número de vizinhos: 3, 5, 7 (define quantos vizinhos considerar para a classificação)
SVM	- C: 0.1, 1, 10 (controla a penalização de erros) - *Kernel*: linear, RBF (define a função de transformação dos dados)
Árvore de Decisão	- Profundidade máxima: sem limite, 5, 10 (controla a profundidade da árvore)
Random Forest	- Número de árvores: 50, 100, 200 - Profundidade máxima das árvores: sem limite, 5, 10
Gradient Boosting	- Número de árvores: 50, 100, 200 - Taxa de aprendizado: 0.01, 0.1, 0.2 (controla a contribuição de cada árvore) - Profundidade máxima: 3, 5, 7 (define a profundidade máxima de cada árvore)

Fonte: Autoria própria.

A escolha criteriosa dos hiperparâmetros é fundamental para otimizar o desempenho dos modelos. O uso do *Grid Search* permitiu explorar sistematicamente diferentes configurações, garantindo que cada algoritmo fosse treinado com os melhores ajustes possíveis. Como métrica principal de avaliação, o *F1-Score* foi utilizado para medir o equilíbrio entre precisão e *recall*, assegurando o desempenho adequado dos modelos de classificação.

4.1.5 Pós-processamento

Após a modelagem, foram realizadas análises pós-processamento para avaliar o desempenho dos modelos:

1. Curva ROC: foram geradas Curvas ROC para todos os modelos, permitindo a comparação da capacidade de cada modelo em distinguir entre as classes. A área sob a curva (AUC) foi calculada para quantificar o desempenho;

2. Matriz de Confusão: as Matrizes de Confusão foram plotadas para cada modelo, mostrando a quantidade de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos. Essa matriz permite entender onde os modelos cometem mais erros;

3. Avaliação de métricas: para cada modelo, foram calculados a acurácia, a precisão, o *recall* e o *F1-Score*.

4.1.6 Interação com o Usuário

Para facilitar a interação com o usuário final, foi implementado um *loop* que permite ao usuário inserir novos dados e obter previsões de falhas de máquinas. O modelo treinado é carregado e utilizado para fazer previsões em tempo real, fornecendo uma resposta imediata ao usuário.

5 RESULTADOS

Neste capítulo, serão apresentados os resultados obtidos a partir da aplicação dos modelos de classificação e das análises realizadas. A estrutura segue a mesma lógica da metodologia, com foco na apresentação dos gráficos, métricas de desempenho e análises pós-processamento. Os resultados são divididos em subseções que refletem as etapas da metodologia: análise exploratória, pré-processamento, modelagem e inferência e interação com o usuário.

5.1 ANÁLISE EXPLORATÓRIA DOS DADOS

A análise exploratória dos dados foi realizada para entender a distribuição das variáveis, identificar possíveis problemas, como valores negativos ou *outliers*, e verificar as relações entre as variáveis. Esta etapa incluiu a verificação dos dados estatísticos, a visualização da distribuição dos atributos numéricos e a criação da matriz de correlação.

5.1.1 Visualização da Distribuição dos Atributos Numéricos

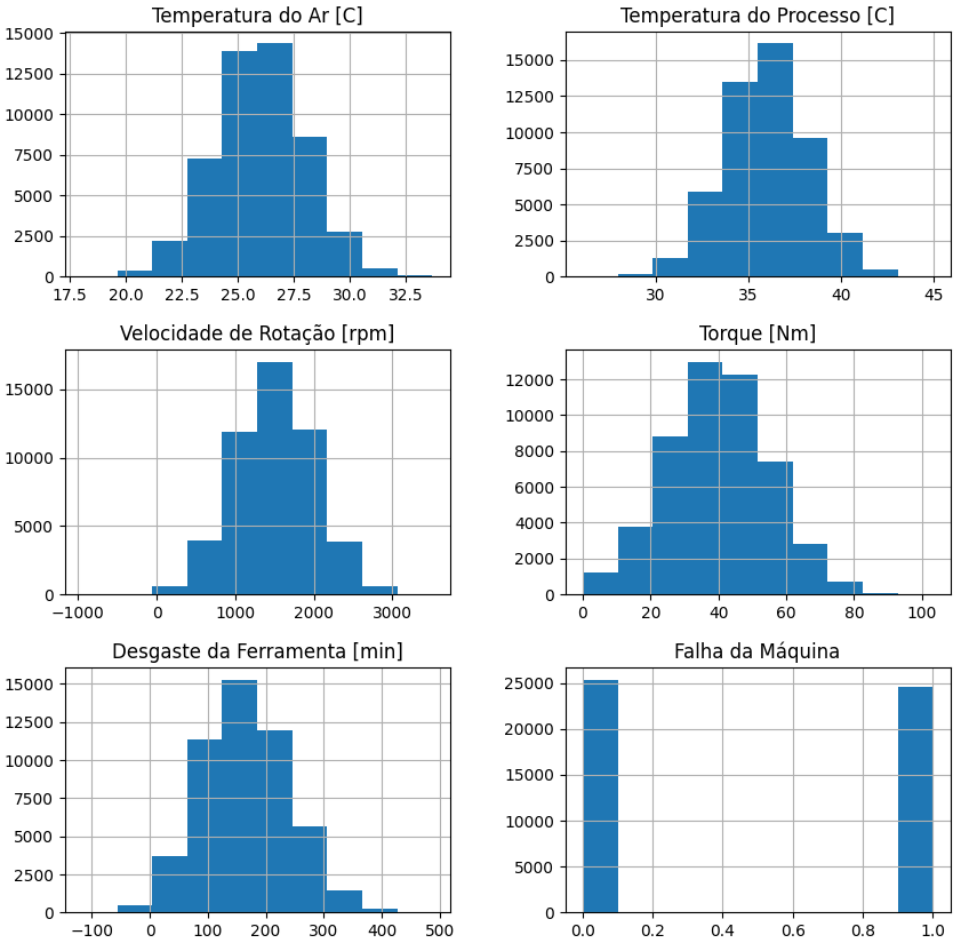
Para entender a distribuição das variáveis numéricas, foram gerados histogramas, como mostrado na figura 10. Observa-se que:

a) A "Temperatura do Ar [ºC]" e a "Temperatura do Processo [ºC]" apresentam distribuições próximas à normal, com médias em torno de 26℃ e 36℃, respectivamente;

b) A "Velocidade de Rotação [rpm]" e o "Torque [Nm]" apresentam distribuições normais, com valores concentrados em torno de 1500 rpm e 40 Nm, respectivamente;

c) O "Desgaste da Ferramenta [min]" apresenta uma distribuição normal, com valores variando entre -116 e 487 minutos, sendo necessário o tratamento dos valores negativos.

Figura 10 – Histogramas das variáveis numéricas



Fonte: Autoria própria

A partir desses gráficos de distribuição, foi possível observar que as variáveis "Velocidade de Rotação" e "Desgaste da Ferramenta" apresentam valores negativos. Esse comportamento não faz sentido do ponto de vista físico, pois a rotação de um motor é medida em rotações por minuto (rpm) e não pode assumir valores negativos nesse contexto, já que representa a magnitude da velocidade de giro. Da mesma forma, o desgaste da ferramenta é uma medida acumulativa de uso ao longo do tempo, sendo sempre um valor positivo ou, no mínimo, zero. Portanto, esses valores precisam ser corrigidos no pré-processamento dos dados.

5.1.2 Verificação dos Dados Estatísticos

A tabela 4 apresenta as estatísticas descritivas das variáveis numéricas do *dataset*. Observa-se que todas as variáveis possuem 50.000 instâncias, sem valores

faltantes. No entanto, foram identificados valores negativos nas colunas "Velocidade de Rotação [rpm]" e "Desgaste da Ferramenta [min]", o que não faz sentido no contexto do problema e exigiu tratamento durante o pré-processamento.

Tabela 4 – Estatísticas Descritivas das Variáveis Numéricas

Estatística	Temp. do Ar [C]	Temp. do Processo [C]	Veloc. Rotação [rpm]	Torque [Nm]	Desgaste da Ferramenta [min]	Falha da Máquina
Contagem	50000.00	50000.00	50000.00	50000.00	50000.00	50000.00
Média	25.99	35.99	1501.29	40.01	165.30	0.49
Desvio Padrão	2.00	2.23	501.31	14.96	74.55	0.50
Mínimo	18.05	26.07	-948.54	0.01	-116.50	0.00
25%	24.65	34.49	1162.20	29.83	112.45	0.00
Mediana	25.99	36.00	1504.97	39.95	162.08	0.00
75%	27.35	37.49	1839.56	50.13	215.81	1.00
Máximo	33.67	44.95	3509.14	103.23	486.94	1.00

Fonte: Autoria própria.

A inspeção dessas estatísticas permitiu identificar possíveis inconsistências nos dados e orientar estratégias de tratamento. A remoção ou correção dos valores negativos foi essencial para evitar distorções nos modelos de aprendizado de máquina, garantindo que os padrões extraídos refletissem melhor as condições reais de operação dos equipamentos.

5.2 PRÉ-PROCESSAMENTO

O pré-processamento dos dados foi uma etapa crucial para garantir a qualidade dos modelos de classificação. As principais etapas realizadas foram: tratamento de valores negativos, checagem de valores nulos e padronização das variáveis. A seguir, são detalhadas essas etapas e seus resultados.

5.2.1 Tratamento de Valores Negativos

Durante a análise exploratória, foram identificados valores negativos nas colunas "Desgaste da Ferramenta [min]" e "Velocidade de Rotação [rpm]". Esses valores não fazem sentido no contexto do problema, pois não existem desgaste negativo nem velocidade de rotação negativa. Foram encontrados 383 valores negativos em "Desgaste da Ferramenta [min]" e 56 em "Velocidade de Rotação [rpm]".

Para corrigir esses valores, os negativos foram substituídos pela média das respectivas colunas. A tabela 5 compara as estatísticas descritivas antes e após a limpeza.

Tabela 5 – Estatísticas Descritivas Antes e Após a Limpeza

Estatística	Desgaste da Ferramenta [min] (Antes)	Desgaste da Ferramenta [min] (Depois)	Veloc. Rotação [rpm] (Antes)	Veloc. Rotação [rpm] (Depois)
Contagem	50000.00	50000.00	50000.00	50000.00
Média	165.30	166.72	1501.29	1503.17
Desvio Padrão	74.55	72.73	501.31	498.11
Mínimo	-116.50	0.01	-948.54	3.88
25%	112.45	114.00	1162.20	1163.98
Mediana	162.08	163.55	1504.97	1504.97
75%	215.81	215.81	1839.56	1839.56
Máximo	486.94	486.94	3509.14	3509.14

Fonte: Autoria própria.

Assim, foi possível observar a eficácia da limpeza e uma baixa alteração nos valores de média e desvio padrão.

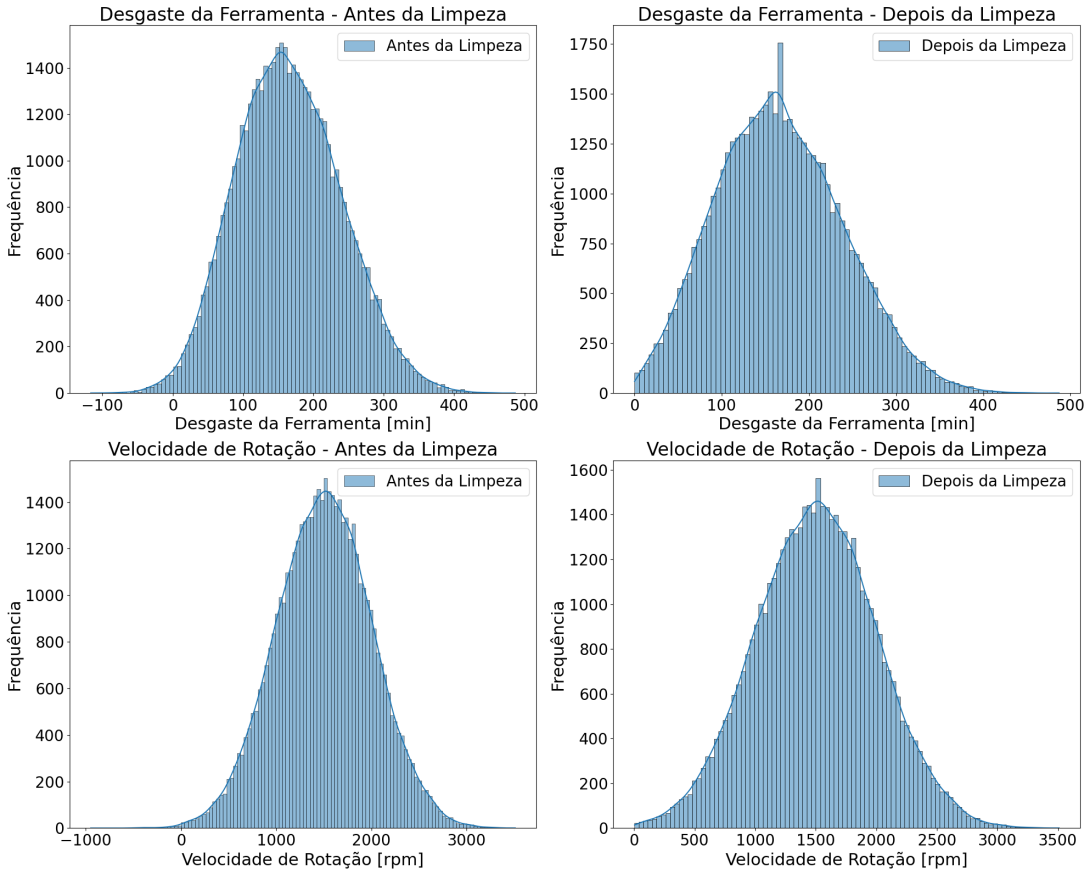
5.2.2 Checagem de Valores Nulos

A checagem de valores nulos revelou que não há dados faltantes no *dataset*. Todas as 50.000 instâncias estão completas, o que dispensou a necessidade de novas alterações.

5.2.3 Visualização da Distribuição Antes e Após a Limpeza

Para avaliar o impacto da limpeza, foram gerados histogramas das variáveis "Desgaste da Ferramenta [min]" e "Velocidade de Rotação [rpm]" antes e após a substituição dos valores negativos. A figura 11 mostra essas distribuições.

Figura 11 – Histogramas das variáveis antes e após a limpeza



Fonte: Autoria própria

Observa-se que, após a limpeza, as distribuições das variáveis se tornaram mais consistentes, sem a presença de valores negativos.

5.2.4 Matriz de Correlação

A matriz de correlação (figura 12) foi criada para avaliar as relações lineares entre as variáveis numéricas. Observa-se que:

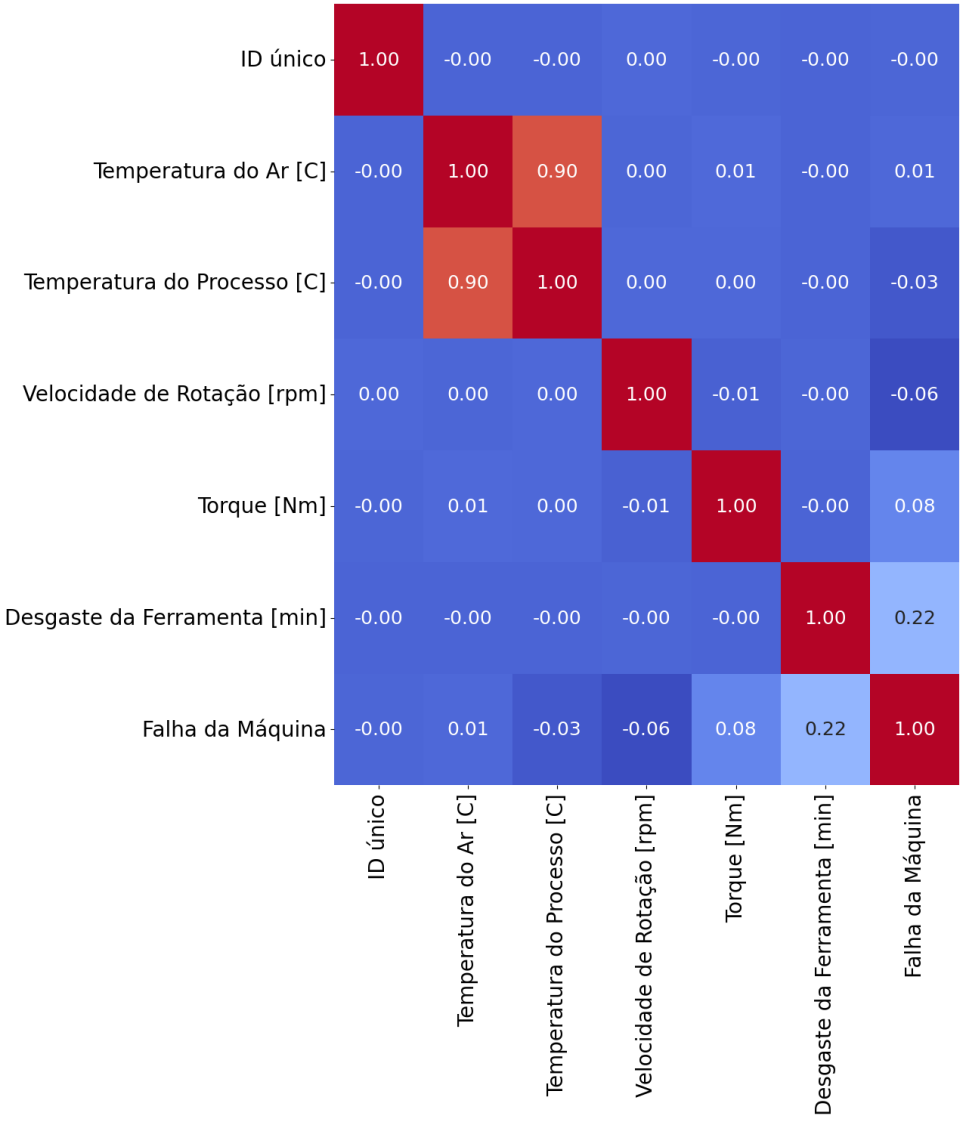
a) A "Temperatura do Ar [℃]" e a "Temperatura do Processo [℃]" apresentam uma correlação positiva elevada, o que era esperado, uma vez que o aumento da temperatura do ar pode influenciar a temperatura do processo;

b) O "Torque [Nm]" e a "Velocidade de Rotação [rpm]" apresentam uma correlação negativa, indicando que, em geral, à medida que o torque aumenta, a velocidade de rotação tende a diminuir;

c) A variável "Falha da Máquina" apresenta correlações fracas com as demais variáveis, sugerindo que a previsão de falhas pode depender de uma combinação

complexa de fatores.

Figura 12 – Matriz de correlação entre as variáveis numéricas



Fonte: Autoria própria

Dessa forma, foi possível observar e identificar padrões de associação entre as variáveis, auxiliando na compreensão da influência de cada fator no comportamento do sistema.

5.3 MODELAGEM E INFERÊNCIA

A modelagem foi realizada utilizando seis algoritmos de classificação: Regressão Logística, *Random Forest*, SVM, KNN, Árvore de Decisão e *Gradient Boosting*. Para cada modelo, foi realizada uma busca de hiperparâmetros utilizando *Grid Search* com

validação cruzada de 5 *folds*. A seguir, são apresentados os resultados detalhados para cada modelo.

5.3.1 Regressão Logística

Os hiperparâmetros testados para a Regressão Logística foram C $\in \{0.1, 1, 10\}$, onde C é um parâmetro de regularização que controla a complexidade do modelo. Valores menores de C aumentam a regularização, tornando o modelo mais simples e prevenindo o *overfitting*, enquanto valores maiores permitem um ajuste mais preciso aos dados, mas podem levar a um modelo mais complexo e suscetível a *overfitting*. O melhor desempenho foi obtido com C igual a 0.1, que proporcionou uma boa generalização sem comprometer a precisão do modelo.

O modelo obteve uma acurácia de 60,27%, com precisão de 60,19% e *recall* de 56,99%. Esses valores refletem um equilíbrio entre a taxa de verdadeiros positivos e a taxa de falsos negativos, conforme indicado pelo *F1-score* de 58,55%. Para uma análise mais detalhada do desempenho do modelo, a Matriz de Confusão foi utilizada, conforme apresentada na tabela 6.

Tabela 6 – Matriz de Confusão para Regressão Logística

	Sem Falha	Falha
Sem Falha	3231	1846
Falha	2118	2805

Fonte: Autoria própria.

No caso deste modelo, a Matriz de Confusão indica que, das previsões feitas, 3.231 foram verdadeiros negativos e 2.805 foram verdadeiros positivos. Por outro lado, houve 1.846 falsos positivos e 2.118 falsos negativos. Esses resultados mostram que o modelo tem uma tendência a cometer um número significativo de erros na classificação da falha da máquina, o que pode indicar a necessidade de ajustes adicionais ou a utilização de técnicas complementares para melhorar seu desempenho.

5.3.2 k-Nearest Neighbours (KNN)

Para o modelo *K-Nearest Neighbors* (KNN), foram testados diferentes valores para o parâmetro número de vizinhos, especificamente os valores 3, 5 e 7. O parâmetro número de vizinhos define quantos vizinhos mais próximos serão considerados pelo

algoritmo para classificar uma nova amostra. Um valor maior tende a suavizar as fronteiras de decisão, enquanto um valor menor pode capturar mais detalhes, mas com risco de *overfitting*. O melhor desempenho foi obtido com esse parâmetro igual a 7. Esse modelo apresentou uma acurácia de 91,01%, com precisão de 94,00% e *recall* de 87,31%. O *F1-score* foi de 90,53%, refletindo um bom equilíbrio entre a taxa de verdadeiros positivos e a taxa de falsos negativos.

A Matriz de Confusão gerada para o modelo KNN é apresentada na tabela 7.

Tabela 7 – Matriz de Confusão para o KNN

	Sem Falha	Falha
Sem Falha	4793	284
Falha	614	4309

Fonte: Autoria própria.

A Matriz de Confusão mostra que o modelo obteve um alto número de acertos na classificação das amostras. Para a classe "Sem Falha", foram corretamente classificadas 4.793 amostras, com apenas 284 falsos positivos. Já para a classe "Falha", o modelo classificou corretamente 4.309 amostras, com 614 falsos negativos. Esses resultados indicam que o modelo tem uma alta capacidade de identificar corretamente ambas as classes, com um número reduzido de erros, o que reforça o bom desempenho geral observado nas métricas de acurácia, precisão e *recall*.

5.3.3 Support Vector Machine (SVM)

No caso do modelo *Support Vector Machine* (SVM), foram testados diferentes valores para os parâmetros C e *kernel*. O parâmetro C controla a penalização por erros de classificação, onde valores maiores priorizam a precisão no treinamento, enquanto o *kernel* define a função usada para transformar os dados em um espaço de maior dimensão, sendo adequado para dados linearmente separáveis. Por outro lado, *Radial Basis Function* (RBF) é adequado para casos mais complexos. Os valores testados para C foram 0,1, 1 e 10, e, para *kernel*, linear e RBF. O melhor desempenho foi obtido com C igual a 10 e *kernel* RBF. Esse modelo apresentou uma acurácia de 93,47%, com precisão de 95,19% e *recall* de 91,35%. O *F1-score* foi de 93,23%, refletindo um bom equilíbrio entre a taxa de verdadeiros positivos e a taxa de falsos negativos.

A matriz de confusão gerada para o modelo SVM é apresentada na tabela 8.

Tabela 8 – Matriz de Confusão para o SVM

	Sem Falha	Falha
Sem Falha	4840	237
Falha	454	4469

Fonte: Autoria própria.

A Matriz de Confusão mostra que o modelo obteve um alto número de acertos na classificação das amostras. O modelo realizou 4.840 classificações de verdadeiros negativos, 4.469 de verdadeiros positivos, 237 de falsos positivos e 454 de falsos negativos. Os resultados indicam que o modelo identifica bem ambas as classes com poucos erros, o que reforça seu bom desempenho nas métricas de acurácia, precisão e *recall*.

5.3.4 Árvore de Decisão

No modelo Árvore de Decisão, foram avaliados diferentes valores para o parâmetro de profundidade máxima: sem limite, 5 e 10. O valor sem limite foi o que obteve o melhor desempenho, permitindo que a árvore crescesse até que cada nó fosse puro ou tivesse menos amostras do que o mínimo necessário para dividir um nó. A tabela 8 mostra a Matriz de Confusão para esse modelo, que alcançou uma acurácia de 91,54%, precisão de 91,18% e *recall* de 91,69%. O *F1-score* foi de 91,43%, indicando um bom equilíbrio entre acertos e erros.

Ao analisar a Matriz de Confusão, apresentada na tabela 9, observa-se que o modelo obteve um bom desempenho, com poucas previsões de falsos positivos e falsos negativos.

Tabela 9 – Matriz de Confusão para a Árvore de Decisão

	Sem Falha	Falha
Sem Falha	4659	418
Falha	449	4474

Fonte: Autoria própria.

Os resultados mostram que a Árvore de Decisão classificou de forma eficiente as amostras. O modelo acertou a maioria das previsões, tanto com falha quanto sem falha, refletindo um ótimo desempenho geral.

5.3.5 Random Forest

No caso do modelo *Random Forest*, diferentes combinações de parâmetros foram avaliadas. O número de estimadores define a quantidade de árvores na floresta, onde valores maiores geralmente melhoram a precisão, mas aumentam o custo computacional. A profundidade máxima controla o tamanho das árvores, sendo que "sem limite" permite que as árvores cresçam até que todas as folhas sejam puras ou contenham um número mínimo de amostras. Para o número de estimadores, os valores 50, 100 e 200 foram testados, enquanto para a profundidade máxima, as opções foram sem limite, 5 e 10. O melhor desempenho foi registrado com 100 estimadores e a profundidade máxima sem limite. Com essa configuração, o modelo alcançou uma acurácia de 94,04%, com uma precisão de 96,48% e *recall* de 91,21%. O *F1-score* obtido foi de 93,77%, o que sugere um desempenho excepcional, particularmente na capacidade de identificar corretamente as classes.

A Matriz de Confusão gerada para o modelo *Random Forest* é apresentada na tabela 10.

Tabela 10 – Matriz de Confusão para o Random Forest

	Sem Falha	Falha
Sem Falha	4911	166
Falha	464	4459

Fonte: Autoria própria.

A Matriz de Confusão revela um desempenho robusto do modelo, com uma alta taxa de classificação correta. Na classe "Sem Falha", 4.911 amostras foram classificadas com precisão, enquanto apenas 166 foram erroneamente identificadas como "Falha". Por outro lado, na classe "Falha", 4.459 amostras foram corretamente detectadas, com 464 classificações incorretas como "Sem Falha". Esses números evidenciam uma eficácia significativa do modelo na distinção entre as duas classes, com uma baixa ocorrência de erros, o que corrobora o desempenho superior demonstrado pelas métricas de acurácia, precisão e *recall*.

5.3.6 Gradient Boosting

Para o modelo *Gradient Boosting*, foram testados diferentes hiperparâmetros, incluindo o número de estimadores, a taxa de aprendizado e a profundidade máxima.

O número de estimadores define quantos modelos base (Árvores de Decisão) serão combinados sequencialmente, enquanto a taxa de aprendizado controla a contribuição de cada modelo no processo de *boosting*, sendo valores menores associados a um aprendizado mais lento, porém mais preciso. A profundidade máxima limita o tamanho das árvores, evitando *overfitting*. Os valores testados para o número de estimadores foram 50, 100 e 200, a taxa de aprendizado variou entre 0,01, 0,1 e 0,2, e a profundidade máxima foi avaliada com os valores 3, 5 e 7. O melhor desempenho foi atingido com 100 estimadores, uma taxa de aprendizado de 0,1 e profundidade máxima de 5. O modelo apresentou uma acurácia de 94,43%, precisão de 96,70% e *recall* de 91,83%. O *F1-score* foi de 94,20%, indicando um equilíbrio muito bom entre a taxa de verdadeiros positivos e falsos negativos.

A Matriz de Confusão gerada para o modelo *Gradient Boosting* é apresentada na tabela 11.

Tabela 11 – Matriz de Confusão para o Gradient Boosting

	Sem Falha	Falha
Sem Falha	5045	32
Falha	243	4680

Fonte: Autoria própria.

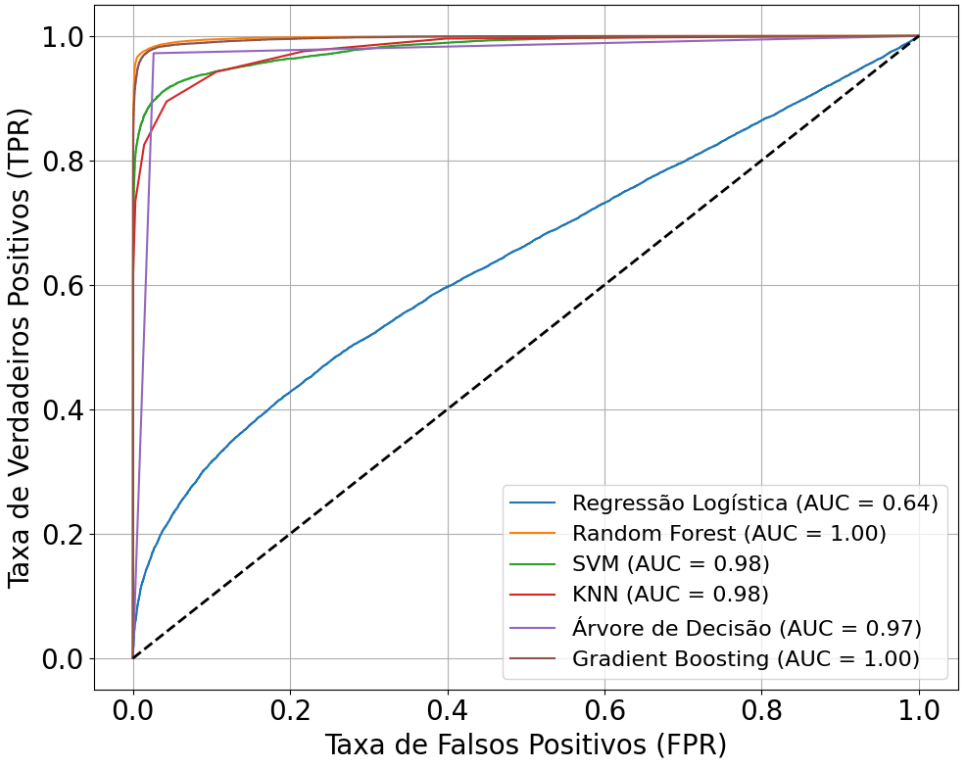
A Matriz de Confusão demonstra um desempenho excepcional do modelo. Para a classe "Sem Falha", 5.045 amostras foram classificadas corretamente, com apenas 32 falsos positivos. Na classe "Falha", 4.680 amostras foram identificadas com precisão, enquanto 243 foram erroneamente classificadas como "Sem Falha". Esses resultados destacam a alta capacidade do modelo em distinguir as classes, com um número extremamente baixo de erros, reforçando o excelente desempenho observado na métrica *F1-score*.

5.3.7 Curvas ROC

As curvas ROC (*Receiver Operating Characteristic*) foram utilizadas para avaliar o desempenho dos modelos de classificação, comparando a taxa de verdadeiros positivos (VP) com a taxa de falsos positivos (FP) em diferentes limiares de decisão. A área sob a curva (AUC) é uma métrica que resume o desempenho do modelo, onde valores próximos a 1 indicam uma capacidade excelente de distinguir entre as

classes, enquanto valores próximos a 0,5 sugerem um desempenho aleatório. A figura 13 apresenta as curvas ROC de todos os modelos avaliados.

Figura 13 – Curvas ROC de todos os modelos



Fonte: Autoria própria

Os melhores modelos foram: *Gradient Boosting*, com AUC = 1, demonstrando uma separação perfeita entre as classes; *Random Forest* e SVM, ambos com AUC = 0,98, mostrando um desempenho quase ideal; KNN, com AUC = 0,97, indicando uma alta capacidade de classificação; Árvore de Decisão, com AUC = 0,91, apresentando um bom desempenho, porém inferior aos anteriores; e Regressão Logística, com AUC = 0,64, revelando uma capacidade limitada de distinção entre as classes.

A peculiaridade das curvas ROC está na sua sensibilidade ao equilíbrio entre VP e FP. Modelos como *Gradient Boosting*, *Random Forest* e SVM apresentam curvas que se aproximam do canto superior esquerdo do gráfico, indicando alta precisão e *recall*. Já a Regressão Logística, com uma curva mais próxima da diagonal, reflete uma performance inferior, possivelmente devido à sua natureza linear e à complexidade dos dados. Esses resultados reforçam a superioridade dos métodos baseados em conjunto de modelos (*Gradient Boosting* e *Random Forest*) e SVM para problemas de

classificação complexos.

5.3.8 Comparação dos Modelos

A tabela 12 apresenta a comparação das métricas de desempenho (acurácia, precisão, *recall* e *F1-Score*) dos modelos avaliados. O *Gradient Boosting* obteve o melhor desempenho, seguido pelo *Random Forest* e pelo SVM. Esses resultados serão discutidos a seguir.

Tabela 12 – Comparação das Métricas de Desempenho

Modelo	Acurácia	Precisão	Recall	F1-Score
Gradient Boosting	94,43%	96,70%	91,83%	94,20%
Random Forest	94,04%	96,48%	91,21%	93,77%
SVM	93,47%	95,19%	91,35%	93,23%
Árvore de Decisão	91,54%	91,18%	91,69%	91,43%
KNN	91,01%	94,00%	87,31%	90,53%
Regressão Logística	60,27%	60,19%	56,99%	58,55%

Fonte: Autoria própria.

O *Gradient Boosting* destacou-se como o modelo mais preciso, com acurácia de 94,43% e *F1-Score* de 94,20%. Seu desempenho superior pode ser atribuído à capacidade de combinar múltiplas Árvores de Decisão sequencialmente, corrigindo erros a cada iteração e capturando relações complexas nos dados. No entanto, essa complexidade adicional resulta em maior custo computacional e reduz a interpretabilidade, tornando a explicação das decisões do modelo mais desafiadora.

O *Random Forest* apresentou um desempenho próximo ao *Gradient Boosting*, com *F1-Score* de 93,77%. Esse modelo, baseado na agregação de várias Árvores de Decisão treinadas em subconjuntos aleatórios dos dados, oferece maior robustez e reduz o risco de *overfitting*. Entretanto, como a previsão final resulta da média de múltiplas árvores, a interpretabilidade do modelo diminui significativamente, o que pode ser um fator limitante em aplicações onde a transparência das decisões é essencial.

O SVM atingiu um *F1-Score* de 93,23%, destacando-se por sua capacidade de encontrar margens de separação ótimas entre as classes. Esse modelo é especialmente eficiente em espaços de alta dimensão, mas seu tempo de processamento aumenta consideravelmente com o tamanho do conjunto de dados. Além disso, interpretar suas decisões pode ser desafiador, pois a fronteira de decisão é determinada por vetores de suporte, dificultando a explicação dos resultados.

A Árvore de Decisão teve um *F1-Score* de 91,43%. Apesar do desempenho inferior aos modelos baseados em conjuntos de árvores, seu grande diferencial é a interpretabilidade. As regras de decisão podem ser visualizadas e compreendidas facilmente, o que é uma vantagem significativa em aplicações onde a transparência das previsões é um requisito fundamental.

O KNN obteve um *F1-Score* de 90,53%, sendo um modelo sensível à escala dos dados e à escolha do número de vizinhos. Embora sua implementação seja simples e ele possa capturar padrões não lineares, seu desempenho pode ser afetado por grandes volumes de dados, tornando-se computacionalmente caro em bases extensas.

Por fim, a Regressão Logística apresentou o pior desempenho, com um *F1-Score* de 58,55%. Esse resultado reflete sua limitação em capturar relações não lineares nos dados, tornando-a mais adequada para problemas linearmente separáveis. No entanto, sua principal vantagem é a interpretabilidade, pois os coeficientes do modelo indicam diretamente a influência de cada variável na previsão.

Esses resultados evidenciam um compromisso entre acurácia e interpretabilidade. Embora modelos como *Gradient Boosting* e *Random Forest* apresentem maior *F1-Score*, sua complexidade dificulta a análise dos fatores que levam a uma falha. No contexto da manutenção preditiva, em que compreender as causas das falhas é essencial para a tomada de decisões assertivas, a interpretabilidade do modelo se torna um critério fundamental. A Árvore de Decisão se destaca nesse aspecto, pois, além de ser mais interpretável que modelos baseados em conjuntos de árvores, também apresenta uma acurácia significativamente maior do que a Regressão Logística (91,54% contra 60,27%). Isso faz com que ela seja uma escolha mais equilibrada, pois mantém um bom nível de previsibilidade sem comprometer a transparência dos resultados. Dessa forma, mesmo que modelos mais complexos ofereçam maior acurácia, a Árvore de Decisão se mostra uma alternativa mais adequada para aplicações que exigem interpretabilidade e embasamento nas decisões.

5.3.9 Interação com o Usuário

Para facilitar o uso do programa pelo usuário, foi criado um *loop* onde podem ser inseridos os valores de cada variável. Com isso, é gerada a previsão. Essa análise é realizada pelo modelo de Árvore de Decisão, visto que se mostrou mais adequado

para essa aplicação. Logo, a figura 14 exibe a interface de operação.

Figura 14 – Interface de operação pelo usuário

```
Insira os dados da máquina:
Temperatura do Ar [C]: 20
Temperatura do Processo [C]: 30
Velocidade de Rotação [rpm]: 1500
Torque [Nm]: 50
Desgaste da Ferramenta [min]: 30

Resultado da previsão: Sem Falha
Probabilidade de falha: 4.96%

Deseja inserir outro dado? (s/n): n
Encerrando as previsões.
```

Fonte: Autoria própria

Assim, é possível ter acesso à previsão e à probabilidade de falha estimadas, auxiliando a tomada de decisão do usuário.

6 CONCLUSÃO

Ao longo deste trabalho, foi possível alcançar os objetivos propostos, culminando em resultados que permitirão a aplicação prática do modelo desenvolvido. A análise exploratória inicial forneceu uma compreensão detalhada do conjunto de dados, destacando características relevantes, padrões e anomalias, o que orientou decisões importantes nas etapas seguintes. Com base nessa análise, os dados foram tratados de forma criteriosa, eliminando inconsistências, como valores negativos e ausentes, garantindo sua qualidade para o processamento subsequente.

O pré-processamento dos dados foi conduzido de maneira estruturada, incluindo a normalização e codificação das variáveis, além da divisão inicial em conjuntos de treino (80%) e teste (20%). Para fortalecer a avaliação do modelo, foi aplicada validação cruzada ao conjunto de treinamento, evitando problemas como *overfitting*. Além disso, o *Grid Search* foi utilizado para otimizar os hiperparâmetros, garantindo que os modelos tivessem um desempenho mais consistente.

Na fase de modelagem, foram testados seis algoritmos de classificação: Regressão Logística, *Random Forest*, SVM, KNN, Árvore de Decisão e *Gradient Boosting*. A avaliação dos modelos com o conjunto de teste revelou diferenças significativas de desempenho entre eles. O *Gradient Boosting* obteve o melhor *F1-score*, demonstrando um bom equilíbrio entre precisão e *recall*, o que indica sua capacidade de prever falhas em equipamentos com alta confiabilidade.

Contudo, em problemas de engenharia mecânica, como a predição de falhas em equipamentos, a interpretabilidade do modelo também desempenha um papel essencial. É importante não apenas atingir altos níveis de acurácia, mas também compreender como o modelo toma suas decisões. A capacidade de explicar as previsões é crucial para que os engenheiros possam confiar no modelo e utilizá-lo para melhorar os processos de manutenção. Nesse contexto, apesar do *Gradient Boosting* e *Random Forest* oferecerem uma acurácia superior, modelos como a Árvore de Decisão se destacam pela clareza e explicabilidade de suas previsões. Embora a Árvore de Decisão tenha apresentado um desempenho ligeiramente inferior em relação aos modelos baseados em conjunto de árvores, ela se mostra mais apropriada para o contexto de engenharia, no qual a transparência das decisões é fundamental para o entendimento das causas das falhas.

Com esses resultados, o próximo passo será implementar o modelo em um ambiente prático, utilizando-o para prever falhas em tempo real e apoiar a tomada de decisões estratégicas. Além disso, o trabalho poderá ser expandido explorando técnicas mais avançadas, como redes neurais, ou incorporando dados reais para aprimorar ainda mais a performance do modelo.

Dessa forma, o estudo reforça a viabilidade do uso de modelos de *Machine Learning* na manutenção preditiva, demonstrando que é possível equilibrar desempenho e interpretabilidade para apoiar a tomada de decisões. Os resultados obtidos evidenciam o potencial dessas técnicas para aprimorar a confiabilidade dos equipamentos e a eficiência dos processos de manutenção, contribuindo para uma gestão mais estratégica e fundamentada.

REFERÊNCIAS

ABRAMAN. Documento nacional. **Associação Brasileira de Manutenção e Gestão de Ativos**, 2021.

ACHOUCH, M.; DIMITROVA, M.; ZIANE, K.; KARGANROUDI, S. S.; DHOUIB, R.; IBRAHIM, H.; ADDA, M. On predictive maintenance in industry 4.0: Overview, models, and challenges. **Appl. Sci**, 2022. Disponível em: <https://doi.org/10.3390/app12168081>.

ALI, T.; KHAN, S.; ALI, A.; AHMAD, N.; AHMAD, S. Unveiling the future: A comprehensivereview of machine learning, deeplearning, multi-model models andexplainable ai in robotics. **Preprints.org**, 2025. Disponível em: <https://www.preprints.org/manuscript/202502.0369/v1>.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2021. ISBN 978-0387310732.

BORLIDO, D. **Indústria 4.0 – Aplicação a Sistemas de Manutenção**. Dissertação (Mestrado) — Faculdade de Engenharia, Universidade do Porto, Porto, Portugal, 2023. Disponível em: <https://repositorio-aberto.up.pt/bitstream/10216/102740/2/181981.pdf>.

BRUCE, A.; BRUCE, P.; GEDECK, P. **Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python**. 2. ed. Sebastopol, Califórnia: O'Reilly Media, 2020.

DAVID, D. **Random Forest Classifier Tutorial: How to Use Tree-Based Algorithms for Machine Learning**. 2020. Disponível em: <https://www.freecodecamp.org/news/how-to-use-the-tree-based-algorithm-for-machine-learning/>. Acesso em: 2025-02-16.

DEMIROVIć, E.; LUKINA, A.; HEBRARD, E.; CHAN, J.; BAILEY, J.; LECKIE, C.; RAMAMOHANARAO, K.; STUCKEY, P. J. Murtree: Optimal classification trees via dynamic programming and search. **Journal of Machine Learning Research 23**, 2022. Disponível em: <https://doi.org/10.48550/arXiv.2007.12652>.

ESCOVEDO, T.; KOSHIYAMA, A. **Introdução a Data Science: Algoritmos de Machine Learning e métodos de análise**. São Paulo, SP: Casa do Código, 2020.

FARIAS, V.; QUELHAS, O. Manutenção 4.0: Uma revisão da literatura da base scopus. **Congresso Internacional de Engenharia Mecânica e Industrial**, 2021.

FERNANDO, F. Tópicos especiais em projetos 4. **Apresentação de slides**, 2024.

FOGLIATTO, F.; RIBEIRO, J. **Confiabilidade e Manutenção Industrial**. 1. ed. Rio de Janeiro: Elsevier: ABEPRO, 2011.

GéRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 3. ed. Sebastopol, Califórnia: O'Reilly Media, 2022.

HOWARD, J.; GUGGER, S. **Deep Learning for Coders with fastai and PyTorch: AI Applications Without a PhD**. Sebastopol, CA: O'Reilly Media, 2020.

IZBICKI, R.; SANTOS, T. M. d. **Aprendizado de máquina: uma abordagem estatística**. São Paulo, SP: UICLAP, 2022.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R.; TAYLOR, J. **An Introduction to Statistical Learning: with Applications in Python**. Suíça: Springer, 2023.

MOUBRAY, J. **Reliability-Centered Maintenance**. 2. ed. Nova York, NY: Butterworth-Heinemann Ltd, 1999.

OSCAR, G. F. **Inteligência Artificial e aprendizagem de máquina: aspectos teóricos e aplicações.** São Paulo, SP: Blucher, 2023.

PASSOS, L. A indústria 4.0: Fundamentos e principais impactos na economia brasileira. **Revista de Administração e Negócios da Amazônia**, 2021.

PICOUTO, G. H. L. **Bioinformatica e a utilização de Genética na Agricultura**. Dissertação (Tese de Graduação) — Escola politécnica e de artes (PUC-GOIÁS), 2023. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/bitstream/123456789/7020/1/BIOINFORMATICA%20E%20A%20UTILIZA%C3%87%C3%83O%20DA%20GEN%C3%89TICA%20NA%20AGRICUTLURA.pdf>. Acesso em: 2025-02-16.

SHMUELI, G.; BRUCE, P. C.; STEPHENS, M. L.; ANANDAMURTHY, M.; PATEL, N. R. **Machine Learning for Business Analytics: Concepts, Techniques and Applications with Jmp Pro**. 2. ed. Hoboken, Nova Jersey: John Wiley Sons, Inc., 2023.

SILVA, T.; NASCIMENTO, G. Importância da manutenção preditiva por análise de vibração num conjunto motor e bomba aplicado (um estudo de caso). **Simpósio Nacional de Ciências e Engenharias (SINACEN)**, 2020. Disponível em: <https://anais.unievangelica.edu.br/index.php/SINACEN/article/download/7601/3759/14218?utm>.

SOUZA, V.; MARCHI, C.; BUENO, N.; FAUSTINO, T.; BARREIRO, T. Utilização das tecnologias da indústria 4.0 na manutenção preditiva através do monitoramento de equipamentos e instalações. **Brazilian Journal of Development**, Curitiba, v. 8, n. 1, 2022. ISSN 7063-7083. Disponível em: <https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/download/43302/pdf/108363?utm>.