



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BAHIA  
(IFBA), CAMPUS JEQUIÉ**

Anne Karoline Cardoso Macedo

**APLICAÇÃO DE APRENDIZADO DE MÁQUINA NA ANÁLISE DE QUALIDADE  
DE ÁGUA DE CALDEIRAS**

Jequié-BA

2023



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BAHIA  
(IFBA), CAMPUS JEQUIÉ**

Anne Karoline Cardoso Macedo

**APLICAÇÃO DE APRENDIZADO DE MÁQUINA NA ANÁLISE DE QUALIDADE  
DE ÁGUA DE CALDEIRAS**

Trabalho de Conclusão de Curso (TCC) apresentado à banca avaliadora, como requisito parcial para conclusão do curso de Engenharia Mecânica do Instituto Federal de Educação, Ciência e Tecnologia da Bahia campus Jequié

Orientador (a): Dr. Marcio Andrade Rocha

Jequié-BA

2023



A banca examinadora, abaixo listada, aprova o Trabalho de Conclusão de Curso **“APLICAÇÃO DE APRENDIZADO DE MÁQUINA NA ANÁLISE DE QUALIDADE DE ÁGUA DE CALDEIRAS”** elaborado por “Anne Karoline Cardoso Macedo” como requisito parcial para obtenção do grau de Bacharel em Engenharia Mecânica, pelo Instituto Federal de Educação, Ciência e Tecnologia da Bahia.

JEQUIÉ-BA, 14/12/2023

Comissão Examinadora

---

Prof. Dr. Márcio Andrade Rocha IFBA  
(Orientador)

---

Prof. Msc. Márcio Henrique Alves dos Santos  
Membro Interno IFBA

---

Prof. Dr. José Josimar Soares  
Membro Externo IFAM.

## RESUMO

A água é um recurso vital para as operações industriais, e muitas indústrias dependem de caldeiras para produzir vapor para diversas finalidades, como geração de energia e aquecimento. A qualidade da água utilizada nas caldeiras é fundamental para garantir sua eficiência e segurança operacional, bem como para evitar a corrosão e o acúmulo de incrustações nos equipamentos. A proposta deste estudo é avaliar a qualidade da água destinada para alimentação de caldeiras, auxiliando na identificação de problemas de qualidade, como a presença de impurezas ou contaminantes. Desta forma, a análise permitirá que as indústrias tomem medidas para evitar danos aos equipamentos, interrupções na operação e possíveis sanções regulatórias, provenientes da não conformidade com as normas regulamentadoras. A partir de dados sintéticos, será desenvolvido um modelo de aprendizado de máquina, que irá classificar a qualidade da água por meio de parâmetros físico-químicos, como a concentração de ferro, cobre, sílica, a alcalinidade e dureza total, e a condutividade, sendo estes parâmetros previstos pelos órgãos regulamentadores. O modelo desenvolvido forneceu uma categorização da qualidade precisa sobre a adequação da água para uso em caldeiras industriais, verificando sua conformidade com as normas e especificações técnicas necessárias para a operação eficiente e segura das caldeiras.

Palavras-chave: máquinas térmicas, análise de dados, python, dados sintéticos.

## **ABSTRACT**

Water is a vital resource for industrial operations, and many industries rely on boilers to produce steam for various purposes such as power generation and heating. The quality of water used in boilers is crucial to ensure their efficiency and operational safety, as well as to prevent corrosion and scale buildup in the equipment. The purpose of this study is to evaluate the quality of water intended for boiler feed, assisting in identifying quality issues such as the presence of impurities or contaminants. This analysis will enable industries to take measures to prevent equipment damage, operational disruptions, and potential regulatory sanctions resulting from non-compliance with regulatory standards. Based on synthetic data, a machine learning model will be developed to classify water quality based on physicochemical parameters such as iron concentration, copper concentration, silica concentration, total alkalinity, total hardness, and conductivity, which are regulated by the governing bodies. The developed model provided an accurate categorization of water quality regarding its suitability for use in industrial boilers, verifying its compliance with the necessary regulations and technical specifications for the efficient and safe operation of boilers.

keywords: thermal machines, data analysis, python, synthetic data.

## LISTA DE ILUSTRAÇÕES

|  |    |
|--|----|
| Figura 1 – Esquema simplificado de uma caldeira de vapor.....              | 14 |
| Figura 2 – Exemplo de matriz de confusão.....                              | 20 |
| Figura 3 – Esquema de funcionamento das iterações do método k-fold.....    | 31 |
| Figura 4 – Matriz de confusão do treinamento com dados desbalanceados..... | 39 |
| Figura 5 – Matriz de confusão do treinamento com dados balanceados.....    | 41 |

## LISTA DE GRÁFICOS

|  |    |
|--|----|
| Gráfico 1 – Tipos de curvas ROC.....   | 22 |
| Gráfico 2 – Proporção do volume de dados entre as classes de conformidade..... | 28 |
| Gráfico 3 – Curvas de aprendizado do modelo treinado com overfitting.....      | 43 |
| Gráfico 4 – Curvas de aprendizado do modelo treinado com ajustes.....          | 44 |

## LISTA DE TABELAS

|   |    |
|---|----|
| Tabela 1 – Quadro de tipos de caldeira.....   | 13 |
| Tabela 2 – Classificação das caldeiras com base da NR 13.....                                   | 14 |
| Tabela 3 – Parâmetros de qualidade da água de caldeira.....                                     | 15 |
| Tabela 4 – Tipos de erro de classificação.....  | 19 |
| Tabela 5 – Valores mínimos considerados na geração de amostras.....                             | 26 |
| Tabela 6 – Medidas de desempenho dos classificadores.....                                       | 32 |
| Tabela 7 – Medidas de desempenho dos regressores.....   | 33 |
| Tabela 8 – Medidas de desempenho dos classificadores após balanceamento.....                    | 34 |
| Tabela 9 – Medidas de desempenho dos regressores após balanceamento.....                        | 35 |
| Tabela 10 – Medidas de desempenho dos classificadores após normalização.....                    | 36 |
| Tabela 11 – Medidas de desempenho dos regressores após normalização.....                        | 37 |
| Tabela 12 – Medidas de desempenho dos classificadores após balanceamento e<br>normalização..... | 38 |
| Tabela 13 – Medidas de desempenho dos regressores após balanceamento e<br>normalização.....     | 38 |
| Tabela 14 – Resultado da validação cruzada com dados balanceados.....                           | 42 |
| Tabela 15 – Medidas de desempenho do modelo após ajustes.....                                   | 44 |
| Tabela 16 – Resultado da validação cruzada após ajustes no modelo.....                          | 45 |

## SUMÁRIO

|  |           |
|--|-----------|
| <b>1 INTRODUÇÃO.....</b>                                 | <b>7</b>  |
| 1.1 Objetivos.....                                       | 8         |
| 1.1.1 Objetivo Geral.....                                | 8         |
| 1.1.2 Objetivos Específicos.....                         | 9         |
| <b>2 REFERENCIAL TEÓRICO.....</b>                        | <b>13</b> |
| 2.1 Caldeiras.....                                       | 13        |
| 2.1.1 Qualidade da água para caldeira.....               | 15        |
| 2.2 Aprendizado de máquina.....                          | 16        |
| 2.3 Aprendizado supervisionado.....                      | 17        |
| 2.4 Árvores de Decisão e Florestas Aleatórias.....       | 17        |
| 2.5 Matriz de confusão.....                              | 19        |
| 2.6 Acurácia.....  | 20        |
| 2.7 Precisão.....  | 21        |
| 2.8 Revocação.....                                       | 21        |
| 2.9 F1 Score.....  | 21        |
| 2.10 A curva ROC.....                                    | 22        |
| <b>3 METODOLOGIA.....</b>                                | <b>23</b> |
| 3.1 Levantamento de requisitos.....                      | 24        |
| 3.2 Geração e pré-processamento dos dados.....           | 25        |
| 3.2.1 Geração de dados sintéticos.....                   | 25        |
| 3.2.2 Balanceamento dos dados.....                       | 27        |
| 3.3 Seleção do modelo de Machine Learning.....           | 28        |
| 3.3.1 Testes com os dados sintéticos.....                | 29        |
| 3.4 Validações após treinamento do modelo escolhido..... | 30        |
| <b>4 RESULTADOS.....</b>                                 | <b>32</b> |
| 4.1 Dados desbalanceados.....                            | 32        |
| 4.2 Dados balanceados.....                               | 33        |
| 4.3 Dados normalizados.....                              | 35        |

|   |           |
|---|-----------|
| 4.4 Dados balanceados e normalizados..... | 37        |
| 4.5 Matriz de confusão.....               | 39        |
| 4.6 Validação cruzada e overfitting.....  | 41        |
| <b>5 CONCLUSÃO.....</b>                   | <b>46</b> |
| <b>REFERÊNCIAS.....</b>                   | <b>47</b> |

## 1 INTRODUÇÃO

Em decorrência da insuficiência de chuvas em determinadas áreas do território brasileiro durante o ano de 2021, os reservatórios das principais usinas hidrelétricas do país enfrentaram uma carência hídrica, gerando uma diminuição na produção de hidreletricidade (BRASIL, 2021).

Com o propósito de contornar esta redução, verificou-se um aumento na oferta de outras fontes energéticas, tais como carvão, vapor, gás natural, energia eólica e solar fotovoltaica, no mesmo período. Conseqüentemente, a participação de renováveis na matriz elétrica brasileira atingiu 78,1% de renovabilidade em 2021, representando uma diminuição em relação ao ano anterior, onde a proporção havia atingido 83,8% (EPE, 2022).

A compensação pela redução no fornecimento de hidreletricidade, refletiu um aumento significativo na produção de termoeletricidade, principalmente a gás natural, obtendo um avanço de 24% em comparação ao ano anterior. Como resultado, a participação da geração termelétrica no total da geração de energia elétrica no Brasil aumentou de 26,2% em 2020 para 31,1% em 2021 (EPE, 2022).

A geração termelétrica é baseada na conversão de energia térmica em energia mecânica, e da conversão desta em energia elétrica. E com o crescimento desse setor no Brasil, faz-se necessário a busca por eficiência operacional na produção de energia, a fim de ter um maior aproveitamento energético dos recursos.

A energia térmica é obtida pela queima de combustíveis fósseis ou renováveis, ou pela fissão nuclear de combustíveis radioativos. Um dos tipos de geração de termoenergia mais comuns é o ciclo a vapor, onde o calor obtido pela queima do combustível é transferido para o fluido, que se expande na forma de vapor a alta pressão e temperatura na caldeira (TOLMASQUIM, 2016).

A água é o principal fluido usado em sistemas de geração de vapor e a qualidade deste afeta diretamente o rendimento e eficiência do ciclo a vapor. Isto porque a água, em decorrência das suas propriedades de solvente e da sua capacidade de transportar partículas, incorpora a si diversas impurezas, as quais definem sua qualidade (ZARPELON; AZZOLINI, 2015). Por isso, é importante ter um controle de qualidade criterioso, garantindo que este fluido trabalhe nos parâmetros esperados.

Entre os pontos que podem ser destacados, a má qualidade da água das caldeiras podem gerar a formação de depósitos e incrustações, que reduzem a transferência de calor e aumentam as perdas de energia. Essas impurezas, ao longo do tempo, também podem estar relacionadas a problemas críticos ao funcionamento das caldeiras, como: sobreaquecimento, corrosão, mudança de estrutura metalográfica e mudança química (IBP, 2020).

Uma maneira de prevenção é recorrer à análise da purga da caldeira e da água de alimentação, a fim de verificar se estas se encontram em parâmetros ideais para o funcionamento da máquina. Caso não estejam, pode-se aplicar medidas corretivas, que, em geral, apresentam um custo financeiro significativamente menor, do que pela reparação dos possíveis problemas causados. Para as indústrias poderem monitorar estes parâmetros, as amostras da água são enviadas a laboratórios especializados, que irão avaliar sua qualidade. Com a expansão da automação industrial, esse processo pode ser revisitado e novas ferramentas para gestão de qualidade podem ser implementadas.

Isso é demonstrado em um estudo realizado na Universidade Estadual de Feira de Santana (UEFS), onde foi proposto o monitoramento automatizado e remoto da qualidade da água da instituição. Foram considerados parâmetros como condutividade, pH, temperatura e oxigênio dissolvido. Os dados foram armazenados em um banco de dados, onde poderiam ser consultados a qualquer momento (LIMA, 2018).

Outro estudo demonstra que é possível identificar regimes de ebulição de forma não intrusiva e com baixo custo, em máquinas térmicas, usando técnicas de aprendizado de máquina em imagens de baixa resolução e velocidade (HOBOLD, 2021). Sendo assim, a tecnologia pode ser utilizada para simplificar processos já existentes. Logo, este estudo se propõe a desenvolver um modelo de aprendizagem de máquina, a fim de colaborar para diminuição dos impactos causados pela má gestão dos parâmetros físico-químicos da água de caldeira.

## **1.1 Objetivos**

### **1.1.1 Objetivo Geral**

Desenvolver um modelo de aprendizado de máquina para avaliação de qualidade de água de caldeiras.

### **1.1.2 Objetivos Específicos**

- Avaliar o algoritmo classificador para aplicação em análises de parâmetros físico-químicos da água;
- Construir um modelo capaz de avaliar a qualidade da água de caldeira com base em normativas.

## 2 REFERENCIAL TEÓRICO

### 2.1 Caldeiras

A caldeira de vapor é um equipamento, que usa da energia química liberada no processo de combustão para promover a mudança de estado físico da água, até que esta se torne vapor em pressões acima da atmosférica. O vapor produzido é utilizado tanto para geração de potência mecânica, quanto para fins de aquecimento em processos industriais (LORA, 2004).

Estas máquinas são empregadas em setores termelétricos, naval, terciário e industrial. Podem ser classificadas como flamotubulares e aquatubulares: nas flamotubulares os gases fluem por dentro de tubos, que estão imersos em água. Nos aquatubulares ocorre o processo contrário, onde a água percorre internamente o corpo tubular, e os gases da combustão ficam externos ao tubo (LORA, 2004).

Como pode ser observado na Tabela 1, cada tipo de caldeira irá se adequar a uma demanda específica de produção de vapor.

| <b>Itens</b>                  | <b>Flamotubulares</b>  | <b>Aquatubulares</b>   |
|-------------------------------|------------------------|------------------------|
| Tipo de vapor produzido       | Vapor saturado         | Vapor superaquecido    |
| Temperatura do vapor          | 150 °C                 | 400 °C                 |
| Exemplo comum de uso do vapor | Aquecimento            | Acionamento de eixos   |
| Combustível                   | Madeira, óleos e gases | Madeira, óleos e gases |
| Capacidade de produzir vapor  | De 100 a 5000 kg/h     | Mais que 5000 kg/h     |

**Fonte: Adaptado de BOTELHO (2018).**

Para uma classificação complementar, as caldeiras podem ser divididas em faixas de pressão de operação, sendo definidos em 4 grupos:

- De baixa a média pressão, onde o valor de pressão é menor que 10 MPa.
- De alta pressão, onde o valor de pressão está entre 10 a 16 MPa;
- De pressão super alta, onde o valor de pressão está entre 17 a 22,1 MPa;

E de pressão supercrítica, onde o valor de pressão está acima de 22,1 MPa (LORA, 2004).

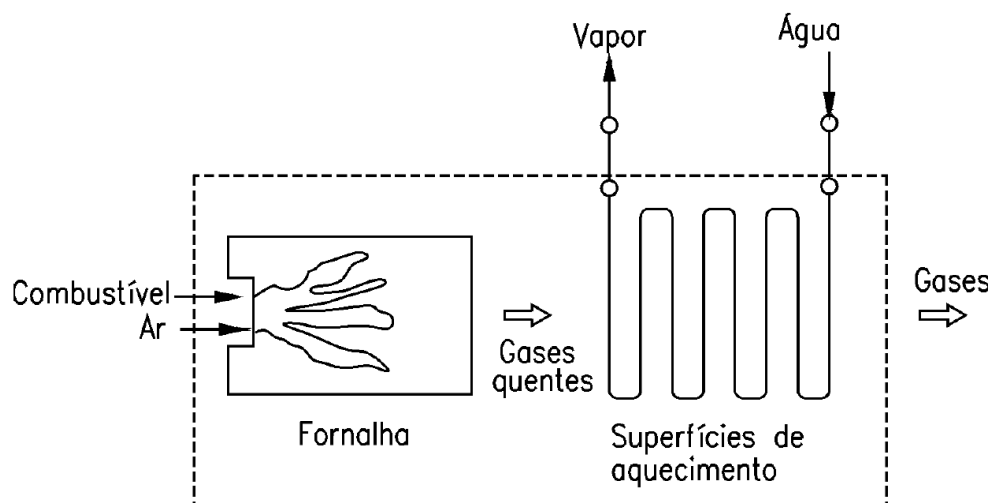
Essas categorias de pressão podem variar a depender do material de referência considerado, contudo, segundo a Norma Regulamentadora 13 (NR 13), às caldeiras se dividem em três grupos, descritos na Tabela 2.

| Tabela 2 – Classificação das caldeiras com base da NR 13 |   |
|--|---|
| Categoria  | Descrição   |
| A  | Pressão interna maior que 19,98 kgf/cm <sup>2</sup>   |
| B  | Pressão interna menor que 5,99 kgf/cm <sup>2</sup> e volume menor que 100 litros  |
| C  | Essa categoria não é descrita na NR 13, mas alguns autores a consideram como a classe de qualquer caldeira que não se enquadram nas categorias A e B. |

Fonte: Adaptado de BRASIL (2022) e BOTELHO (2018).

Em relação ao seu funcionamento básico, a caldeira possui dois principais componentes: a fornalha e as superfícies de aquecimento. A fornalha é onde ocorre o processo de combustão do combustível, gerando como resultado gases aquecidos, que irão trocar calor com o fluido de trabalho através das superfícies de aquecimento, que, em geral, são tubulares (LORA, 2004). Este processo pode ser observado no esquema simplificado da Figura 1.

Figura 1 – Esquema simplificado de uma caldeira de vapor



Fonte: Lora (2004, p. 123).

Vale enfatizar que o fluido de trabalho pode estar em fase líquida, em mistura líquido-vapor, em vapor superaquecido ou ar (LORA, 2004).

### 2.1.1 Qualidade da água para caldeira

Segundo Botelho (2018), a qualidade da água a qual a caldeira será alimentada é vital para a produção de vapor. A água de alimentação pode ter origem de uma rede de saneamento, poços profundos, corpos de água doce e destiladores de água do mar.

Contudo, na água potável há presença de sólidos e gases dissolvidos, e, por isso, ainda necessita ser tratada e ter seus parâmetros físico-químicos balanceados para uma operação segura. Uma baixa qualidade na água pode ocasionar em acúmulo de sólidos, incrustações no material metálico do equipamento e corrosão pela presença de oxigênio dissolvido (BOTELHO, 2018).

**Tabela 3 – Parâmetros de qualidade da água de caldeira**

| Pressão (psi) | Ferro<br>(ppm<br>Fe) | Cobre<br>(ppm<br>Cu) | Dureza                               |                                      | Alcalinidade<br>total (ppm<br>CaCO <sub>3</sub> ) | Condutividade<br>específica<br>(micro-ohms/cm)<br>(não neutralizada) |
|---------------|----------------------|----------------------|--------------------------------------|--------------------------------------|---|--|
|               |                      |                      | total<br>(ppm<br>CaCO <sub>3</sub> ) | Sílica<br>(ppm<br>SiO <sub>2</sub> ) |   |  |
| 0 a 300       | 0,1                  | 0,05                 | 0,3                                  | 150                                  | 700   | 7000   |
| 301 a 450     | 0,05                 | 0,025                | 0,3                                  | 90                                   | 600   | 6000   |
| 451 a 600     | 0,03                 | 0,02                 | 0,2                                  | 40                                   | 500   | 5000   |
| 601 a 750     | 0,025                | 0,02                 | 0,2                                  | 30                                   | 400   | 4000   |
| 751 a 900     | 0,02                 | 0,015                | 0,1                                  | 20                                   | 300   | 3000   |
| 901 a 1000    | 0,02                 | 0,015                | 0,05                                 | 8                                    | 200   | 2000   |
| 1001 a 1500   | 0,01                 | 0,01                 | 0                                    | 2                                    | 0   | 150  |
| 1501 a 2000   | 0,01                 | 0,01                 | 0                                    | 1                                    | 0   | 100  |

**Fonte: Guidelines for Water Quality in Modern Industrial Water Tube Boilers for Reliable Continuous Operation (ASME, 2011).**

Impurezas na água, ao longo do tempo, também podem estar relacionadas a problemas críticos ao funcionamento da caldeira, como: sobreaquecimento, corrosão, mudança de estrutura metalográfica e mudança química (IBP, 2020). Uma maneira de prevenção é recorrer à análise de amostras de água da caldeira, a fim de verificar se seus parâmetros estão adequados para o funcionamento da máquina.

O tratamento da água é uma etapa prevista pela Norma Regulamentadora 13, podendo ser realizado na água de alimentação ou na água que já esteve em operação (BRASIL, 2022).

Os padrões de qualidade da água de alimentação de caldeira são estabelecidos pela The American Society of Mechanical Engineers (ASME), sendo eles: ferro, cobre, condutividade, alcalinidade, dureza total e sílica. Os valores de qualidade para cada parâmetro são estabelecidos por faixas de pressão no qual a caldeira opera, como pode ser observado na Tabela 3.

## **2.2 Aprendizado de máquina**

O aprendizado de máquina é a ciência que concede a habilidade de aprendizagem aos computadores, ou seja, a capacidade de aprender conforme a experiência com determinados dados ou tarefas, sem que este conhecimento ou experiência adquirida fosse explicitamente programada (GÉRON, 2021).

Diferentemente de um algoritmo com regras pré-definidas, a experiência adquirida pelo aprendizado de máquina é adaptativa. Se as regras mudam, na programação tradicional seria necessário atualizar o algoritmo, enquanto no aprendizado de máquina, o mesmo algoritmo pode aprender os novos padrões e estabelecer novas regras (GÉRON, 2021).

Esta diferença faz com que o aprendizado de máquina seja uma alternativa para problemas muito complexos para abordagem tradicional, como o reconhecimento de voz e imagens. Um exemplo de aplicação de aprendizado de máquina na indústria, é a classificação automática de produtos em linha de produção por meio de imagens (GÉRON, 2021).

## 2.3 Aprendizado supervisionado

O aprendizado de máquina pode ser categorizado em tipos de sistemas, considerando critérios como: capacidade de aprendizado gradativo em tempo real, se há supervisão humana no processo, se apenas comparam ou reconhecem padrões nos dados.

No aprendizado supervisionado, como o próprio nome indica, tem-se a supervisão humana no processo de aprendizado. Para isto, no treinamento, é fornecido um conjunto de dados com as soluções desejadas, chamadas de rótulos, para o algoritmo ter a experiência guiada em um padrão de predição almejado (GÉRON, 2021).

Entre as tarefas de aprendizado supervisionado mais comuns, pode ser citada a classificação, seja na classificação de rótulos ou de números para regressão (HARRISON, 2019).

## 2.4 Árvores de Decisão e Florestas Aleatórias

Árvores de decisão são algoritmos de aprendizado de máquina muito versáteis, pois são capazes de trabalhar em tarefas de classificação e regressão - onde comumente se trabalha com rótulo binário -, como em tarefas que exigem uma múltipla saída ou múltiplos rótulos (GÉRON, 2021).

O modelo de árvore de decisão utiliza da medida de índices, chamada de medida de impureza de Gini, para tomada de decisões, ou seja, ele percorre os atributos e encontra um valor que tenha menor probabilidade de erro de classificação. As principais vantagens deste algoritmo é possuir suporte para dados não numéricos e para relacionamentos não lineares, além de não exigir muitas etapas de preparação e tratamento dos dados (HARRISON, 2019).

O maior problema das árvores de decisão é a sensibilidade às variações dos dados de treinamento: caso ocorra uma alteração nos dados, o modelo resultante pode ter um desempenho completamente diferente do anterior à mudança. Isso se dá pelo comportamento estocástico aplicado na maioria dos algoritmos de árvores de decisão (GÉRON, 2021).

O algoritmo CART (Árvores de Classificação e Regressão) é utilizado pela biblioteca *python Scikit-Learn* para árvores de decisão, este funciona repartindo o

conjunto de dados em dois subconjuntos. Esta divisão tem como parâmetros uma única característica  $k$  e um limiar  $t_k$ .

De acordo com Géron (2021), os parâmetros são escolhidos considerando o par  $(k, t_k)$  que produz os subconjuntos mais puros, através de uma função de custo que o modelo tenta minimizar.

Para classificação, essa função de custo é definida pela Equação 1.

$$J(k, t_k) = \frac{m_{\text{esquerdo}}}{m} G_{\text{esquerdo}} + \frac{m_{\text{direito}}}{m} G_{\text{direito}} \quad (1)$$

Sendo que  $G_{\text{direito}}$  e  $G_{\text{esquerdo}}$  calculam a impureza de cada subconjunto, representados por esquerdo e direito. Já o  $m_{\text{direito}}$  e o  $m_{\text{esquerdo}}$  são o número de instâncias de cada subconjunto.

Para regressão funciona da mesma forma, com exceção de que, ao invés de buscar reduzir impureza, o algoritmo irá tentar reduzir o erro quadrático médio (MSE). Essa função de custo é definida pela Equação 2.

$$J(k, t_k) = \frac{m_{\text{esquerdo}}}{m} MSE_{\text{esquerdo}} + \frac{m_{\text{direito}}}{m} MSE_{\text{direito}} \quad (2)$$

Sendo que:

$$MSE_{\text{nó}} = \sum_{i \in \text{nó}} (\hat{y} - y^i)^2 \quad (3)$$

$$\hat{y}_{\text{nó}} = \frac{1}{m_{\text{nó}}} \sum_{i \in \text{nó}} (y^i) \quad (4)$$

Um modelo *random forest* (floresta aleatória) nada mais é que um conjunto de árvores de decisão, sendo um algoritmo que busca corrigir a tendência das árvores de decisão a superadequação aos dados. Seu funcionamento se dá ao agregar previsões de um grupo de preditores, ou ensemble. O método irá obter as previsões individualmente e prever a classe majoritária entre os resultados (GÉRON, 2021).

Ao criar uma floresta aleatória, considera-se apenas um subconjunto aleatório para cada nó das divisões, contudo é possível tornar o processo de árvores ainda mais aleatório. Em um modelo *extra trees* (árvores extras), obtemos essa maior

aleatoriedade definindo limiares arbitrários para cada característica  $k$ . Neste algoritmo tem-se uma menor variância e um treinamento mais rápido, dado que não é necessário calcular os limiares de cada característica em cada nó (GÉRON, 2021).

## 2.5 Matriz de confusão

Um classificador binário pode ter quatro resultados de classificação, sendo eles: *true positive* (verdadeiro positivo), *true negative* (verdadeiro negativo), *false positive* (falso positivo) e *false negative* (falso negativo). Entre elas, as duas primeiras indicam uma classificação correta, enquanto, por sua vez, as duas últimas indicam uma classificação incorreta. Essas classificações incorretas são conhecidas, respectivamente, como erro tipo 1 e erro tipo 2, como pode ser observado na Tabela 4 (HARRISON, 2019).

| Tabela 4 – Tipos de erro de classificação |                         |                         |
|---|-------------------------|-------------------------|
| Real                                      | Previsto como negativo  | Previsto como positivo  |
| Verdadeiramente negativo                  | Verdadeiro negativo     | Falso-positivo (tipo 1) |
| Verdadeiramente positivo                  | Falso-negativo (tipo 2) | Verdadeiro positivo     |

Fonte: Adaptado de HARRISON (2019)

Sabendo a contabilização de cada classificação, pode-se construir uma de matriz de confusão, onde cada linha da matriz representa uma classe real e cada coluna representa uma classe predita. Um classificador perfeito teria valores diferentes de zero apenas na diagonal principal (do superior esquerdo até o inferior direito) (GÉRON, 2021).

Figura 2 – Exemplo de matriz de confusão.

|             |   |                 |    |
|-------------|---|-----------------|----|
| Rótulo Real | 0 | 27              | 3  |
|             | 1 | 5               | 25 |
|             |   | 0               | 1  |
|             |   | Rótulo Previsto |    |

Fonte: Adaptado de HARRISON (2019)

A matriz de confusão é uma das formas de avaliar o desempenho de um classificador, indicando os totais de classes corretas e erros tipo 1 e 2, como visto no exemplo da Figura 2.

## 2.6 Acurácia

A acurácia é o percentual de classificações corretas, sendo sua fórmula definida pela Equação 5.

$$acurácia = \frac{tp+tn}{tp + tn + fp + fn} \quad (5)$$

Onde  $tp$  é o total de verdadeiros positivos,  $tn$  é o total de verdadeiros negativos, o  $fp$  é o total de falsos positivos e o  $fn$  é o total de falsos negativos. A medida resultante, indica quanto o classificador está acertando a predição de classes (HARRISON, 2019). Quanto mais próximo do valor 1, ou 100%, mais acurado é o classificador.

Contudo, a acurácia não deve ser o único critério de qualidade de um classificador, pois um alto valor de acurácia não se traduz em qualidade de predição. Um classificador pode apresentar alta acurácia para classes desbalanceadas, ou seja, se uma classe A é predominante, o classificador pode sempre prever e classificar como A que, no fim, ele ainda acertará a maioria das vezes.

Um classificador deste tipo não é muito útil, por estar enviesado pela formação da base de dados usada no treinamento do aprendizado. Logo, a acurácia, ao ser observada isoladamente, não é medida ideal para classificadores, principalmente, considerando um conjunto de dados assimétricos (GÉRON, 2021).

## 2.7 Precisão

Outro meio de avaliar o desempenho de um modelo classificador é observar a acurácia de predições positivas, ou seja, quantas predições positivas estavam corretas (HARRISON, 2019). Esta medida é nomeada de precisão, varia entre os valores de 0 a 1, sendo expressa pela Equação 6.

$$precisão = \frac{tp}{tp + fp} \quad (6)$$

A precisão, assim como a acurácia, não é uma medida analisada isoladamente. Segundo Géron (2021), uma forma de obter precisão perfeita, sem necessariamente se traduzir em um bom desempenho, é realizar uma única predição positiva e garantir que esteja correta. Portanto, a precisão é observada juntamente a uma métrica chamada *recall* (revocação).

## 2.8 Revocação

A revocação, ou também chamada de sensibilidade do classificador, mede a taxa de verdadeiros positivos, ou seja, quantas classes positivas foram corretamente detectadas pelo modelo (GÉRON, 2021). A revocação é definida pela Equação 7, e também varia entre os valores de 0 a 1.

$$revocação = \frac{tp}{tp + fn} \quad (7)$$

## 2.9 F<sub>1</sub> Score

O F1 Score (pontuação F1) é a média harmônica entre a precisão e a revocação, pertinente ao combinar as duas métricas para simplificar a comparação de desempenho entre classificadores. Diferente da média comum, a harmônica dá um peso maior aos valores mais baixos, logo, o modelo só terá uma pontuação F1 alta se a revocação e a precisão forem altas (HARRISON, 2019).

O F1 Score é definido pela Equação 8.

$$F_1 = \frac{2}{\frac{1}{precisão} + \frac{1}{revocação}} = \frac{2tp}{tp + \frac{fn + fp}{2}} \quad (8)$$

Essa medida favorece classificadores com valores semelhantes entre precisão e revocação, pois ao aumentar um indicador, reduz-se o valor do outro - esse fenômeno se chama *trade-off* de precisão/revocação (GÉRON, 2021).

## 2.10 A curva ROC

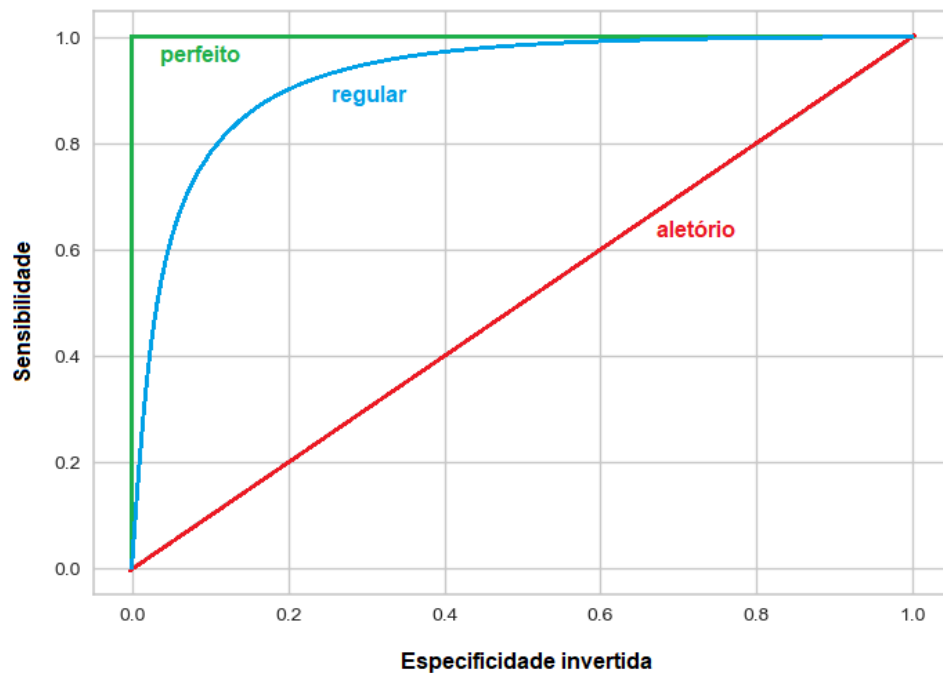
A curva de característica de operação (ROC) representa a taxa de verdadeiros positivos (revocação) em relação à taxa de falsos positivos (GÉRON, 2021). A taxa de falsos positivos (FPR) é dada pela Equação 9.

$$FPR = 1 - \text{especificidade} \quad (9)$$

Sendo a especificidade a taxa de verdadeiros negativos (TNR), dada pela proporção de instâncias negativas corretamente classificadas. Logo, a curva ROC irá representar graficamente a relação da sensibilidade (revocação) e a medida que especificidade invertida (FPR) muda (HARRISON, 2019). Assim como para o F1 Score, na curva ROC, quanto maior a revocação, maior a taxa de falsos positivos.

Uma forma de utilizar a curva ROC para comparação de desempenho é calculando a área sob a curva (AUC). A ROC AUC com valor 1 indica um classificador perfeito, enquanto um ROC AUC de 0,5 indica um comportamento aleatório na predição (GÉRON, 2021).

Gráfico 1 – Tipos de curvas ROC.



Fonte: Autor.

No Gráfico 1, pode-se identificar graficamente as curvas de desempenho do modelo, onde a curva verde representa um classificador perfeito, a azul um comportamento regular e em vermelho um comportamento aleatório. Caso a área sob a curva esteja abaixo do aleatório, tem-se um modelo com péssimo desempenho.

### 3 METODOLOGIA

Para um algoritmo classificar a qualidade do fluido de operação de caldeiras, este precisa ser treinado e ensinado a entender quando esta qualidade será ideal ou não. Para isto, este experimento irá se basear na análise dos parâmetros padrões estabelecidos pela *The American Society of Mechanical Engineers* (ASME), sendo eles: ferro, cobre, condutividade, alcalinidade, dureza total e sílica (ASME, 2011).

O algoritmo classificador deve ser capaz de indicar, em classificação binária, se a qualidade da água obedece aos padrões ou não. Desta forma, o modelo de aprendizagem de máquina poderá ser implementado em sistemas de controle já existentes e colaborar com o diagnóstico, através dos dados do sistema, da água da caldeira.

#### 3.1 Levantamento de requisitos

Na primeira etapa do desenvolvimento, foram identificados quais são os requisitos do projeto. Para ambiente de desenvolvimento, foi utilizado o software Visual Studio Code como editor de código-fonte e Python como linguagem de desenvolvimento.

Para construção e treinamento do modelo de aprendizado de máquina, é preciso ter dados de treino consistentes, para o modelo conseguir identificar os padrões necessários para classificar novos dados de teste.

Para obter os dados, há algumas alternativas:

- Coletar os dados de caldeiras em funcionamento;
- Utilizar dados abertos disponíveis em plataformas de organizações ou governamentais;
- Ou sintetizar dados pseudo-aleatórios, com base em regras que delimitam o cenário e o comportamento dos valores gerados.

Para este estudo, optou-se por sintetizar os dados a serem utilizados no treino do modelo. Isto se deu, pois, para a alternativa de coleta de dados em máquinas reais, seria preciso uma amostra grande e variada de caldeiras diferentes, de forma que cobrisse o máximo de cenários possíveis.

Esta variedade na amostra demandaria um maior tempo de estudo de caso e custo para realização das análises químicas das amostras.

Não foram encontrados dados consistentes em plataformas de dados abertos, principalmente, pois não era encontrado dados dos parâmetros necessários para a análise desejada. Logo, optou-se por simular e gerar dados sintéticos, se baseando nos parâmetros estabelecidos pela ASME (2011), descritas na Tabela 3.

Apesar de optar intencionalmente por um modelo classificador, testes devem ser realizados com modelos de regressão linear, a fim de avaliar o comportamento e decidir pelo modelo com melhores previsões.

Para aumentar o número de possibilidades ao selecionar o modelo a ser utilizado, a biblioteca python *Lazy Predict* será aplicada, por conseguir, com pouco código, construir testes de variados modelos de aprendizagem de máquina, gerando um relatório com as métricas de avaliação de cada um.

Logo, para os modelos que serão testados, deve-se considerar que:

- O modelo deve, a partir dos dados de entrada, classificar a qualidade da água da caldeira de forma binária - 0 ou 1. Sendo 1 o indicador de qualidade positiva e 0 o indicador negativo;
- Será considerado o modelo que obter as melhores métricas de avaliação, como o F1 Score, ROC AUC e acurácia;
- Modelos com métricas abaixo ou próximas a 50% serão descartados, pois por se tratar de uma classe binária, 50% de precisão é, em tese, uma previsão pseudo-aleatória.

## **3.2 Geração e pré-processamento dos dados**

### **3.2.1 Geração de dados sintéticos**

Como discutido, os dados utilizados serão sintetizados a partir dos parâmetros físico-químicos estabelecidos para faixas de pressão estabelecidos pela ASME (2011). Ou seja, para cada intervalo de pressão, serão gerados dados pseudo-aleatórios a partir dos critérios definidos a seguir.

Para cada parâmetro em um determinado intervalo, será considerado seu valor máximo como o limite suportado para que a qualidade da água esteja dentro

do ideal. E, para que amostras de água com baixa qualidade também sejam geradas, calcula-se uma variação entre o valor máximo e mínimo do parâmetro com base no desvio-padrão. O valor do desvio será a tolerância acima e abaixo dos limites de qualidade da água.

$$tolerância = desvio\ padrão = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} \quad (10)$$

O valor mínimo dos parâmetros, para cada faixa de pressão, não são descritos na norma representada na Tabela 3, pois, idealmente, este representaria a menor quantidade possível de impurezas no fluido. Como é preciso um valor mínimo para cálculo da tolerância, será considerado o menor valor para o parâmetro, independente do intervalo de pressão (Tabela 5).

| <b>Tabela 5 – Valores mínimos considerados na geração de amostras</b> |                                 |
|---|---------------------------------|
| <b>Parâmetro</b>  | <b>Valor mínimo considerado</b> |
| Ferro (ppm Fe)  | 0,01                            |
| Cobre (ppm Cu)  | 0,01                            |
| Dureza total (ppm CaCO <sub>3</sub> )                                 | 0                               |
| Sílica (ppm SiO <sub>2</sub> )  | 1                               |
| Alcalinidade total (ppm CaCO <sub>3</sub> )                           | 0                               |
| Condutividade específica (micro-ohms/cm)<br>(não neutralizada)        | 100                             |

**Fonte: Autor.**

Com a tolerância, calculam-se os novos limites máximo e mínimo para o parâmetro, da forma a seguir:

$$limite\ máximo = valor\ máximo + tolerância \quad (11)$$

$$limite\ mínimo = valor\ mínimo - tolerância \quad (12)$$

Em casos onde o novo limite inferior seja menor que zero, o valor é corrigido para ser igual a zero, a fim de se aproximar mais de um cenário real.

Com os limites calculados, números pseudo-aleatórios são escolhidos entre esses limites e organizados para formar os novos dados sintéticos. O processo é repetido para todas as faixas de pressão, gerando 50 amostras por intervalo e um total de 400 amostras sintéticas.

### 3.2.2 Balanceamento dos dados

Ao sintetizar os dados, entende-se que as proporções entre amostras com qualidade de água ideal e não ideal ficam desbalanceadas, pois cada amostra é um número qualquer em um intervalo possível dentro ou fora do padrão recomendado.

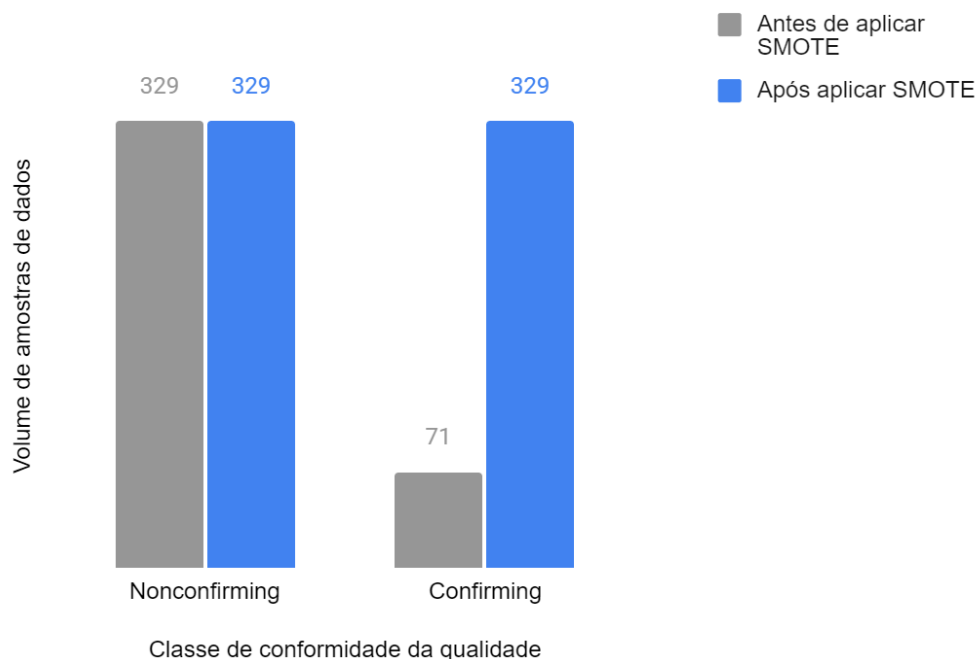
Modelos de aprendizado de máquina podem ser sensíveis a este desbalanceamento, tendendo a ficarem enviesados com a classe com maior proporção de dados. Para os dados gerados, sendo a classe *confirming* representando amostras que estão conforme os padrões de qualidade, e *nonconfirming* aquelas que não estão, pode ser visto na Gráfico 2 que há um desequilíbrio na proporção volumétrica das classes.

Para balancear, foi utilizado o *Synthetic Minority Oversampling Technique* (SMOTE) do pacote *python imbalanced*, uma biblioteca de código aberto, que fornece ferramentas para lidar com classificação de classes desbalanceadas (LEMAITRE; NOGUEIRA; ARIDAS, 2017).

O SMOTE é uma técnica de *oversampling*, onde são gerados novos dados sintéticos da classe minoritária. Isso é feito por meio da criação de novos exemplos a partir dos dados da classe minoritária, gerando sintéticos por meio de uma combinação linear de dados existentes e seus vizinhos mais próximos, com os pesos determinados aleatoriamente (LEMAITRE; NOGUEIRA; ARIDAS, 2017).

Após a aplicação da técnica, pode-se ver na Gráfico 2 que os dados sintéticos gerados ganham uma proporção balanceada.

**Gráfico 2 – Proporção do volume de dados entre as classes de conformidade.**



**Fonte: Autor.**

O *oversampling*, em geral, é uma estratégia útil para melhorar o desempenho de modelos de aprendizado de máquina em tarefas de classificação onde as classes estão desbalanceadas. Será testado o desempenho dos modelos com e sem a aplicação desta técnica, para metrificar o impacto nos dados sintéticos gerados.

### **3.3 Seleção do modelo de Machine Learning**

Como já citado, a biblioteca *Lazy Predict* permitirá a testagem e avaliação de vários modelos ao mesmo tempo, reduzindo a complexidade do desenvolvimento, por não ser necessário a criação e adaptação de treinos e testes individuais. Desta forma, garante-se uma maior variabilidade de algoritmos e possibilidades de escolha.

O *Lazy Predict* fornece o teste de 61 distintos algoritmos, sendo, entre estes, 32 algoritmos de regressão linear e 29 algoritmos classificadores, sendo apenas os últimos de interesse do estudo. As seguintes etapas devem ser seguidas para avaliar os algoritmos classificadores:

- Dividir as amostras sintéticas em um conjunto de teste e outro para treino, numa proporção de, consecutivamente, 30% e 70%.
- Realizar o treinamento supervisionado do algoritmo com os dados de treino.
- Com o algoritmo treinado, aplica-se os dados de testes para predição.
- Com o resultado da predição do modelo, verifica-se se a classificação predita corresponde a classificação real. Esta comparação gera as medidas de desempenho do modelo, que serão utilizadas para comparar a predição dos algoritmos.

As medidas de desempenho avaliadas serão a acurácia, o ROC AUC e o F1 Score do modelo de aprendizado de máquina e, com base no resultado de cada algoritmo, será escolhido aquele com melhores indicadores.

### 3.3.1 Testes com os dados sintéticos

Para este trabalho, tem-se maior interesse na aplicação de classificadores, ao invés de modelos de regressão linear. Pois a regressão obtém melhor desempenho em dados numéricos que estejam em uma mesma escala linear, e que não tenham a presença de muitos dados *outliers*.

Como os dados foram gerados de forma pseudo-aleatória, se pressupõe que teremos a presença de *outliers* e cenários incomuns em comparação com a operação real.

Modelos de floresta de árvores aleatórias, em geral, conseguem lidar com bom desempenho com dados em escalas distintas e com cenários mais diversos. Contudo, por serem modelos propensos a *overfitting*, também é de interesse do estudo experimentar e visualizar qual será o comportamento do modelo com dados mais variados e sem um padrão único, afinal, os dados simulam aleatoriamente caldeiras de diversas faixas de pressão de operação.

Espera-se que, com essa variabilidade, o modelo não tenda a se sobreajustar, sendo capaz de ser inserido em cenários reais e diferentes dos encontrados nos dados.

Para comprovar a hipótese, e com auxílio do *Lazy Predict*, serão testados tanto modelos de regressão linear, quanto outros modelos categóricos além da floresta randômica. Para comparação, dos 61 modelos treinados em cada teste,

serão considerados apenas os 11 classificadores e 11 de regressão linear com melhores desempenhos.

Também serão testados com os dados gerados sem tratamento, com e sem balanceamento, normalizados e não-normalizados. A fim de avaliar se estas etapas de tratamento podem influenciar positivamente, ou não, no desempenho dos modelos.

### **3.4 Validações após treinamento do modelo escolhido**

Com os resultados dos testes, o modelo que apresentar melhor desempenho será escolhido para construção do classificador de qualidade de água de caldeira, porém, serão necessárias novas etapas de validação e ajuste do comportamento do modelo.

A matriz de confusão é um dos meios para avaliar este comportamento, principalmente para identificarmos as recorrências de erros do tipo 1 e 2. Para o modelo escolhido serão construídas duas matrizes: uma com dados desbalanceados e outra com dados balanceados.

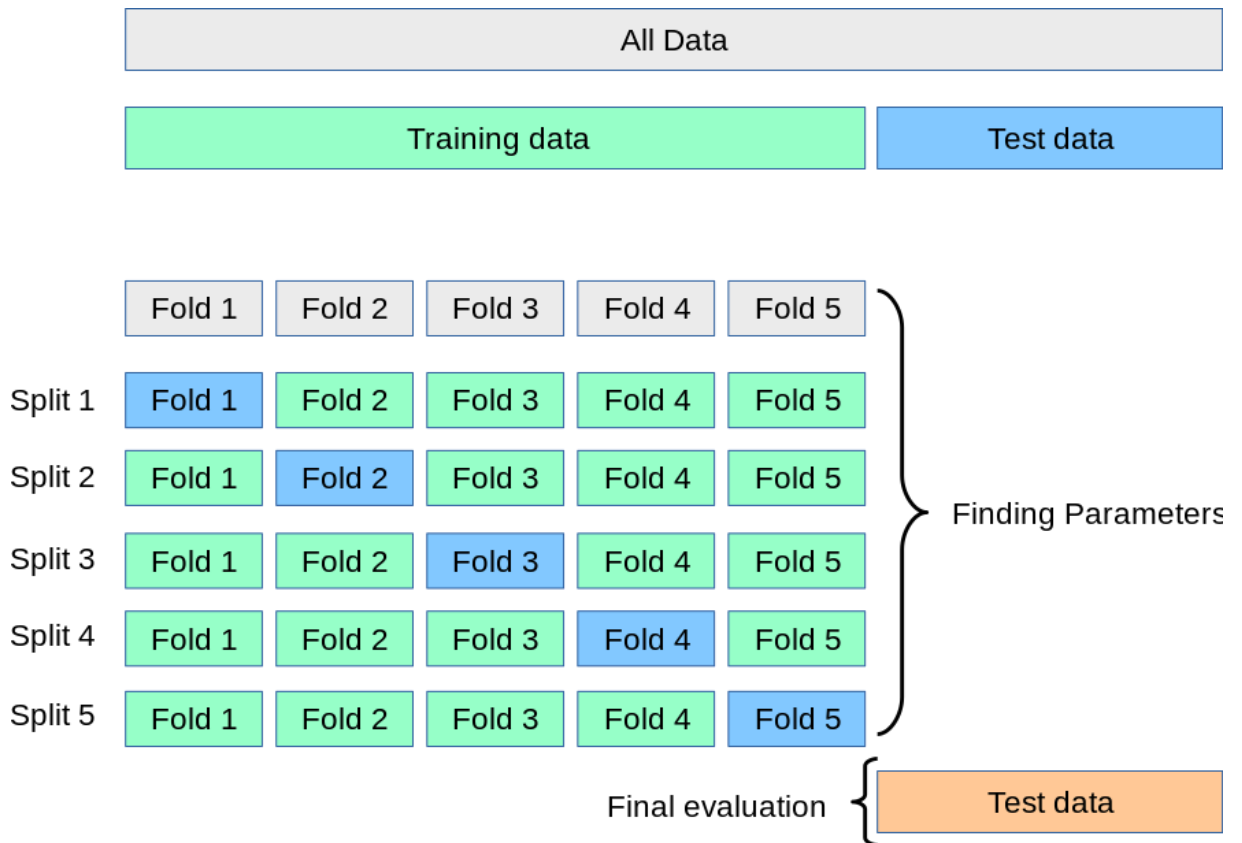
Para essa construção, os conjuntos de dados serão divididos nas proporções de 70% para treino e 30% para teste, e, com base no resultado de testes, as matrizes são elaboradas.

Outra avaliação a ser realizada é pela validação cruzada, pois este auxilia a identificação de *overfitting* no modelo. O método adotado será o *k-fold*, que consiste em repartir os dados de treinamento em um número *k* de subconjuntos, com aproximadamente o mesmo número de amostras em cada.

O número *k* de subconjuntos também terá um número *k* de iterações, para garantir que todos os subconjuntos sejam utilizados para teste e treino durante a validação, como pode ser visto no esquema da Figura 3.

Após a validação cruzada e, a depender dos resultados, o modelo pode passar por ajustes de parâmetros, visando refino do desempenho e redução de erros.

Figura 3 – Esquema de funcionamento das iterações do método *k-fold*.



Fonte: PEDREGOSA (2011).

## 4 RESULTADOS

### 4.1 Dados desbalanceados

O primeiro teste executado ocorreu com os dados gerados sem nenhum tipo de tratamento ou balanceamento. Entende-se que, sem o balanceamento dos dados, os modelos tendem a enviesar seus resultados para a classe majoritária, então não se espera medidas de desempenho muito equilibradas entre si.

Como pode ser visto na Tabela 6, entre os 11 modelos classificadores com melhores desempenhos, obteve-se medidas de desempenho regulares. A média de acurácia entre esses 11 modelos foi de 88%, porém tem-se uma média mais baixa de 77% para a ROC AUC.

Esses dados mostram que, apesar de uma acurácia regular, o viés pelo desbalanceamento dos dados fez com que os modelos perdessem sensibilidade e precisão na predição das categorias de qualidade.

**Tabela 6 – Medidas de desempenho dos classificadores.**

| Ordem | Modelo                        | Acurácia | ROC AUC | F1 Score | Tempo gasto |
|-------|-------------------------------|----------|---------|----------|-------------|
| 1     | AdaBoostClassifier            | 88%      | 83%     | 89%      | 0,17        |
| 2     | LGBMClassifier                | 91%      | 80%     | 91%      | 0,11        |
| 3     | LabelPropagation              | 86%      | 79%     | 87%      | 0,06        |
| 4     | QuadraticDiscriminantAnalysis | 80%      | 79%     | 82%      | 0,02        |
| 5     | XGBClassifier                 | 88%      | 78%     | 89%      | 0,22        |
| 6     | LabelSpreading                | 85%      | 77%     | 86%      | 0,03        |
| 7     | BaggingClassifier             | 89%      | 76%     | 89%      | 0,06        |
| 8     | GaussianNB                    | 67%      | 73%     | 72%      | 0,02        |
| 9     | LogisticRegression            | 91%      | 73%     | 90%      | 0,03        |
| 10    | LinearSVC                     | 91%      | 73%     | 90%      | 0,03        |
| 11    | PassiveAggressiveClassifier   | 57%      | 72%     | 63%      | 0,02        |

Fonte: Autor.

Para os modelos de regressão linear, na Tabela 7, pode-se observar que os melhores desempenhos ainda apresentam comportamento de predição muito impreciso. Isso se dá principalmente pela medida do *R-Squared*, onde o modelo com melhor medida conseguiu explicar a variância dos dados em apenas 50%.

Dado que o *R-Squared* é uma medida baseada na correlação dos dados, quanto mais variância for explicada pelo modelo, mais ajustados os dados estarão em relação à função de regressão.

**Tabela 7 – Medidas de desempenho dos regressores.**

| Ordem | Modelo                        | Adjusted  |           | RMSE | Tempo gasto |
|-------|-------------------------------|-----------|-----------|------|-------------|
|       |                               | R-Squared | R-Squared |      |             |
| 1     | ExtraTreesRegressor           | 0,50      | 0,53      | 0,24 | 0,17        |
| 2     | SVR                           | 0,44      | 0,47      | 0,25 | 0,03        |
| 3     | MLPRegressor                  | 0,40      | 0,44      | 0,26 | 0,34        |
| 4     | NuSVR                         | 0,32      | 0,36      | 0,28 | 0,02        |
| 5     | RandomForestRegressor         | 0,30      | 0,34      | 0,28 | 0,28        |
| 6     | HistGradientBoostingRegressor | 0,30      | 0,34      | 0,28 | 0,45        |
| 7     | GradientBoostingRegressor     | 0,30      | 0,34      | 0,28 | 0,19        |
| 8     | LGBMRegressor                 | 0,26      | 0,30      | 0,29 | 0,07        |
| 9     | BaggingRegressor              | 0,25      | 0,30      | 0,29 | 0,05        |
| 10    | OrthogonalMatchingPursuitCV   | 0,17      | 0,22      | 0,31 | 0,02        |
| 11    | Lars                          | 0,17      | 0,22      | 0,31 | 0,02        |

**Fonte: Autor.**

Com isso, para os testes com dados sem tratamento, os modelos classificadores possuem melhor desempenho, porém ainda são passíveis de aperfeiçoamento.

## 4.2 Dados balanceados

O segundo teste executado ocorreu com os dados após aplicação da técnica de SMOTE, ou seja, com as classes equilibradas. Entende-se que, com o balanceamento dos dados, os modelos tendem a fazer previsões mais precisas e acuradas. Logo, em relação ao teste anterior, espera-se medidas de desempenho mais equilibradas entre si.

Como pode ser visto na Tabela 8, entre os modelos classificadores, obteve-se medidas de desempenho mais altas. A média de acurácia entre esses 11 modelos foi de 89%, porém, modelos como *ExtraTreesClassifier* e *RandomForestClassifier* já apresentam um equilíbrio entre as três métricas e uma boa relação entre sensibilidade e precisão, observados pela ROC AUC de 94% e 91%, respectivamente.

**Tabela 8 – Medidas de desempenho dos classificadores após balanceamento.**

| Ordem | Modelo                 | Acurácia | ROC AUC | F1 Score | Tempo gasto |
|-------|------------------------|----------|---------|----------|-------------|
| 1     | ExtraTreesClassifier   | 93%      | 94%     | 93%      | 0,20        |
| 2     | RandomForestClassifier | 91%      | 92%     | 91%      | 0,36        |
| 3     | DecisionTreeClassifier | 90%      | 90%     | 90%      | 0,02        |
| 4     | XGBClassifier          | 90%      | 90%     | 90%      | 0,10        |
| 5     | LGBMClassifier         | 90%      | 90%     | 90%      | 0,12        |
| 6     | BaggingClassifier      | 89%      | 89%     | 89%      | 0,07        |
| 7     | LabelSpreading         | 87%      | 87%     | 87%      | 0,04        |
| 8     | LabelPropagation       | 86%      | 87%     | 86%      | 0,03        |
| 9     | AdaBoostClassifier     | 86%      | 86%     | 86%      | 0,17        |
| 10    | NuSVC                  | 86%      | 86%     | 86%      | 0,04        |
| 11    | SVC                    | 86%      | 86%     | 86%      | 0,03        |

**Fonte: Autor.**

O tempo gasto para execução dos testes é relativamente baixo para todos os modelos, chegando ao máximo de 36 segundos para classificadores, onde os dois

citados são os com maiores tempos de execução. Mesmo sendo os mais demorados, ainda possuem um bom tempo de resposta para o intuito de aplicação do estudo.

Para os modelos de regressão linear, na Tabela 9, observa-se que os modelos, apesar de alcançarem melhores médias, continuam apresentando comportamento de predição impreciso em relação aos classificadores.

O tempo gasto para execução dos testes também é relativamente baixo para os regressores, porém, em comparação, temos um acréscimo no tempo de resposta que chega a 71 segundos para o modelo *HistGradientBoostingRegressor*.

Pensando em aplicações para resposta em tempo-real, é interessante manter modelos com o menor tempo possível, e que ainda apresentem um desempenho satisfatório.

**Tabela 9 – Medidas de desempenho dos regressores após balanceamento.**

| Ordem | Modelo                        | Adjusted  |           | RMSE | Tempo gasto |
|-------|-------------------------------|-----------|-----------|------|-------------|
|       |                               | R-Squared | R-Squared |      |             |
| 1     | ExtraTreesRegressor           | 0,78      | 0,79      | 0,23 | 0,22        |
| 2     | HistGradientBoostingRegressor | 0,69      | 0,70      | 0,27 | 0,71        |
| 3     | LGBMRegressor                 | 0,69      | 0,70      | 0,27 | 0,10        |
| 4     | XGBRegressor                  | 0,69      | 0,70      | 0,27 | 0,12        |
| 5     | RandomForestRegressor         | 0,69      | 0,70      | 0,27 | 0,40        |
| 6     | BaggingRegressor              | 0,64      | 0,66      | 0,29 | 0,07        |
| 7     | GradientBoostingRegressor     | 0,64      | 0,66      | 0,29 | 0,27        |
| 8     | DecisionTreeRegressor         | 0,60      | 0,62      | 0,31 | 0,03        |
| 9     | NuSVR                         | 0,58      | 0,59      | 0,32 | 0,05        |
| 10    | SVR                           | 0,57      | 0,59      | 0,32 | 0,04        |
| 11    | MLPRegressor                  | 0,53      | 0,55      | 0,34 | 0,43        |

Fonte: Autor.

### 4.3 Dados normalizados

Nos testes anteriores, a hipótese em evidência era de que o balanceamento dos dados interferiria no desempenho dos modelos. Porém, outro fator pode influenciar as métricas de um modelo, principalmente, neste caso, de regressão linear.

Regressores, em geral, são mais sensíveis que os classificadores para classes com distintas escalas. Nos dados observados, parâmetros como condutividade e cobre, tem escalas numéricas com dimensionamento desproporcional. Enquanto uma classe pode alcançar valores na casa de sete mil, outra não ultrapassará um valor absoluto de 0,1.

Essa desproporção entre as classes pode fazer com que as de maiores valores tenham um maior peso no resultado dado pelo modelo. Sendo assim, uma técnica aplicada para evitar este cenário é a da normalização.

**Tabela 10 – Medidas de desempenho dos classificadores após normalização.**

| Ordem | Modelo                        | Acurácia | ROC AUC | F1 Score | Tempo gasto |
|-------|-------------------------------|----------|---------|----------|-------------|
| 1     | AdaBoostClassifier            | 88%      | 83%     | 89%      | 0,22        |
| 2     | LGBMClassifier                | 91%      | 80%     | 91%      | 0,10        |
| 3     | LabelPropagation              | 86%      | 79%     | 87%      | 0,04        |
| 4     | QuadraticDiscriminantAnalysis | 80%      | 79%     | 82%      | 0,02        |
| 5     | XGBClassifier                 | 88%      | 78%     | 89%      | 0,13        |
| 6     | LabelSpreading                | 85%      | 77%     | 86%      | 0,03        |
| 7     | BaggingClassifier             | 89%      | 76%     | 89%      | 0,05        |
| 8     | GaussianNB                    | 67%      | 73%     | 72%      | 0,02        |
| 9     | LogisticRegression            | 91%      | 73%     | 90%      | 0,04        |
| 10    | LinearSVC                     | 91%      | 73%     | 90%      | 0,03        |
| 11    | PassiveAggressiveClassifier   | 57%      | 72%     | 63%      | 0,02        |

**Fonte: Autor.**

Para manter a validade da hipótese em que os classificadores obtiveram, na maioria dos casos, um melhor desempenho pelas características e propriedades dos

dados observados, o terceiro teste avaliará os resultados para dados normalizados e sem aplicação da técnica de SMOTE.

Como pode ser visto na Tabela 10, entre os modelos classificadores, o desempenho geral decaiu em relação ao teste com dados balanceados. Era esperado essa queda, dado a observação do comportamento enviesado destes com classes desiguais.

Já para os modelos regressores (Tabela 11), obteve-se desempenho fora do esperado pela hipótese, já que os valores preditos e reais se distanciaram, como indicado pela métrica RMSE. O esperado era que os regressores possuíssem melhores resultados com a normalização das escalas, mas, isto pode ter ocorrido pela presença de *outliers* causados pela randomização na geração dos dados.

**Tabela 11 – Medidas de desempenho dos regressores após normalização.**

| Ordem | Modelo                        | Adjusted  |           | RMSE | Tempo gasto |
|-------|-------------------------------|-----------|-----------|------|-------------|
|       |                               | R-Squared | R-Squared |      |             |
| 1     | ExtraTreesRegressor           | 0,50      | 0,53      | 0,24 | 0,16        |
| 2     | SVR                           | 0,44      | 0,47      | 0,25 | 0,03        |
| 3     | MLPRegressor                  | 0,40      | 0,44      | 0,26 | 0,29        |
| 4     | NuSVR                         | 0,32      | 0,36      | 0,28 | 0,02        |
| 5     | RandomForestRegressor         | 0,30      | 0,34      | 0,28 | 0,31        |
| 6     | HistGradientBoostingRegressor | 0,30      | 0,34      | 0,28 | 0,48        |
| 7     | GradientBoostingRegressor     | 0,30      | 0,34      | 0,28 | 0,20        |
| 8     | LGBMRegressor                 | 0,26      | 0,30      | 0,29 | 0,07        |
| 9     | BaggingRegressor              | 0,25      | 0,30      | 0,29 | 0,04        |
| 10    | OrthogonalMatchingPursuitCV   | 0,17      | 0,22      | 0,31 | 0,02        |
| 11    | Lars                          | 0,17      | 0,22      | 0,31 | 0,02        |

**Fonte: Autor.**

Outro grande fator de influência, pode ser o desbalanceamento dos dados, pois já foi observado no teste anterior que este tratamento beneficia o desempenho de todos os modelos - classificadores ou regressores.

Para validar a influência do balanceamento, será necessário um novo teste com dados balanceados e normalizados.

#### 4.4 Dados balanceados e normalizados

Dado o resultado do terceiro teste, o quarto teste se propõe a validar se a queda no desempenho ocorreu pelo desbalanceamento dos dados, ou se foi uma influência direta da técnica de normalização aplicada. Para isso, os modelos serão treinados e testados com dados balanceados e, logo após, normalizados.

Como pode ser visto nas Tabelas 12 e 13, tanto os modelos classificadores, quanto os regressores, não obtiveram uma diferença significativa no desempenho em relação ao teste com dados apenas balanceados.

**Tabela 12 – Medidas de desempenho dos classificadores após balanceamento e normalização.**

| Ordem | Modelo                 | Acurácia | ROC AUC | F1 Score | Tempo gasto |
|-------|------------------------|----------|---------|----------|-------------|
| 1     | ExtraTreesClassifier   | 93%      | 93%     | 93%      | 0,20        |
| 2     | LGBMClassifier         | 90%      | 91%     | 90%      | 0,11        |
| 3     | RandomForestClassifier | 90%      | 91%     | 90%      | 0,32        |
| 4     | XGBClassifier          | 89%      | 89%     | 89%      | 0,10        |
| 5     | BaggingClassifier      | 88%      | 88%     | 88%      | 0,06        |
| 6     | AdaBoostClassifier     | 86%      | 86%     | 86%      | 0,15        |
| 7     | LabelSpreading         | 86%      | 86%     | 86%      | 0,05        |
| 8     | LabelPropagation       | 85%      | 86%     | 85%      | 0,03        |
| 9     | DecisionTreeClassifier | 85%      | 85%     | 85%      | 0,03        |
| 10    | SVC                    | 85%      | 85%     | 85%      | 0,04        |
| 11    | NuSVC                  | 85%      | 85%     | 85%      | 0,05        |

**Fonte: Autor.**

Contudo, diferente do terceiro teste, o RSME nos modelos regressores apresenta valores baixos e mais próximos aos alcançados no teste dois. Ou seja, o tratamento de balanceamento tem influência direta no desempenho dos modelos,

diferentemente da normalização, que, para este caso, não teve impacto positivo significativo.

Como observado nos quatro testes realizados, o modelo de melhor desempenho com os dados foi o *ExtraTreesClassifier* e, por isto, este foi escolhido para construção do modelo para categorização da qualidade da água de caldeira.

**Tabela 13 – Medidas de desempenho dos regressores após balanceamento e normalização.**

| Ordem | Modelo                        | Adjusted  |           | RMSE | Tempo gasto |
|-------|-------------------------------|-----------|-----------|------|-------------|
|       |                               | R-Squared | R-Squared |      |             |
| 1     | ExtraTreesRegressor           | 0,77      | 0,78      | 0,24 | 0,21        |
| 2     | HistGradientBoostingRegressor | 0,70      | 0,71      | 0,27 | 0,71        |
| 3     | LGBMRegressor                 | 0,69      | 0,70      | 0,27 | 0,10        |
| 4     | BaggingRegressor              | 0,68      | 0,69      | 0,28 | 0,05        |
| 5     | RandomForestRegressor         | 0,68      | 0,69      | 0,28 | 0,39        |
| 6     | XGBRegressor                  | 0,64      | 0,66      | 0,29 | 0,12        |
| 7     | GradientBoostingRegressor     | 0,62      | 0,63      | 0,30 | 0,26        |
| 8     | NuSVR                         | 0,58      | 0,59      | 0,32 | 0,05        |
| 9     | SVR                           | 0,57      | 0,59      | 0,32 | 0,04        |
| 10    | MLPRegressor                  | 0,51      | 0,53      | 0,34 | 0,45        |
| 11    | ExtraTreeRegressor            | 0,48      | 0,49      | 0,36 | 0,01        |

Fonte: Autor.

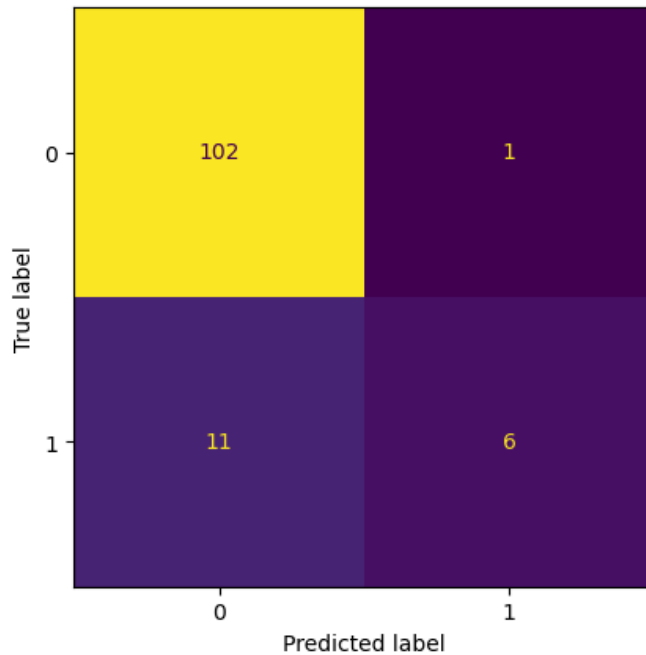
#### 4.5 Matriz de confusão

A matriz de confusão é uma meio para avaliar o desempenho de um modelo, principalmente para identificarmos as recorrências de erros do tipo 1 e 2. Para o modelo *ExtraTreesClassifier* foram construídas duas matrizes: uma com dados desbalanceados e outra com dados balanceados.

Para essa construção, os conjuntos de dados foram divididos nas proporções de 70% para treino e 30% para teste, e, com base no resultado de testes, as matrizes são elaboradas.

Como pode ser visto na Figura 4, para dados desbalanceados, entre as 120 amostras de teste, 103 estavam fora do padrão de qualidade, enquanto apenas 17 atendiam o padrão.

**Figura 4 – Matriz de confusão do treinamento com dados desbalanceados.**



**Fonte: Autor.**

Observa-se que é predominante o erro tipo 1, ou seja, o modelo teve baixa sensibilidade ao tentar decidir se uma amostra atende o padrão, quando de fato isto é verdadeiro. Em números, têm-se 35% para a medida de sensibilidade e 86% de precisão, um desequilíbrio esperado, dado que uma classe tem vantagem em volumetria de amostras.

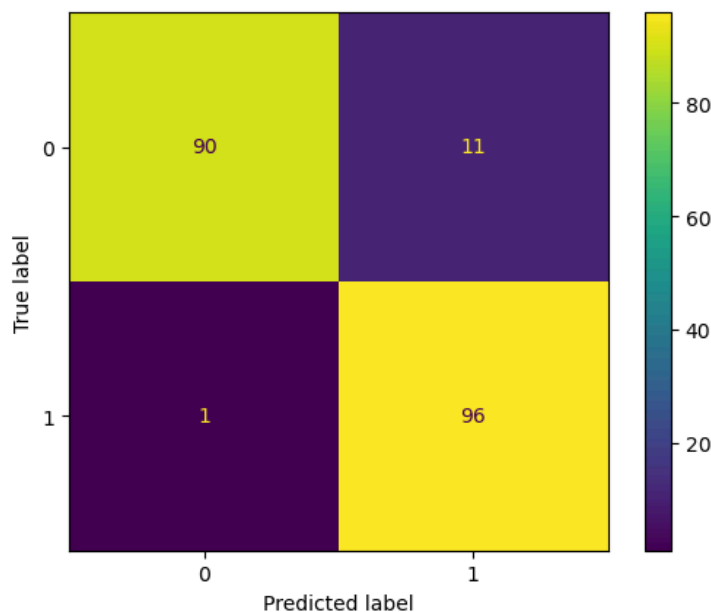
Para os dados balanceados, é visto na figura 5 que de 198 amostras de teste, 101 são amostras fora do padrão de qualidade, enquanto 97 atendem o padrão. Aqui já se observa um equilíbrio entre as classes e diminuição significativa do erro tipo 1, e, conseqüentemente, melhora no indicador de sensibilidade.

Em números, têm-se 98,9% de sensibilidade e 89,7% de precisão, que juntos compõem um cenário mais favorável em comparação com a matriz anterior. Porém, ter a sensibilidade muito próxima de 100% pode ser um indicativo de *overfitting* do modelo, podendo ser uma oportunidade para ajustar os dados, ganhar mais precisão e tornar estas métricas mais próximas.

Para erros do tipo 2, tem-se um baixo volume de erros em relação ao total. Não é trivial controlar estas duas medidas de erro, pois, em geral, elas não são aperfeiçoadas simultaneamente.

Entende-se que, para este estudo, há um maior peso no erro tipo 2, pois em um cenário de manutenção e monitoramento preventivo, o aumento de erros tipo 1 significa mais falsos alertas, porém o tipo 2, significaria que amostras fora do padrão não estariam sendo corretamente detectadas.

**Figura 5 – Matriz de confusão do treinamento com dados balanceados.**



Fonte: Autor.

#### 4.6 Validação cruzada e *overfitting*

Árvores de decisão e florestas aleatórias são algoritmos que possuem uma tendência a se sobreajustarem aos dados. Este fator pode ser um problema para aplicação do modelo treinado em novos dados - gerados em cenários distintos.

Por isso, será necessário avaliar se há *overfitting* no modelo, e para isso, foi utilizado o método de *k-fold* em 10 iterações para avaliar o modelo. Vale enfatizar que, cada vez que for executado, a validação cruzada irá retornar métricas diferentes, pois os subconjuntos são repartidos de forma aleatória.

Na Tabela 14, observa-se o resultado obtido em cada iteração. A depender de como os subconjuntos foram separados, vê-se que o comportamento do modelo

alcança medidas abaixo de 50% para a revocação e a precisão, enquanto o melhor resultado alcança valores acima de 90%.

Essa discrepância entre os resultados das iterações é um grande indicativo de que o modelo se sobreajustou aos ruídos dos dados e, por isso, não soube lidar com os novos dados de forma satisfatória.

Dado este resultado, foi necessário ajustar os parâmetros de criação do modelo, definindo novos valores limite para o número de árvores na floresta, a profundidade máxima das árvores e o número mínimo de amostras necessárias para dividir um nó interno.

| <b>Iteração</b> | <b>Tempo gasto</b> | <b>Precisão</b> | <b>Sensibilidade/Revocação</b> | <b>Acurácia</b> |
|-----------------|--------------------|-----------------|--------------------------------|-----------------|
| 1               | 0,0210             | 41,0%           | 48,5%                          | 80,0%           |
| 2               | 0,0220             | 92,3%           | 57,1%                          | 85,0%           |
| 3               | 0,0180             | 92,3%           | 57,1%                          | 85,0%           |
| 4               | 0,0210             | 41,3%           | 50,0%                          | 82,5%           |
| 5               | 0,0180             | 41,3%           | 50,0%                          | 82,5%           |
| 6               | 0,0180             | 58,6%           | 54,1%                          | 80,0%           |
| 7               | 0,0180             | 41,0%           | 48,5%                          | 80,0%           |
| 8               | 0,0190             | 95,8%           | 78,6%                          | 92,5%           |
| 9               | 0,0210             | 91,3%           | 91,3%                          | 95,0%           |
| 10              | 0,0180             | 91,0%           | 56,3%                          | 82,5%           |

**Fonte: Autor.**

No primeiro treinamento, foram criadas 100 árvores na floresta, e pelo menos 2 amostras eram necessárias para dividir um nó interno. Já para a profundidade, as árvores foram expandidas até que todas as folhas fossem puras ou contenham menos amostras que o mínimo estipulado.

No ajuste, esses parâmetros foram configurados da seguinte forma:

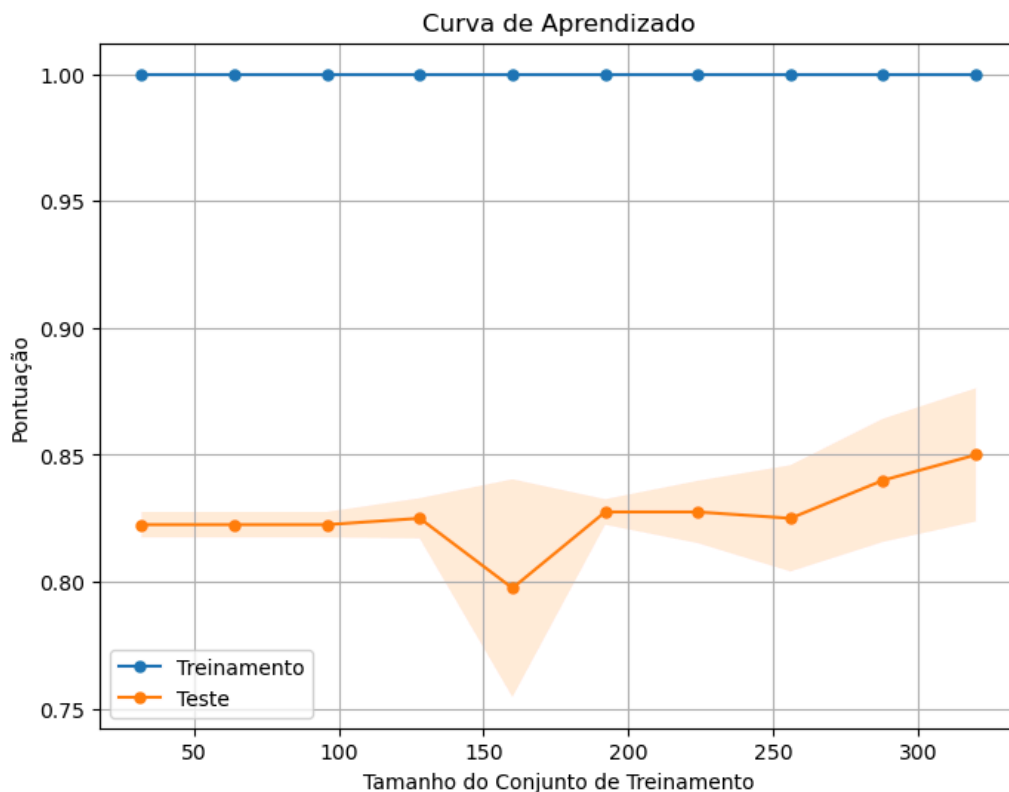
- 50 árvores na floresta;
- No mínimo 15 amostras para dividir um nó interno;

- Profundidade máxima de cada árvore em 30;
- E a semente - para controlar a aleatoriedade - de valor 42.

Para facilitar a análise de overfitting do modelo, foram construídos dois gráficos de curva de aprendizado do modelo, um antes e outro após os ajustes. O gráfico compara a curva de aprendizado do modelo no treinamento e no teste, e, como pode ser visto na Gráfico 3, para o modelo sem ajustes a curva de treinamento é linear e possui uma pontuação constante no valor 1 (ou 100%).

Essa aprendizagem constante é um indicativo do comportamento já notado de overfitting, dado que o modelo se adaptou completamente aos dados de treinamento. Nesse cenário, o modelo está apenas se especializando nos dados de treinamento, em vez de generalizar a partir deles.

**Gráfico 3 – Curvas de aprendizado do modelo treinado com *overfitting*.**

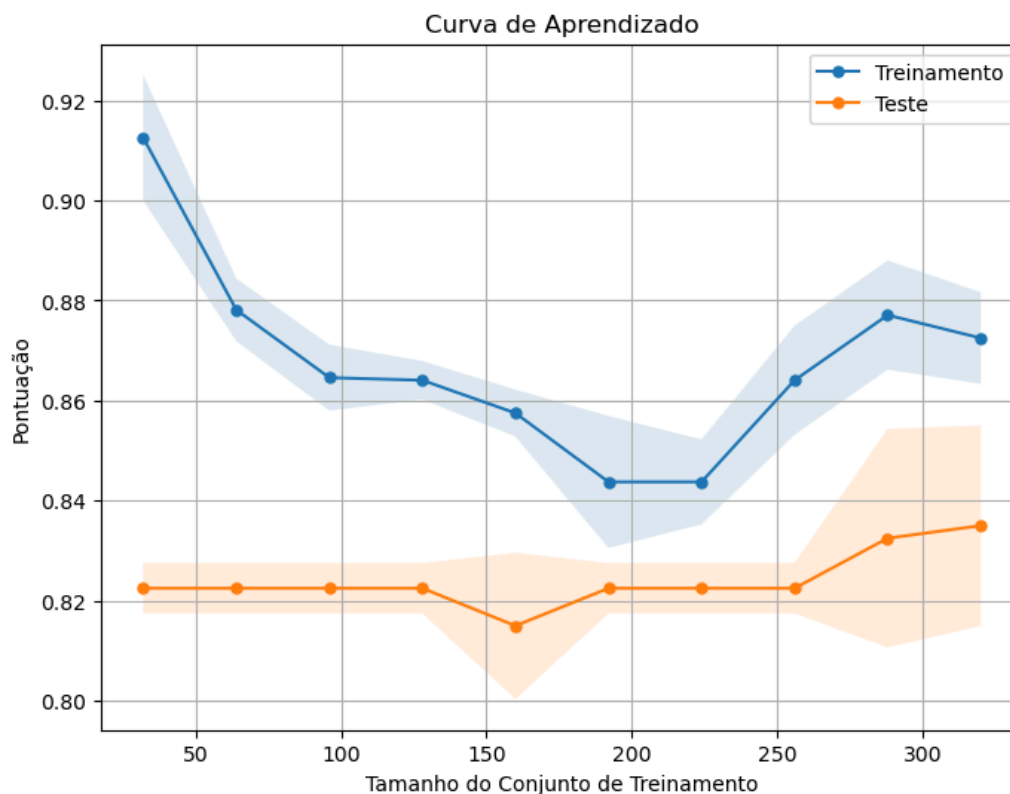


**Fonte: Autor.**

Na Gráfico 4 já observa-se um comportamento mais generalizado do modelo, as curvas de aprendizado de treinamento e teste variam e tendem a se aproximar à medida que o tamanho do conjunto de treinamento cresce.

Apesar de obter medidas finais de acurácia e sensibilidade inferiores ao modelo pré-ajustado (Tabela 15), isso se traduz numa melhor generalização do modelo aos dados. Isso pode resultar em um melhor comportamento ao ser implementado em máquinas em operação para testes.

**Gráfico 4 – Curvas de aprendizado do modelo treinado com ajustes.**



Fonte: Autor.

**Tabela 15 – Medidas de desempenho do modelo após ajustes.**

| Métrica   | Após o ajuste |
|-----------|---------------|
| Acurácia  | 89,90%        |
| Precisão  | 84,68%        |
| Revocação | 96,91%        |
| ROC AUC   | 98,10%        |
| F1 Score  | 90,38%        |

Fonte: Autor.

Ao repetir a validação cruzada com os ajustes, obtém-se um cenário de valores mais constantes durante as primeiras sete iterações, onde o modelo tem um desempenho baixo (inferior a 50%) em sensibilidade e precisão. Contudo, as

iterações oito e nove alcançam os melhores resultados, com métricas, em sua maioria, acima de 80%.

Isso pode indicar que, apesar das iterações iniciais não funcionarem bem, há uma melhoria na resposta do modelo, que levou a iterações significativamente melhores, o que condiz com o comportamento observado na curva de aprendizagem.

Contudo, ainda seria tem-se oportunidade para realizar análises e testes mais profundos para entender o que está acontecendo e se essas melhorias são consistentes.

**Tabela 16 – Resultado da validação cruzada após ajustes no modelo**

| Iteração | Tempo gasto | Precisão | Sensibilidade/ |          |
|----------|-------------|----------|----------------|----------|
|          |             |          | Revocação      | Acurácia |
| 1        | 0,0140      | 41,25%   | 50,00%         | 82,50%   |
| 2        | 0,0140      | 41,25%   | 50,00%         | 82,50%   |
| 3        | 0,0150      | 41,25%   | 50,00%         | 82,50%   |
| 4        | 0,0120      | 41,25%   | 50,00%         | 82,50%   |
| 5        | 0,0110      | 41,25%   | 50,00%         | 82,50%   |
| 6        | 0,0120      | 41,25%   | 50,00%         | 82,50%   |
| 7        | 0,0110      | 41,25%   | 50,00%         | 82,50%   |
| 8        | 0,0110      | 95,83%   | 78,57%         | 92,50%   |
| 9        | 0,0120      | 97,14%   | 85,71%         | 95,00%   |
| 10       | 0,0120      | 91,03%   | 56,25%         | 82,50%   |

**Fonte: Autor.**

## 5 CONCLUSÃO

O experimento mostrou-se eficiente na automatização do processo de análise de dados e tomada de decisão, para a classificação normativa da qualidade da água, evidenciando o potencial da aplicação de modelagem de aprendizado de máquina no auxílio no controle de qualidade e na regulamentação de processos industriais.

O modelo em sua primeira versão resultou em um cenário de sobreajuste (*overfitting*), onde a curva de aprendizado estava constante em um cenário *perfeito* - igual a 100%. Caso este modelo fosse aplicado em cenários reais, sua incapacidade de generalizar os dados resultaria em falsos alarmes do sistema e classificações incorretas, que poderiam prejudicar a operação.

Após os ajustes, o segundo modelo gerado já foi mais capaz de generalizar, e, apesar de ter uma menor acurácia, pelos resultados obtidos, entende-se que isso se torna uma oportunidade de melhoria. Pois o modelo já consegue ser testado em ambientes operacionais e ter um desempenho razoável, podendo gerar insumos para ser aperfeiçoado.

Um cenário ideal seria testar o modelo em diferentes caldeiras em operação, armazenar os dados analisados e os resultados de classificação, para que pudessem ser avaliados e serem base para um novo treinamento do modelo.

Ainda há possibilidade de entender profundamente as análises de validação cruzada, com um número mais adequado de iterações ou aplicando outros tipos de análise de erro, a fim de encontrar as razões que levaram o modelo ajustado a ter um baixo desempenho em suas primeiras iterações.

Ao identificar esses problemas, podem ser aplicadas penalidades ao modelo, para evitar repetir estes cenários. Além também, de poder apontar para falhas no pré-processamento dos dados.

Um importante ponto é que os dados serem gerados de forma pseudo-aleatório é um fator limitante para o algoritmo, e este pode ser o principal fator para ter ocorrido o *overfitting* inicial. Logo, obter novos dados de operações industriais é indispensável para a evolução deste estudo.

Uma evolução deste trabalho seria explorar outros algoritmos, com intuito de retornar sugestões de tratativas para melhorar a qualidade da água, para a caldeira operar em um cenário idealmente recomendado. Dessa forma, o responsável

técnico pela caldeira teria não apenas a informação de conformidade, como também um auxílio com relação a quais e como os parâmetros deveriam ser regulados.

Vale enfatizar ainda que são necessários testes mais rigorosos, principalmente em cenários reais, para verificar se as hipóteses levantadas são válidas. Para classificação de qualidade de água de caldeira, a automatização pode ser integrada a sistemas de controle, dado que ele exige pouco recurso computacional para ser executado.

Também podem ser pensadas outras formas de disponibilizar o modelo, como em uma *Application Programming Interface* (API), que facilitaria não só o compartilhamento do recurso entre distintos sistemas, como também facilitaria a captação de dados de diferentes máquinas para evolução do algoritmo.

Outra oportunidade seria utilizar modelos de aprendizado por reforço, onde o algoritmo poderia aprender durante seu funcionamento, a partir da resposta de retorno do usuário ou do sistema sobre suas previsões.

## REFERÊNCIAS

ASME, American Society of Mechanical Engineers. Guidelines for Water Quality in Modern Industrial Water Tube Boilers for Reliable Continuous Operation. New York: ASME Press, 2011.

BRASIL. Ministério do Trabalho e Emprego. Norma Regulamentadora 13: Caldeiras e Vasos de Pressão. Brasil: 2022.

BRASIL. Resolução ANA nº 77, de 1º de junho de 2021. [S.l.: s.n.], 2021. Diário Oficial [da] República Federativa do Brasil, Poder Executivo, Brasília, DF, 25 abr.

BOTELHO, Manoel H. C.; BIFANO, Hercules M. Operação de caldeiras - gerenciamento, controle e manutenção. 2ª ed. São Paulo: Edgard Blücher, 2018.

EPE, Empresa de Pesquisa Energética. Balanço Energético Nacional 2022. EPE, Rio de Janeiro, 2022.

GÉRON, Aurélien. Mãos à Obra: Aprendizado de Máquina com Scikit-Learn, Keras & TensorFlow. 2ª ed. Califórnia: O'Reilly, 2019. Traduzido por Cibelle Ravaglia - Rio de Janeiro: Alta Books, 2021.

HARRISON, Matt. Machine learning – guia de referência rápida: trabalhando com dados estruturados em Python. Califórnia, EUA: O'Reilly, 2019. Traduzido por Lúcia A. Kinoshita - São Paulo: Novatec, 2020.

HOBOLD, Gustavo M.; DA SILVA, Alexandre K. Machine learning classification of boiling regimes with low speed, direct and indirect visualization. International Journal of Heat and Mass Transfer, v. 171, p. 121166, Jan. 2021.

IBP, Instituto Brasileiro de Petróleo e Gás. Inspeção de caldeiras. 3. ed. Rio de Janeiro: IBP, 2020.

LEMAITRE, G.; NOGUEIRA, F.; ARIDAS, C. K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. Journal of Machine Learning Research, v. 18, n. 17, p. 1–5, 2017.

LIMA, Ellen Lima de. Módulo de sensores para monitoramento da qualidade da água com transmissão sem fio utilizando plataforma de prototipagem. Dissertação (Mestrado em Computação Aplicada). Universidade Estadual de Feira de Santana, Feira de Santana, 2018.

LORA, Electo E. S.; NASCIMENTO, Marco A. R. Geração Termelétrica - Planejamento, Projeto e Operação: Volume 1. Rio de Janeiro: Interciência, 2004. 2 volumes.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, v. 12, p. 2825-2830, 2011.

TOLMASQUIM, Mauricio Tiomno. Energia Termelétrica: Gás Natural, Biomassa, Carvão, Nuclear. Rio de Janeiro: Empresa de Pesquisa Energética - EPE, 2016.

ZARPELON, Willian; AZZOLINI, José Carlos. Caldeiras de alta pressão: caracterização e avaliação da qualidade do tratamento das águas de abastecimento. Unoesc & Ciência - ACET Joaçaba, v. 6, n. 2, p. 141–154, 2015.