UNIVERSIDADE FEDERAL DO PAMPA

LUCAS RIBEIRO LOPES LEITES

ESTUDO DA APLICAÇÃO DE TÉCNICAS DE ANÁLISE DE DADOS E *MACHINE LEARNING* EM UMA INDÚSTRIA DE ARROZ PARBOILIZADO: MANUTENÇÃO PREDITIVA E ESTUDO DE FALHAS EM UMA CALDEIRA

Bagé
2023

LUCAS RIBEIRO LOPES LEITES

ESTUDO DA APLICAÇÃO DE TÉCNICAS DE ANÁLISE DE DADOS E *MACHINE LEARNING* EM UMA INDÚSTRIA DE ARROZ PARBOILIZADO: MANUTENÇÃO PREDITIVA E ESTUDO DE FALHAS EM UMA CALDEIRA

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Química da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Bacharel em Engenharia Química.

Orientador: Alexandre Denes Arruda

Bagé
2023

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais).

L533e Leites, Lucas Ribeiro Lopes
ESTUDO DA APLICAÇÃO DE TÉCNICAS DE ANÁLISE DE DADOS E MACHINE LEARNING EM UMA INDÚSTRIA DE ARROZ PARBOILIZADO: MANUTENÇÃO PREDITIVA E ESTUDO DE FALHAS EM UMA CALDEIRA / Lucas Ribeiro Lopes Leites.
116 p.

Trabalho de Conclusão de Curso(Graduação)--
Universidade Federal do Pampa, ENGENHARIA QUÍMICA, 2023.
"Orientação: Alexandre Denes Arruda".

1. Análise de Dados. 2. Machine Learning. 3. Caldeira. 4. Indústria de arroz. 5. Manutenção preditiva. I. Título.

SERVIÇO PÚBLICO FEDERAL

MINISTÉRIO DA EDUCAÇÃO

Universidade Federal do Pampa

LUCAS RIBEIRO LOPES LEITES

ESTUDO DA APLICAÇÃO DE TÉCNICAS DE ANÁLISE DE DADOS E MACHINE LEARNING EM UMA INDÚSTRIA DE ARROZ PARBOILIZADO: MANUTENÇÃO PREDITIVA E ESTUDO DE FALHAS EM UMA CALDEIRA

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Química da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia Química.

Trabalho de Conclusão de Curso defendido e aprovado em 10 de fevereiro de 2023.

Banca examinadora:

_____

Prof. Dr. Alexandre Denes Arruda

Orientador

(UNIPAMPA)

_____

Prof. Dr. Rodolfo Rodrigues

(UFSM)

Dra. Andressa Apio

(CEO Latos)

Assinado eletronicamente por **Andressa Apio**, **Usuário Externo**, em 10/02/2023, às 19:54, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **Rodolfo Rodrigues**, **Usuário Externo**, em 10/02/2023, às 19:59, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **ALEXANDRE DENES ARRUDA**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 10/02/2023, às 20:06, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1047402** e o código CRC **249B1131**.

Referência: Processo nº 23100.002091/2023-90 SEI nº 1047402

Dedico este trabalho a minha mãe, Rita, meu maior exemplo de coragem e força que não está mais aqui.

AGRADECIMENTO

A minha família, em especial minha avó Edy, meu tio Paulo e meu irmão Márcio, pelo carinho, amor e apoio prestados durante toda minha jornada.

A minha namorada, Evelyn, pelo companheirismo, amor e apoio.

A Universidade Federal do Pampa, pelo ensino e infraestrutura de qualidade.

Aos meus amigos e colegas de Engenharia Química João, Rayan e Thays, por tornarem a jornada mais leve.

Ao Professor Doutor Alexandre Denes Arruda, que sempre mostrou disponibilidade, apoio e amizade.

Aos professores da graduação de Engenharia Química da UNIPAMPA, pelos ensinamentos profissionais e éticos compartilhados.

“A persistência é o caminho do êxito”.

Charles Chaplin

RESUMO

A ascensão da Indústria 4.0 e suas tecnologias junto ao *Big Data* trazem consigo diversas oportunidades para a indústria. Aliados a isso, a Análise de Dados e a *Machine Learning* utilizam desses grandes volumes de dados gerados para auxiliar nas tomadas de decisão. Apesar da ascendência da Indústria 4.0, a realidade brasileira mostra que a maioria das indústrias não estão prontas para essa mudança e são caracterizadas em sua maioria como Indústrias 2.0. Nesse sentido, uma indústria de beneficiamento de arroz apresentou-se promissora para fins de estudo devido suas diversas áreas e geração de dados, por isso, o objetivo do presente trabalho é o estudo da aplicação de técnicas de Análise de Dados e *Machine Learning* através da linguagem de programação Python em uma indústria de beneficiamento de arroz na Região da Campanha que possui características de uma Indústria 3.0. A área selecionada como objeto de estudo para aplicações das técnicas mencionadas é a de uma caldeira mista que apresenta coleta de dados pela própria empresa de hora em hora através de anotações em planilhas impressas e uma recorrência alta de falhas. A partir disso, os dados são digitalizados através do Microsoft Excel e estudados através da plataforma Jupyter Notebook. As técnicas de Análise de Dados e *Machine Learning* possuem finalidade de efetuar um estudo de falhas da caldeira e posteriormente uma classificação dessas falhas, somado a isso, efetuar um estudo de manutenção preditiva através da predição da vida útil remanescente da caldeira. Para isso, é utilizado Regressão Linear, XGBoost Regressor e Florestas Aleatórias Classificador e Regressor. Os resultados obtidos neste estudo mostram a validade de trabalhar com Análise de Dados em uma Indústria 3.0 para compreender o processo da caldeira, suas falhas e o comportamento das variáveis de processo perante a algumas das falhas apresentadas. Em contrapartida, acredita-se que as adversidades encontradas nos dados fazem com que os algoritmos de *Machine Learning* para classificação das condições de falhas e predição da vida útil remanescente não obterem bons resultados.

Palavras-Chave: Análise de Dados; *Machine Learning*. Caldeira. Indústria de arroz; Indústria 4.0. Manutenção preditiva. Estudo de falhas.

ABSTRACT

The rise of Industry 4.0 and its technologies along with Big Data bring with them many opportunities for industry. Allied to this, Data Analysis and Machine Learning use these large volumes of data generated to assist in decision making. Despite the ascendancy of Industry 4.0, the Brazilian reality shows that most industries are not ready for this change and are mostly characterized as Industry 2.0. In this sense, a rice processing industry presented itself as promising for study purposes due to its several areas and data generation, therefore, the objective of this paper is to study the application of Data Analysis and Machine Learning techniques through Python programming language in a rice processing industry in Campanha Region that has characteristics of an Industry 3.0. The area selected as the object of study for the application of the mentioned techniques is a mixed boiler that presents data collection by the company itself every hour through annotations on printed spreadsheets and a high recurrence of failures. From this, the data is digitized through Microsoft Excel and studied using the Jupyter Notebook platform. The Data Analysis and Machine Learning techniques are intended to perform a study of boiler failures and then a classification of these failures, in addition to this, to perform a predictive maintenance study by predicting the remaining life of the boiler. For this, Linear Regression, XGBoost Regressor and Random Forests Classifier and Regressor are used. The results obtained in this study show the validity of working with Data Analysis in an Industry 3.0 to understand the boiler process, its failures and the behavior of process variables in the face of some of the faults presented. On the other hand, it is believed that the adversities found in the data make the Machine Learning algorithms for classification of fault conditions and prediction of remaining useful life do not obtain good results.

Keywords: Data Analysis; Machine Learning; boiler; rice industry; predictive maintenance; failure study.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

PCA – *Principal Component Analysis*

CPS - Sistemas Ciber Físicos

CSV – Valores separados por vírgula

CV – Validação Cruzada

ERP – Planejamento de Recursos Empresariais

FN – Falsos Negativos

FP – Falsos Positivos

IA – Inteligência Artificial

IoS – Internet de Serviços

IoT – Internet das Coisas

MAE – Erro Absoluto Médio

MES – Sistema de Execução da Produção

ML – *Machine Learning*

MSE – Erro Quadrático Médio

RMSE – Raiz do Erro Quadrático Médio

VN – Verdadeiros Negativos

PTV_pca – Coluna da pressão, temperatura, vazão de vapor formados pelo PCA

VP – Verdadeiros Positivos

VUR – Vida útil remanesce

LISTA DE SÍMBOLOS

i – Índice

n – Número de pontos

R^2 – R-quadrado / coeficiente de determinação

S_i – Valor predito

s_i – Valor real

x_i – Variável independente

y_i – Variável dependente

β_0 – Coeficiente linear

β_i – Coeficiente angular

ε_i – Erro

d_i – Dado de posição “i”

μ – Média aritmética

σ – Desvio padrão

q_3 – Terceiro quartil

q_1 – Primeiro quartil

A_{iq} – Intervalo interquartil

lim_{sup} – Limite superior

lim_{inf} – Limite inferior

$taf_{max,ciclo}$ – tempo até falha máximo relativo ao ciclo

taf_i – tempo até falha da posição “i”

SUMÁRIO

1 INTRODUÇÃO

O setor industrial está em constante evolução, com cada vez mais sistemas tecnológicos dinâmicos e complexos que permitem desenvolver fábricas mais inteligentes, onde máquinas, módulos de produção, sensores, controladores e produtos estão conectados através de Sistemas Ciber Físicos, Internet das Coisas e Fabricação em Nuvem (PEREIRA; ROMERO, 2017). Neste meio, a Indústria 4.0 emerge, onde sistemas virtuais e físicos de produção colaboram de forma integral e flexível, permitindo a customização de produtos e fabricação de novos formatos operacionais (SCHWAB, 2016).

O avanço tecnológico promovido pela Indústria 4.0 aliados a era digital, faz com que diversos equipamentos e sistemas como celulares, sensores, ERP (*Enterprise Resource Planning* – Planejamento de Recursos Empresariais), *web sites*, satélites, entre outros, produza e colete uma quantidade gigantesca de dados em uma velocidade e variedade muito grande, criando o que se chama "*Big Data"* (TAURION, 2013). Através disso, buscar métodos para manipular, explorar e transformar esses dados em informações úteis se torna fundamental em um mundo cada vez mais conectado, onde tomadas de decisões rápidas são cada vez mais exigidas (SCHWAB, 2016). Sendo assim, surgem técnicas como a Análise Exploratória de Dados e a *Machine Learning* como aliadas para realizar essas funções.

A Análise Exploratória de Dados utiliza de uma série de métodos gráficos e estatísticos que findam encontrar tendências, semelhanças, correlações, agrupamentos, entre outros para transformar os dados em conhecimentos úteis e aplicáveis (LOPES *et al*, 2019). Aliado a isso, a *Machine Learning* de Aprendizado Supervisionado, área anexa a IA (Inteligência Artificial), fundamenta seu aprendizado através de dados históricos, onde a partir deles, extrai informações para prever ou classificar alguma situação (MONARD; BARANAUSKAS, 2003).

Na indústria, as oportunidades de utilização dessa ferramenta são inúmeras, buscar possíveis falhas em sistemas ou equipamentos, realizar previsões de quando efetuar uma manutenção preditiva, prever demandas de produtos, planejamento logístico, entre outros, são alguns exemplos de oportunidades de aplicação da *Machine Learning* no setor industrial (SUGAHARA, 2020). Nesse sentido, a indústria de beneficiamento de arroz abre portas para diversas aplicações, visando que seu processo seja composto por diversas etapas, onde cada uma delas gera uma

quantidade significativa de dados, formando assim, oportunidades de estudar e analisar seu processo através de técnicas de Análise de Dados e *Machine Learning*.

Apesar da ascensão da Indústria 4.0 no mundo, a realidade brasileira dentro da indústria ainda é diferente, visto que as empresas brasileiras ainda não estão prontas para essa mudança e ainda possuem um caminho pela frente para atingirem essa mudança, além da falta de capital humano (SANTA RITA, 2019). Reforçando esse pensamento, pesquisas realizadas alegam que as indústrias brasileiras ainda estão em um processo de familiarização com os efeitos da digitalização e que a pouca utilização de tecnologias digitais no Brasil reflete negativamente no cenário global (CONFEDERAÇÃO NACIONAL DA INDÚSTRIA, 2016). Em vista disso, grande parte das indústrias brasileiras estão englobadas na classe de Indústria 3.0, que são indústrias que apresentam robôs e equipamentos mais sofisticados, que resultam na redução do trabalho manual, aumento da escala de produção e maior eficiência na indústria, características essas que pertencem a automação industrial (PASQUINI, 2020). No entanto, apesar da tecnologia avançada, não apresentam tecnologias como a Internet das Coisas, coleta de dados com armazenamento em nuvem e uso da IA, que são características da Indústria 4.0.

Fundamentado nos conceitos apresentado anteriormente, o presente Trabalho de Conclusão de Curso tem como objetivo o estudo da aplicação de técnicas de Análise de Dados e *Machine Learning* através da linguagem de programação Python em uma indústria de beneficiamento de arroz na Região da Campanha que possui características de uma Indústria 3.0.

Este trabalho está dividido em sete capítulos. O primeiro, resume-se a introdução, onde consta-se a definição do tema em linhas gerais, delimitação do tema, motivação para o estudo e objetivos. Em seguida, encontra-se o objetivo geral e objetivos específicos, delimitados no segundo capítulo. No terceiro capítulo, apresenta-se a revisão bibliográfica, contendo os conceitos que embasaram o desenvolvimento do trabalho. Em sequência, no quarto, expõe-se a metodologia que é utilizada para alcançar os objetivos descritos no segundo capítulo. A seguir, no quinto capítulo, são apresentados os resultados e discussões obtidos no presente trabalho. No sexto capítulo, encontram-se as considerações finais deste trabalho. Por fim, no sétimo, encontram-se sugestões e propostas para estudos futuros.

2 OBJETIVOS

No presente capítulo serão apresentados o objetivo geral e os objetivos específicos deste Trabalho de Conclusão de Curso.

2.1 Objetivo Geral

Estudo da aplicação de técnicas de Análise de Dados e *Machine Learning* através da linguagem de programação Python em uma indústria de beneficiamento de arroz na Região da Campanha com características de um Indústria 3.0.

2.2 Objetivos Específicos

- **i.** Coleta de dados das áreas da indústria de beneficiamento de arroz na Região da Campanha, que são: secagem, caldeira e tratamento de água de processo;
- **ii.** Seleção da melhor área de estudo referente ao item i para aplicação das técnicas de Análise de Dados e *Machine Learning*;
- **iii.** Definir o que será estudado na área selecionada do item ii;
- **iv.** Coleta e inserção dos dados da área selecionada em planilha digital;
- **v.** Tratamento dos dados coletados no ambiente Python referente ao item iv;
- **vi.** Análise Exploratória dos dados tratados visando o estudo das correlações e a melhor compreensão dos dados;
- **vii.** Seleção e estudo da aplicação das técnicas de *Machine Learning* utilizando os dados tratados da área selecionada no item ii;
- **viii.** Avaliação de desempenho dos algoritmos através de métricas para modelos de *Machine Learning*;
- **ix.** Discutir a viabilidade de aplicar técnicas de Análise de Dados e *Machine Learning* em uma Indústria 3.0 de acordo com a experiência vivenciada.

3 REVISÃO BIBLIOGRÁFICA

Neste capítulo, será exposto a revisão bibliográfica abordando os diversos assuntos que compõe o presente trabalho.

3.1 Revoluções Industriais e suas características

Ao longo dos séculos, os processos de fabricação passaram por diversas mudanças, que contribuíram na sua evolução e também da sociedade. Quando abordado esse tópico, as Revoluções Industriais tornam-se protagonistas durante esse processo de transformação (PASQUINI, 2011).

A Primeira Revolução Industrial, ocorrida na Inglaterra durante o século XVIII protagonizou mudanças significativas na humanidade, como: surgimento de máquinas a vapor, substituindo processos que até então, eram artesanais e manufaturados; o desenvolvimento do telégrafo; separação e especialização do trabalho (PASQUINI, 2020 *apud* DECICINO, 2011).

Em meados do século XIX e XX, principalmente nos Estados Unidos, a Segunda Revolução Industrial trouxe avanços comparada a primeira, como o surgimento do Fordismo, que introduziu linhas de montagem de produção em massa e otimização do trabalho, além da fabricação do aço e a descoberta da eletricidade, fomentando a criação de novos motores, materiais e do rádio (SOUSA, 2016).

Na década de 1970, iniciou-se a Terceira Revolução Industrial, trazendo diversas mudanças para a indústria e sociedade. Entre elas, a globalização, que segundo Medeiros e Rocha (2004), uma de suas consequências mais visíveis são as novas tecnologias, o desemprego e as novas formas de organização do trabalho. Visando o setor tecnológico, algumas das principais inovações foram a telefonia celular, criação de robôs utilizados na indústria, desenvolvimento biotecnológico, foguetes de longo alcance e uso da energia atômica. Através dos robôs e máquinas mais sofisticadas a indústria diminuiu o número de trabalhos manuais e permitiu o aumento da escala de produção e maior eficiência nos processos industriais, caracterizando a automação dos processos industriais (PASQUINI, 2020).

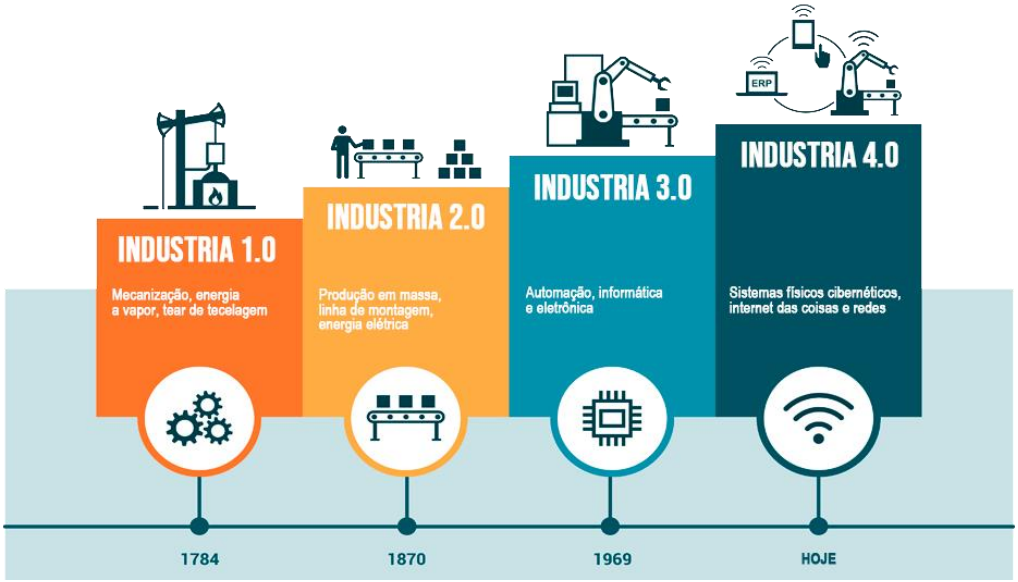
O debate sobre a Indústria 4.0 e uma possível Quarta Revolução Industrial teve início na virada do século XX, mas tomou destaque na Feira de Hannover em 2011 na Alemanha, que abordou seu impacto na organização das cadeias globais de

valores. Para Schwab (2016), a Indústria 4.0 fundamenta-se na revolução digital, que é descrita principalmente pela difusão global da internet, por sensores menores e mais potentes, pela Inteligência Artificial e *Machine Learning*.

O setor industrial será impactado com o complexo sistema tecnológico proposto pela Indústria 4.0. Estas novas ideias colocam a indústria perante a novos paradigmas que englobam um grupo de avanços tecnológicos futuros, tais como os Sistemas Ciber Físicos (CPS), Internet das Coisas (IoT), Internet de Serviços (IoS), Robótica, *Big Data*, Fabricação em Nuvem e Realidade Aumentada. O acolhimento dessas novas tecnologias é essencial para o desenvolvimento de fábricas mais inteligentes, onde módulos de produção, sensores, máquinas e produtos aptos a trocar informações de forma autônoma, são capazes de realizar ações e controlar-se de forma independe, corroborando para um ambiente de manufatura inteligente (PEREIRA; ROMERO, 2017).

A Figura 1 ilustra todas as fases da evolução da indústria e suas principais integrações tecnológicas.

Figura 1 – Fases industriais



Fonte: Adaptado de Yadin (2021)

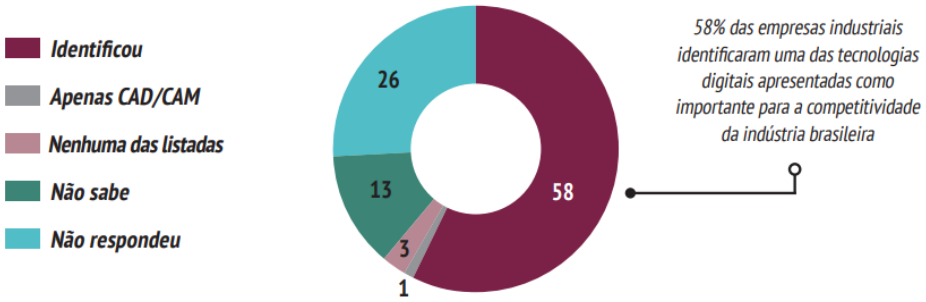
Segundo Santa Rita (2019), através do jornal Correio Braziliense, as empresas brasileiras ainda não estão prontas para a Indústria 4.0 e possuem um grande caminho pela frente para atingirem essa transformação. O artigo cita que segundo o Segundo Secretário Especial do Ministério da Economia em exercício na época, a

maioria das empresas brasileiras estariam ainda em uma Indústria 2.0 e não teriam conhecimento do que é a Indústria 4.0. Além disso, um outro obstáculo que o secretário ressalta é a falta de capital humano para coordenar esse avanço. No entanto, para contrapor esse atraso, foi criado pelo Ministério da Economia junto ao da Ciência, Tecnologia e Inovações a Câmara Brasileira da Indústria 4.0 em 3 de abril de 2019, que objetiva fomentar diálogos entre a academia, indústria e setor público a fim de produzir parcerias público-privadas direcionadas a adesão de tecnologia 4.0 nas indústrias brasileiras (BRASIL, [2022?]).

O Estúdio ABC (2017), através da revista brasileira Exame, publicou um artigo online onde cita a fala do professor da Poli-USP Eduardo de Senzi Zancul, que segundo ele, a Indústria 4.0 baseia-se em duas linhas no qual o Brasil ainda precisa evoluir muito: processos integrados que permitem uma produção customizada e produtos inovadores. O artigo também cita algumas indústrias brasileiras que já possuem características de uma 4.0, como a Ambev, que adotou tecnologias no processo de resfriamento de cerveja para diminuir a variação de temperatura e a Volkswagen Brasil, que utiliza simulação 3D para projetar e criar seus produtos, oferecendo a empresa diversas vantagens.

Uma pesquisa realizada pela Confederação Nacional da Indústria (CNI) visou medir o conhecimento de 2225 empresas brasileiras em relação a tecnologias digitais com potencial de alavancar a competitividade da indústria no ano de 2016. Em vista disso, a pesquisa consistiu em mostrar 10 tecnologias potenciais às empresas e questionar se haviam identificado alguma. Vale ressaltar que foram feitas pesquisas com diversas categorias de empresas, sendo as que obtiverem maior desconhecimento foram as pequenas empresas (57%) e as grandes empresas com menor desconhecimento (32%). Além do mais, a pesquisa relacionou que o desconhecimento dessas tecnologias está diretamente ligado ao uso de poucas tecnologias digitais pelas empresas (CONFEDERAÇÃO NACIONAL DA INDÚSTRIA, 2016). A Figura 2 demonstra o parâmetro geral dos resultados em percentuais obtidos pela pesquisa.

Figura 2 – Resultado pesquisa CNI



Fonte: CONFEDERAÇÃO NACIONAL DA INDÚSTRIA (2016)

Segundo a própria Confederação Nacional da Indústria (2016), as indústrias brasileiras ainda estão em um processo de familiarização com os efeitos da digitalização ou da manufatura avançada no que diz a respeito aos setores e os modelos de negócio. Além disso, ressalta que a pouca utilização de tecnologias digitais no Brasil reflete negativamente a capacidade de disputa do país no cenário da economia global. Ademais, dados de pesquisa indicam que apenas 13% das empresas de manufatura passaram por transformação digital em seus processos (HALL, 2020).

3.2 *Big Data*

O avanço da tecnologia e a entrada na era digital fez com que novos conceitos e questões surgissem. As diversas fontes de coleta de dados, tais como sensores, ERP, MES (Manufacturing Execution – Sistema de Execução da Produção), mídias sociais, satélites, *web sites*, câmeras, celulares, entre outros, fazem com que o volume, variedade e velocidade de dados coletados diariamente seja gigantesco, gerando o que se denomina “*Big Data”* (TAURION, 2013).

A Figura 2 ilustra cada etapa da estrutura hierárquica dos sistemas de automação.

Figura 3 – Estrutura hierárquica dos sistemas de automação

Fonte: SKA ([2023?])

Apesar dos avanços de tecnologias para sistemas computacionais, tais como *hardwares* e tecnologias da *internet*, a geração de dados possui uma escala muito maior que a capacidade de processamento desses dados por esses sistemas. Por isso, um dos maiores problemas da análise de dados concentra-se em encontrar informações úteis a partir de dados produzidos (TSAI *et al*, 2015).

As oportunidades geradas através do *Big Data* são inúmeras, encontrar formas para usufruir melhor dos dados gerados pode levar governos a encontrar métodos inovadores para beneficiar seus cidadãos, influenciar e otimizar as tomadas de decisões na indústria e auxiliar empresas em seus processos de negócio (SCHWAB, 2016).

3.3 Análise Exploratória de Dados

Em vista da grande geração de volume de dados causada pela era do *Big Data*, o manuseio e análise desses dados de forma inteligente se transforma num dos principais desafios computacionais da atualidade. Para este propósito, a Análise Exploratória de Dados torna-se ideal, visto que objetiva transparecer informações ocultas e desconhecidas dos dados de forma que o analista obtenha uma representação imediata, objetiva e compreensível. Para isso, obter visualizações através de gráficos se faz essencial nesta abordagem, visto a capacidade do cérebro humano de adquirir uma compreensão direta e confiável de tendências, distinções,

agrupamentos, correlações e semelhanças através de imagens, em oposição a um encadeamento de números (LOPES *et al*, 2019).

Em suma, pode-se descrever a Análise Exploratória de dados como um grupo de métodos apropriados para a coleta, exploração, apresentação e interpretação de conjuntos de dados numéricos. Tais métodos os quais, possibilitam a exploração dos dados objetivando encontrar padrões de interesse e a exibição dos dados caracterizados por estes padrões (LOPES *et al*, 2019).

Nesse sentido, há diversos *softwares* direcionados para trabalhar com planilhas e análise de dados. No entanto, utilizar uma linguagem de programação para realizar essas análises ao invés de *softwares* de planilhas possui uma série de benefícios, tais como a automação, reprodutibilidade, acessibilidade e compatibilidade com diversos sistemas operacionais. Somando a isso, utilizar uma linguagem de programação para análise de dados faz com que o analista registre o procedimento realizado na análise através da sequência e lógica de programação utilizada, além de ser possível realizar comentário ao longo do código (CHEN, 2018).

3.4 Padronização de dados

A padronização de dados é uma técnica utilizada na preparação dos dados que tem como principal objetivo colocar todas as colunas (*features)* dos dados em uma só escala. Através disso, a padronização pode impactar positivamente o desempenho dos algoritmos de *Machine Learning* e para alguns, é essencialmente necessária. Essencialmente, a padronização consiste em colocar a média dos dados em 0 e o desvio padrão em 1, após isso, todas colunas estão padronizadas (IPNET, 2021).

A principal técnica de padronização utilizada é o Z-Score, que contabiliza o número de desvios padrões que um ponto de uma coluna está da média dessa coluna, portanto, os pontos com valores próximos a média, também serão próximos a zero. Além do mais, como é uma técnica que centraliza em zero, possui valores positivos e negativos que variam de acordo com o valor do ponto analisado ser ou não maior que a média (DATA SCIENCE, 2020). O modelo de cálculo utilizado para a padronização através do Z-Score é representado pela Equação 1.

$$Z - Score = \frac{x_i - \mu}{\sigma} \quad (1)$$

Na Equação 1 o x_i é o dado de posição (*index) "i*" a ser padronizado, o μ é a média aritmética da coluna e σ o desvio padrão da respectiva coluna.

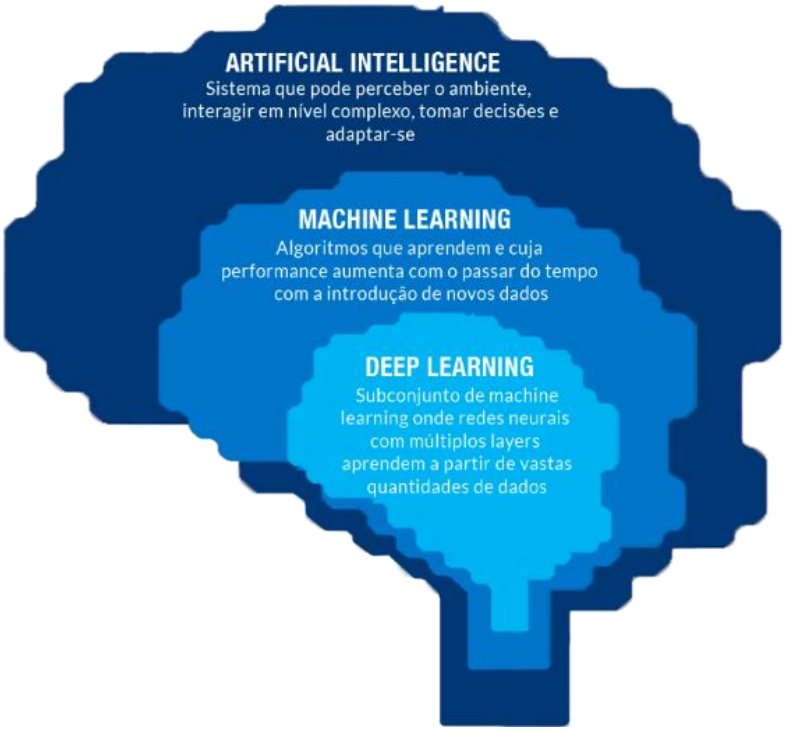
3.5 *Machine Learning*

A Inteligência Artificial (IA) é uma abrangente área da ciência, onde procura-se simular habilidades de raciocínio humano para que o computador execute tarefas sem que seja previamente programado para realizá-las (DICK, 2019). Visto isso, pode-se dizer que o principal objetivo da IA é a criação de sistemas para realizar tarefas que, momentaneamente, são melhores executadas por humanos do que por máquinas, ou que ainda não existe solução algorítmica acessível com a tecnologia computacional atual (SICHMAN, 2021 *appud* RICH; KNIGHT, 1991).

Anexo às áreas da IA, a *Machine Learning,* ou Aprendizado de Máquina, tem como objetivo desenvolver métodos computacionais que visam o aprendizado e criação de sistemas com capacidade de obter conhecimento de forma automática. Esses sistemas de aprendizagem são programas que adquirem conhecimento para a tomada de decisão com base em histórico de dados conhecidos ou experiências acumuladas a partir de resoluções de problemas anteriores bem-sucedidas (MONARD; BARANAUSKAS, 2003).

A Figura 4 representa bem as áreas e subáreas da inteligência artificial, esclarecendo melhor os conceitos citados acima.

Figura 4 – Áreas e subáreas da inteligência artificial



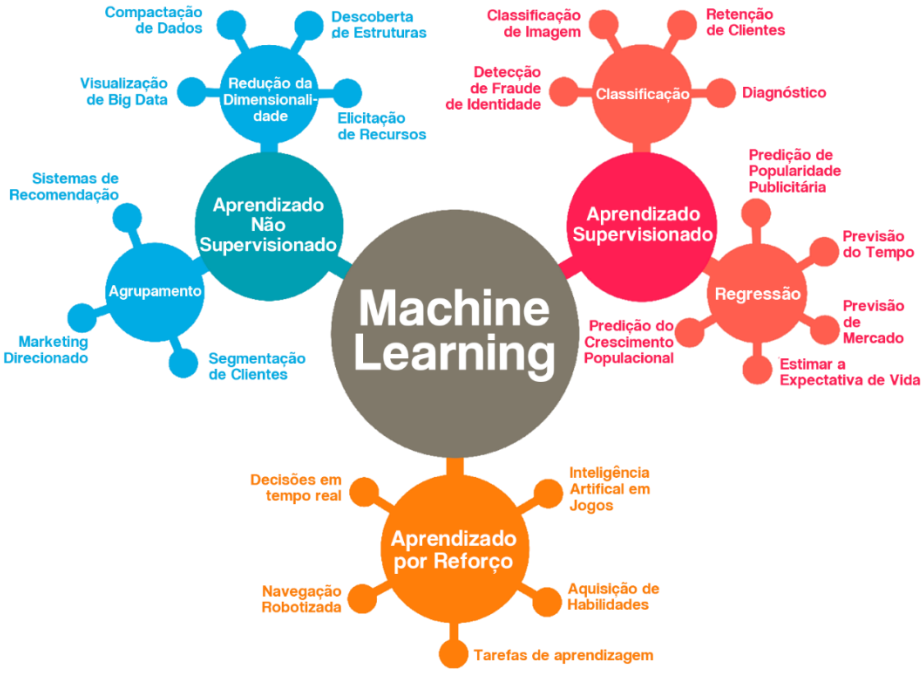
Fonte: Adaptado de Nascimento (2017)

O Aprendizado de Máquina é realizado através do processo de indução, que segundo Ribeiro (2019) é o raciocínio que vai do particular ao geral, de maneira a postular algo que represente o todo e, por conta disso, a indução desempenha papel fundamental nas ciências experimentais. A partir dessa definição, pode-se dizer que, o conjunto de informações induzidas no sistema de aprendizagem é dado como verdade e, a partir disso, um conceito de aprendizagem é gerado. Consequentemente, as hipóteses construídas a partir dessa suposição podem ser ou não verdadeiras. Por conta disso, a escolha do conjunto e exemplos a serem induzidos devem ser suficientes para representar o todo, caso contrário, a pressuposição estimada pode ter pouco valor (MONARD; BARANAUSKAS, 2003).

O treinamento e avaliação do algoritmo da *Machine Learning* é efetuado pela indução de dados divididos em duas categorias: dados de treinamento e dados de teste. Os dados de treinamento serão o conjunto de informações que ensinará um conceito para o programa e os dados de teste serão utilizados para verificar a veracidade do modelo a partir de métricas (SILVEIRA; BULLOCK, 2017).

No geral, classifica-se a *Machine Learning* em três categorias de aprendizado: Aprendizado Supervisionado, Aprendizado não Supervisionado e Aprendizado por Reforço. Na Figura 5, pode-se ver as categorias de aprendizado e suas aplicações.

Figura 5 – Aprendizado supervisionado e suas aplicações



Fonte: Adaptado de Lorberfeld (2019)

3.6 Aprendizado supervisionado

A aprendizagem supervisionada é a categoria de aprendizado mais utilizada no treinamento de algoritmos de *Machine Learning*. O aprendizado consiste em mapear uma função hipotética utilizando como referência uma base de dados rotulada com pares de entrada e saída, o qual, a partir dessa função, poderá mapear uma predição futura ou de situações invisíveis durante o treinamento (HADRI, 2021).

O conjunto de treinamento pode ter qualquer valor de entrada e saída, não se limitando a apenas números, mas também pode conter palavras, imagens, entre outros. Quando as saídas forem de valores finitos (por exemplo, rótulos ou números de 0 a 10), o obstáculo do aprendizado será de classificação, podendo ser chamado de classificação binária, se houver apenas dois valores, ou booleana, com mais de dois. Em contrapartida, quando a saída for um valor numérico, como temperatura e concentração, a problemática da aprendizagem é chamada de regressão e suas predições são uma aproximação do valor verdadeiro (RUSSEL; NORVIG, 2013).

3.7 Seleção de *features*

Os conceitos em relação ao Aprendizado de Máquina estão relacionados justamente ao “aprender”, no caso como o algoritmo compreende as informações

passadas a ele, ou seja, o conhecimento extraído pelo algoritmo através dos dados. Portanto, quanto mais significante são os *features* passados ao algoritmo, maior o nível de aprendizado do algoritmo e mais confiável ele se torna. Por isso, realizar a seleção de *feature* é uma etapa tão importante (LEE, 2000).

Em Python, um dos métodos utilizados para seleção de *features* chama-se *Recursive Feature Elimination* (RFE), ou Eliminação Recursiva de Recursos, onde o algoritmo de *Machine Learning* desejado é treinado com o conjunto de *features* iniciais e partir disso, uma pontuação é fornecida pelo próprio método RFE e a *feature* de pior desempenho é eliminada. Em seguida, o processo repete-se até que a quantidade estabelecida de *features* a serem selecionadas seja atingida (SCIKIT-LEARN, 2023).

3.8 Modelo de Regressão Linear

A regressão é um método estatístico que visa modelar relações entre variáveis e prever o resultado de uma ou mais variáveis dependentes em relação a um grupo de variáveis independentes. Em vista disso, a regressão linear é definida como a relação entre uma variável dependente (y) e uma independente (x), podendo ter comportamento crescente ou decrescente. A Equação 2 representa o modelo padrão da regressão linear, onde β_0 é o coeficiente linear, β_i o coeficiente angular e ε_i, sendo $i = 1, \ldots, n$, variáveis aleatórias relacionadas ao erro (RODRIGUES, 2012).

$$y_i = \beta_0 + \beta_i x_i + \varepsilon_i \quad i = 1, \ldots, n \tag{2}$$

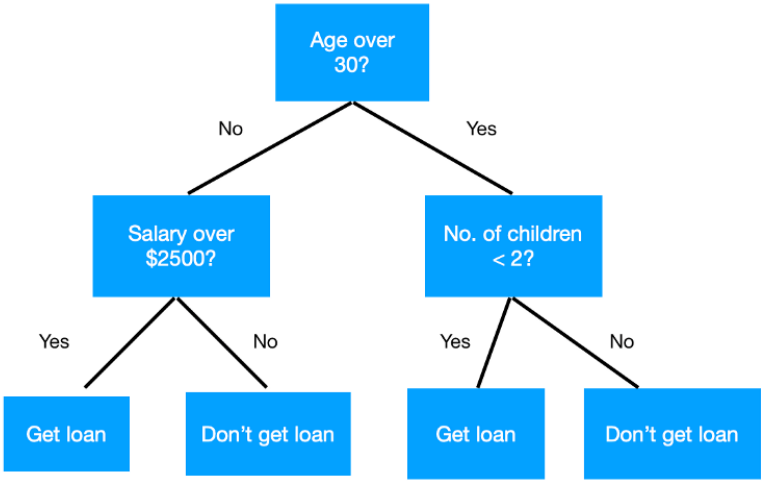
O modelo de ML de Aprendizado Supervisionado Regressão Linear é utilizado para realizar previsões, modelar e avaliar a correlação entre duas variáveis, ou seja, o quão bem uma variável independente x descreve uma dependente y. Essa modelagem é obtida a partir de um ajuste realizado em um gráfico de dispersão relacionando x e y. Junto a isso, para determinar a existência de uma correlação entre duas variáveis, utiliza-se o coeficiente de Pearson, que mede o grau de correlação e a direção da mesma, podendo ser crescente ou decrescente. Os valores atribuídos para esse coeficiente podem ser de -1 a 1, sendo -1 uma correlação decrescente e 1 uma correlação crescente (GOMES, 2019).

3.9 Modelo de Árvores de Decisões

O modelo de Aprendizado Supervisionado Árvore de Decisões pode ser dividido em dois tipos: em árvore de decisão de variável categórica ou contínua. Assim sendo, os de variáveis categóricas utilizados para classificação e os de variáveis contínuas, utilizados para previsões (PEREIRA; MUNIZ; VARGAS, 2020).

A árvore de decisão assemelha-se com a estrutura de um fluxograma no qual cada nó interno representa uma “prova” em um atributo, cada ramo retrata o resultado da prova e cada nó de folha simboliza uma classe ou um valor. Em suma, realiza uma verificação se o atributo cumpre uma condição e, segundo o resultado, determinará qual ramificação posterior será percorrida até alcançar o nó de folha (VISHAL, 2018). Para compreender melhor, a Figura 7 elucida a estrutura da árvore de decisões com os termos comumente utilizados.

Figura 6 – Estrutura do modelo de Árvore de Decisões



Fonte: Borcan (2020)

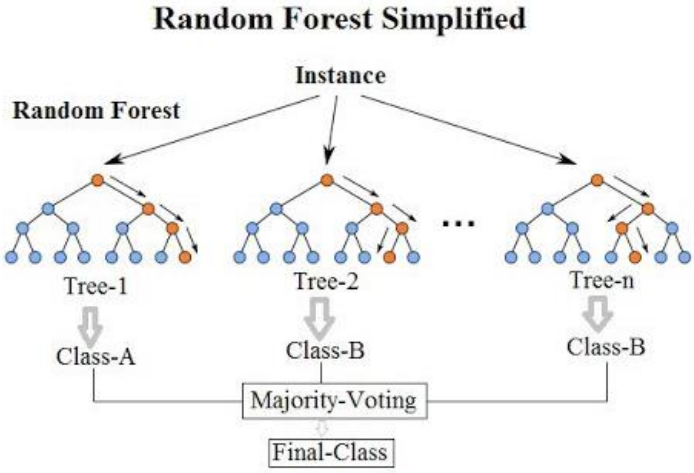
O nó raiz representa todo conjunto de amostras e se divide em dois ou mais conjuntos homogêneos. Essas divisões, são chamadas de subnós de decisão, que também é dividido em dois ou mais nós. Os nós que não se dividem são chamados de nó folha ou terminal (PEREIRA; MUNIZ; VARGAS, 2020).

3.10 Modelo de Florestas Aleatórias

O modelo de *Machine Learning* de Aprendizado Supervisionado Florestas Aleatórias ou Florestas Randômicas é um algoritmo utilizado tanto para classificação, quanto para regressão e é composto por uma associação de árvores de decisão

individuais independentes entre si. Visto isso, cada árvore de decisão que compõe a floresta depende de um subconjunto de características aleatórias e distintas das outras para prever uma classe ou número. Dessa maneira, faz com que cada árvore de decisão realize sua decisão de forma independente das outras e, a partir do conjunto de decisões individuais, uma única decisão e mais precisa será realizada (DATA SCIENCE TEAM, 2020). Para representar isso, a Figura 8 ilustra o formato de raciocínio para tomada de decisão do modelo de Florestas Aleatórias, neste exemplo, para classificação.

Figura 7: Estrutura do modelo de Florestas Aleatórias



Fonte: Webdesign em Foco (2021)

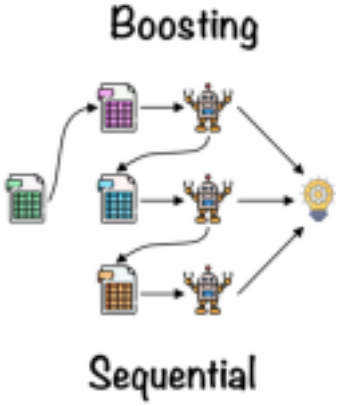
A quantidade de árvores que são criadas é determinada de acordo com um número estipulado, o qual cada árvore define aleatoriamente um subgrupo de características que a árvore utilizará a partir de um grupo total de características. Dessa maneira, torna o método de Florestas Aleatórias muito eficiente, pois ao criar árvores com números reduzidos de características, descarta a chance de que uma árvore seja muito mais forte que outra e influenciando muito a tomada de decisão da floresta, de modo que as árvores criadas sejam distintas entre si e com alto valor de variância (COSTA; PRADO; SILVA; SILVA; ULTIMURA, 2020).

3.11 Modelo de XGBoost

O modelo de aprendizado supervisionado *Extreme Gradient Boosting*, XGBoost, é baseado em árvore de decisão e utiliza a estrutura de *Gradient Boosting*, porém, primeiramente deve-se compreender o que significa um *boosting*. Um

aumento, *boosting*, é uma metodologia para aumentar a performance de um estimador de acordo com as predições já realizadas pelo estimador anterior. Portanto, em termos de uma classificação de árvore de decisão, a primeira árvore é treinada e retira suas próprias conclusões em relação a separação do conjunto de dados. Em seguida, uma segunda árvore é treinada com ponderações de pesos maior para aqueles dados mais difíceis de se classificar, já para os mais fáceis, o peso é diminuído. Esse processo repete-se até o modelo atingir a melhor performance possível, sendo o último modelo as previsões ponderadas de todos os anteriores. A Figura 8 ilustra um exemplo do funcionamento do *boosting* (DATA SCIENCE, 2021).

Figura 8 – Estrutura de um *boosting*



Fonte: Adaptado de Kumar (2022)

O *Gradient Boosting* é baseado na ideia de que o próximo modelo reduzirá os erros de previsão e para isso é calculada uma função de perda (*loss*) para reduzir o erro do próximo modelo (DATA SCIENCE, 2021). O XGBoost é uma forma mais regularizada de *Gradient Boosting*, utilizando a chamada regularização avançada, resultando na melhor da generalização do modelo. Além do mais, oferece um desempenho mais elevado com uma alta velocidade de treinamento (KHANDELWAL, 2020).

3.12 Modelo de PCA

O modelo de aprendizado de máquina não supervisionado *Principal Component Analysis*, ou Análise de Componente Principal, é uma técnica de redução de dimensionalidade que visa selecionar os dados mais representativos através de combinações lineares das variáveis de origem. O PCA realiza a identificação das

dimensões em que os dados se encontram mais dispersos. Onde a partir disso, o algoritmo consegue identificar as dimensões que mais caracterizam o conjunto de dados originários e seleciona as componentes principais (BI4ALL, 2018).

3.13 Métricas de desempenho de *Machine Learning*

Após o treinamento do modelo de *Machine Learning* se faz necessário maneiras de avaliar o desempenho do aprendizado do algoritmo. Para isso, as métricas são parâmetros quantificáveis para medir, explicar, comparar e analisar o desempenho de um determinado processo (D'ANGELO, 2020).

A seguir será apresentado métricas utilizadas para avaliação de desempenho de algoritmos de Aprendizado de Máquina:

i. **R-quadrado:** Coeficiente de determinação, ou r-quadrado, é uma métrica utilizada para regressões que ilustra a variância da variável resposta ou ajuste do modelo aos dados. Essa métrica é quantificada entre 0 e 1, sendo que quanto mais perto de 1, melhor o modelo representa os dados e explica sua variabilidade. Em contraponto, não expõe situações de *overfitting*, o que faz com que necessite de outras métricas para analisar a eficiência do modelo de regressão (MINITAB BLOG EDITOR, 2019). A Equação 3 representa o modelo para determinar o R-quadrado.

$$R^2 = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i y_i}{\sqrt{n\sum_{i=1}^{n} {x_i}^2 - \sum_{i=1}^{n} {x_i}^2}\sqrt{n\sum_{i=1}^{n} {y_i}^2 - \sum_{i=1}^{n} {y_i}^2}} \quad (3)$$

Onde, n é o número de pontos, x_i é o valor da variável independente na posição i e y_i é o valor da variável dependente no índice i.

ii. **Erro Quadrático Médio:** O Erro Quadrático Médio, ou do inglês *Mean Square Error* (MSE), é uma das métricas mais utilizadas para avaliar o desempenho de modelos de regressão e compreender erros de previsão. A métrica apresenta como principal objetivo descobrir a diferença média de um valor estimado e seu valor real e, quanto maior o valor do MSE, menor o desempenho do modelo. Junto a isso, como considera o erro ao quadrado, predições longínquas do valor real aumentam o valor da métrica com

facilidade, tornando o método efetivo para avaliar circunstâncias que maiores erros não são tolerados. Como desvantagem, a métrica é afetada significativamente por *outliers* ou valores nulos, devido sua sensibilidade a erros (AZANK, 2020). O modelo para calcular o MSE é ilustrado pela Equação 4.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(S_i - s_i)^2 \quad (4)$$

Onde, *n* é o número de pontos, S_i é o valor da predição na posição *i* e s_i é o valor real da variável na posição *i*.

iii. Raiz do Erro Quadrático Médio: Do inglês, *Root Mean Squared Error* (RMSE), tem como objetivo facilitar a avaliação da métrica MSE colocando os valores na mesma unidade do valor previsto para ser comparado a outras métricas, como o Erro Absoluto Médio (VARGAS JUNIOR, 2020). A Equação 5 representa o modelo para determinar a RMSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(S_i - s_i)^2} \quad (5)$$

iv. Erro Absoluto Médio: O Erro Absoluto Médio, ou do inglês *Mean Absolute Error* (MAE), junto ao MSE é muito utilizado para avaliar modelos de regressão medindo o erro entre o valor predito e o real. No entanto, como não eleva esse erro ao quadrado, acaba por não levar tanto em consideração erros maiores e, por meio disso, torna-se uma boa opção para modelos que não necessitam tanto de delicadeza e prever uma tendência é mais importante (AZANK, 2020). A Equação 6 representa o modelo para determinar a MAE.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|S_i - s_i| \quad (6)$$

v. **Acurácia:** A Acurácia é a métrica mais simples para testar o desempenho de algoritmos de classificação, que basicamente, representa as previsões corretas do modelo no geral. No entanto, é uma métrica eficiente para classes balanceadas, ou seja, que para todas as classes o modelo esteja prevendo igualmente (SCUDILIO, 2020). O modelo para calcular a Acurácia é ilustrado pela Equação 7.

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{7}$$

Onde, *VP* são os verdadeiros positivos, *VN* os verdadeiros negativos, *FP* os falsos positivos e *FN* os falsos negativos.

vi. **Precisão:** A Precisão é uma métrica eficiente para avaliar o desempenho do algoritmo de classificar apenas uma classe, pois mede quantas predições classificadas como classe 1, realmente são da classe 1 (BITTAR, 2020). A Precisão pode ser calculada através da Equação 8.

$$Precisão = \frac{VP}{VP + FP} \tag{8}$$

vii. **Revocação:** A Revocação, também conhecida como *Recall* e Sensibilidade, é uma métrica para classificação para avaliar o desempenho do algoritmo a respeito de uma classe. Essa métrica mede quantos valores esperados de classe 1, realmente foram classificados como classe 1, por isso, é mais utilizada em situações que falsos negativos se tornam mais prejudiciais que falsos positivos (RODRIGUES, 2019). A Equação 9 ilustra o método para cálculo da Revocação.

$$Revocação = \frac{VP}{VP + FN} \tag{9}$$

viii. **F1-Score:** A F1-Score é uma métrica para avaliar modelos de classificação, utilizada no geral para avaliar conjuntos de dados com classes desproporcionais. Essa métrica realiza a média harmônica entre a Revocação e a Precisão, o que faz com que o valor calculado se aproxime

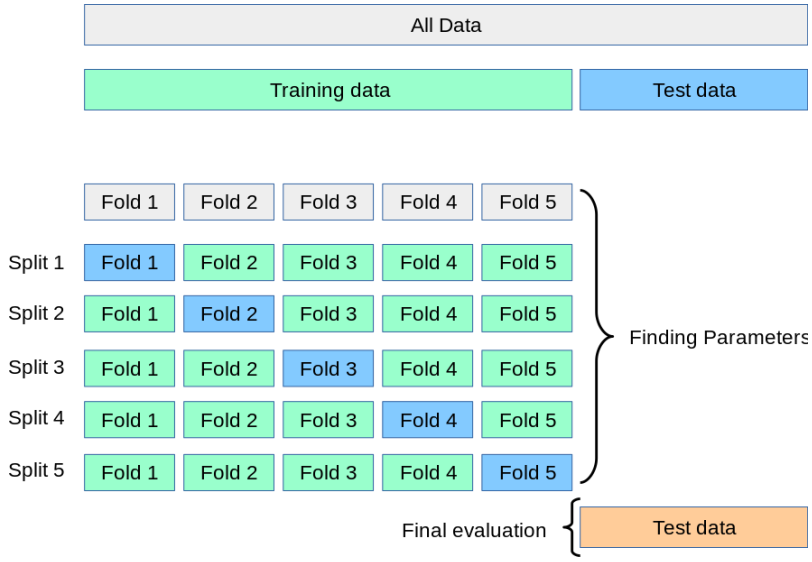
do menor valor entre as duas. Assim dizendo, quando o F1-Score é baixo, visto que seu valor máximo é 1, indica que a Precisão ou a Revocação está baixa. A Equação 10 representa o modelo para cálculo do F1-Score (LEAL, 2017).

$$F1 - Score = \frac{2 * Precisão * Revocação}{Precisão + Revocação} \quad (10)$$

ix. Validação Cruzada: Um modelo que é treinado com os mesmos dados que são testados, terá uma pontuação perfeita, mas isso não significa que será útil para prever dados não conhecidos. Essa situação é denominada de *overfitting* ou sobreposição, que é a não generalização dos dados. Para impedir essa situação, divide-se os dados em um percentual para treino e um para teste. Todavia, ao considerar diferentes hiperparâmetros para modelos de *Machine Learning*, há risco desses hiperparâmetros serem ajustados até o modelo funcionar idealmente de forma que os dados de teste "vazem" no modelo, causando um *overfitting*. Como solução para esse problema, pode-se dividir os dados também em dados para validação, porém, uma terceira divisão nos dados pode reduzir o número de dados consideravelmente a ponto de o modelo não ser treinado de forma eficiente. Para isso, a Validação Cruzada se torna a solução, não sendo mais necessário uma terceira divisão nos dados para validação (SCIKIT-LEARN, [2022?]).

A respeito de *k-fold Cross-Validation* (CV), o conjunto de treinamento é dividido em *k* divisões exclusivas (dobras – *folds*) e é treinado utilizando as dobras de treinamento "*k-1*". Após isso, o modelo é validado com a dobra remanescente. Essa dinâmica será repetida *k* vezes, toda vez validada com uma dobra diferente da anterior (MONARD; BARANAUSKAS, 2003). Para ilustrar melhor o funcionamento da Validação Cruzada, a Figura 9 exemplifica sua utilização com *k* = 5 dobras.

Figura 9 – Estrutura do procedimento da Validação Cruzada



Fonte: Scikit-learn ([2022?])

x. **Matriz de confusão:** A matriz de confusão é um método de avaliação do desempenho utilizado para classificações binárias e booleana que permite a visualização do desempenho do algoritmo em relação aos seus acertos e erros na predição, além de mostrar os seus falsos positivos e falsos negativos (IBM, 2021). Além do mais, é uma ótima aliada para compreender melhor as métricas de Precisão e Revocação.

3.14 Python

O Python é uma linguagem de programação desenvolvida visando a simplicidade na programação e versatilidade, sendo capaz de ser utilizada para inúmeras tarefas, desenvolvimento e criação de diversos programas. Ademais, é uma das linguagens em maior ascensão no mundo e no Brasil, o que resulta na sua valorização e requisição no mercado de trabalho. Exemplos de sites que foram criados ou utilizam Python em sua construção são o Instagram, Spotify, Google, Netflix e Uber (HASHTAG TREINAMENTOS, 2021).

As aplicações do Python são inúmeras, sendo utilizadas em diversos domínios e aplicativos. Algumas das suas utilizações são: Desenvolvimento da Web e da Internet, tais como sites, aplicativos, jogos *web frameworks*, *web servers*, entre outros; Computação Científica e Numérica, por exemplo análise de dados e ciência de dados, possuindo diversas bibliotecas para tal como SciPy, Numpy, Pandas, Scikit-learn, Seaborn entre outras; Educação, por ser uma linguagem de programação intuitiva,

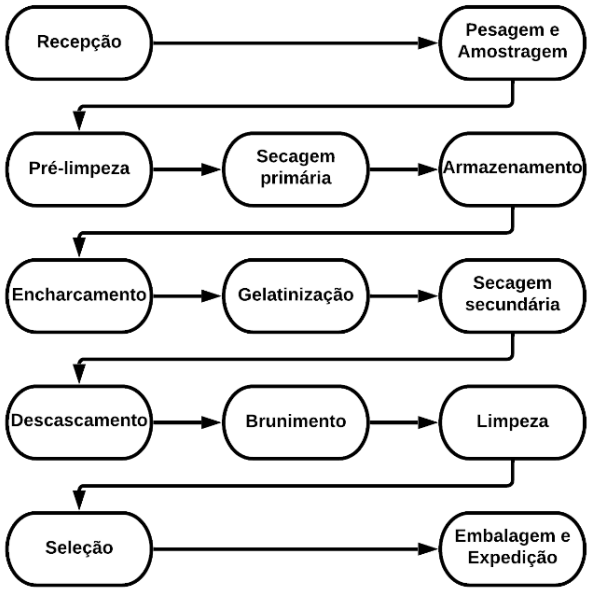
facilita o ensino na programação; Desenvolvimento de Software, tal como controle e gerenciamento de erros, testes, rastreamento de *bugs*, entre outros. Aplicações de Negócios, tais como gerenciamento corporativo, criação de ERP e *e-commerce*, entre outros (PYTHON SOFTWARE FOUNDATION, 2022).

As técnicas de *Machine Learning* podem ser aplicadas em diversas linguagens de programação, no entanto, algumas características diferem o Python de outras, tornando-o a mais promissora linguagem para ML e IA. Primeiramente, por sua clareza em suas sintaxes, torna-se intuitivo e de fácil aprendizado. Como também, os interpretadores em linguagem C e C++, permitem o Python ser interativo e multiplataforma. Também, por ser de código aberto, está em constante atualização, recebendo melhorias, novos recursos e bibliotecas. Além do mais, principalmente pela gama de bibliotecas dedicadas a área de dados e *Machine Learning,* que possibilita os mais diversos processamentos e manipulação de dados, além de diversos modelos de ML (ROJAS, 2020).

3.15 Processo de beneficiamento do arroz

O processo de beneficiamento do arroz é composto por diversas etapas, compreendidas desde sua recepção até sua expedição. A Figura 10 mostra um diagrama de blocos que exemplifica esse processo.

Figura 10 – Diagrama de blocos do beneficiamento do arroz



Fonte: Autor (2023)

i. **Recepção:** O arroz *in natura* chega na indústria de caminhão, onde será coletado dados básicos para a identificação do transportador, informações do fornecedor e verificação da nota fiscal (DA EIRA, 2010)

ii. **Pesagem e Amostragem:** Em seguida, o caminhão é pesado em uma balança, onde após será coletado amostras representativas do arroz para análise em laboratório, os quais os principais parâmetros analisados são a umidade e a pureza. Junto a isso, um teste de rendimento num protótipo do engenho é realizado, classifica-se o arroz e o percentual de grãos existentes. Um teste de secagem com os grãos “verdes” é realizado e após 48 horas, será realizada uma nova análise do arroz. Neste contexto, o produto será enviado para a “caixa de depósito verde” que prossegue para os secadores na produção (SAIDELLES *et al*, 2012).

iii. **Moagem:** Dando continuidade no processo, o caminhão prossegue para a moega, que de acordo com a origem do arroz é decidido o tipo de moega que se destinará o produto (DA EIRA, 2010).

iv. **Pré-limpeza:** Nesta etapa, o arroz é transportado através do elevador de canecas até peneiras para a retirada de impurezas que podem vir a atrapalhar a secagem ou corroborar para proliferação de microrganismos e insetos. Normalmente, nessa etapa o teor de impureza do arroz é reduzido em 2% (BARRIGOSSI, 2019).

v. **Secagem primária:** Caso o arroz chegue no moinho com uma umidade superior a 13% ou 14% é realizado um processo de secagem, que é um processo crucial para o armazenamento do arroz, que preserva sua qualidade e impede o desenvolvimento de fungos no armazenamento. Para a secagem o ar é aquecido por caldeiras, onde será utilizado para reduzir a umidade do grão dentro do secador. Após a secagem, a umidade de saída ideal do grão será entre uma faixa de 12% a 13% (DOS SANTOS, 2020).

vi. **Armazenamento:** O armazenamento é fundamental para a preservação da qualidade do grão de arroz, sendo dependente dos processos anteriores a ele e, quando bem-feito, reduz as perdas de grão durante a armazenagem. Após a secagem, o arroz é transportado para os silos de armazenagem, geralmente, é realizado em silos a granel, onde o grão permanece durante um período de 48h a 72h. Parâmetros importantes de controle de armazenamento podem reduzir as perdas de grãos e evitar o surgimento de

pragas, tais como: controle químico, manipulação de umidade relativa, controles físicos, aeração e temperatura (CHINELATO, 2020).

vii. **Encharcamento:** No encharcamento o arroz com casca é colocado em tanques com água a temperaturas ao redor de 65°C por horas, fazendo com que os sais e vitaminas solubilizem em água e penetre o arroz, o que facilita futuramente o processo de gelatinização. Após o encharcamento, o ideal é que o grão obtenha uma umidade de 30 a 32% (NITZKE; BIEDRZYCKI, 2005).

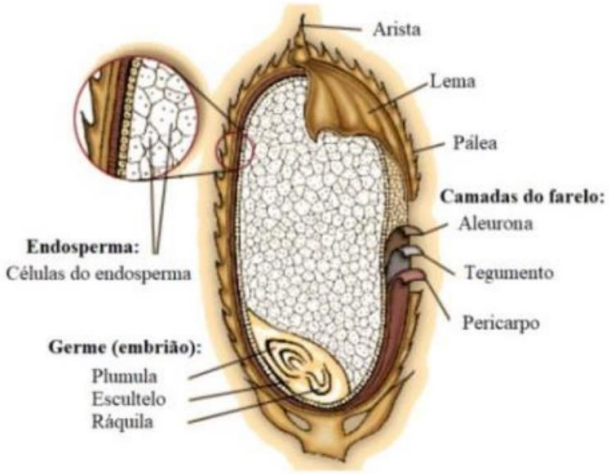
viii. **Gelatinização:** No processo de gelatinização do amido é onde ocorre uma modificação na estrutura do amido, modificando de cristalino para amorfo. Nessa etapa o arroz fica mais compacto e permite que índice de quebra diminua ao longo do processo, característica diferencial econômica do arroz parboilizado. Além disso, nesta etapa os nutrientes, vitaminas e sais minerais, são fixados através de um processamento térmico através de autoclaves utilizando vapor e água (NITZKE; BIEDRZYCKI, 2005).

ix. **Secagem secundária:** A secagem secundária possui parâmetros similares ao da primária e utiliza ar quente para secar os grãos. Porém, normalmente é utilizado dois secadores, um contínuo e outro intermitente. Após a secagem, a umidade de saída ideal do grão será entre uma faixa de 12% a 13% (NITZKE; BIEDRZYCKI, 2005).

x. **Descascamento:** Em seguida, o grão do arroz passa entre um vão de dois roletes de borracha que rotacionam em sentidos opostos e velocidades distintas, descascando o arroz e separando a casca do resto do grão. Nesta etapa do processo, deve-se tomar cuidado principalmente com a umidade, para evitar quebra do grão. Em seguida, o arroz inteiro é separado da casca através da câmara de palha. Na sequência, o arroz descascado é separado através de uma máquina do arroz com casca, chamado de “marinheiro” (BARRIGOSSI, 2019).

xi. **Brunimento:** Após o descasque, o arroz irá para o processo de brunimento, o qual determinará se o arroz sairá em seu formato original ou quebrado. Nesse processo, o arroz entrará em uma máquina que através da abrasão com uma superfície áspera, tem como objetivo remover do arroz a película de tegumento e o germe (LUZ *et al*, 2005). A Figura 11 mostra a estrutura do grão de arroz.

Figura 11 – Estrutura do grão de arroz



Fonte: Montiel (2020)

xii. **Limpeza:** Após o brunimento, o grão é transportado para a limpeza em um "flutuador", onde será removida as impurezas remanescentes, como poeira, palha de arroz, pedras, entre outros. Em sucessão, o arroz entra em um classificador tipo "*trieur*", para separação dos grãos quebrados (SAIDELLES *et al.*, 2012).

xiii. **Seleção:** Nesta etapa, o arroz será selecionado de acordo com sua fragmentação para aumentar seu valor comercial. Junto a isso, remove-se também partículas estranhas que possam ter passado em processos anteriores através de uma série de peneiras. O processo de seleção que distingue o arroz em tipos, classes, grupos e subgrupos (DA EIRA, 2010).

xiv. **Embalagem e Expedição:** O processo de embalagem do arroz é totalmente automatizado, onde o arroz preenche as embalagens contendo as diversas especificações dele, como número do lote, classe, tipo, peso líquido, entre outros. Em seguida o arroz será expedido em fardos através de caminhões (SAIDELLES *et al*, 2012).

3.16 Caldeiras

Segundo Brasil (2006, p. 7), na NR-13: Manual Técnico de Caldeiras e Vasos de Pressão, é considerado caldeira "todos os equipamentos que simultaneamente geram e acumulam vapor de água ou outro fluido", além de poder utilizar qualquer fonte de energia (BRASIL, 2006). Majoritariamente, as caldeiras são utilizadas para

produzir vapor em temperatura e pressão elevadas e podem ser encontradas para suprir necessidades domésticas, como calefação residencial e aquecimento de água, mas principalmente na indústria, para atender as mais diversas necessidades de processos (BOTELHO, 2015).

As caldeiras que produzem vapor através da queima de combustíveis são classificadas de acordo com o posicionamento relativo entre os gases de combustão e a água, que se dividem em dois grupos: caldeiras flamotubulares e aquatubulares (INSTITUTO BRASILEIRO DE PETRÓLEO E GÁS, 2020).

Caldeiras do tipo flamotubular são muito utilizadas, principalmente em situações em que não há necessidade de vapor a altas pressões (até 19 kgf/cm²) ou vazões (até 10 t/h). Possuem um corpo cilíndrico com um feixe de tubos em seu interior e a fornalha em uma de suas extremidades, onde os gases de combustão escoam pelo interior dos tubos, que por sua vez, estão submersos em água contida dentro do corpo da caldeira. Essa classe de caldeira produz somente vapor saturado, sendo inviável produzir vapor superaquecido, também opera a baixas pressões, possui maior tolerância a águas não tratadas e tem baixo custo (INSTITUTO BRASILEIRO DE PETRÓLEO E GÁS, 2020).

As caldeiras da classe aquatubular são utilizadas quando há necessidade de alta produção de vapor a pressões e temperaturas elevadas, normalmente utilizadas em grandes plantas industriais. Essa classe de caldeiras possui um aspecto de construção que permite uma área de troca térmica superior às caldeiras do tipo flamotubular, justificando dessa forma a possibilidade de condições operacionais superiores. Essencialmente, nas caldeiras aquatubulares a água escoa por dentro dos tubos, que passam por dentro da fornalha, onde está contida a chama e os gases de combustão. Há caldeiras dessa classe que podem produzir até 750 t/h de vapor, que normalmente se encontra superaquecido, pode chegar a temperaturas superiores a 400 °C e pressões de 200 kgf/cm², porém também há caldeiras aquatubulares com capacidades bem inferiores. Dentre as caldeiras aquatubulares, pode-se dividi-las em três tipos diferenciados pela geometria do feixe de tubos e pelo escoamento em seu interior, que são: caldeiras de tubos retos, quando os tubos que separam os tambores são retos; caldeiras de tubos curvos, quando os tubos separadores são curvos; caldeiras com circulação positiva, que envolve a circulação positiva da água, podendo ser natural devido ao projeto, ou pode ser forçada através de uma bomba (DIÓRIO, 2019). Geralmente, as caldeiras aquatubulares são compostas pelas componentes:

i. **Fornalha:** Compartimento onde acontece a combustão do combustível e onde encontra-se os tubos que criam as paredes de água.
ii. **Tubulão de vapor:** Também conhecido como tambor superior, é um vaso de pressão responsável pela separação das fases vapor-água e onde é realizada a alimentação de água na caldeira.
iii. **Tubulão de lama:** Também conhecido como tambor inferior, nem toda caldeira possui essa componente, no entanto, é responsável pela coletar e separar sólidos suspensos na água e purgar do sistema.
iv. **Superaquecedor:** Componente com maior temperatura da caldeira, é um sistema de serpentinas onde o vapor saturado torna-se superaquecido.
v. **Economizador:** Componente que faz parte da integração energética da caldeira e evita choque térmico, ele promove a troca de calor entre os gases de combustão (antes de saírem pela chaminé) e a água antes de ingressar no tubulão de vapor.
vi. **Pré-aquecedor de ar:** O pré-aquecedor de ar tem a função de aquecer o ar antes de entrar na fornalha e pode utilizar como fluído de troca térmica o vapor ou os gases de combustão, onde algumas caldeiras possuem os dois tipos de pré-aquecedores.
vii. **Chaminé:** Possibilita a circulação dos gases de combustão pelas componentes da caldeira, podendo essa circulação ser natural, pelas diferenças de temperatura e densidade, ou forçada, pela influência de sopradores e exaustores.
viii. **Reaquecedor:** Complexo de serpentinas para reaquecer o vapor após processos intermediários da caldeira, nem toda caldeira possui reaquecedores.
ix. **Dessuperaquecedor:** Componente utilizado para controle de temperatura, onde reduz a temperatura de vapor superaquecido através da injeção de água.
x. **Precipitador eletrostático:** Presente em caldeiras que utilizam combustíveis sólidos particulados, tem a função de coletar as cinzas presente nos gases antes destes entrarem para as chaminés.
xi. **Queimadores:** Promove a proporção adequada de ar para garantir a combustão completa do combustível, o que aumenta a geração de energia térmica e diminui a emissão de poluentes pela combustão incompleta.

xii. **Coletores:** Tubulações que possuem a função de coletar as diversas correntes de água e vapor advindas da caldeira.

xiii. **Tubos:** Normalmente feitos de aço carbono, possuem a função de conduzir água e vapor, assim como realizar as trocas térmicas. As paredes d'águas que compõem a fornalha são constituídas por tubos que conduzem água interligados por aletas que impedem a passagem dos gases de combustão.

xiv. **Válvulas de Segurança:** Dispositivos de segurança para evitar que a pressão da caldeira exceda os limites do projeto, normalmente caldeiras aquatubulares possuem uma válvula no tubulão de vapor e outra no superaquecedor (INSTITUTO BRASILEIRO DE PETRÓLEO E GÁS, 2020).

3.17 Manutenção preditiva

A palavra manutenção deriva do latim, *manus tenere*, que significa "manter o que se tem", atualmente, pode definir-se por conjunto de técnicas e procedimentos necessários para cuidar, manter a integridade e bom funcionamento de equipamentos, máquinas, peças e ferramentas. Com o avanço tecnológico desde o surgimento da mecanização, industrialização e automatização, as técnicas de manutenção também evoluíram e não apenas sobre procedimentos básicos como montagem, desmontagem e substituições, mas também no gerenciamento da manutenção e também no desenvolvimento de diferentes tipos de manutenção, para atender as mais variadas necessidades da indústria. Atualmente, há cinco tipos de manutenção: manutenção corretiva, manutenção preventiva, manutenção preditiva, manutenção produtiva total e manutenção centrada na confiabilidade (DE ALMEIDA, 2015).

Para entender melhor os tipos de manutenção, é interessante saber os conceitos de: defeito, quando há qualquer desvio das especificações do equipamento; falha, evento no qual o equipamento perde a capacidade de executar as funções demandadas; pane, é o estado em que o equipamento se encontra após a falha. Tendo em vista esses conceitos, discute-se agora sobre o que é a Manutenção corretiva, que pode ser definida como o ato de restituir o equipamento a seu estado normal depois do estado de pane. Porém, também há quem defina que pode ser a ação para corrigir a falha ou defeito, quando é identificado através de monitoramento do equipamento ainda em funcionamento (GREGÓRIO; DA SILVEIRA, 2018).

Antes de falar sobre manutenção preditiva e preventiva, é interessante distinguir os conceitos da prevenção e da predição. A prevenção constitui na troca de uma peça ou componente que se supõe no limite de sua vida útil e afirma-se em estatísticas com confiabilidade deveras duvidosa, além de ser frequente o aparecimento de rupturas nas peças substituídas. Com relação a predição, a substituição da peça é baseada em dados numéricos oriundos do aferimento de parâmetros relativos à própria peça, o que faz com que a substituição seja feita somente quando for necessária, independente do tempo de uso. A partir disso, pode-se dizer que são dois conceitos antagônicos (NEPOMUCENO, 1989).

Manutenção preventiva se faz necessária para evitar paradas indesejadas no equipamento. Para isso, é planejado e controlado manutenções predefinidas no equipamento, com intuito de manter o estado de conservação do equipamento. Planeja-se a manutenção preventiva através do histórico de operações de manutenção corretiva já feitas e informações de vida útil do equipamento, estabelecidas pelo fabricante (DE ALMEIDA, 2015).

> A manutenção preditiva é um tipo de manutenção planejada que aplica de forma sistemática técnicas de análise, buscando reduzir manutenções preventivas e corretivas e utilizando o componente durante toda a sua vida útil (GREGÓRIO; DA SILVEIRA, 2018, p. 23).

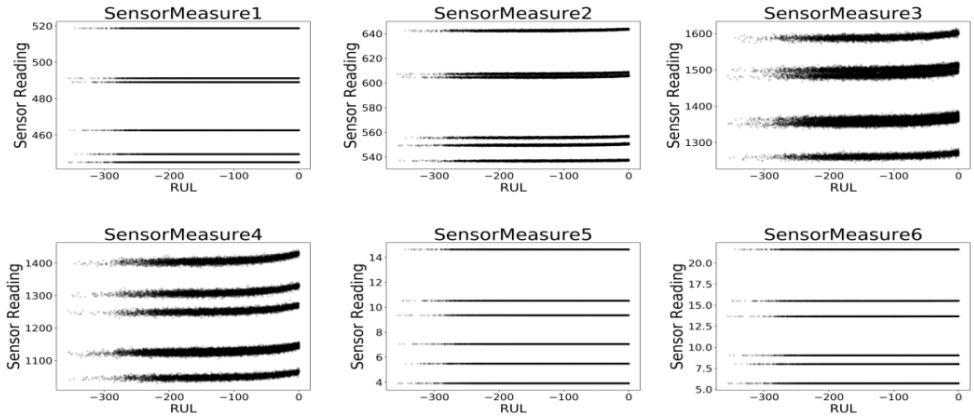
A manutenção preditiva utiliza de dados dos fenômenos ocorridos durante o tempo de operação de um equipamento, principalmente dos fenômenos antecedentes a falhas e defeitos. Os fenômenos podem ser observados através de instrumentos específicos por meio da temperatura, vibração, ruído, entre outros, onde esse tipo de manutenção baseia suas inspeções. Essa análise permite indicar as condições atuais do funcionamento da máquina ou equipamento, além de observar o desenvolvimento de uma falha e com isso, planejar um processo de manutenção corretiva planejada. A manutenção preditiva também permite indicar o tempo de vida útil remanescente (VUR) do equipamento (DE ALMEIDA, 2016). Nepomuceno (1989), em relação à manutenção preditiva, diz que “o fundamento do método consiste em admitir que a existência de um defeito ou irregularidade dá origem a uma reação sobre determinados parâmetros, que podem ser medidos e verificados de maneira precisa”. Esses parâmetros devem constituir a base da manutenção e dependem do equipamento, a qual sua operação define o que medir.

Kardec e Nascif (2001) possuem a visão de que, para implementar a manutenção preditiva, necessita-se de alguns pré-requisitos básicos, que são eles: o equipamento deve ser merecedor dessa tecnologia, devido aos custos incluídos; o equipamento em questão deve admitir algum tipo de medição ou monitoramento; as falhas precisam ser provenientes de causas que possam ser monitoradas e seu desenvolvimento acompanhado; implementação de um programa sistematizado de monitoramento, análise e diagnóstico.

A implementação da manutenção preditiva possui diversos benefícios, como reduzir custos com manutenção, melhoria no processo, segurança pessoal, segurança da planta, entre outras. Pode reduzir significativamente falhas catastróficas, o que diminui o número de acidentes de trabalho e danos a planta. Além do mais, reduz muito as paradas inesperadas na planta, que além de comprometer a segurança, pode causar prejuízos significativos para a empresa devido a produção. Também, evita intervenções desnecessárias na planta, o que reduz gastos com manutenção e mantém o equipamento operando por mais tempo de forma segura (KARDEC; NASCIF, 2001).

Rosa (2019) realizou um trabalho na área de manutenção preditiva em motores de turbinas através de dados disponibilizados pelo Centro de Prognósticos e Excelência da NASA. O objetivo, principalmente, foi prever com maior precisão possível a vida útil remanescente, ou *remaining useful life* (RUL), dos motores utilizando algoritmos de *Machine Learning* em Python e comparar seus desempenhos, que foram: *Extreme Gradient Boosting*, Árvore de Decisão, Máquina Suporte de Vetor e Florestas Aleatórias. Na Figura 12 mostra o comportamento de 6 dos 21 sensores presentes na base de dados.

Figura 12 – Dados brutos dos motores de turbina

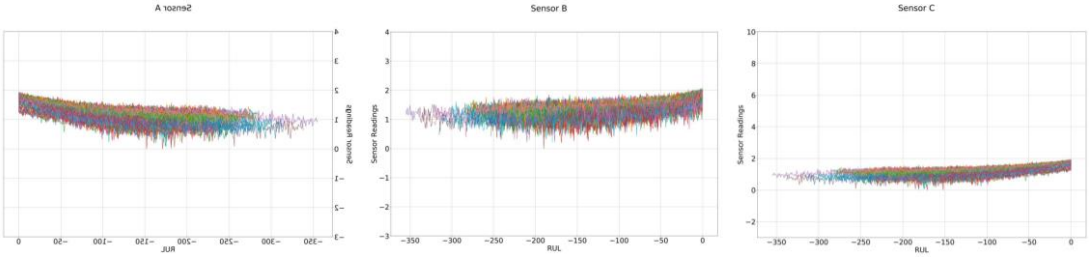


Fonte: Rosa (2019)

Um ponto interessante a se reforçar na visualização dos dados, principalmente para o sensor 2, 3 e 4 é a mudança comportamental nos parâmetros a serem medidos conforme o motor aproxima-se do seu final de vida útil.

Antes do treinamento dos algoritmos, os dados passaram por uma série de procedimentos preparatórios, como o *clustering* dos dados através de *K-Means Clustering*, uma técnica de aprendizado não-supervisionado; padronização dos dados a fim de remover a separação dos dados por agrupamentos; criação de grupos de colunas com comportamento semelhante nos parâmetros; eliminação de *outliers*; redução de dimensionalidade para cada grupo de colunas através do algoritmo PCA (*Principal Component Analysis*); transformação em log; *feature extraction* através da ferramenta "*tsfresh*"; treinamento dos modelos preditivos. Após esses pré-processamentos, os dados dos sensores resumiram-se a três representantes do conjunto, mostrados na Figura 13.

Figura 13 – Dados processados dos motores de turbina

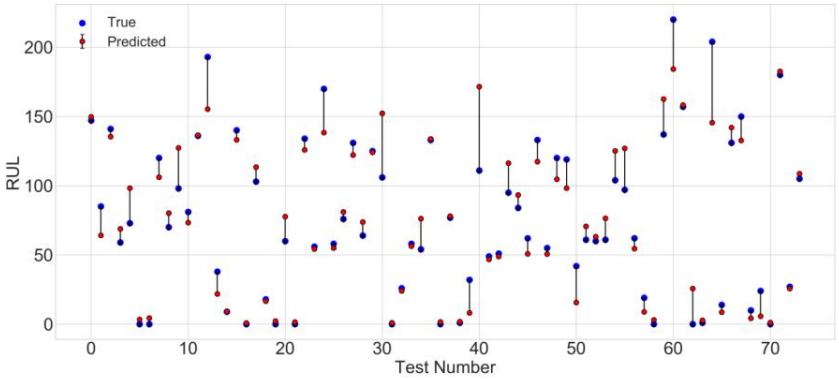


Fonte: Rosa (2019)

Dos algoritmos testados, o Máquina de Vetor Suporte foi o que obteve pior desempenho, seguido de Árvore de Decisão. Discrepante dos outros, os algoritmos

de Florestas Aleatórias e *Extreme Gradient Boosting* apresentaram melhores resultados, sendo este último o melhor. Na Figura 14 é mostrado o gráfico de desempenho do algoritmo *Extreme Gradient Boosting* para as primeiras predições, onde os pontos azuis são os reais e vermelho o predito.

Figura 14 – Desempenho do algoritmo *Extreme Gradient Boosting*



Fonte: Rosa (2019)

O autor concluiu que a análise comparativa dos algoritmos foi realizada com sucesso e que apesar de ter obtido bons resultados, acredita que utilizar técnicas de *deep learning*, como LSTM (*Long Short-Term Memor*), poderia melhorar o desempenho nas predições. Também, cogitou modificar as etapas de pré-processamento, como na utilização do pacote *Feature Tools*, para selecionar de forma semiautomática as *features* a serem extraídas.

4 METODOLOGIA

Neste capítulo será apresentado a metodologia que será utilizada para alcançar os objetivos propostos neste trabalho.

4.1 Seleção da área a ser estudada

Como primeira etapa, para a seleção da área a ser estudada será realizada uma pesquisa em campo nas áreas propostas, que são a secagem, o tratamento de água de processo e a caldeira. Os critérios a serem analisados serão principalmente a qualidade de coleta de dados realizada pela própria indústria, armazenamento e integridade desses dados, além de possíveis problemas enfrentados rotineiramente pelos operadores que possam ser alvo de estudo.

4.2 Coleta de dados

A segunda etapa a ser realizada para a aplicação das técnicas de Análise de Dados e *Machine Learning* na indústria de arroz é a coleta de dados, etapa crucial para obter bons resultados de aprendizado do algoritmo.

Na etapa de coleta de dados, será realizada uma pesquisa de campo dentro da indústria de arroz, onde se buscará dados históricos da área selecionada. Esses dados poderão estar em formato de planilhas digitais ou manuscritas, nuvens, sistemas ERP ou em algum sistema de banco de dados coletados por sensores.

Após a identificação da localização dos dados de histórico, se iniciará a coleta de dados, que dependerá do local onde eles se encontrarão. No caso da indústria de arroz em estudo, os dados são armazenados em planilhas impressas e preenchidas manualmente. Em consequência, a metodologia de coleta de dados será realizada através de uma reescrita dos dados de processo em uma planilha digital utilizando o *software* editor de planilhas Microsoft® Excel® LTSC MSO (16.0.14326.20450) 64 *bits*.

Os dados de histórico de processos serão coletados de acordo com a data e hora da coleta realizada pelos funcionários da indústria.

4.3 Tratamento de dados coletados

Em sequência da coleta dos dados, os dados serão exportados utilizando o Microsoft Excel em um arquivo de formato CSV (*comma-separated-values*, valores separados por vírgulas), formato esse que pode ser lido por diversos *softwares* e linguagens de programação.

Para o tratamento dos dados, será utilizada a plataforma Jupyter Notebook, versão 6.4.8, inserida no *software* Anaconda 2.3.2, que é voltado para ciência de dados em linguagem Python, que será utilizada na versão 3.9.12.

A primeira etapa do tratamento dos dados é constituída pela leitura dos dados em um arquivo CSV pela plataforma Jupyter. Em direção a essa finalidade, a biblioteca utilizada será a *Pandas*, que permite a análise e manipulação de planilhas, tabelas, séries temporais, matrizes, entre outros. Em seguida, para evitar erros de leitura durante o código, os títulos de cada coluna serão renomeados, substituindo espaços por traços baixos “_” e removendo acentos.

Após a leitura e renomeação das colunas, será realizada uma análise que forneça uma visão geral dos dados importados, tais como quantidade de dados nulos, não nulos, linhas e colunas, também o tipo de dados em cada coluna, podendo ser classificado em: “int” (*integer* – inteiro), que são números que não possuem decimais; “float” (ponto flutuante), que representa o conjunto dos números reais; “str” (*string*), que representa um conjunto de caracteres, podendo conter letras, números e símbolos.

A seguir à identificação do tipo de dados de cada coluna, os dados reconhecidos de maneira indevida – como por exemplo um dado que deveria ser *float* ser reconhecido como *string* – serão convertidos no seu tipo correto para possibilitar a leitura e reconhecimento verdadeiro de seus valores em análise e manipulações futuras.

Como última etapa do tratamento de dados, serão excluídos os valores “NaN”, valores esses nulos ou vazios em caso de muitos dados ou pouca importância da coluna. Senão, os dados serão preenchidos e, para tal, serão utilizados métodos estatísticos que melhor se adequarem a cada situação. Comumente, para preencher os valores nulos de colunas tipo “*int*” ou “*float*”, pode-se utilizar o método da mediana, que busca uma tendência central dos valores analisados e trabalha melhor com dados que apresentam alguma discrepância entre valores do que o método da média. Em

contraponto, para as colunas tipo “str”, tais como dados de Hora, pode-se preencher os valores nulos ao utilizar o método estatístico de moda, que basicamente é o valor com maior frequência em um conjunto de dados.

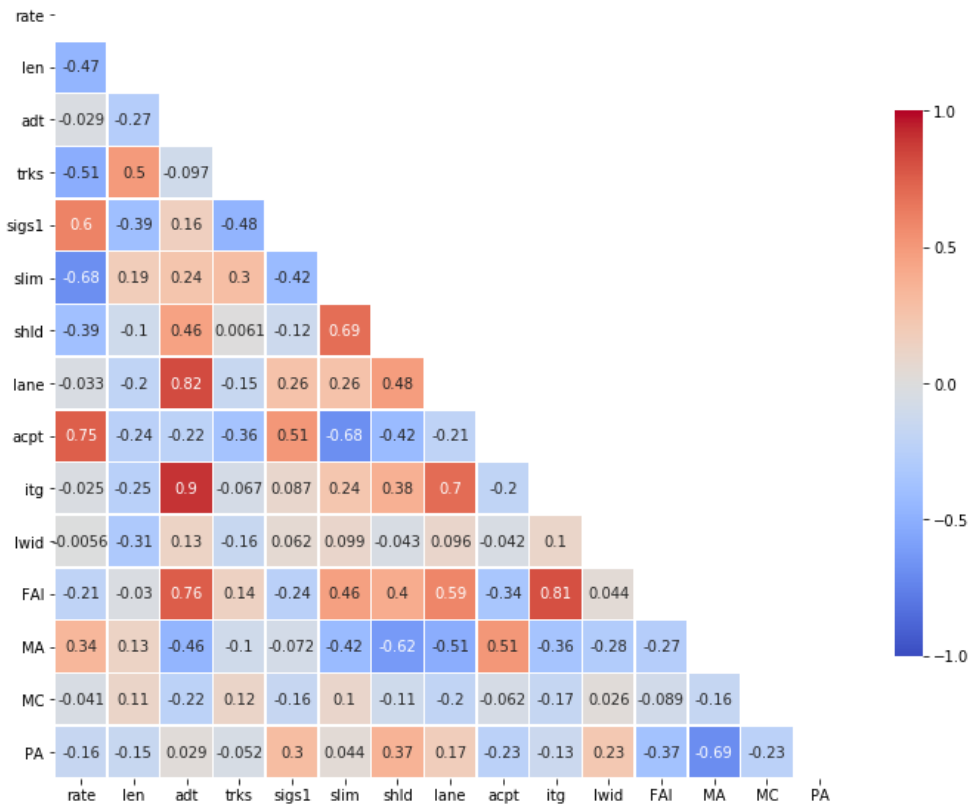
4.4 Análise Exploratória de Dados

A próxima etapa em sequência do tratamento de dados, será a Análise Exploratória de Dados, etapa crucial para a exploração dos dados coletados em busca de obter conhecimento sobre os dados através de correlações, divisões, padrões e tendências dentro da área do processo em análise da indústria de arroz. Para tal, o uso de bibliotecas em Python que atuarão como ferramentas de análise será essencial, tais como: Numpy, Seaborn e Matplotlib. Essas bibliotecas poderão produzir diversas variedades de gráficos que auxiliarão na exploração e análise de dados.

Vale ressaltar que as imagens de gráficos utilizados para os exemplos não possuem nenhuma correlação com o presente trabalho e podem ser de diferentes dados obtidos em diferentes fontes, por isso, possuem finalidade puramente ilustrativa.

Uma ferramenta gráfica útil para início da análise exploratória dos dados são os *heatmaps* ou mapas de calor, que pode ser encontrado na biblioteca Seaborn e faz correlações entre as distintas colunas da base de dados podendo representar um valor quantitativo e visual através de diferença de intensidade de cores. Na Figura 15, é ilustrado um *heatmap* correlacionando diversos dados distintos, onde observa-se os valores do coeficiente de Pearson, onde mais próximos de 1 com tendência à coloração rosa avermelhado e com proximidade ao -1, tendem a coloração de azul. Os valores das correlações obtidas entre 1 e -1, indicam tendência ao comportamento crescente e decrescente respectivamente. Através disso, pode-se extrair informações prévias de dados correlacionados com tendência linear.

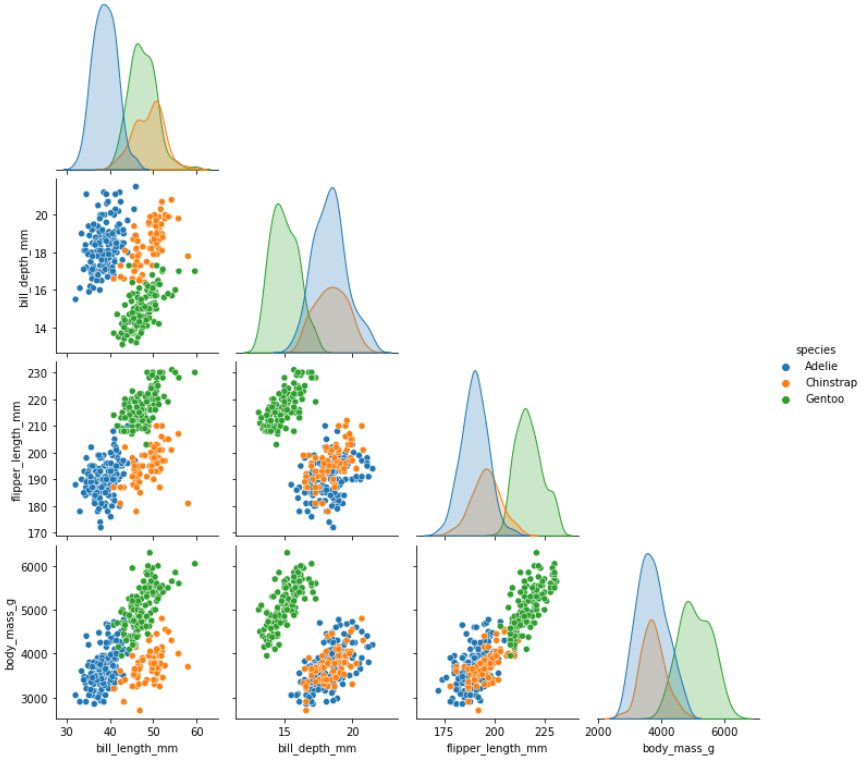
Figura 15 – Exemplo de utilização da função *heatmap*



Fonte: Kho (2019)

Outro tipo de gráfico que poderá ser utilizado como ferramenta preliminar de exploração na análise de dados é o *pair plot* ou diagrama de pares, função da biblioteca Seaborn, que assim como *heatmap* correlaciona todas colunas da base de dados entre si em apenas um lugar, disponibilizando uma visão geral dos dados. No entanto, as representações das correlações realizadas pelo *pair plot* é realizada através de gráficos de dispersão e uma distribuição normal para cada coluna da base de dados. Somando a isso, uma possibilidade que esse tipo de gráfico fornece é classificar os dados de acordo com um rótulo, como por exemplo, a Figura 16 mostra as relações entre as colunas da base de dados divididos em subgrupos através do atributo do *pair plot "*hue*"*. Esse tipo de análise permite encontrar padrões de classificação que se distinguem de acordo com o rótulo divisor.

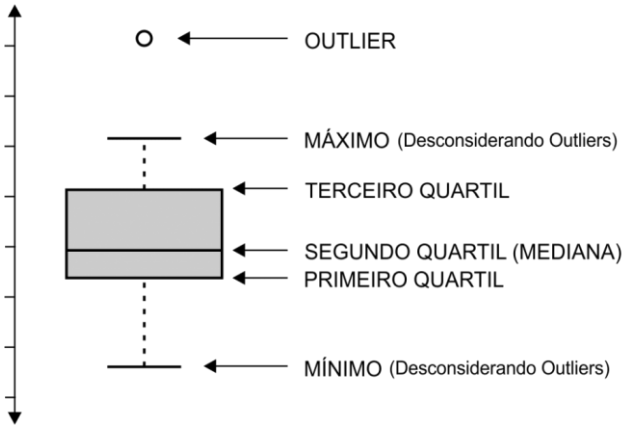
Figura 16 – Exemplo de utilização da função *pair plot*



Fonte: Adaptado de Seaborn ([2023?])

Um gráfico muito utilizado pela comunidade que trabalha com dados é o *box plot*, ou diagrama de caixas, da biblioteca Seaborn, que possibilita diversas análises visuais e estatísticas. A Figura 17 mostra o gráfico *box plot* com informações do que cada representação nele significa.

Figura 17 – Gráfico *box plot* com indicação de suas informações estatísticas

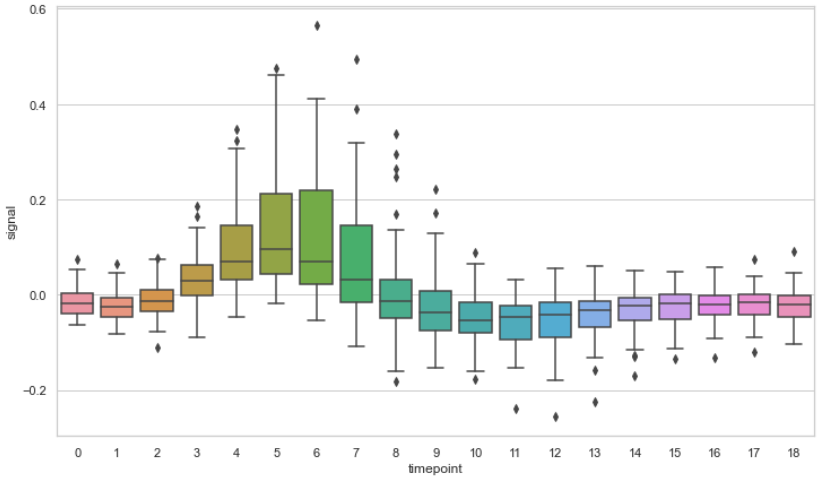


Fonte: Vizoná (2021)

Algumas das informações apresentadas na Figura 17 são os quartis, que dividem os dados em quatro grupos de 25% dos dados totais. A partir disso, o primeiro

quartil significa que 25% dos dados estão abaixo dele, o segundo quartil, ou mediana, representa que 50% dos dados estão abaixo dele e o terceiro quartil, 75% dos dados abaixo dele. Além disso, do primeiro quartil até o terceiro quartil é chamado de intervalo interquartil, que é a região de maior concentração de dados e de acordo com sua amplitude, pode-se analisar a dispersão dos dados (VIZONÁ, 2021). Além do mais, a partir dos máximos (limite superior) e mínimos (limite inferior), consegue-se determinar se os dados analisados possuem *outliers*, o qual é uma das funções para qual o *box plot* é mais utilizado na comunidade de dados. Além do mais, como mostra a Figura 18, também é possível utilizar o *box plot* para analisar dados muito dispersos e a partir do seu intervalo interquartil e mediana, consegue-se ter uma visão mais clara de seu comportamento.

Figura 18 – Gráfico *box plot* com indicação de suas informações estatísticas



Fonte: Adaptado de GeeksforGeeks (2020)

Ademais, outros tipos de gráficos mais comuns como gráficos de linhas, gráficos de dispersões de pontos e histogramas podem ser enquadrados ao longo das análises dos dados.

4.5 Padronização de dados

A próxima etapa que será realizada é a padronização dos dados que colocará todas as *features* em apenas uma escala, etapa crucial para poder obter um melhor desempenho na redução de dimensionalidade com PCA e também no treinamento dos modelos.

Para realizar o processo de padronização é utilizado a função *stats* que contém o Z-Score da biblioteca SciPy, que com o comando em Python "*DataFrame.apply(stats.zscore)*", após o *Data Frame* padronizado é armazenado em uma variável com nome diferente para sinalizar que estava padronizado.

4.6 *Principal Component Analysis*

Os procedimentos iniciais para realizar a etapa do PCA é realizado na análise de dados, que através da identificação de *features* que possuam um alto grau de correlação e um comportamento semelhante, onde serão selecionadas, agrupadas e reduzidas a apenas uma *feature* de saída representante das de entrada.

O modelo aprendizado não supervisionado PCA está presente na biblioteca Scikit-learn e é chamado través da função "*PCA()*", além disso, o parâmetro "*n_components*" é colocado com valor igual a 1, visto que do grupo selecionado, o desejado é o retorno de apenas uma *feature* por grupo, se houve mais de um. Após aplicar o método com a função "*fit_transform()*" nos dados, o PCA retorna um *array*, que será convertido em *Data Frame* e renomeado através da biblioteca Pandas.

4.7 Tratamento de *outliers*

A etapa de tratamento de *outliers* consistirá em três etapas: análise dos dados através de do gráfico *box plot*, cálculo dos limites superior e inferior e eliminação dos *outliers*.

Primeiramente, para identificar se os dados possuem *outliers* é plotado um *box plot* exatamente como é mostrado na etapa da análise exploratória de dados. A seguir, com a visualização é identificado as colunas que possuem outliers para calcular os limites superiores e inferiores dessa coluna, ou apenas um dos limites, dependendo da existência de outliers para os dois ou apenas para um.

Para realizar o cálculo dos limites superior e inferior, primeiramente é realizado o cálculo da amplitude do intervalo interquartil, onde será necessário obter o primeiro e terceiro quartil, que é obtido através da função "*percentile()*" da biblioteca Numpy. A amplitude do intervalo interquartil é demonstrada na Equação 11.

$$A_{iq} = q_3 - q_1 \tag{11}$$

A_{iq} representa a amplitude do intervalo interquartil, q_3 o terceiro quartil e q_1 o primeiro quartil. Após isso, o cálculo para o limite superior e inferior é dado as Equações 12 e 13, respectivamente.

$$lim_{sup} = q_3 + 1{,}5A_{iq} \quad (12)$$

$$lim_{inf} = q_1 - 1{,}5A_{iq} \quad (13)$$

O lim_{sup}, limite superior, é calculado a partir de uma adição de 50% a mais do intervalo interquatil no terceiro quartil. Já o limite inferior é calculado com uma subtração de 50% a mais do intervalo interquartil a partir do primeiro quartil.

Sabendo os limites, basta filtrar o *Data Frame* da seguinte forma: "*DataFrame [(DataFrame[coluna]<*lim_{sup} *& DataFrame[coluna]<*lim_{inf}*)]*".

4.8 Seleção de *features*

A seleção de *features* será realizada a partir do método *Recursive Feature Elimination* (RFE) da biblioteca Scikit-learn e será efetuada para os três diferentes modelos de Aprendizado de Máquina, visto que diferentes modelos podem se adequarem melhor a diferentes *features*.

Para realizar a seleção, a função "*RFE()*" é preenchida com os seguintes atributos: o modelo a ser utilizado; "*n_feature_to_select*", número de *features* a serem selecionadas no final e por padrão são selecionados metade das *features* originais, porém será deixado em 1 *feature* e o motivo se discutirá posteriormente; "*step*" em 1, eliminando uma *feature* a cada ciclo de avaliação da seleção. Com a função "*fit()*" e atribuindo os dados, a seleção é feita. O atributo "*n_feature_to_select*" é posicionado em 1 para posteriormente emitir um *ranking* das melhores *features*, com a função "*ranking_*", caso contrário, todas as *features* selecionadas serão as primeiras e as outras seguem de acordo com sua ordem de eliminação.

4.9 Divisão dos dados

Após definir um alvo de estudo, os dados deverão ser divididos para realizar o treinamento e teste dos algoritmos de *Machine Learning*. Essa divisão deverá ser

realizada corretamente para garantir uma confiabilidade na medição do desempenho do algoritmo, o qual por senso, terá de manter maior parte do conjunto para o treinamento e um percentual menor para o teste.

A ferramenta utilizada para dividir os dados fará parte da bilbioteca "*sklearn.model_selection", "train_test_split()"*, onde poderá receber como argumento, além das variáveis dependente e independente, o tamanho dos dados de teste "*test_size"* e controlar a aleatoriedade dos dados com "*random_state"*, além de outros argumentos. Dessa forma, as divisões serão realizadas de modo que 80% dos dados serão para treino e 20% para teste.

A respeito de regressões em série temporal é crucial que os dados não sejam ordenados de forma aleatória, por isso, para esses casos os dados serão divididos em "passado" e "futuro" respeitando a série. Para realizar essa divisão, é necessário calcular quantos dados representam 20% do total. A partir disso, sabendo o número do maior *index* é possível calcular o *index* que dividirá os dados em 80% para treino e 20% para teste. A divisão é realizada utilizando filtros de condições, como por exemplo "*DataFrame*[:*index*]".

4.10 Seleção de parâmetros

Os parâmetros são argumentos ajustáveis que não são aprendidos diretamente pelo estimador o qual permitem controlar a etapa de treinamento do algoritmo. Nesta etapa, para seleção dos melhores parâmetros será utilizada a função "*GridSearchCV()"*, integrante da biblioteca "*sklearn.model_selection"*. Essa função realiza uma série de treino e teste em cima dos dados com os possíveis valores de parâmetros especificados, em seguida definirá o conjunto de parâmetros para atingir o melhor valor da métrica escolhida no argumento "*scoring"*. Esse método utiliza os conceitos de validação cruzada, portanto realiza diversos treinos e teste de acordo com o número de dobras selecionadas e combinando todas as probabilidades de conjuntos de parâmetros selecionados, o que pode causar um maior custo computacional e dependendo da quantidade de parâmetros e o estimador, pode levar diversos minutos em processo de execução do algoritmo. Além disso, os dados a serem utilizados para a seleção de parâmetros já serão utilizados de acordo com a seleção de *features* realizada pelo RFE de acordo com seus respectivos algoritmos.

Para casos de classificação a métrica a ser aperfeiçoada será a acurácia, já em casos de regressão o R-quadrado.

4.11 Treinamento dos modelos

Na etapa conseguinte, será executado o treinamento dos algoritmos de *Machine Learning* apresentados na revisão bibliográfica: Regressão Linear, Florestas Aleatórias e XGBoost o qual para os dois primeiros, será utilizado a biblioteca *Scikit-learn* e para a última, a biblioteca XGBoost.

As funções para treinar os algoritmos dos modelos serão: para o modelo de Regressão Linear, será utilizada a função *LinearRegression()*; para a Florestas Aleatórias, a função *RandomForestRegressor()* será utilizada em casos de regressão e *RandomForestClassifier()* pra casos de classificação; para o XGBoost, a função *XGBRegressor()* será utilizada em casos de regressão e *XGBClassifier()* pra casos de classificação.

4.12 Avaliação dos modelos

Como última etapa, o algoritmo de *Machine Learning* será submetido a avaliações utilizando métricas de acordo com seus dados de teste para medir seu desempenho de predição ou classificação. Para tal, será utilizada a biblioteca "*sklearn.metrics"*.

Para os modelos de regressão, poderá ser utilizado como métrica para avaliação de seu desempenho a raiz do erro quadrático médio elevando o resultado do erro quadrático médio (*"mean_squared_error()")* na potência de ½, o erro absoluto médio (*"mean_absolute_error()")* e o coeficiente de determinação, também chamado de r-quadrado (*"r2_score()").*

No sentido dos modelos de classificação de duas classes, poderá ser recorrido como métrica para avaliação de seu desempenho a acurácia, precisão, revocação e F1-Score*.* A função "*classification_report()"* permite uma visão completa de todas as métricas citadas, além de servir tanto para classificações binárias ou multiclasse.

5 RESULTADOS E DISCUSSÕES

A indústria de beneficiamento de arroz em questão, que se enquadra na região da campanha do Rio Grande do Sul, enquadra-se nos parâmetros de uma Indústria 3.0, no qual possui a maior parte de seu processo automatizado.

5.1 Escolha da área de estudo e motivação

Ao total, são coletados dados suficientes para realizar uma análise prévia de cada área posta em pauta, onde primeiro são os dados de tratamento de água de processo, em seguida a secagem e pôr fim a caldeira. Para os dois primeiros, as coletas são interrompidas por motivos de inconsistência na coleta de dados, diversos dados faltantes e algumas colunas com baixa variabilidade nos dados, principalmente quando se tratava sobre o tratamento de água.

Pelas dificuldades apresentadas na coleta de dados nas outras áreas (secagem e tratamento de água), os esforços são direcionados ao estudo da caldeira e após a coleta de dados, identificou-se diversas oportunidades na área e a caldeira é escolhida como objeto de estudo.

As motivações da escolha da caldeira da indústria de beneficiamento de arroz são diversas, como: a importância do equipamento para todo processo industrial; a rigidez e cuidado no monitoramento e operação de caldeira, por ser um equipamento que trabalha em alta pressão e temperatura, o que faz com que maiores cuidados e uma atenção especial sejam demandados; devido ao monitoramento mais rigoroso comparado a outras etapas do processo, a qualidade dos dados de coleta também é superior às outras áreas; a ocorrência de falhas recorrentes fazem da caldeira uma boa oportunidade para um estudo de manutenção preditiva; a caldeira, apesar de ser muito estudada na área de Engenharia Mecânica, também é estudada na Engenharia Química devido a sua grande aplicação em processos industriais e engloba conceitos estudados nas disciplinas curriculares durante a graduação.

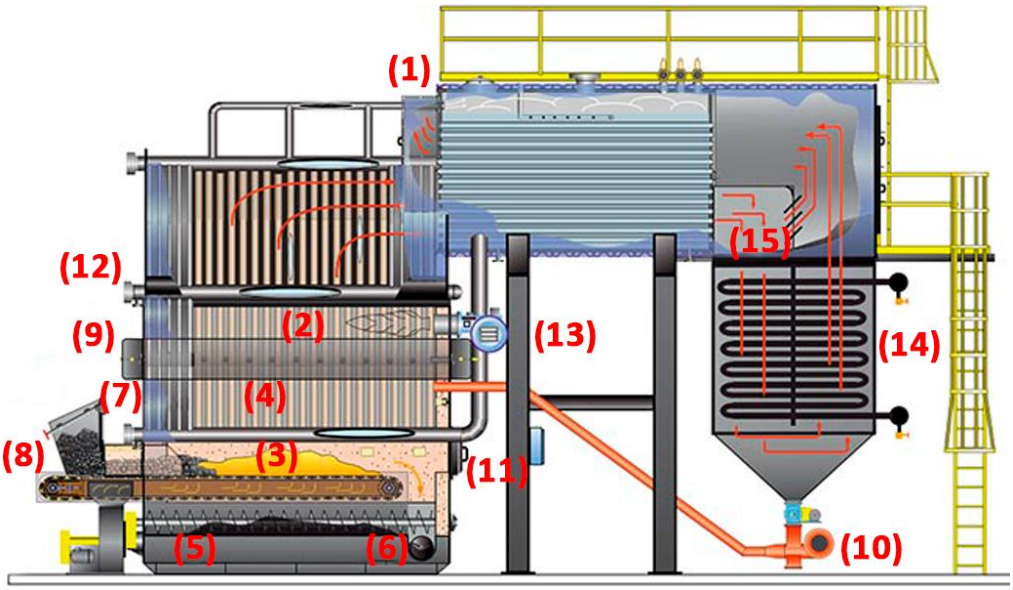
Devido aos motivos apresentados acima e principalmente devido às recorrentes falhas e a coleta de dados mais precisa, o foco é direcionado ao estudo da caldeira e suas falhas, visando realizar dois principais estudos: estudo de falhas e manutenção preditiva. Primeiro, o estudo de falhas terá como principal objetivo caracterizar essas falhas através da análise de dados e por fim, classificar essas diferentes falhas na caldeira. Em seguida, realizar o estudo da manutenção preditiva

através do tempo de vida útil remanescente na caldeira (VUR), onde é visado compreender o comportamento dos dados ao longo da VUR e por fim, utilizar um algoritmo de regressão para predizer a mesma.

5.2 Processo da caldeira

O equipamento em estudo é uma caldeira do tipo mista de biomassa, ou seja, ela possui uma parte aquatubular e outra flamotubular. O combustível da caldeira é casca de arroz, proveniente da própria indústria, que são transportados por um elevador de canecos até a caixa de cascas da caldeira onde a partir dali, é controlada a injeção de casca na caldeira, ou seja, a alimentação. A Figura 19 mostra uma representação mais próxima encontrada da caldeira mista em estudo.

Figura 19 – Representação caldeira mista



Fonte: Adaptado de Hurst Boiler ([2023?])

Cada ponto numerado na Figura 19 é representado pelas seguintes componentes: (1) projeto misto de tubo de fogo e água; (2) seção de tubos de água; (3) sistemas de grelha de corrente; (4) câmara de combustão, fornalha; (5) parafuso de coleta de cinzas primário; (6) parafuso de coleta de cinzas secundário; (7) Controle de ar; (8) alimentação; (9) sistema de entrada de ar sobre fogo; (10) soprador de reinjeção de cinzas; (11) portas corta-fogo; (12) portas de inspeção do lado da água; (13) queimador de gás/óleo de reserva; (14) superaquecedor; (15) desviador de gás de combustão (HURST BOILER, [2023?]). Apesar das similaridades entre as

componentes principais da caldeira mista em estudo e a apresentada na Figura 19, há componentes que não existem na caldeira em estudo, tais como o (10), (13) e (15).

O processo inicia na parte aquatubular da caldeira, onde é realizada a combustão da casca de arroz no forno sobre uma grelha móvel com o auxílio da injeção de ar para realizar a combustão. A partir disso, as cinzas formadas saem da grelha e são transportadas pelo *redler*, um transportador de corrente de arraste.

Os gases formados na combustão trocam calor com a água, o qual escoa nas paredes de água dentro do forno e inicia seu processo de vaporização. A partir disso, a água prossegue até o casco da parte flamotubular da caldeira e os gases de combustão entram nos feixes de tubos, onde trocam calor entre si e é formado o vapor saturado. Em seguida, o vapor saturado percorre o superaquecedor, troca calor com os gases de combustão e é formado o vapor superaquecido. Por fim, o vapor produzido na caldeira é destinado a seus devidos usos no processo industrial de beneficiamento do arroz.

Os gases de combustão direcionam-se a um ciclone para a retirada de partículas mais grosseiras e, para as partículas menores, é filtrado através de um filtro manga, que após isso, direciona-se à chaminé. Para realizar esse processo de retirada dos gases da caldeira é utilizado um exaustor.

O vapor produzido na caldeira é utilizado para aquecer o ar utilizado na secagem de grãos e também é utilizado diretamente no processo de parboilização do arroz, o que acarreta em perdas de vapor de água, caracterizando um ciclo de água da caldeira não totalmente fechado. Além do mais, esporadicamente o vapor é utilizado na turbina de geração de energia e, quando feito, a caldeira opera em parâmetros de processos mais elevados do que somente para processo. Após ser utilizado nos processos, o vapor percorre um condensador e após liquefeito, é direcionado para um tanque de água dentro da sala da caldeira, que é responsável de armazenar a água para alimentação da caldeira. O volume de água perdido no processo é reposto diretamente no tanque, onde é bombeada água tratada para ele.

5.3 Registro, monitoramento e controle da caldeira

A caldeira opera 24 horas por dia e seu monitoramento é realizado através de um painel que indica os parâmetros de processo da caldeira e é monitorado por operadores que se revezam em turno de 8 em 8 horas. O controle do processo da

caldeira é realizado através de um controlador proporcional integral derivativo (PID), onde a partir do *set point* de pressão de vapor e alimentação de casca fornecidos pelo operador, o controlador irá manipular, por exemplo, a velocidade dos ventiladores para a entrada de ar na caldeira a fim de aumentar a queima e atingir o novo *set poin*t. Com relação a isso, os *set points* variam de acordo com as necessidades do processo, como a quantidade de equipamentos utilizando vapor no processo de produção do arroz ou a geração de energia através da turbina. A Figura 20 apresenta o painel de visualização dos parâmetros de operação da caldeira.

Figura 20 – Painel da caldeira

Fonte: Autor (2022)

Para fins de registro do monitoramento da caldeira é realizado dois procedimentos: o primeiro é a coleta de dados de hora em hora utilizando planilhas impressas em papel; o segundo é realizar anotações em um caderno de registros dos acontecimentos, anormalidades, falhas e procedimentos realizados na caldeira. Vale ressaltar que, no momento que acontece a falha, não é mais realizada coleta de dados, portanto se a 13h30min ou 14h a caldeira falhou por duas horas, os dados de 14h e 15h serão vazios e o último dados coletado é das 13h, somente coletado novamente quando voltar à operação normal.

5.4 Procedimento de identificação de falhas

A identificação das falhas na caldeira é realizada pelos operadores, através do seu conhecimento sobre o comportamento da caldeira e não possui nenhum indicador explícito de falha ou alarme, a não ser na alteração dos parâmetros de processo, sons

e ruídos da caldeira e equipamentos interligados que podem indicar que certa falha está acontecendo.

5.5 Parâmetros da caldeira e coleta de dados

Ao todo são anotadas 11 colunas de dados pelos operadores na planilha, que são eles: data, horário, pressão de vapor, depressão, vazão de vapor, temperatura de vapor, ar primário, ar secundário, alimentação e tiragem.

i. **Data:** data no formato de dia/mês/ano que foi realizada a coleta;
ii. **Horário:** horário que foi realizada a coleta em horas;
iii. **Pressão de vapor:** pressão de vapor da saída da caldeira em kilograma força por centímetro quadrado (kgf/cm²);
iv. **Depressão:** diferença de pressão em milímetros de coluna de água (mmca), que simboliza a retirada de gases de dentro da caldeira;
v. **Vazão de vapor:** vazão do vapor de saída da caldeira em ton/h.
vi. **Temperatura de vapor:** temperatura de vapor da saída da caldeira em graus celsius (°C);
vii. **Ar primário:** ar que entra por baixo da grelha no sentido vertical e mede o percentual da potência do ventilador que está sendo utilizado para inserir ar na caldeira;
viii. **Ar secundário:** ar que entra por cima da grelha no sentido horizontal e mede o percentual da potência do ventilador que está sendo utilizado para inserir ar na caldeira;
ix. **Alimentação:** alimentação de casca de arroz em percentual inseridas no forno da caldeira;
x. **Tiragem:** mede o percentual da potência do exaustor que está sendo utilizado para retirar os gases da caldeira.

Ao todo são digitalizados 7 meses de dados através do Microsoft Excel mantendo a originalidade dos dados e respeitando os tempos de paradas rotineiras e tempos de pane da caldeira, onde para essas paradas, para não deixar vazio, os parâmetros são preenchidos com zeros.

Para cada falha ou pane ao longo dos dados coletados é considerado um ciclo de operação da caldeira, portanto sempre que acontece uma nova falha ou pane, um novo ciclo começa após a caldeira voltar a funcionar novamente. As paradas rotineiras

(chamadas paradas normais), que normalmente acontecem em sábados ou domingos, não são consideradas falhas. Simultaneamente, para cada ocorrência com a caldeira, em sua respectiva hora (linha) é criado um rótulo da “condição” da caldeira, onde pode-se identificar todas as ocorrências através dos dados, falhas ou normais.

Com propósito de criar a coluna de vida útil remanescente (VUR) pelo Python a seguir, é criada uma coluna chamada “tempo até falha”, que basicamente é um contador em horas do tempo de duração de cada ciclo que inicia em 1 hora e vai até a caldeira falhar ou entrar em pane e para o ciclo seguinte, o contador reinicia.

5.6 Descrição das falhas na caldeira

Esta etapa tem como objetivo realizara a descrição das falhas que são apresentadas ao longo deste trabalho com objetivo de melhorar a compreensão sobre essas ocorrências.

i. **Elevador:** Esta falha representa as falhas no elevador de cascas, um elevador de canecos que transporta as cascas para a caixa de cascas da caldeira. Normalmente, ocorre devido a “embuchar” (emperrar ou trancar) devido as cascas ou por algum problema nas componentes do mesmo, como a sua bucha;
ii. **Rosca:** Problemas na rosca que há no galpão, como possui baixa ocorrência (será visto futuramente), estão associados a rosca trancar;
iii. **Pressão positiva:** As falhas de pressão positiva na caldeira estão associadas quando a depressão da caldeira se aproxima do valor positivo ou torna-se positivo, no geral está associada a saturação do filtro manga, problemas no exaustor ou excesso de entrada de ar;
iv. ***Redler*:** Falhas ditas como *redler* estão associadas ao transportador de correias das cinzas, onde geralmente a corrente do *redler* rompe;
v. **Caldeira furada:** Problemas de caldeira furada estão associados a algum furo na tubulação de água da parte aquatubular ou da parede que separa a parte flamotubular da fornalha na aquatubular;
vi. **Pistão:** Problemas no pistão da caixa de casca que auxilia a injeção de cascas para alimentação da caldeira comumente estão associados ao mesmo trancar devido a casca ou algum problema mecânico;

vii. **Grelha:** Problemas relacionados a grelha estão relacionados a parada da rotação da grelha, não há muitos registros em relação a essa falha;

viii. **Caixa de cinzas:** Os problemas de caixa de cinza, que possuem baixa ocorrência, normalmente são identificados por baixa pressão na caldeira devido a caixa de cinza estar cheia;

ix. **Exaustor:** A falha no exaustor não está claramente descrita, mas é provável que seja algum problema de funcionamento relacionado a tiragem da caldeira.

5.7 *Data Frames*

Ao todo, são utilizados dois *Data Frames* com os mesmos dados de processo e colunas da caldeira, porém se distinguem de forma que um apresenta os tempos de pane, ou seja, o intervalo em que a caldeira fica parada após falha e também os intervalos de parada normais. Esse é chamado de “dados com tempo de pane”, possuindo 12 colunas com 4753 linhas (4753 horas contendo tempo de operação e tempo de pane) e é mais utilizado para classificação de falhas e paradas. O segundo, não apresenta tempo de pane e nem indica paradas normais, além de só retratar a operação da caldeira até as falhas. Esse é chamado de “dados sem tempo de pane”, possuindo 12 colunas com 3081 linhas (3081 horas somente de operação) e é mais utilizado para as regressões na predição da vida útil remanescente.

5.8 Estudo de falhas

Esta etapa tem como objetivo apresentar o estudo sobre as falhas, onde primeiramente é explorada os dados buscando explorar e caracterizar as falhas e em seguida, realizar um tratamento e processamento de dados a fim de realizar duas classificações com o algoritmo de *Machine Learning* Florestas Aleatórias. A primeira, tem como objetivo classificar o estado de “operando” ou “parada” da caldeira. A segunda, classificar cada condição em que a caldeira se passa, como condições normais, falhas, panes e paradas normais.

5.8.1 Análise das condições de falha

Primeiramente, é feita a leitura do *Data Frame* “dados com tempo de pane.csv” e com o comando “*DataFrame.info()*” e “*DataFrame.isnull()*”, verifica-se que todos os tipos de cada coluna estão corretos e que não há valores vazios.

Figura 21 – a) Primeiras nove colunas dos dados condições de falha, b) últimas três colunas dos dados condições de falha

	Horas	P_vap_kgf	Depressão_mmca	Vazão_vap_tonh	T_vap_C	Ar_primário_percent	Ar_secundário_percent	Alimentação_percent	Tiragem
0	0.0	8.3	-6.0	9.4	171.0	53.5	51.6	35.0	75.0
1	1.0	9.3	-5.0	10.4	176.0	52.3	51.6	37.0	71.0
2	2.0	9.0	-8.0	10.6	173.0	53.5	51.6	10.0	73.0
3	3.0	10.4	-10.0	11.3	180.0	26.4	51.6	10.0	49.0
4	4.0	9.0	-8.0	10.5	174.0	24.3	51.6	10.0	40.0

(a)

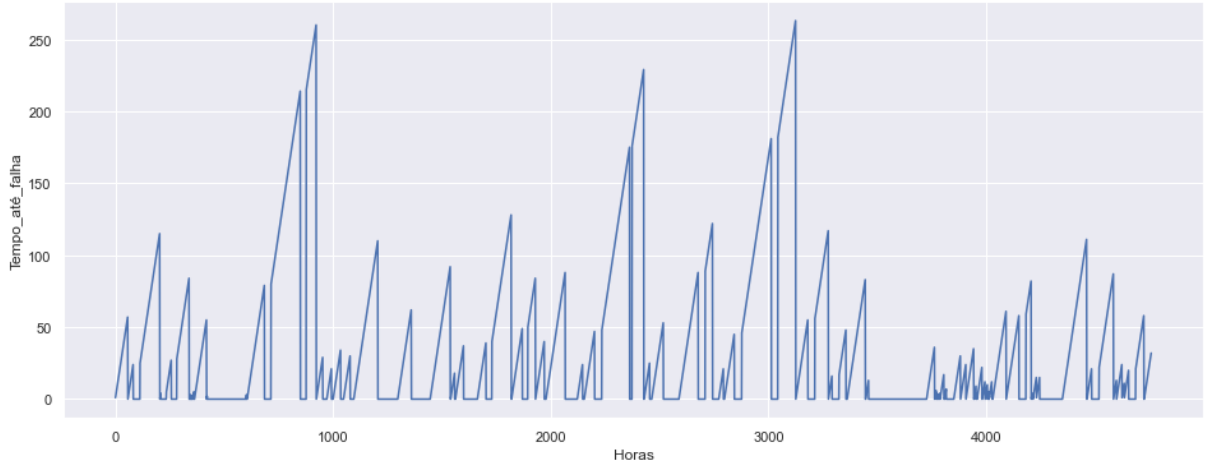
Tempo_até_falha	Ciclo_falha	Condição
1.0	1.0	Normal
2.0	1.0	Normal
3.0	1.0	Normal
4.0	1.0	Normal
5.0	1.0	Normal

(b)

Fonte: Autor (2023)

Para ter um parâmetro geral de falhas e do tempo em que a caldeira fica parada, o gráfico *line plot* do Seaborn ilustrado na Figura 22 consegue transmitir essa perspectiva.

Figura 22 – Tempo até falha ao longo das horas



Fonte: Autor (2023)

Cada “pico” do gráfico representa um ciclo de falha da caldeira que atinge seu tempo de operação até ocorrer a falha. As interrupções ao longo dos ciclos são as paradas normais, que não são consideradas falhas, portanto, o ciclo de vida da caldeira continua. Após cada ciclo pode ser observado continuidades fixadas em tempo até falha de 0 que indicam, no eixo da abscissa, o tempo em que a caldeira permanece parada. Um ponto que vale ressaltar na análise desse gráfico é que para cada triângulo retângulo formado pelos ciclos, os catetos adjacentes e opostos possuem valor iguais e poderão ser visualizados com mais clareza futuramente, quando se tratar da regressão.

No total, há 18 tipos de condições em que a caldeira pode se encontrar ao longo do período de operação estudado que são ilustradas com a Figura 23. Através do Python, pode-se utilizar o comando “*unique()*” para obter tal resultado.

Figura 23 – Condições da caldeira

	Condição
0	Normal
1	Elevador
2	Iniciando
3	Encerrando
4	ParadaNormal
5	Rosca
6	PressãoPositiva
7	Redler
8	CaldeiraFurada

(a)

	Condição
9	Pistão
10	CondiçãoNãoID
11	CaixaCinza
12	Grelha
13	CaldReiniciada
14	Manutenção
15	RecÁgua
16	DadosSet
17	Exaustor

(b)

Fonte: Autor (2023)

Das condições apresentadas acima, a “normal” (operação normal), “iniciando”, “encerrando” e “parada normal” não são consideradas falhas. Além disso, há a condição “DadosSet” que representa todo o mês seis de coleta, mês em que não há dados. No entanto, a caldeira operou nesse mês e também houve falhas, por isso será considerado como uma condição de falha. A partir disso, a caldeira apresenta no total 14 condições de falhas distintas em locais diferentes.

Sabendo todos os tipos de falhas que podem acontecer durante a vida da caldeira, faz-se interessante saber as falhas com maior frequência de ocorrência para identificar as maiores causas de parada da caldeira. Para isso, utilizou-se o *Data Frame* sem os tempos de pane, comandos como "*groupby()*" aliados a "*sum()"* e "*sort_values()*". A Figura 24 ilustra as falhas mais ocorrentes durante a vida da caldeira.

Figura 24 – Frequência de ocorrência de falhas em ordem decrescente

	Contador
Condição	
Redler	21
Elevador	13
CondiçãoNãoID	12
PressãoPositiva	7
Grelha	3
RecÁgua	1
Pistão	1
CaixaCinza	1
Manutenção	1
CaldReiniciada	1
Exaustor	1
DadosSet	1
CaldeiraFurada	1
Rosca	1

Fonte: Autor (2023)

A partir da Figura 24 observa-se que falhas no *redler* são as mais frequentes e que são os maiores motivos de falhas na caldeira. Em segundo, falhas no elevador de cascas e como terceiro, as condições não identificadas, que são paradas que aconteceram na caldeira, mas não foram registradas nos dados, portanto, não se sabe o motivo. Problemas de pressão positiva também possuem números significativos de ocorrência de falhas durante o tempo de operação estudada. As outras falhas possuem frequências significativamente menores. Além do mais, vale acrescentar que acontecem 17 paradas normais ao longo dos dados, que não são mostradas na Figura 24 por não ser contabilizada no *Data Frame* utilizado.

A fim de comprovar a veracidade da análise realizada acima, a soma de todas falhas devem resultar 65, que é o número de ciclos da caldeira, onde após realizar essa soma, o resultado é de 65.

Uma informação importante de se obter é o tempo no total (em horas) que a caldeira ficou parada por cada tipo de falha. Para isso, utilizou-se o *Data Frame* com os tempos de pane, comandos como "*groupby()*" aliados a "*sum()*" e "*sort_values()*". A Figura 25 demonstra o tempo total em que a caldeira ficou parada por pane.

Figura 25 – Tempo total de pane por tipo de falha

Condição	Contador
CondiçãoNãoID	630
ParadaNormal	492
CaldeiraFurada	178
Redler	163
PressãoPositiva	128
Elevador	56
Grelha	6
CaldReiniciada	4
Pistão	3
CaixaCinza	2
Manutenção	2
Exaustor	1
RecÁgua	1
DadosSet	1
Rosca	1

Fonte: Autor (2023)

Condições não identificadas são as predominantes e são os maiores motivos da caldeira ficar parada por tempos longos, infelizmente não se consegue realizar maiores análises sobre essas condições. Em segundo lugar, as paradas normais representam maior motivo da caldeira ficar parada e em terceiro e quarto, caldeira furada e *redler*, respectivamente. De uma forma geral, pode-se observar que as falhas e condições que ocorrem com maior frequência são responsáveis por deixar a caldeira parada por maior tempo. No entanto, existem algumas exceções como a falha de caldeira furada (ocorre apenas uma vez) e falhas no elevador (segunda mais frequente).

Sabendo o tempo total que a caldeira fica parada por tipo de falha e a frequência da ocorrência das mesmas, consegue-se obter o tempo médio que a caldeira fica parada por falha ou condição. Para realizar isso, é dividido o tempo total de cada falha pela sua frequência utilizando estruturas de repetição e condições em Python. A Figura 26 demonstra os resultados obtidos.

Figura 26 – Tempo médio em horas de parada por condição

CONDIÇÃO	TEMPO MÉDIO
CaldeiraFurada	178.00
CondiçãoNãoID	52.50
ParadaNormal	28.94
PressãoPositiva	18.29
Redler	7.76
Elevador	4.31
CaldReiniciada	4.00
Pistão	3.00
CaixaCinza	2.00
Grelha	2.00
Manutenção	2.00
DadosSet	1.00
Exaustor	1.00
RecÁgua	1.00
Rosca	1.00

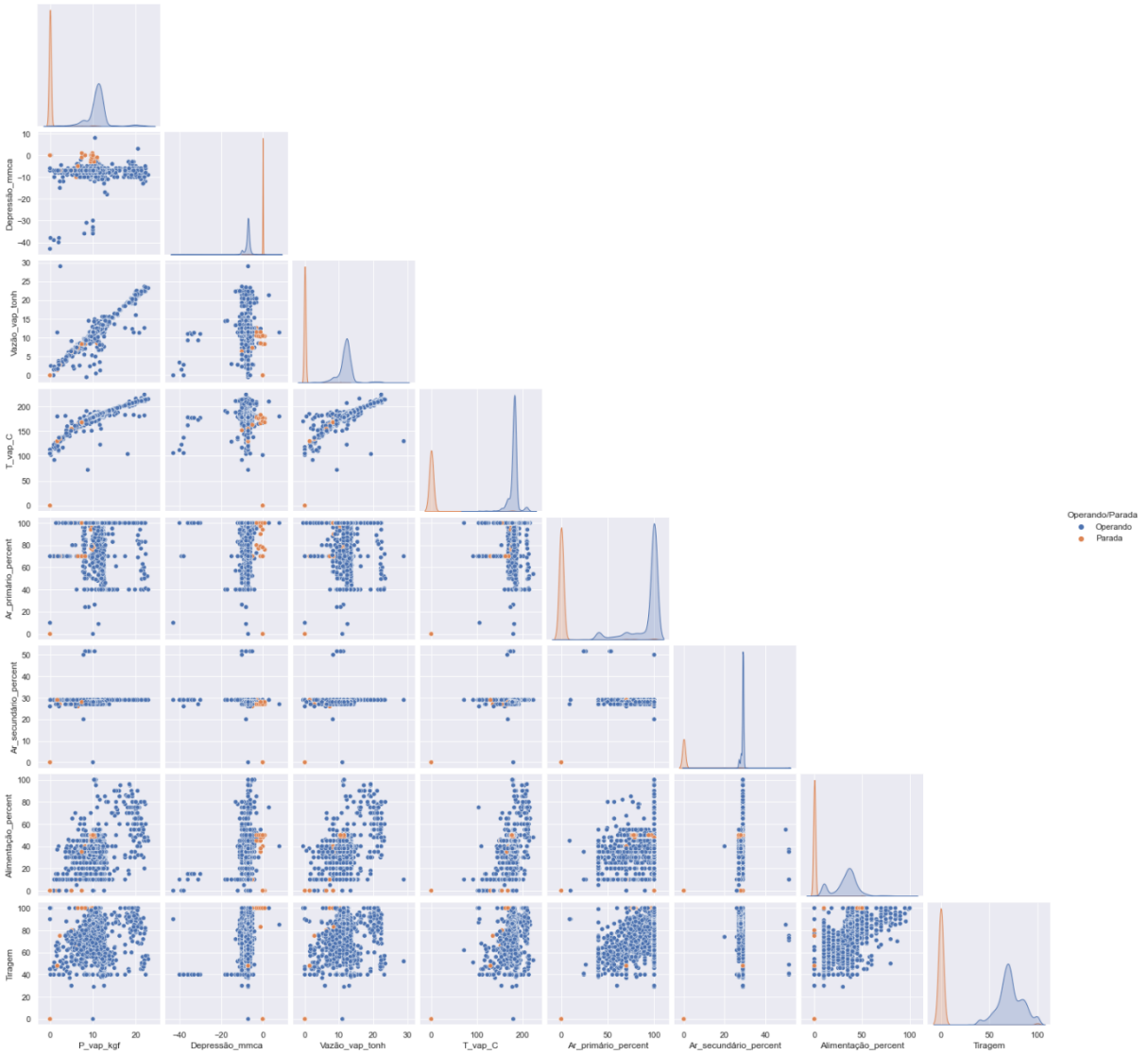
Fonte: Autor (2023)

Com maior tempo médio de parada, a falha caldeira furada é a que mais demora a ser corrigida para a caldeira voltar a operar. Em segundo, as condições não identificadas. Em terceiro, paradas normais, com algumas horas a mais do que um dia inteiro e, respectivamente a seguir, falhas de pressão positiva, *redler* e elevador. Um ponto interessante das últimas duas é de que, apesar de serem as mais frequentes, são corrigidas relativamente rápido comparado a outras.

5.8.2 Análise dos dados para classificação de “operando” e “parada”

Para classificar o estado de “operando” ou “parada” da caldeira é criada a coluna “Operando/Parada” na base de dados com tempo de pane. A fim de analisar a distribuição dos dados perante a esses novos rótulos criados, pode-se utilizar o *pair plot* do Seaborn com seu atributo “hue” indicando a coluna “Operando/Parada”. A Figura 27 mostra o resultado do *pair plot* rotulando os dados como proposto anteriormente.

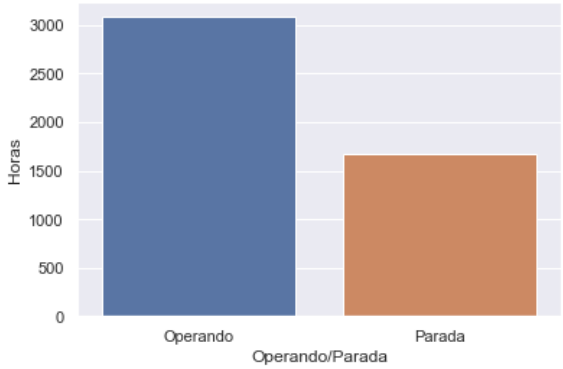
Figura 27 – *Pair plot* para rótulos “operando” e “parada”



Fonte: Autor (2023)

Observando os gráficos de distribuição normal de cada coluna, consegue-se perceber a separação dos dados com relação aos dados de “operando” (em azul) e “parada” (em laranja), separação essa que já era esperada, devido a caldeira quando operando apresenta valores diferentes de zero e quando está parada, possui valores iguais a zero. Com relação a dispersão de dados, os dados de rótulo “operando” possuem uma dispersão muito maior que dos dados de rótulo “parada”, o que fornece uma falsa impressão que há poucos dados de “parada”. Para esclarecer melhor essa questão, um gráfico do tipo histograma pode ser útil, como a Figura 28 prova.

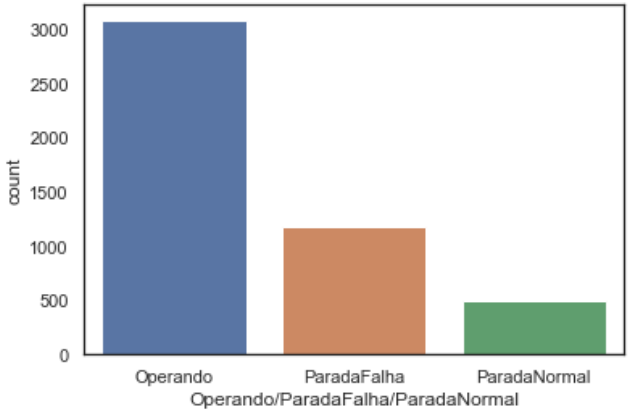
Figura 28 – Histograma para “operando” e “parada”



Fonte: Autor (2023)

Com o gráfico de histograma consegue-se observar a quantidade de dados para cada rótulo, onde o rótulo “operando” mantém predominância na quantidade de dados, portanto, os dados estão desbalanceados. Porém, como a diferença não é de grande discrepância e os dados de cada rótulo possuem características bem distintas, não será realizado o balanceamento. A partir desse gráfico consegue-se obter a informação de quanto tempo a caldeira ficou operando e parada durante todo espectro de análise e, como pode-se observar, a caldeira fica parada aproximadamente metade do tempo em que opera, portanto pode-se caracterizar com uma relação de ½, ou seja, para cada hora parada, ela opera duas horas. Entretanto, verifica-se que nem todos dados que estão caracterizado como “parada”, significam pane devido a falha, as paradas normais estão contabilizadas também e como visto anteriormente, ela é o segundo maior motivos da caldeira ficar parada. Por esse motivo, realizar a distinção entre parada devido a falha e parada normal é crucial para uma melhor compreensão do tempo total em que a caldeira está parada, o que é visualizado na Figura 29.

Figura 29 – Histograma para “operando”, “parada” e “parada normal”



Fonte: Autor (2023)

Na Figura 29 pode-se observar que o tempo em que a caldeira fica parada devido a falhas é majoritário em relação ao tempo de paradas normais e corresponde a aproximadamente 2/5 (2h de falha para 5h de operação) do tempo de operação, fração consideravelmente alta se tratando de tempo de operação e tempo de parada devido a falhas. Vale acrescentar que incluso na “ParadaFalha” está o rótulo “condições não identificadas”, visto que o mesmo apresenta características mais similares a falhas do que paradas normais.

5.8.3 Seleção de *features* para classificação de “operando” e “parada”

Para a seleção de *features*, primeiramente é realizada a divisão dos dados em variáveis de saída (rótulos “Operando/Parada”) e variáveis de entrada (todas *features* com exceção dos rótulos). Na Figura 30 pode-se visualizar os dados utilizados para essa divisão.

Figura 30 – Dados para “operando” e “parada”

	P_vap_kgf	Depressão_mmca	Vazão_vap_tonh	T_vap_C	Ar_primário_percent	Ar_secundário_percent	Alimentação_percent	Tiragem	Operando/Parada
0	8.3	-6.0	9.4	171.0	53.5	51.6	35.0	75.0	Operando
1	9.3	-5.0	10.4	176.0	52.3	51.6	37.0	71.0	Operando
2	9.0	-8.0	10.6	173.0	53.5	51.6	10.0	73.0	Operando
3	10.4	-10.0	11.3	180.0	26.4	51.6	10.0	49.0	Operando
4	9.0	-8.0	10.5	174.0	24.3	51.6	10.0	40.0	Operando

Fonte: Autor (2023)

Os dados de y estão contidos na coluna de rótulos “Operando/Parada”, já as variáveis de entrada, X, serão todas as outras. A seguir, é chamada a função RFE da biblioteca Scikit-learn e é passado os parâmetros necessários para fazer a seleção, que são: classificador, que é o Classificador de Florestas Aleatórias; número de *features* a serem selecionadas, que serão 5 das 8; o “*step*”, que é o número e *features* a serem eliminadas por interação, que será 1. Após isso, o algoritmo retorna as *features* selecionadas, que a seguir, serão ordenadas também na ordem de prioridade: temperatura; depressão; ar primário; ar secundário; tiragem.

5.8.4 Seleção de parâmetros para classificação de “operando” e “parada”

Para realizar a seleção de parâmetros do estimador de Florestas Aleatórias, é utilizada a função “*GridSearchCV()*” da biblioteca Scikit-learn. O número de dobras que o modelo fará serão cinco, portanto, terão 80% de dados para treino e 20% para teste, além do algoritmo priorizar os parâmetros que maximizam a acurácia do modelo.

Os parâmetros selecionados para o estimador de Florestas Aleatórias foram: “*gini*”, para o critério de divisão (*criterion*); 6, para a profundidade máxima da árvore (*max_depth*); 20, para o número de árvores (*n_estimators*).

5.8.5 Divisão de dados para classificação de “operando” e “parada”

A divisão dos dados em treino e teste foi realizada utilizando a função “*train_test_split*”, que divide o *Data Frame* de maneira randomizada, onde 80% dos dados serão para treino e 20% para teste.

5.8.6 Treinamento e avaliação do modelo para classificação de “operando” e “parada”

No treinamento do algoritmo de Florestas Aleatórias, primeiramente foi passado para o algoritmo os parâmetros selecionados e em seguida realizado o ajuste com a função “*fit()*” utilizando os dados de treino. Em seguida, com a função “*predict()*”, foi realizada a classificação utilizando a variável correlacionada “X” dos dados de teste.

Para avaliar o modelo, foi utilizado a função “*classification_report()*” da Scikit-learn que após informar os valores preditos e os valores reais, ela informa diversas métricas de avaliação, como mostra a Figura 31.

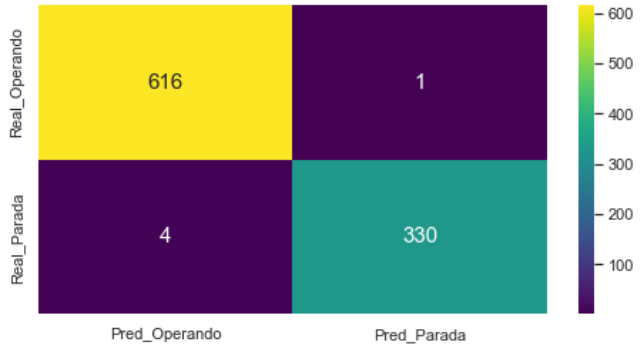
Figura 31 – Métricas de avaliação para “operando” e “parada”

	precision	recall	f1-score	support
Operando	0.99	1.00	1.00	617
Parada	1.00	0.99	0.99	334
accuracy			0.99	951
macro avg	1.00	0.99	0.99	951
weighted avg	0.99	0.99	0.99	951

Fonte: Autor (2023)

Ao visualizar as métricas de avaliação do modelo, observa-se valores próximos a 1 para todas métricas, apresentando uma precisão e revocação altos, que significa baixos ou nenhum falsos positivos e falsos negativos. Por causa disso, a métrica "*f1_score"* também é muito boa, visto que ela é a média harmônica entre as duas anteriores. Por fim, a acurácia total do modelo foi de 0,99. Além do mais, para classificações, é interessante realizar a matriz de confusão para se ter uma visualização melhor dos acertos (verdadeiros positivos), falsos positivos e falsos negativos. A Figura 32 mostra a matriz de confusão para a classificação do estado de "Operando" e "Parada" da caldeira.

Figura 32 – Matriz de confusão para "operando" e "parada"



Fonte: Autor (2023)

Com a matriz de confusão verifica-se os acertos e erros da predição, onde o número de erros é muito inferior perante os acertos e apesar do desbalanceamento dos dados, as duas classes conseguiram ser distintas uma da outra pelo modelo de Florestas Aleatórias.

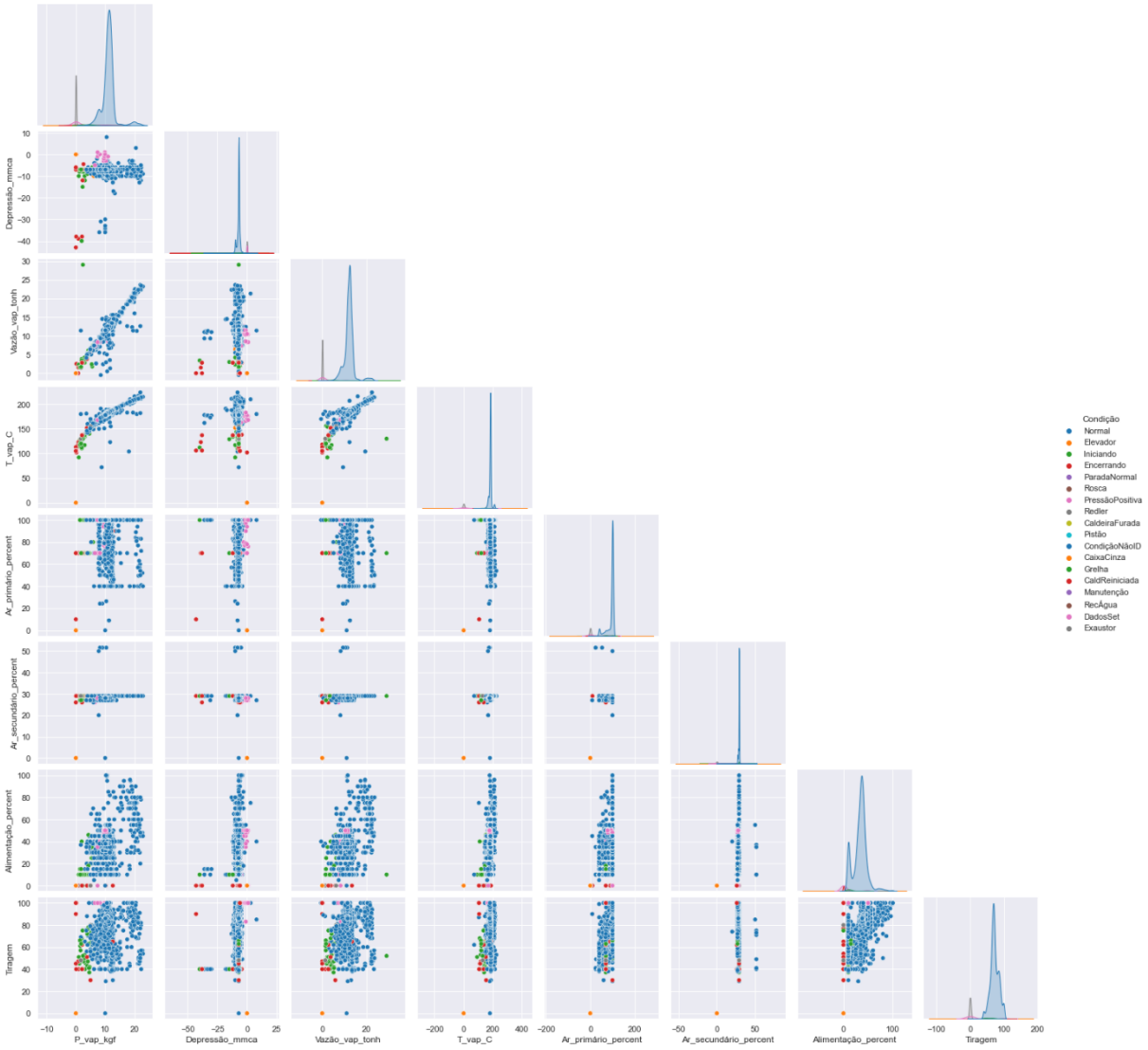
O resultado obtido através da classificação do estado da caldeira de "Operando" e "Parada" de certa forma já era esperado, visto que como pode-se observar na etapa de análise, os dados de caldeira operando possuíam valores normalmente diferentes de zero e de caldeira parada, igual a zero.

5.8.7 Análise dos dados para classificação das condições de falha

Para classificar cada condição em que a caldeira se passa, os rótulos já foram criados inicialmente na digitalização dos dados no em planilhas no Microsoft Excel. Para iniciar as análises, um gráfico do tipo *pair plot* pode fornecer um parâmetro geral

das distribuições dos dados e auxiliar os próximos passos a serem dados. A Figura 33 mostra o gráfico *pair plot* dividido pelos 18 rótulos da coluna “Condição”.

Figura 33 – *Pair plot* para as condições

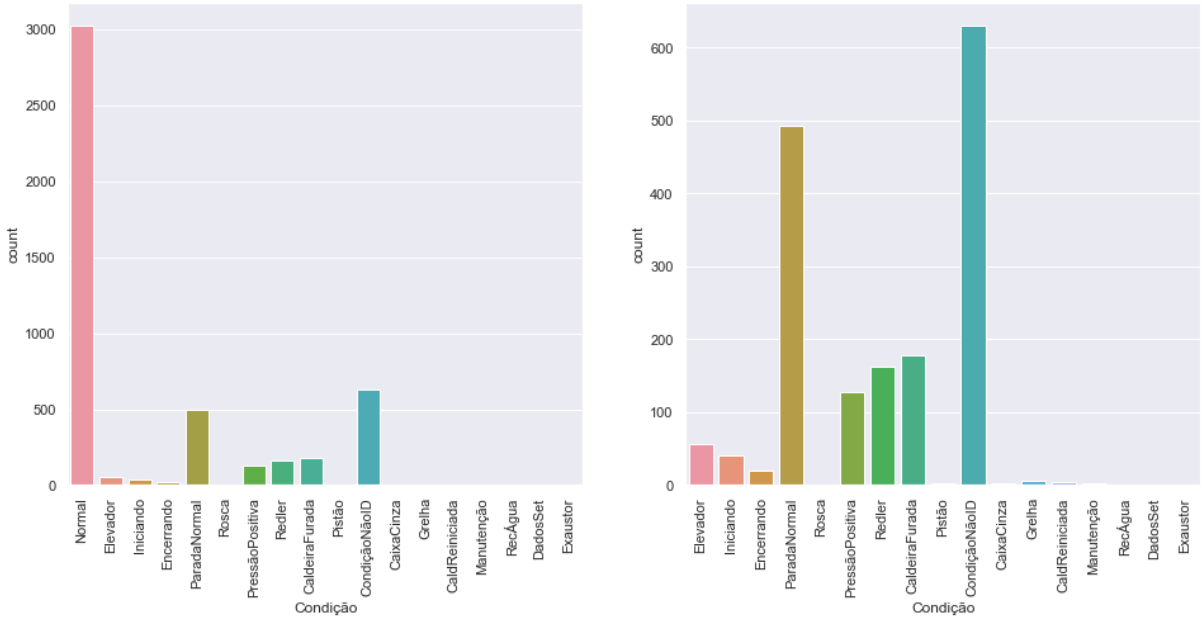


Fonte: Autor (2023)

Ao observar o gráfico *pair plot* acima percebe-se que na legenda há diversos rótulos com diversas cores, no entanto, ao visualizar as distribuições normais e os gráficos de dispersão, não se visualiza a separação desses dados em todos os rótulos, havendo a maior predominância de dados no rótulo de condição “Normal” (em azul). Primeiramente, isso ocorre devido a uma predominância numérica de quantidades de dados para o rótulo “Normal” e, depois disso, pela maioria das falhas não refletirem diretamente nos dados e serem rotuladas somente após a caldeira estar parada, ou

seja, quando os valores são iguais a zero. No entanto, há rótulos como “Encerrando” (em vermelho) e “Iniciando” (em verde) que se consegue distinguir em alguns dos gráficos de dispersão de pontos. Para ter uma perspectiva da quantidade de dados para cada rótulo apresentado, é utilizado dois gráficos de histograma, o primeiro com todas as condições e o segundo, sem a condição normal, que claramente é a que possui mais dados. A Figura 34 representa esses resultados.

Figura 34 – a) Histogramas das condições da caldeira visão geral, b) Histogramas das condições sem “Normal”



(a) (b)

Fonte: Autor (2023)

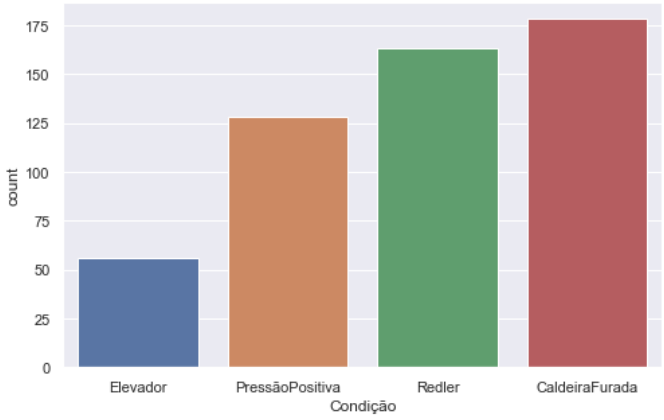
Com a Figura 34 pode-se observar claramente a predominância na quantidade de dados para o rótulo “Normal” e essa análise é a mesma apresentada na Figura 25, com o “tempo total de pane por tipo de falha”, no entanto com a diferença de que agora é contabilizada condições de “Iniciando”, “Encerrando”, “Normal” e “Parada Normal”.

5.8.8 Tratamento de dados para classificação das condições de falha

Como visto na Figura 34, há aquelas falhas ou rótulos que possuem pouquíssimos dados, como: grelha, caldeira reiniciada, manutenção, recuperando água, dados de setembro e exaustor. Rótulos esses que devido a ter poucos dados, dificilmente conseguirão ficar no treino e no teste ao mesmo tempo, por isso devem

ser excluídos. Além do mais, os rótulos de “condição não identificada”, justamente por possuir a possibilidade de ter diversas condições distintas uma da outra dentro do mesmo rótulo, também serão excluídas. Somando a isso, condições de “Normal”, “Parada Normal”, “Encerrando” e “Iniciando” não são consideradas falhas e serão excluídas. A Figura 35 representa o histograma com as mudanças realizadas nos dados.

Figura 35 – Histogramas das condições de falha da caldeira filtrados

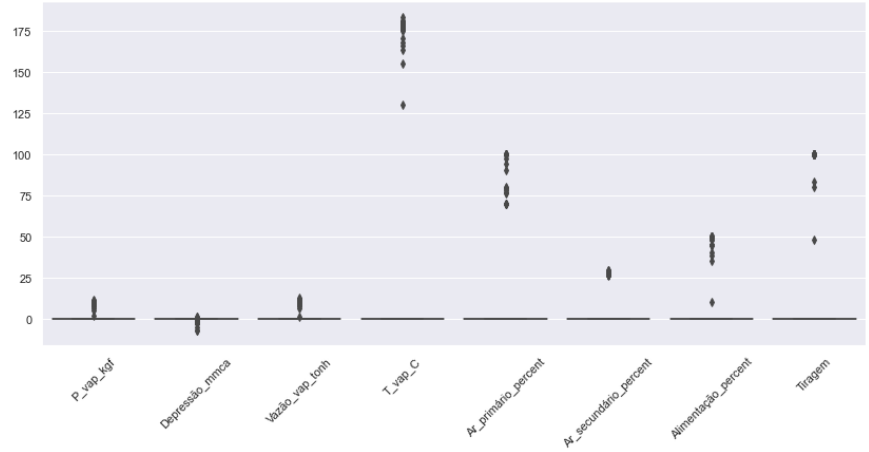


Fonte: Autor (2023)

No total, 14 condições foram excluídas, contendo falhas que continham poucos dados e condições de operação que não são falhas. Das condições restantes, as falhas no elevador são as que menos possuem dados e pode-se considerar que as outras três possuem um número de dados próximos. Todavia, é interessante aplicar o balanceamento de dados para garantir a aprendizagem equivalente do estimador para todas as classes.

No sentido de analisar os *outliers* das variáveis de entrada, o *box plot* do Seaborn pode auxiliar nesse processo. A Figura 36 mostra o gráfico *box plot* para a análise de *outliers* dos dados.

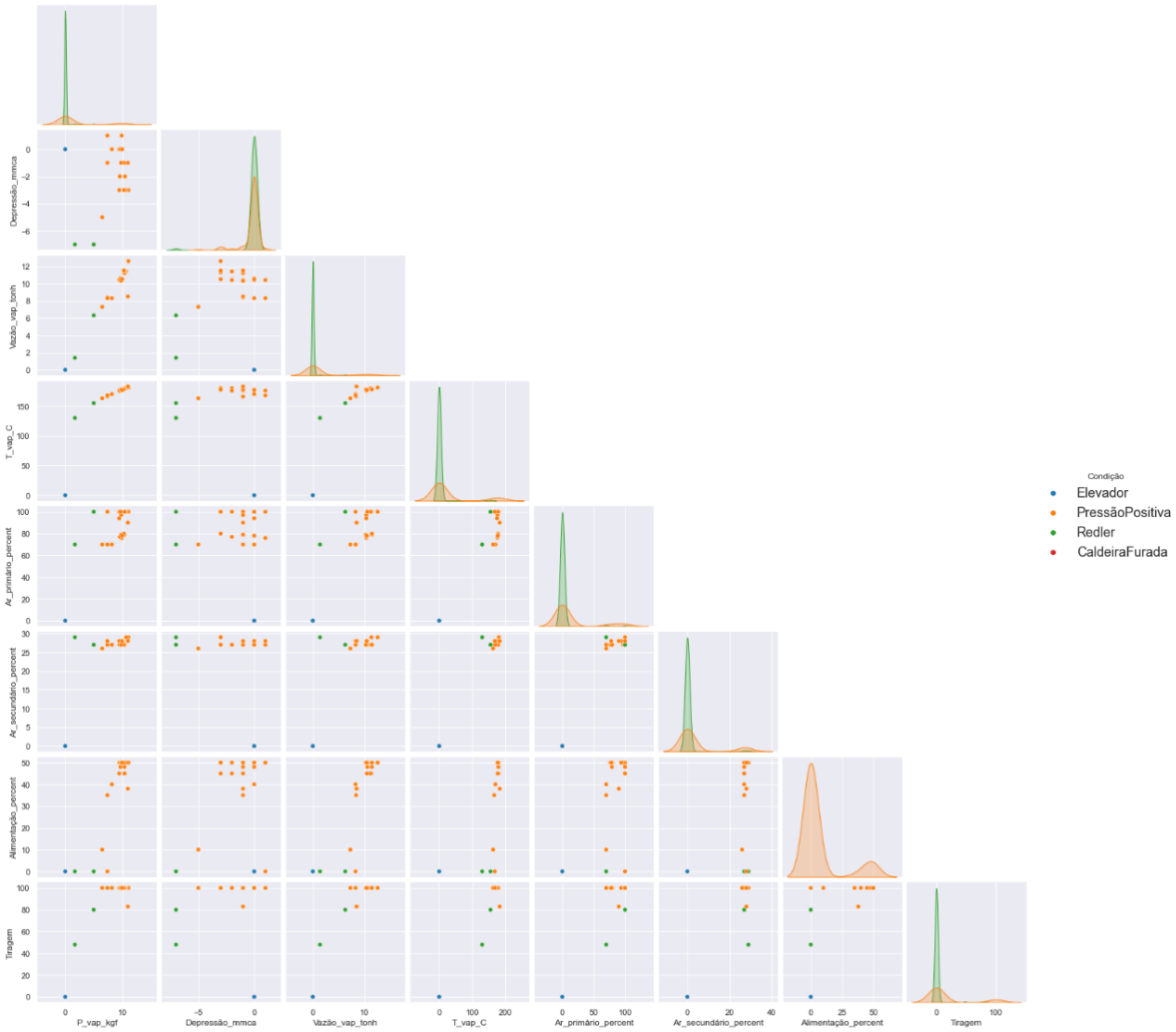
Figura 36 – Análise de *outlier* pelo *box plot* das condições de falha da caldeira



Fonte: Autor (2023)

Como pode-se observar, a predominância de dados iguais a zero faz o *box plot* não apresentar um perfil de intervalo interquartil visível e considera os dados diferentes de zero como *outliers*. No entanto, excluir esses dados diferentes de zero que podem caracterizar alguma das quatro condições não é o ideal. Um gráfico de *pair plot* pode facilitar as visualizações da distribuição e dispersão dos dados perante a as condições escolhidas, como mostra a Figura 37.

Figura 37 – *Pair plot* das condições filtradas de falha da caldeira



Fonte: Autor (2023)

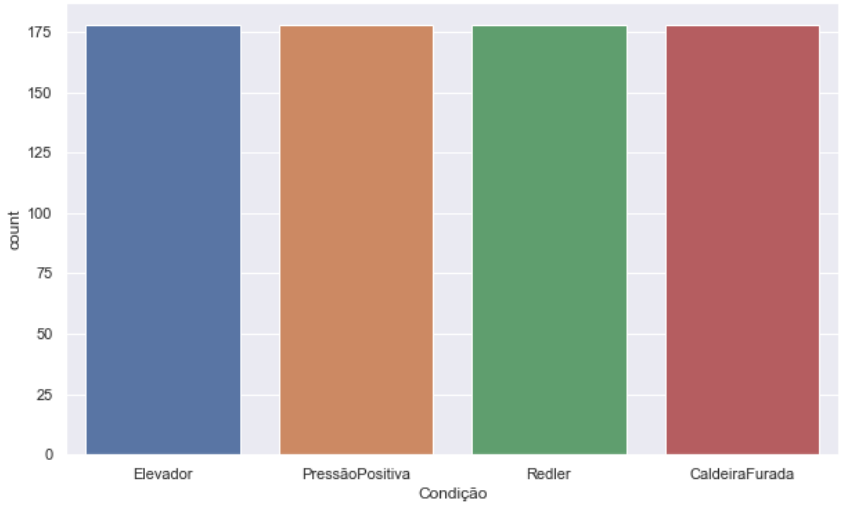
Analisando o gráfico, nota-se nitidamente que a melhor distribuição e dispersão dos dados se encontra na condição de falha de pressão positiva, justamente por se distinguir das demais, que retirando alguns dados de falhas no *redler*, aparentemente os dados das outras condições estão concentrados em zero.

5.8.9 Balanceamento de dados para classificação das condições de falha

Para o balanceamento de dados, primeiramente é realizada a divisão dos dados em variáveis de saída (y) e variáveis de entrada (X). No que se diz a respeito do balanceamento de dados, a técnica a ser utilizada será a SMOTE (*Synthetic Minority Oversampling Technique* ou Técnica de Superamostragem de Minoria Sintética) da biblioteca Imblearn, que seleciona as classes minoritárias e utiliza a técnica de “k vizinhos próximos” para encontrar vizinhos da mesma classe. Em

seguida, seleciona um vizinho aleatoriamente e cria um dado sintético entre esses dados. Dessa forma, a técnica se repete até todas classes estiverem balanceadas com a classe majoritária (caldeira furada). A Figura 38 mostra o resultado do balanceamento de dados através de um histograma.

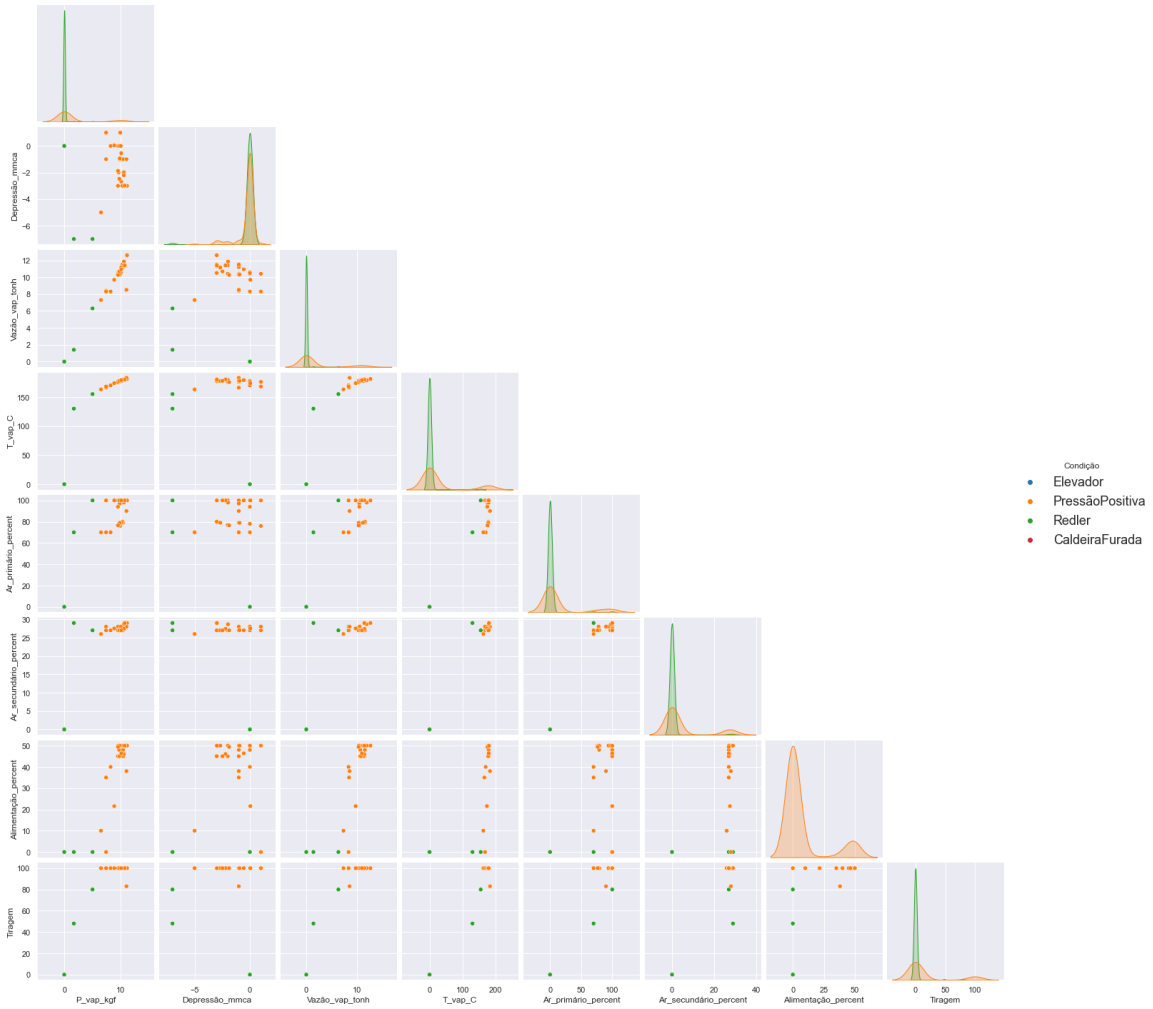
Figura 38 – Histogramas das condições de falha da caldeira balanceadas



Fonte: Autor (2023)

Como pode-se observar, todas as classes foram balanceadas igualadas ao número de dado da classe majoritária. A seguir, o gráfico *pair plot* pode permitir a visualização desses novos dados criados e como está a dispersão desses dados, como mostra a Figura 39.

Figura 39 – *Pair plot* das condições de falha da caldeira filtrados e balanceados



Fonte: Autor (2023)

Visualmente pode-se ver um preenchimento dos “vazios” da dispersão de pontos para a classe de “pressão positiva”, em laranja, já para as outras classes não houve uma mudança visualmente perceptível.

5.8.10 Seleção de *features* para classificação das condições de falha

Com relação a seleção de *features*, ela foi realizada da mesma maneira e com os mesmos parâmetros da classificação de “Operando” e “Parada”. Portanto, é utilizado a função RFE, com os parâmetros de Florestas Aleatórias, número de *features* a serem selecionadas igual a 5 e “*step”* igual a 1. Após isso, o algoritmo retorna as *features* selecionadas, que a seguir, são ordenadas também na ordem de prioridade: pressão de vapor, tiragem, vazão de vapor, alimentação e temperatura de vapor.

5.8.11 Seleção de parâmetros para classificação das condições de falha

Para realizar a seleção de parâmetros do estimador de Florestas Aleatórias, é utilizada a função “*GridSearchCV()*”, a mesma que foi utilizada para a classificação anterior. O número de dobras para a seleção de parâmetros serão cinco e o algoritmo prioriza os parâmetros que maximizem a acurácia do modelo.

Os parâmetros selecionados para o estimador de Florestas Aleatórias foram: “*log_loss*”, para o critério de divisão (*criterion*); 8, para a profundidade máxima de cada árvore; ‘50’, para o número de estimadores, “*n_estimators”*.

5.8.12 Divisão de dados para classificação das condições de falha

A divisão dos dados em treino e teste é realizada utilizando a função “*train_test_split*”, que divide o *Data Frame* de maneira aleatória, onde 80% dos dados serão para treino e 20% para teste.

5.8.13 Treinamento e avaliação do modelo para classificação das condições de falha

Após passar os parâmetros selecionados para o modelo de Florestas Aleatórias, o algoritmo foi treinado com os dados de treino. Em seguida, utilizou-se os dados de teste para realizar a classificação das condições de falha e a função “*classification_report()*” para obter as métricas de avaliação. A Figura 40 expressa os resultados para as métricas de avaliação do modelo de classificação das condições de falha.

Figura 40 – Métricas de avaliação para condições de falha

	precision	recall	f1-score	support
CaldeiraFurada	0.00	0.00	0.00	42
Elevador	0.26	1.00	0.42	36
PressãoPositiva	1.00	0.21	0.34	29
Redler	0.00	0.00	0.00	36
accuracy			0.29	143
macro avg	0.32	0.30	0.19	143
weighted avg	0.27	0.29	0.17	143

Fonte: Autor (2023)

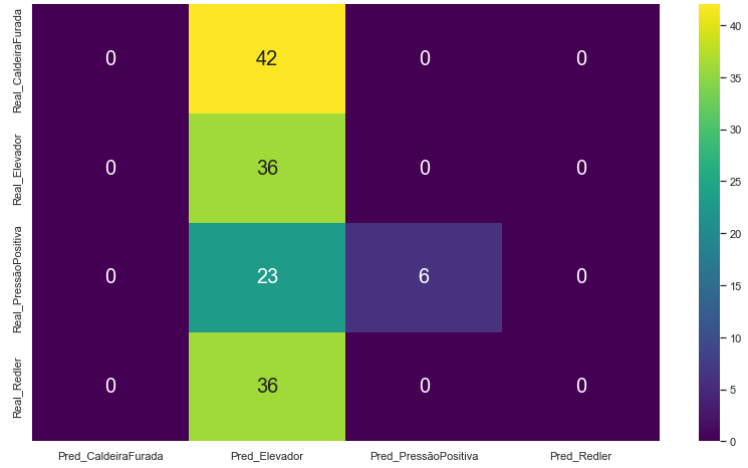
Analisando os resultados obtidos através das métricas de avaliação, pode-se afirmar que para as classes de “caldeira furada” e “*redler*” o algoritmo teve um péssimo desempenho na estimativa, visto que não conseguiu acertar nenhum dos dados que pertenciam a essas classes.

Para a classe de condição de falha “elevador”, o algoritmo possui um desempenho melhor do que os analisados anteriormente, porém, ainda ruim. Possui uma precisão baixa, 0,26, o que significa que há muitos dados pertencentes a outras classes que foram classificados como falhas no elevador (falsos positivos). No entanto, o *recall* apresentou um resultado de 1, significando que todos os dados de teste da classe de falhas no elevador, foram classificados corretamente. Por possuir o *recall* com bom desempenho e a precisão com um mau, o f1_score acaba “pesando” sempre para a métrica com pior desempenho, por isso seu resultado foi de 0,42.

Com uma avaliação similar em alguns pontos com a classificação de falhas no elevador, as falhas de “pressão positiva” obtiveram também um resultado ruim. Ao contrário da classe avaliada anteriormente, a precisão foi de 1, o que significa que não há falsos positivos. No entanto, apresentou *recall* com um baixo desempenho, de 0,21, que indica que há diversos falsos negativos na classificação. A fim avaliar essas duas métricas em uma só, o f1_score apresentado para essa classe foi de 0,34, um pouco pior que o para falhas no elevador.

Para avaliar como o algoritmo classificou os dados, a matriz de confusão pode auxiliar a entender melhor o seu desempenho. A Figura 41 mostra a matriz de confusão para classificação de condições de falha.

Figura 41 – Matriz de confusão para condições de falha



Fonte: Autor (2023)

Com a matriz de confusão consegue-se aumentar a compreensão de como o algoritmo está classificando cada classe. Primeiramente, todos os dados das condições de falha de caldeira furada e no *redle*r estão sendo confundidos com falhas no elevador, o que explica o péssimo desempenho deles e o motivo da precisão da classe “elevador” ser tão baixa. Somando a isso, a maioria dos dados da classe “pressão positiva” também são confundidos com a classe “elevador”, porém, nenhum dos dados é confundido com a classe “pressão positiva”, o que consolida o resultado da sua precisão de 1.

O baixo desempenho nessa predição é principalmente devido a maioria das condições de falhas possuírem dados iguais a zero (devido a ser identificada somente em pane e não durante a operação), o que possibilita a confusão do algoritmo perante as classes. Até mesmo para as falhas de pressão positiva, que apesar de não aparentar nos gráficos de *pair plot* visualizados anteriormente, também possuem dados iguais a zero. Visto isso, em cada vez que o algoritmo é executado o algoritmo pode ter um desempenho diferente similar ao obtido para classe “elevador”, no entanto dessa vez com a classe de “caldeira furada” e “*redler*”, ou até mesmo com falhas de pressão positiva, que apesar de menos prováveis ainda possuem um número considerável de dados iguais a zero.

5.9 Estudo de manutenção preditiva através da vida útil remanescente

Nesta etapa será realizado o estudo da manutenção preditiva através da vida útil remanescente (VUR) da caldeira.

5.9.1 Pré-tratamento de dados para manutenção preditiva

Para realizar o estudo da vida útil remanescente foi utilizado o *Data Frame* “dados sem tempo de pane”, em formato CSV, onde a leitura do mesmo pela função “*head()*” é mostrada na Figura 42.

Figura 42 – a) Primeiras nove colunas dos dados para manutenção preditiva, b) Últimas duas colunas dos dados para manutenção preditiva

	Horas	P_vap_kgf	Depressão_mmca	Vazão_vap_tonh	T_vap_C	Ar_primário_percent	Ar_secundário_percent	Alimentação_percent	Tiragem
0	0.0	8.3	-6.0	9.4	171.0	53.5	51.6	35.0	75.0
1	1.0	9.3	-5.0	10.4	176.0	52.3	51.6	37.0	71.0
2	2.0	9.0	-8.0	10.6	173.0	53.5	51.6	10.0	73.0
3	3.0	10.4	-10.0	11.3	180.0	26.4	51.6	10.0	49.0
4	4.0	9.0	-8.0	10.5	174.0	24.3	51.6	10.0	40.0

(a)

	Tempo_até_falha	Ciclo_falha
0	1.0	1.0
1	2.0	1.0
2	3.0	1.0
3	4.0	1.0
4	5.0	1.0

(b)

Fonte: Autor (2023)

Após visualizar os dados, uma verificação se há dados nulos com a função "*isnull()*" aliado a "*sum()"* é realizada e os resultados são expressos pela Figura 43.

Figura 43 – Análise de colunas com dados nulos

```
Horas                 3
P_vap_kgf             3
Depressão_mmca        3
Vazão_vap_tonh        3
T_vap_C               3
Ar_primário_percent   3
Ar_secundário_percent 3
Alimentação_percent   3
Tiragem               3
Tempo_até_falha       3
Ciclo_falha           3
dtype: int64
```

Fonte: Autor (2023)

Como pode-se visualizar, os dados possuem alguns valores vazios, onde através da função "*dropna()*", serão retirados. Com isso, através da função "*shape"*, obtém-se o tamanho do *Data Frame*, que é de 11 colunas com 3078 linhas.

Ao observar as figuras acima, observa-se que ainda não há a coluna da vida útil remanescente (VUR), portanto ela deverá ser criada logo no início para prosseguir para a etapa de análise de dados. A VUR da caldeira será calculada utilizando a coluna "tempo até falha" (em horas) – que funciona como um contador de tempo para ciclo de falha – onde para cada linha será calculado utilizando a Equação 6.

$$VUR = taf_i - taf_{max,ciclo} \quad (6)$$

Onde a taf_i é o tempo até a falha de índice *i* e o $taf_{max,ciclo}$ é o tempo até a falha máxima do ciclo em que o índice *i* está inserido. Dessa forma, a coluna VUR é criada, utilizando funções como "*groupby()*", "*transform()*" e "*max()*". Um ponto interessante a

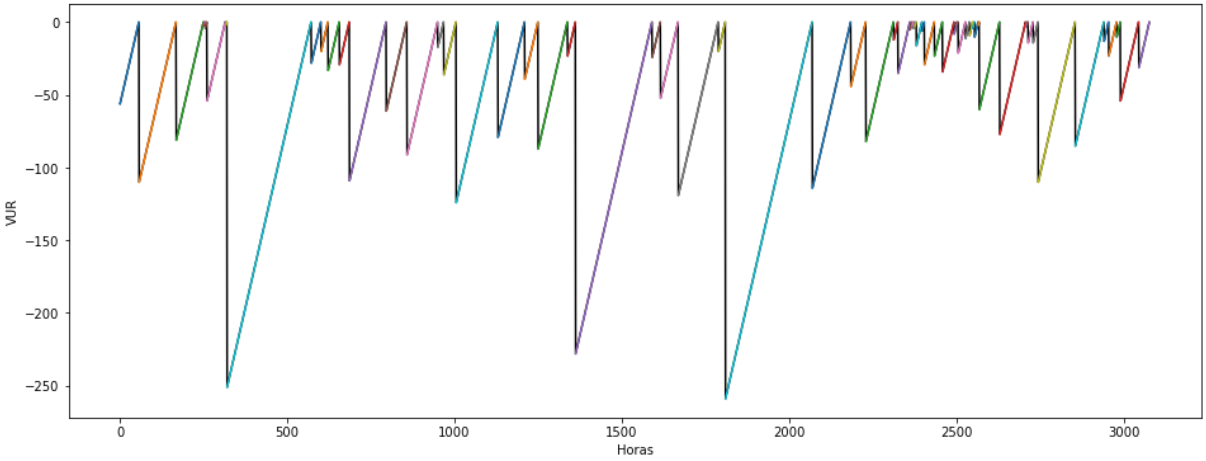
esclarecer é que o VUR terá valor negativo, então quanto mais negativo o valor, mais vida útil resta para a caldeira e no momento em que chega a zero, significa que ocorreu a falha. A VUR foi deixada negativo por questão de leitura da série temporal em gráficos, visto que quando positiva, a leitura correta seria feita da direita para esquerda no eixo das abscissas, o que tornaria a interpretação mais difícil e atípica.

A coluna “tempo até falha” é retirada com a função “*drop()*”, visto que seu propósito foi cumprido e não será mais necessária no *Data Frame*.

5.9.2 Análise de dados para manutenção preditiva

Como já visto anteriormente, há 65 ciclos de vida na caldeira, o qual agora há 65 ciclos de vida útil remanescente. Para poder visualizar esses ciclos e suas respectivas VUR, pode-se utilizar um gráfico de linhas do Seaborn através da função “*lineplot()*”, como mostra a Figura 44.

Figura 44 – Vida útil remanescente ao longo das horas



Fonte: Autor (2023)

O gráfico da VUR ao longo das horas é muito similar ao do tempo até falha ao longo das horas, apresentado na Figura 22, no entanto, ele não apresenta os tempos de pane (entre ciclos) e de parada normal (ao longo dos ciclos). Ao observar o gráfico, consegue-se notar a presença de três grandes ciclos que se destoam perante aos demais e podem ser considerados como *outliers*, assunto que será visto mais adiante. Além do mais, apesar de ser difícil observar, nota-se ciclos extremamente curtos, principalmente a cerca de 2500 horas. Assim como os ciclos muito longos, ciclos

pequenos demais podem atrapalhar no aprendizado e do modelo e é bem improvável que o modelo consiga predizer tais ciclos.

A função “*describe()*” da biblioteca Pandas pode mostrar diversas informações úteis e pode fornecer uma visão geral dos dados, apresentando máximos, mínimos, desvio padrão, média e intervalo interquartil. A Figura 45 mostra o *describe* para a o *Data Frame* em análise.

Figura 45 – a) *Describe* dos dados para manutenção preditiva, b) Continuação do *describe* dos dados para manutenção preditiva

	Horas	P_vap_kgf	Depressão_mmca	Vazão_vap_tonh	T_vap_C	Ar_primário_percent	Ar_secundário_percent	Alimentação_percent	Tiragem
count	3078.00000	3078.000000	3078.000000	3078.000000	3078.000000	3078.000000	3078.000000	3078.000000	3078.000000
mean	1538.50000	11.032609	-7.350065	11.915221	180.715237	92.185737	28.811761	33.386940	72.051170
std	888.68639	2.411878	1.912538	2.488483	9.444528	16.505818	1.334097	13.919643	12.555209
min	0.00000	0.100000	-38.000000	-0.500000	72.000000	0.000000	0.000000	0.000000	0.000000
25%	769.25000	10.300000	-7.000000	11.300000	179.000000	96.000000	29.000000	28.000000	65.000000
50%	1538.50000	11.100000	-7.000000	12.300000	182.000000	100.000000	29.000000	35.000000	71.000000
75%	2307.75000	11.800000	-7.000000	12.500000	184.000000	100.000000	29.000000	40.000000	80.000000
max	3077.00000	25.000000	8.000000	23.600000	224.000000	100.000000	51.600000	100.000000	100.000000

(a)

	Ciclo_falha	VUR
count	3078.000000	3078.000000
mean	27.905458	-56.553606
std	17.306406	57.667146
min	1.000000	-259.000000
25%	15.000000	-78.000000
50%	25.000000	-37.000000
75%	33.000000	-14.000000
max	65.000000	0.000000

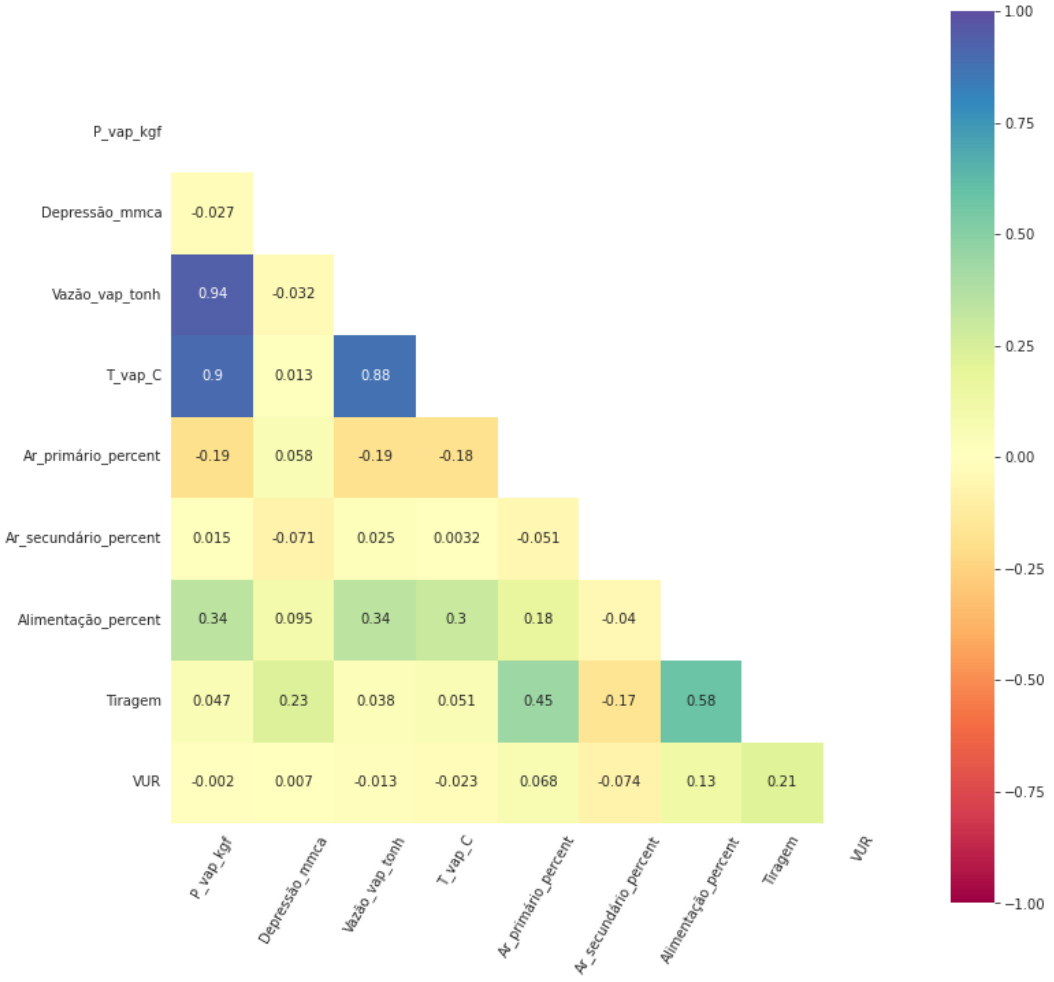
(b)

Fonte: Autor (2023)

Em uma análise prévia, observa-se que colunas de depressão e ar secundário possuem desvio padrão (std) relativamente baixos, principalmente comparadas a outras colunas. Além disso, nota-se que o intervalo interquartil (25%, 50% e 75%) está concentrado em apenas um valor para as duas, o que fortifica a pouca variação nos dados para ambas colunas e ajuda a desqualificar ambas colunas como boas *features*. Em relação ao ar secundário, essa estatística reforça o que já era pressuposto na coleta de dados, onde praticamente não havia variação, além de não variar também quando se aproximava de falhas. Em contraponto, a depressão da caldeira, em uma observação na própria coleta de dados, responde a falhas do tipo “pressão positiva”, fato confirmado pela própria empresa.

Para se extrair mais informações, uma análise do coeficiente de linearidade de Pearson através de um mapa de calor é de extrema importância nesta etapa do processo de análise. O Seaborn com a função “*heatmap*()” pode fornecer tais informações, como mostra a Figura 46.

Figura 46 – *Heatmap* dos dados para manutenção preditiva

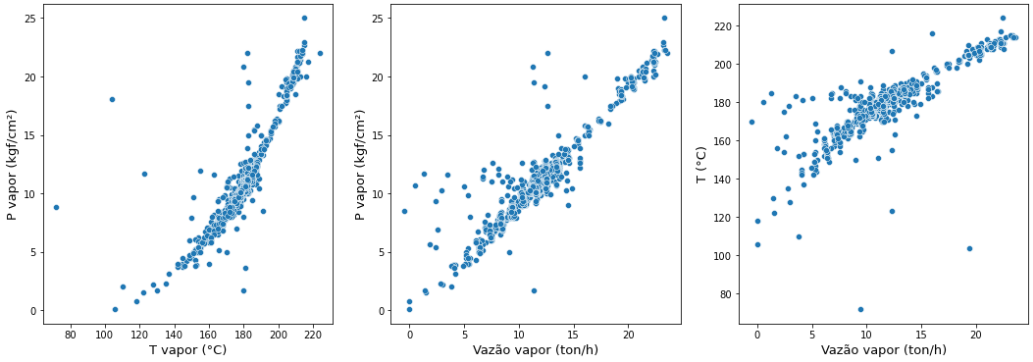


Fonte: Autor (2023)

Ao analisar os valores das correlações pelo mapa de calor produzido acima, observa-se claramente três fortes correlações com tendências lineares positivas, que são: vazão de vapor, pressão de vapor e temperatura de vapor. Além de serem três parâmetros correlacionados ao próprio vapor, são variáveis comumente correlacionadas em equações na literatura. Nesse sentido, pode-se cogitar em realizar alguma técnica de redução de dimensionalidade nos dados para essas três *features*, como PCA, visto que aparentemente elas contêm a mesma informação. Além disso, os parâmetros de ar primário e alimentação possuem uma leve tendência linear positiva com relação a tiragem.

A Figura 47 mostra três gráficos de dispersão de pontos correlacionando as colunas temperatura de vapor, vazão de vapor e pressão de vapor entre si.

Figura 47 – a) Correlação pressão e temperatura, b) Correlação pressão e vazão, c) Correlação temperatura e vazão



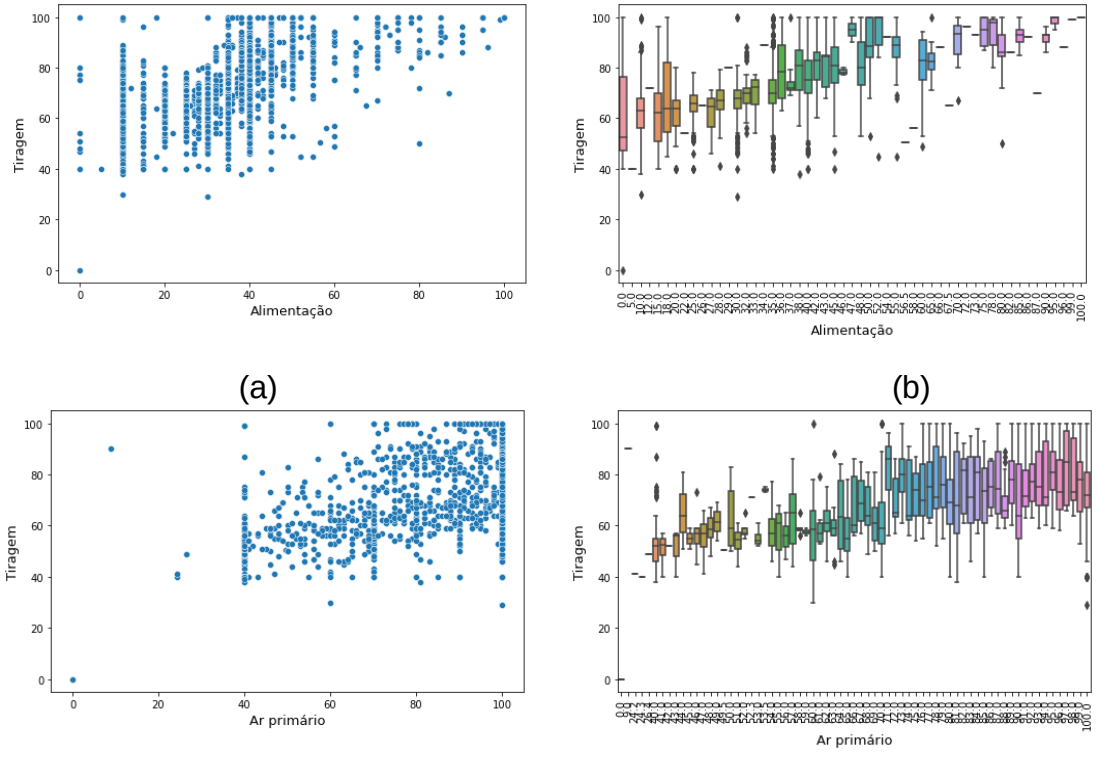
(a) (b) (c)

Fonte: Autor (2023)

Como pode-se observar, as três colunas estão altamente correlacionadas entre si, o que confirma a análise realizada através do mapa de calor.

A Figura 48 mostra três gráficos de dispersão de pontos e outro *box plot* correlacionando as colunas de alimentação e ar primário com a tiragem.

Figura 48 – a) Dispersão de ponto tiragem e alimentação, b) *Box plot* tiragem e alimentação, c) Dispersão de ponto tiragem e ar primário, d) *Box plot* tiragem e ar primário



(a) (b)

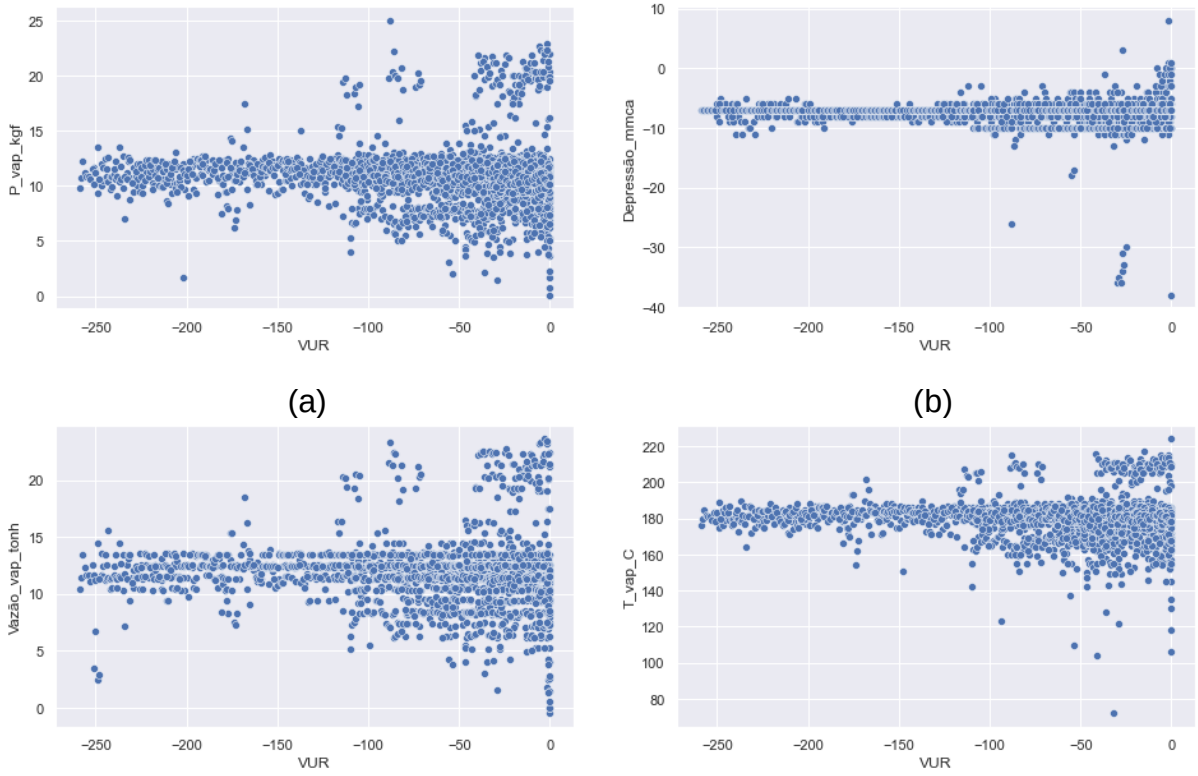
(c) (d)

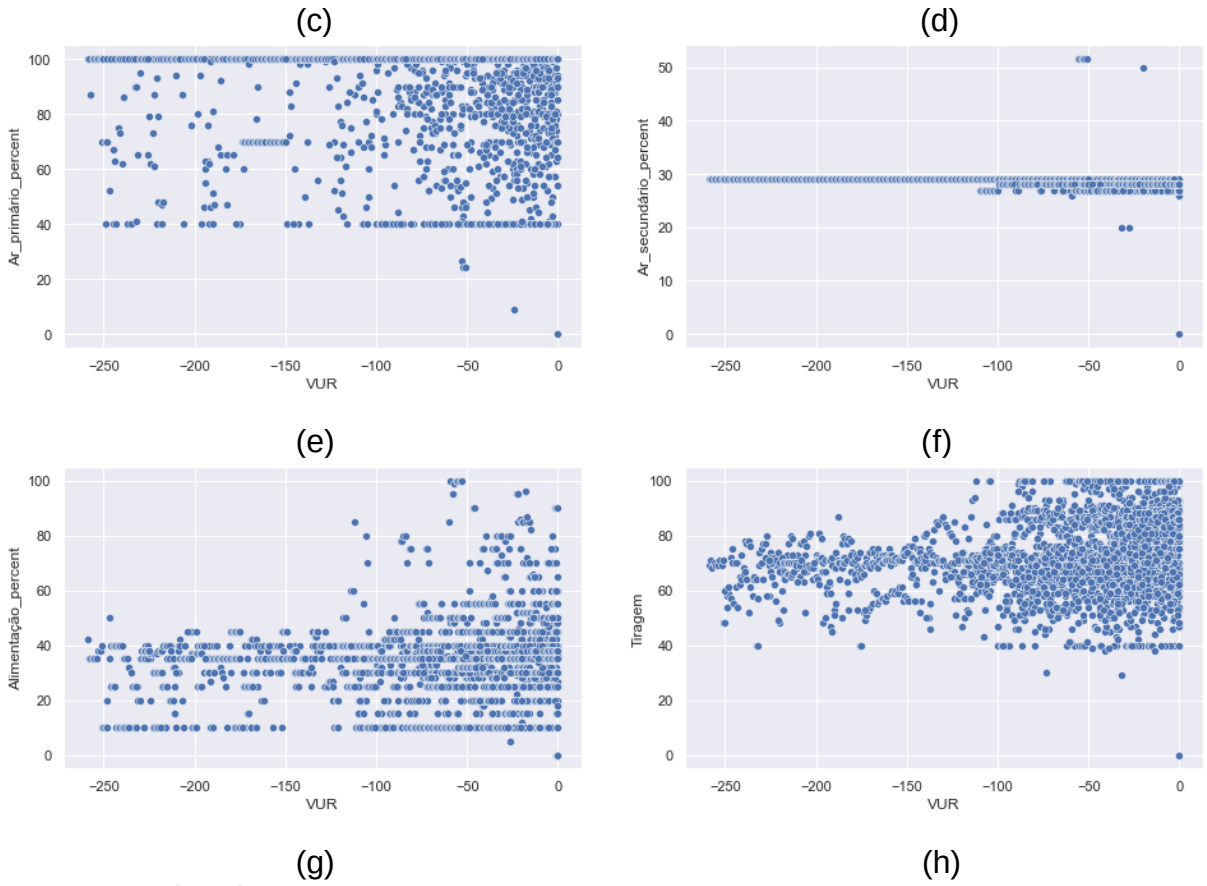
Fonte: Autor (2023)

As tendências lineares positivas da alimentação e ar primário com a tiragem são mais difíceis de ver com o gráfico de dispersão de pontos, onde não estão totalmente definidas pela dispersão ser muito alta. Por isso, é melhor observada através do *box plot*, que consegue definir melhor o gráfico em lugares com maior concentração de dados.

Ao observar o mapa de calor é possível perceber que não há nenhuma correlação linear entre as colunas de parâmetros da caldeira com a vida útil remanesce e em termos de aprendizado de um algoritmo para regressão, isso mostra que talvez as falhas, ou pelo menos a maioria delas, não transparecem diretamente nos dados. Em consequência disso, será feito um gráfico de dispersão de pontos para cada coluna de dados em relação a VUR, a fim de analisar seu comportamento. A Figura 49 mostra um gráfico de dispersão de pontos para cada coluna de dados com relação a VUR.

Figura 49 – a) Pressão pela VUR, b) Depressão pela VUR, c) Vazão pela VUR, d) Temperatura pela VUR, e) Ar primário pela VUR, f) Ar secundário pela VUR, g) Alimentação pela VUR, h) Tiragem pela VUR





Fonte: Autor (2023)

Em uma análise visual, os dados claramente não possuem um comportamento definido conforme a VUR decai, fato esse que dificulta o aprendizado e predição do algoritmo.

Os gráficos da pressão (Figura 49a), vazão (Figura 49c) e temperatura (Figura 49d) de vapor, como já visto anteriormente, possuem um comportamento muito similar, o que reforça a afirmação de que eles possuem a mesma informação, portanto, será realizado o PCA nas etapas seguintes. Além do mais, observa-se que a caldeira, quando operando a vazão, temperatura e pressão de vapor elevadas, possui um tempo de vida útil remanescente menor.

Reforçando o comentado a respeito do ar secundário, pode-se notar agora, no gráfico (Figura 49f), que ele praticamente se mantém constante durante toda vida útil remanescente, portanto, não irá auxiliar no aprendizado do algoritmo e será excluído futuramente. Junto a isso, o ar primário, visível no gráfico (Figura 49e), opera em uma faixa de grande dispersão e visualmente não aparenta ter nenhum comportamento que influencie a VUR, portanto, também será excluído.

Junto ao ar secundário, analisou-se também a depressão, que não possuía um intervalo interquartil definido e através do gráfico (Figura 49b), consegue-se observar essa constância nos dados. No entanto, é possível identificar que, para depressões maiores (mais próximas de zero), a caldeira pode estar próxima a falha.

Também, pode-se perceber que alimentações maiores da caldeira, visualizada no gráfico (Figura 49g), podem significar um VUR menor, assim como para a tiragem (Figura 49h).

A diferença de escala entre os dados pode dificultar o aprendizado do algoritmo de *Machine Learning,* o que pode levar ao estimador ser influenciado mais por uma *feature* do que por outra devido a escala. Por isso, uma técnica de padronização de dados como a z-score se faz necessária.

5.9.3 Padronização de dados para manutenção preditiva

A padronização visa transformar todas as colunas de dados que possuem diferentes escalas em uma mesma ordem de grandeza. Para isso, pode-se utilizar a metodologia z-score, que contabiliza quantos desvios padrão o ponto "*n*" está distante da média, que estará próxima a zero e, dessa forma, deixará todos as colunas de dados na mesma escala.

A fim de efetuar a padronização, a função "*stats"* da biblioteca SciPy será utilizada. Onde através dela, com auxílio da função "*apply()*", será aplicada a padronização através do z-score. Além do mais, as colunas da VUR, horas e ciclo de falha devem ser mantidas em seu estado original pois não necessitam padronização.

Para verificar se o método foi realizado com sucesso, os desvios padrão das colunas devem ser iguais a 1 e as médias iguais a 0. A Figura 50 mostra a função "*describe()*" para o *Data Frame* padronizado.

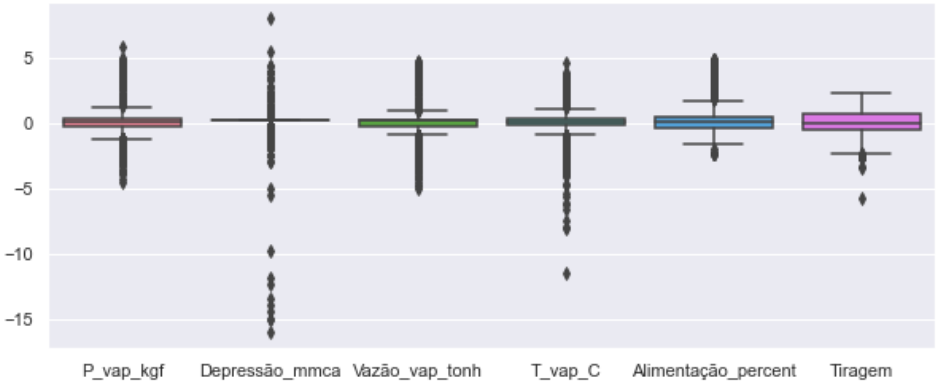
Figura 50 – *Describe* para os dados padronizados

	P_vap_kgf	Depressão_mmca	Vazão_vap_tonh	T_vap_C	Alimentação_percent	Tiragem
count	3.078000e+03	3.078000e+03	3.078000e+03	3.078000e+03	3.078000e+03	3.078000e+03
mean	8.894409e-16	-1.478764e-15	2.308456e-18	-2.790346e-16	4.151253e-16	-1.399069e-15
std	1.000162e+00	1.000162e+00	1.000162e+00	1.000162e+00	1.000162e+00	1.000162e+00
min	-4.533557e+00	-1.602840e+01	-4.989883e+00	-1.151279e+01	-2.398938e+00	-5.739679e+00
25%	-3.037998e-01	1.830666e-01	-2.472675e-01	-1.816413e-01	-3.870656e-01	-5.617043e-01
50%	2.794591e-02	1.830666e-01	1.546491e-01	1.360546e-01	1.159026e-01	-8.373738e-02
75%	3.182234e-01	1.830666e-01	2.350324e-01	3.478519e-01	4.751655e-01	6.332130e-01
max	5.792027e+00	8.027323e+00	4.696306e+00	4.583797e+00	4.786321e+00	2.226436e+00

Fonte: Autor (2023)

Como pode-se observar, as colunas possuem médias com valores somente nas 15°, 16° e 18° casa decimal, que é praticamente igual a zero aos critérios dessa análise. Além do mais, os desvios padrão são todos iguais e muito próximos a 1. Portanto, pode-se confirmar o sucesso da padronização através da técnica z-score. Para ter uma análise mais visual, o *box plot* mostra os dados já padronizados para todas colunas na Figura 51.

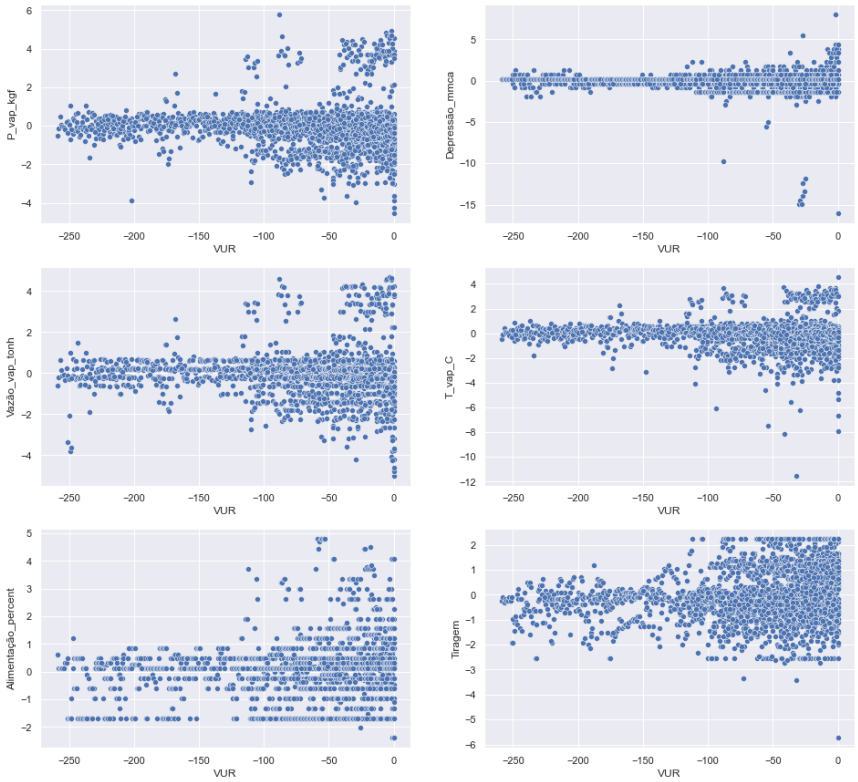
Figura 51 – *Box plot* para os dados padronizados



Fonte: Autor (2023)

Com a Figura 51 acima a padronização dos dados torna-se mais visível. A seguir, a Figura 52 mostra um gráfico de dispersão de pontos para cada coluna de dados com relação a VUR a fim de validar a conservação do comportamento dos dados após a padronização.

Figura 52 – Comportamento dos dados padronizados ao longo da VUR



Fonte: Autor (2023)

Ao observar a Figura 52 e comparar com a Figura 49, pode-se perceber que o comportamento e informações foram mantidas após a padronização.

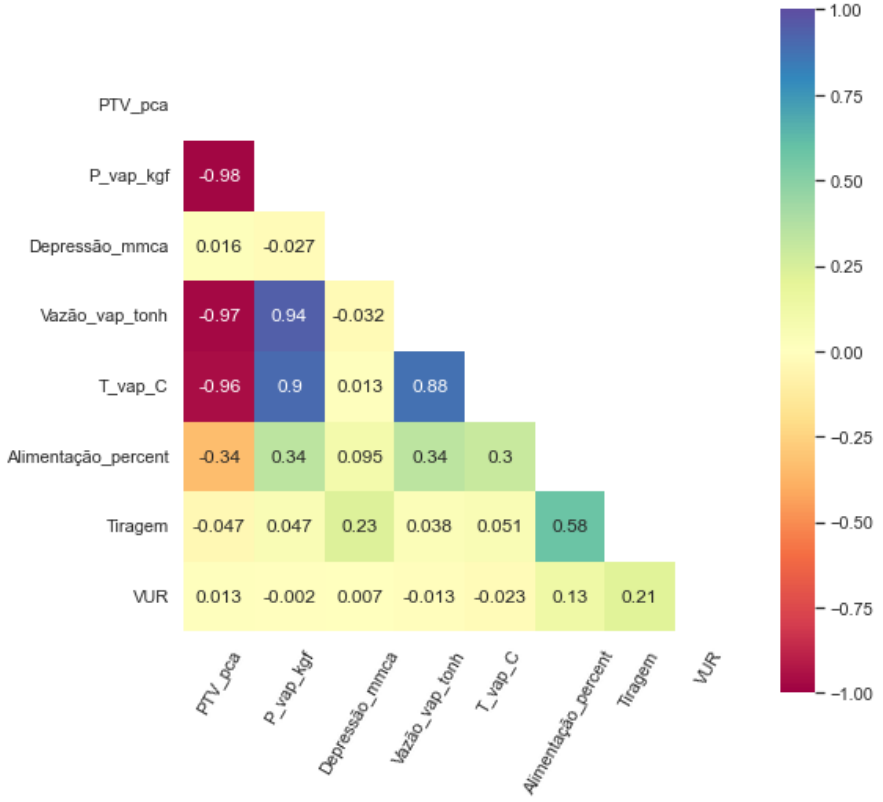
5.9.4 *Principal Component Analysis* para manutenção preditiva

A *Principal Component Analysis* (PCA), ou Análise de Componentes Principais, é uma técnica de aprendizado não supervisionado para redução de dimensionalidade, que de uma forma geral, visa preservar os dados que mais se diferenciam dos demais e eliminar aqueles que possuem a mesma informação. A partir disso, como visto anteriormente, as colunas de dados de pressão, vazão e temperatura de vapor estão altamente correlacionadas e possuem um comportamento muito similar, o que justifica a transformação dessas três colunas em apenas uma preservando os seus dados mais significativos, ou componentes principais.

Para aplicar a redução de dimensionalidade é utilizada a função PCA, disponível através da biblioteca Scikit-learn, com o parâmetro "*n_components*" (número de componentes) igual a 1, visto que o objetivo é retornar somente uma coluna. Em seguida, a função "*fit_transform()*" aplica a redução nas três colunas e

transforma em apenas uma, que será chamada de “PTV_pca” (abreviação para pressão, temperatura e vazão de vapor transformadas pelo PCA). A seguir, a nova coluna criada é adicionada ao *Data Frame* e para verificar o sucesso da técnica PCA em preservar as características das três colunas reduzidas, analisar a correlação da nova coluna criada com as colunas originais através de um mapa de calor pode ser útil para obter essa informação. A Figura 53 mostra um mapa de calor com o coeficiente de Pearson para analisar a correlação da coluna “PTV_pca” com as originais.

Figura 53: *Heatmap* para analisar as correlações da coluna criada com PCA



Fonte: Autor (2023)

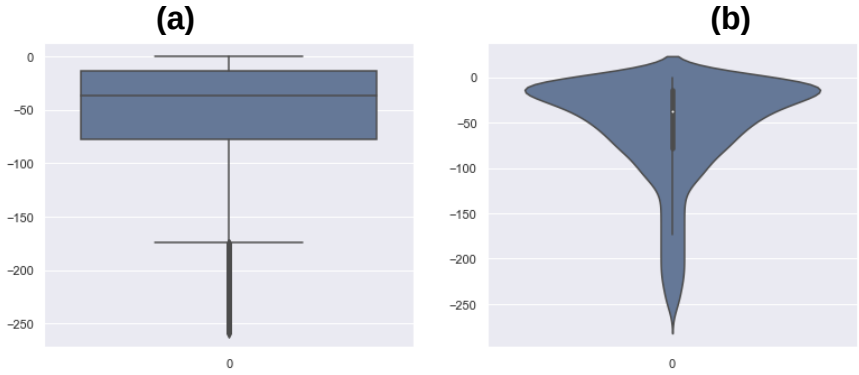
Ao observar o mapa de calor apresentado acima, pode-se perceber a alta correlação linear decrescente (negativa) entre a coluna “PTV_pca” com as colunas de pressão, temperatura e vazão de vapor, que além de afirmar o sucesso da técnica PCA, prova a ortogonalidade da nova coluna de dados em relação as colunas originais, característica do método PCA. A partir disso, as colunas de pressão, temperatura e vazão de vapor não serão mais necessárias, portanto, serão excluídas das análises posteriores.

5.9.5 Tratamento de *outliers* de dados para estudo da VUR

Nesta etapa será realizado a análise de distribuição, densidade e dos *outliers* dos dados e para isso, será utilizado ferramentas gráficas do tipo *box plot* e *violin plot*. Além disso, será executada a retirada dos *outliers* caso haja necessidade.

Primeiramente, será realizada a análise para os dados da vida útil remanescente através dos gráficos citados acima. A Figura 54 representa os gráficos *box plot* e *violin plot* para a VUR.

Figura 54 – Análise de *outliers* para a VUR



Fonte: Autor (2023)

Ao analisar os gráficos acima consegue-se notar uma concentração maior dos dados nas VUR menores, como mostra o intervalo interquartil do gráfico (Figura 54a) com uma amplitude baixa e assimétrica. Reforçando isso, observa-se no gráfico (Figura 54b) uma densidade dos dados bem definida que decresce conforme a VUR aumenta. A respeito dos *outliers*, consegue-se observar que os valores absolutos maiores para a VUR são considerados *outliers*, o que já era esperado, visto que os ciclos com vida útil remanescente longas se diferenciavam dos demais.

Para tratar os *outliers*, uma função para calcular o limite superior e inferior foi criada, onde o valor do limite inferior, que é o que contém os *outliers*, será utilizado para filtrar os dados e eliminar os *outliers*.

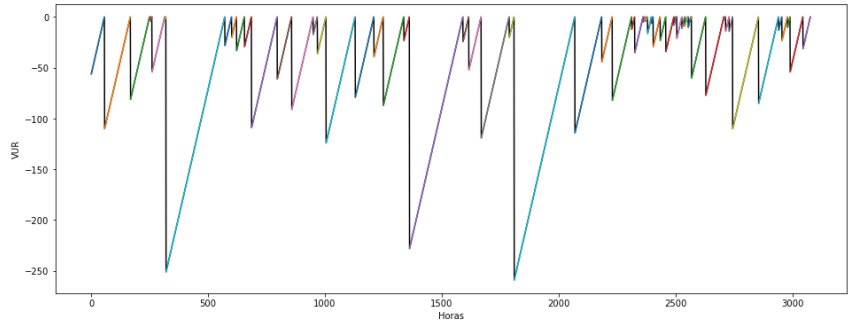
O limite inferior calculado para a VUR pela função é de -174 horas, portanto, todo *index* do *Data Frame* referente aos dados menores que isso. ou com mais horas de vida útil, serão excluídos. A partir disso, um novo gráfico *box plot* é feito para verificar se há novos *outliers* e se houver, novamente é executada a função e excluído

os outliers. Esse processo foi repetido três vezes, sendo o último limite inferior calculado de -139 horas.

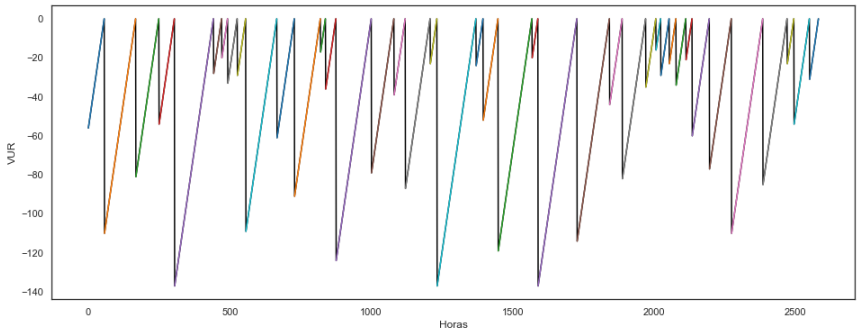
Como mencionado anteriormente, não será somente os ciclos maiores que será considerado *outlier*, mas também aqueles ciclos muito pequenos. Por isso, foi estipulado uma vida útil remanescente mínima de 15 por ciclo, ciclos com VUR menor que isso serão excluídos. Para realizar esse processo de exclusão, não é possível apenas filtrar os dados como da forma anterior, senão serão excluídas todas VUR menor que 15 e isso afetaria todos os ciclos, portanto, serão identificados e excluídos os ciclos inteiros que estão dentro dessa condição, sem afetar os demais. Vale ressaltar que durante todo processo de exclusão de dados é utilizada a função "*reset_index()*" para resetar cada linha em seu novo *index,* visto que ao excluir os dados são formadas "lacunas" nos *index* das linhas.

Para visualizar os ciclos depois da retirada de *outliers* da VUR será plotado um gráfico de linhas da VUR ao longo das horas, além de compará-lo ao antes do tratamento. A Figura 55 mostra os ciclos antes e depois do tratamento das *outliers* da VUR.

Figura 55 – a) Ciclos da VUR antes do tratamento de *outliers*, b) Ciclos da VUR depois do tratamento de *outliers*



(a)



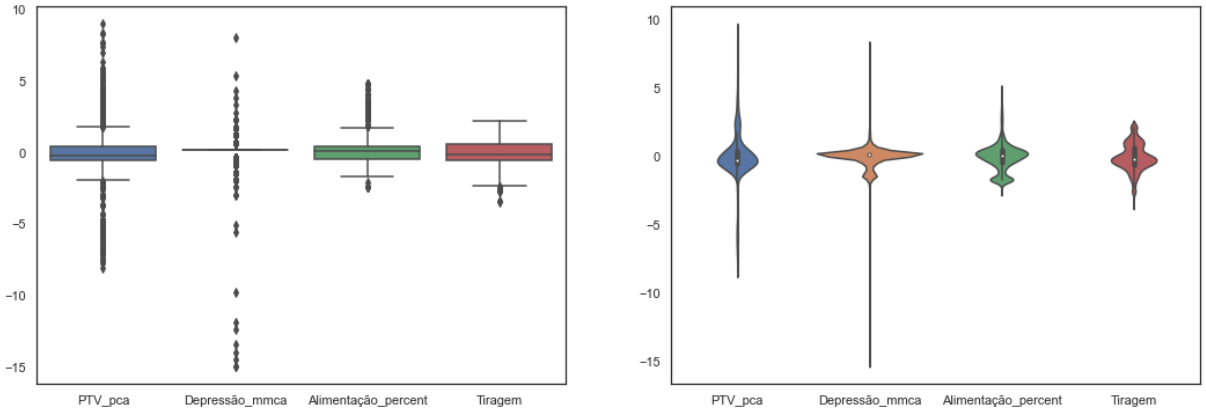
(b)

Fonte: Autor (2023)

Com a eliminação dos *outliers* a grande diferença de VUR entre os ciclos foi retirada e os ciclos estão mais padronizados, além de ter facilitado também a sua visualização. Ao total, foram excluídos 496 dados, que se aproxima cerca de 1/6 do total de dados, que é uma parcela excluída de tamanho considerável devido ao *Data Frame* não possuir muitos dados. Além do mais, foram retirados 24 ciclos e restam 41.

Da mesma forma que foi realizada para o VUR, a análise de distribuição, densidade e dos *outliers* será realizada para todas as colunas de parâmetros da caldeira. A Figura 56 representa os gráficos *box plot* e *violin plot* para os parâmetros da caldeira.

Figura 56 – a) *Box plot* para os parâmetros da caldeira, b) *Violin plot* para os parâmetros da caldeira



(a) (b)

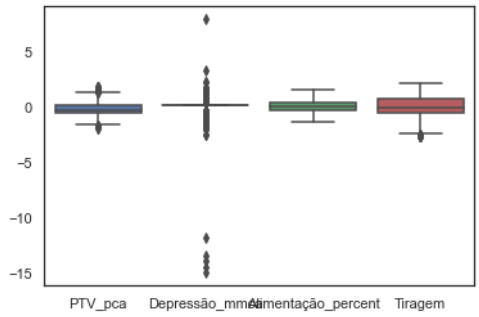
Fonte: Autor (2023)

Analisando os parâmetros PTV (pressão temperatura e vapor), alimentação e tiragem observa-se no gráfico (Figura 56a) que os três apresentam simetria na distribuição dos dados e amplitudes pequenas no intervalo interquartil, o que representa dados bastante concentrados. O que reforça isso são os limites superiores e inferiores que também estão próximos ao primeiro quartil e terceiro quartil. Para o gráfico (Figura 56b), observar-se que o parâmetro PTV possui uma distribuição com densidade bem definida, já para a alimentação e principalmente tiragem, as distribuições possuem a densidade variando, com uma tendência a uma distribuição bimodal. O parâmetro PTV é o que mais possui *outlier* dos três, tanto no limite superior, como inferior. A alimentação possui alguns *outliers* também e a tiragem muito poucos.

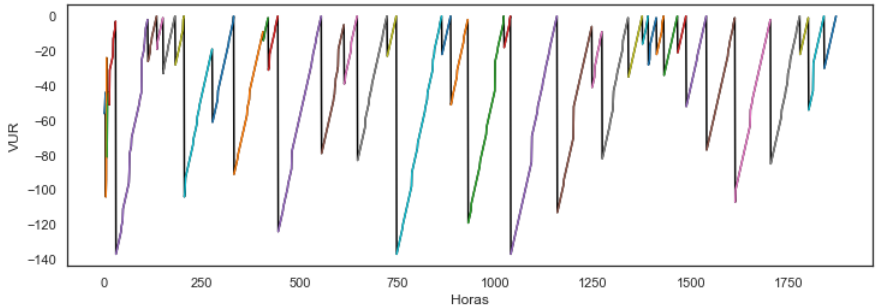
Na análise do gráfico (Figura 56a) para a coluna de depressão, observa-se que o intervalo interquartil e os limites superiores e inferiores estão todos concentrados em apenas um ponto, ressaltando a baixa variância dos dados, que já era esperada devido a análises estatísticas anteriores. O que ressalta também é seu *violin plot* (Figura 56b) que é “comprido” e com amplitude baixa, ressaltando a concentração dos pontos em um lugar só. Como possui quase todos os dados em apenas um lugar, todos os demais são considerados *outliers*. A partir disso, a coluna de depressão da caldeira não se idealiza muito bem para o treinamento do algoritmo, mas essa decisão será deixada para a seleção de *features*. Além do mais, se os *outliers* foram excluídos, a coluna de depressão se tornará constante, portanto, não se filtrará os outliers para depressão.

Utilizando a função para identificação do limite superior e inferior aliados a estrutura de repetição “*for”*, consegue-se os respectivos valores para as quatro colunas de parâmetros da caldeira em questão. Em seguida, os dados são filtrados. A Figura 57 expressa o resultado obtido para a primeira exclusão de *outliers* dos dados e também o comportamento dos ciclos em função do tempo após esse tratamento de *outliers*.

Figura 57 – a) Resultado do tratamento de *outliers box plot*, b) Resultado tratamento de *outliers* para ciclos da VUR



(a)



(b)

Fonte: Autor (2023)

Como pode-se observar no gráfico (Figura 57a) a filtragem realizada excluiu quase todos *outliers* presentes nos parâmetros da caldeira (com exceção da coluna depressão). Em contrapartida, ao analisar o gráfico (Figura 57b), observa-se que a eliminação dos dados desconfigurou o comportamento dos ciclos de falha da caldeira, onde alguns são interrompidos antes de chegar em VUR igual a zero, outros apresentam falhas ao longo de suas vidas úteis. Além do mais, foram excluídos 709 dados, que somando aos já excluídos anteriormente, custa pouco mais de 1/3 do total de dados. Dessa forma, a decisão tomada é de que não serão excluídos os *outliers* das colunas dos parâmetros da caldeira devido à desconfiguração dos ciclos de vida útil da caldeira e da grande perda de dados que o tratamento causa.

5.9.6 Seleção de *features* para estudo da VUR

Nesta etapa a seleção de *features* será realizada com o método RFE, que já foi utilizado anteriormente para o estudo de falhas e será realizado para os três modelos de *Machine Learning* diferentes que irão predizer a VUR, que são: Regressão Linear, Regressor de Florestas Aleatórias e Regressão XGBoost. Para todos, as três primeiras *features* serão selecionadas e a última excluída do treino e teste dos dados.

A ordem de prioridade das *features* obtidas para o modelo de Regressão Linear foi de: tiragem, depressão, alimentação e PTV. Portanto, a coluna “PTV_pca” será excluída com a função “*drop()*”.

Para o Regressor de Florestas Aleatórias, a ordem obtida foi de: PTV, tiragem, alimentação e depressão. A partir disso, a coluna “Depressão_mmca” será excluída.

A ordem obtida para o modelo XGBoost foi: Tiragem, alimentação, depressão PTV. Portanto, a coluna “PTV_pca” será excluída.

5.9.7 Seleção de parâmetros para classificação das condições de falha

Para selecionar os melhores parâmetros para os algoritmos de predição será utilizada a função “*GridSearchCV()*”, onde como critério de avaliação será utilizado o R^2.

Para Regressão Linear, as melhores combinações de parâmetros foram: “*fit_interception*”, para calcular a intercepção do eixo y, igual a *True*; “*positive*”, para forçar os coeficientes a serem positivos, igual a *True*; “*copy_X*”, será *False*.

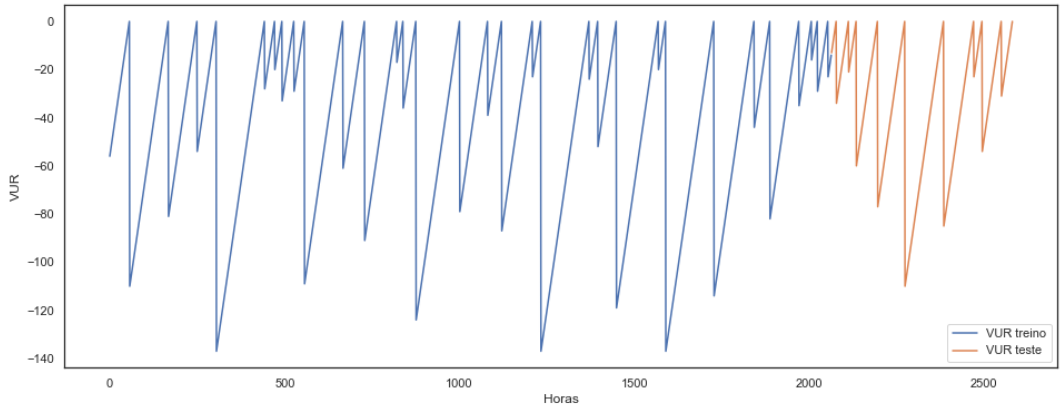
Os parâmetros selecionados para o estimador de Regressão de Florestas Aleatórias foram: “*criterion*”, que mede a qualidade da divisão, será “*squared_error*”; “*max_depth*”, profundidade máxima da árvore, será 5; “*n_estimators*”, número de árvores, será 500.

Para XGBoost, os parâmetros selecionados foram: “*eta*”, utilizado para evitar excesso de ajuste, será 0,2; “*gamma*”, redução mínimo de perda requerida para fazer a divisão adicional do nó folha da árvore, será 2; “*max_depth*”, profundidade da árvore, será 6; “*min_child_weight*”, mínimo de amostras para realizar uma divisão em um nó folha, será 2; “*subsample*”, razão de subamostras das instâncias de treinamento, será 0,1.

5.9.8 Divisão de dados para estudo da VUR

Diferente das outras divisões dos dados em treino e teste realizadas para classificação no estudo de falhas, com a vida útil remanescente não será possível fazer com a função “*train_test_split()*” devido à randomização dos dados. Por isso, a divisão será feita “manualmente” respeitando a série temporal e calculando quantos dados representam 20% do total, sabendo isso, consegue-se dividir os dados pelo *index* divisor de primeiros 80% de dados para treino e últimos 20% para teste. A Figura 58 representa um gráfico de linhas para a VUR em função do tempo divididos em treino e teste.

Figura 58 – Divisão da VUR em treino teste na série temporal



Fonte: Autor (2023)

Como pode-se notar, a divisão dos dados em treino e teste está respeitando a ordem cronológica, característica que é fundamental para predições em série temporal.

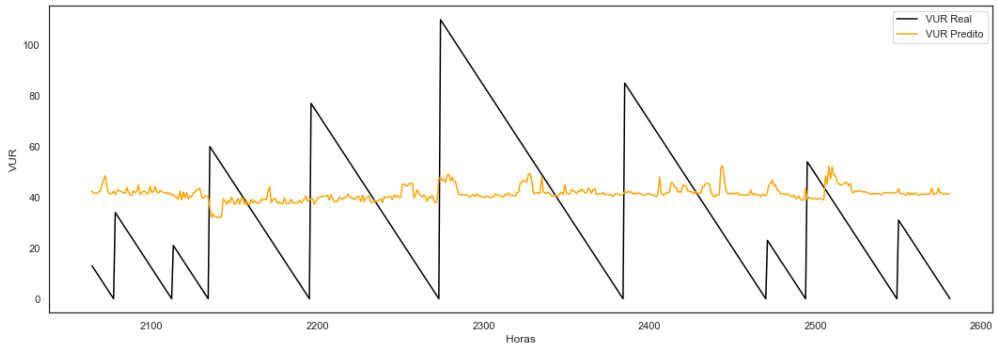
5.9.9 Treinamento e avaliação Regressão Linear para estudo da VUR

Para o treinamento e avaliação do modelo de Regressão Linear será utilizada a função "*LinearRegression()*" da biblioteca Scikit-learn com os parâmetros selecionados, que com auxílio da função "*fit()*" e com os dados de treino selecionados pelo RFE, o modelo é treinado. Em seguida, é realizada a predição com a função "*predict()*" utilizando os dados de teste.

Uma função criada para avaliar o modelo tem como saída as métricas do erro absoluto médio (MAE), raiz do erro quadrático médio (RMSE) e o R^2, além de utilizar as funções fornecidas pela biblioteca Scikit-learn para cálculo das respectivas métricas.

Ao inserir os valores de VUR real e predito na função de avaliação, o retornado é um MAE de 23,90, RMSE de 27,46 e R^2 -7,24%. Ao avaliar as métricas, percebe-se um erro absoluto médio muito alto, que diz basicamente que, em média, a predição está 23,90 horas diferente da real. Além disso, possui uma raiz do erro quadrático médio consideravelmente superior ao erro absoluto médio e com isso pode-se dizer que há diversos erros de predição com grande diferença entre o real e o predito. Por fim, foi obtido um r-quadrado negativo de -7,24%, um valor atípico, que significa que a predição está pior do que se traçasse uma linha constante da média e a utilizasse como preditor. A Figura 59 apresenta a relação da VUR real e predita ao longo das horas através de um gráfico de linhas.

Figura 59 – Comparação do predito e real para Regressão Linear



Fonte: Autor (2023)

No gráfico a linha preta retrata os valores de VUR real e a linha amarela retrata o predito, além da VUR estar representada em sua forma positiva para mais fácil interpretação na comparação das linhas.

A respeito dos resultados, o gráfico permite visualizar a discrepância dos valores reais e os preditos através do modelo de regressão linear. Além disso, pode-se observar uma certa continuidade na predição com alguns ruídos, como se o estimador estivesse “viciado” em predizer alguma métrica como média ou mediana aprendida com o treino. Para sanar essa questão, a média da VUR utilizada no treino (“*y_train”*) foi obtida e seu valor foi de -44,83 horas. Comparando com a média da VUR predita, que foi de -41,40 horas, percebe-se uma grande proximidade entre as duas médias, o que pode validar a hipótese de o modelo estar “viciado” em predizer a média.

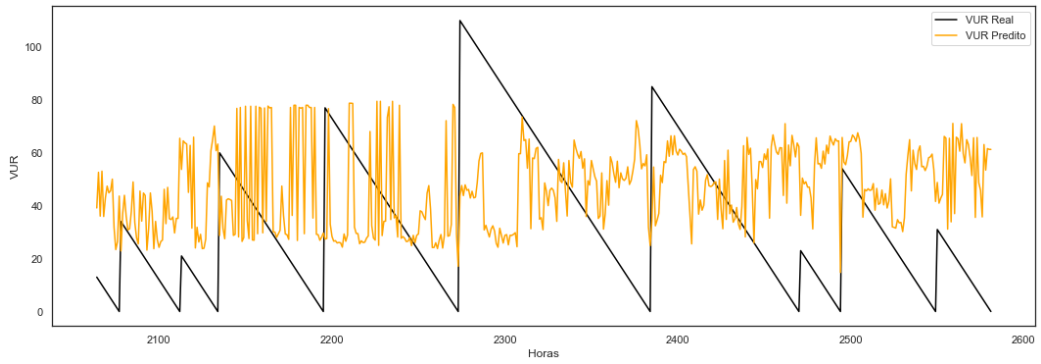
Considera-se que esse resultado pode ser decorrente do modelo de Regressão Linear não ser ideal para esse tipo de predição, mas principalmente pela não idealidade dos dados, como a falta de correlação dos mesmos com a vida útil remanescente, fato que dificulta a compreensão do algoritmo a respeito do que deve ser predito.

5.9.10 Treinamento e avaliação Florestas Aleatórias para estudo da VUR

O treinamento do modelo de Florestas Aleatórias para regressão utilizará a função “*RandomForestRegressor()*” da biblioteca Scikit-learn com os parâmetros selecionados e os dados para treino e teste de acordo com o resultado da seleção de *features*.

Após realizada a predição utilizando os dados de teste é fornecido para a função de avaliação do modelo a VUR predita e a real, o que a mesma retorna com os seguintes resultados: MAE de 28,62, RMSE de 34,63 e R^2 -70,58%. Ao avaliar as métricas, percebe-se que o algoritmo para regressão através de Florestas Aleatórias apresentou um pior desempenho comparado a Regressão Linear, visto que apresentou erros bem maiores que o estimador anterior, além de um r-quadrado ainda mais negativo, que como dito anteriormente, diz que uma linha contínua da média como predição teria menor variabilidade do que a predição com o algoritmo. A Figura 60 apresenta a relação da VUR real e predita ao longo das horas através de um gráfico de linhas.

Figura 60 – Comparação do predito e real para Florestas Aleatórias



Fonte: Autor (2023)

A linha preta representa os valores da VUR real e a linha amarela o predito e assim como o gráfico anterior, a VUR está sendo representada em sua forma positiva para mais fácil interpretação na comparação das linhas.

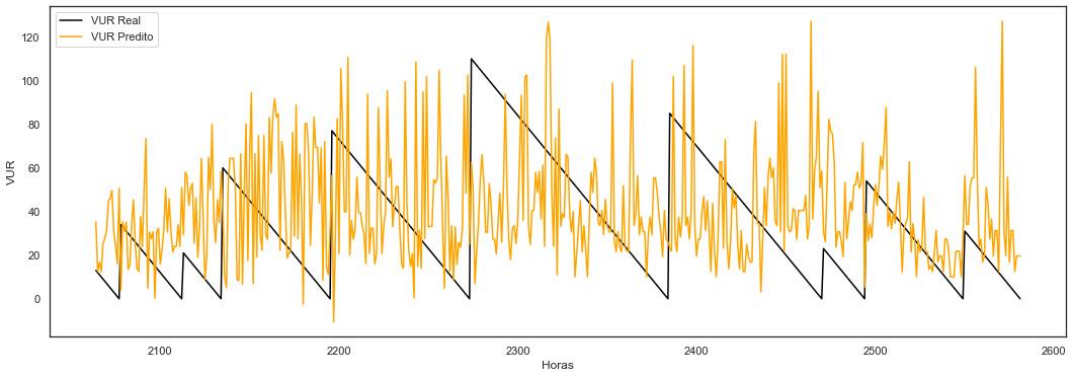
Ao observar a Figura 60, nota-se uma significativa variação na predição da vida útil remanescente sem um padrão claro, o que explica o valor obtido do R^2. Ademais, o gráfico aparentemente respeita uma faixa para predição máxima e mínima. Ao analisar a média da predição, o resultado obtido é de -46,11 horas, valor muito próximo à média dos dados de treino, que é -44,83 horas. No entanto, apresenta um comportamento com maiores ruídos, ao contrário da Regressão Linear e o que pode justificar isso é o fato de que cada árvore de decisão das Florestas Aleatórias possui diversas árvores com diversos “nós folhas” com diferentes valores de VUR. Além disso, o fato de que os dados não transparecem as mudanças que ocorrem na vida útil remanescente da caldeira, por exemplo, um valor inserido no algoritmo para o parâmetro “PTV” (pressão temperatura e vazão de vapor) quando a caldeira está prestes a falhar (VUR próxima a zero) poderia ser predito facilmente pelo estimador como uma VUR no seu início de ciclo ou no seu final, justamente pela maioria das falhas não impactarem diretamente nos seus parâmetros.

5.9.11 Treinamento e avaliação XGBoost para estudo da VUR

O algoritmo de aumento de gradiente XGBoost utilizará a função “*XGBoostRegressor()*” da biblioteca XGBoost com os parâmetros selecionados. Em seguida, com auxílio da função “*fit()*”, o algoritmo é treinado com os dados de treinado através das *features* selecionadas para o mesmo.

Após realizar a predição, a função de avaliação de desempenho retorna os seguintes valores: MAE de 27,76, RMSE de 35,30 e R^2 -77,27%. Ao observar as métricas de avaliação, observa-se um desempenho do algoritmo muito similar ao obtido através da Regressão por Florestas Aleatórias, no entanto com alguns resultados inferiores, como o R^2, mas com um desempenho melhor na raiz do erro quadrático médio. Com intuito de explorar melhor esse resultado, visualizar graficamente torna-se mais intuitivo, por isso, a Figura 61 apresenta a relação da VUR real e predita ao longo das horas através de um gráfico de linhas.

Figura 61 - Comparação do predito e real para XGBoost



Fonte: Autor (2023)

Ao analisar a Figura 61, observa-se um comportamento de predição com variação extremamente altas, com um comportamento similar a ruídos de grande amplitude. Visualizar o comportamento do gráfico para a predição do algoritmo XGBoost esclarece que apesar do desempenho ser semelhante ao obtido com o Florestas Aleatórias, seus comportamentos são distintos. Apesar disso, os dois algoritmos utilizam de árvores de decisão para realizar suas predições, o que pode justificar alguma similaridade entre o resultado de ambos. Todavia, assim como os outros, o algoritmo XGBoost não apresentou um bom resultado na predição e acredita-se que os motivos sejam similares aos demais.

Por apresentar um comportamento gráfico similar a ruídos, não parece seguir um padrão, no entanto, como a média das predições dos dois estimadores anteriores está muito próxima à da média dos dados de treino, é válido o estudo para as predições do XGBoost. Ao realizar a média dos valores preditos, obteve-se um valor de -41,04 horas, valor muito próximo aos outros.

Aparentemente, apesar do comportamento gráfico para as três predições serem distintas entre os três algoritmos, apresentando menos ou mais ruído, os três aparentam, de certa forma, oscilar ao redor da média da VUR utilizada nos dados de treino.

6 CONSIDERAÇÕES FINAIS

Neste desfecho final é visado apresentar as considerações finais de todas etapas e cumprimento de objetivos que levaram aos resultados obtidos no presente trabalho de conclusão de curso.

No que diz a respeito da qualidade da coleta de dados realizada dentro da área selecionada, a caldeira, pode-se afirmar que foi de qualidade regular devido a diversas variáveis que a afeta, tais como a variedade de operadores que realiza a coleta, o intervalo de coleta estabelecido de hora em hora e a ausência da coleta de dados durante o processo de falha. Somando a isso, não foi possível obter os registros de operação da caldeira durante boa parte da coleta de dados, o que compromete a identificação de falhas e paradas normais nos dados. Além do mais, o mês faltante ao longo do tempo de coleta também afeta negativamente a qualidade dos dados. Acerca do repasse dessas informações para planilhas digitais é esperado algumas falhas durante a digitação, apesar das checagens e correções de erros (principalmente os mais grosseiros), visto que ao total foram 27771 dados digitalizados manualmente dos parâmetros da caldeira e nesses números não está incluso dados de horário de coleta, além dos dados de tempo até falha, ciclos de falha e condições que foram criadas depois.

Com relação ao estudo de falhas da caldeira mista, a análise de dados realizada possibilita a melhor compreensão dos ciclos de falhas e do processo da caldeira, onde é possível identificar as falhas mais recorrentes, falhas que mais consomem tempo de manutenção e tempo de manutenção médio por falha, além do tempo em que a caldeira fica parada devido a falhas com relação ao tempo em operação, que é significativamente alto.

No que diz a respeito do desempenho do algoritmo de Florestas Aleatórias para classificações dos estados de “operando” e “parada” para a caldeira, pode-se salientar o ótimo desempenho do algoritmo, que obteve resultados excelentes em todas as métricas. Em contraponto, o mesmo algoritmo para classificação das condições de falhas selecionadas (elevador, pressão positiva, *redler* e caldeira furada) obteve um baixo desempenho, apresentando métricas de avaliação ruins, com alto índice de confusão entre classes e uma acurácia geral de 29%. O que explica esses dois resultados antagônicos é que, o mesmo motivo pelo qual um possui ótimo desempenho, a fácil distinção entre a caldeira em estado operacional ou parada (seja

por falha ou parada normal), é o que dificulta a aprendizagem do outro, visto que diversas falhas são somente identificadas quando a caldeira está parada (com valores iguais a zero).

A análise de dados realizada para o estudo da manutenção preditiva possibilita a compreensão das correlações entre os diversos parâmetros da caldeira, o comportamento dos diversos ciclos da caldeira e suas vidas úteis remanescente e, principalmente, o comportamento dos parâmetros da caldeira conforme a VUR decresce. Sendo que, para o último é essencial apresentar significativa correlação e comportamento dos dados definidos, fato que não ocorre e impacta os resultados obtidos posteriormente com as regressões.

No que cerne os desempenhos obtidos através dos regressores de Regressão Linear, Florestas Aleatórias e *Extreme Gradient Boosting* para predizer a vida útil remanescente da caldeira, é possível afirmar que todos os algoritmos não obtiveram sucesso em suas estimações e suas métricas de avaliação apresentaram resultados adversos, com grandes erros e coeficiente de determinação (R-quadrado) negativo. Para elucidar esse resultado, é admissível alegar que ele é causado pela falta de correlação dos parâmetros de processo da caldeira com a vida útil remanescente da mesma, o que implica diretamente em um dos princípios da aplicação da manutenção preditiva, que diz que as falhas a serem estudadas devem ser provenientes de causas que sejam monitoradas, ou seja, as falhas devem afetar diretamente os parâmetros mensurados. Além do mais, acredita-se que o estudo simultâneo de diversos tipos de falhas dificulta o aprendizado do algoritmo, pois cada falha possui características únicas e ocorrem em diferentes componentes da caldeira, que podem afetar, ou não, as variáveis de processo.

Ao longo do desenvolvimento do presente trabalho, as falhas de pressão positiva apresentam-se como um potencial objeto de estudo e possui interferência direta nas variáveis de processo da caldeira, principalmente na depressão e tiragem do exaustor. Infelizmente, o tempo de coleta mostrou-se muito curto para obter uma quantidade significativa de dados para possibilitar um estudo profundo sobre essa falha.

Para este estudo, acredita-se fortemente que a escolha de outros algoritmos de *Machine Learning* e outros tipos de processamento de dados não terão um aumento significativo no desempenho das predições. Todavia, acredita-se que direcionar os esforços para aumentar a qualidade dos dados é a peça chave para o

sucesso do treinamento. Com esse propósito, o ideal é realizar um estudo em campo individual para uma ou cada componente que apresenta falha, com isso obter o melhor local para instalação de um sensor, que poderá medir parâmetros como temperatura, vibração ou ruídos. A partir disso, coletar essas informações e armazená-las em nuvem. No entanto, são ações inviáveis economicamente para fins de estudo, sendo só possível com o investimento por parte da empresa em tecnologia de uma Indústria 4.0.

Com relação a viabilidade de aplicar as técnicas de Análise de Dados em uma indústria com características de uma Indústria 3.0 é de total validez, apesar das diversidades durante a coleta de dados é possível retirar diversas informações valiosas, que não só podem esclarecer pontos do processo que não são compreendidos apenas visualizando ou monitorando o processo, mas também aprimorar o conhecimento do processo como um todo. Em divergência, a viabilidade de aplicar técnicas de *Machine Learning* tornam-se mais restritas a predições mais simples, como caldeira “operando” ou “parada” e aponta dificuldades para trabalhar com predições mais complexas, tais como a predição da vida útil remanescente para manutenção preditiva. Ademais, o principal fator que contribui para a inviabilidade parcial da aplicação de algoritmos de *Machine Learning* é a não idealidade dos dados para os fins propostos, principalmente em questões de coleta.

O presente trabalho possibilitou o desenvolvimento em diversas áreas, tais como: manipulação de dados, Análise de Dados, Ciência de Dados, coleta de dados, programação em Python, *Machine Learning*, manutenção, manutenção preditiva, beneficiamento do arroz e processo de caldeiras industriais. Além disso, possibilitou desenvolvimento de habilidades pessoais como pensamento crítico e analítico.

7 SUGESTÕES PARA TRABALHOS FUTUROS

Neste tópico será abordado recomendações e pontos a serem levados em consideração para realizar estudos futuros.

i. Se possível, os dados devem ser coletados pessoalmente e em um intervalo de tempo menor do que de hora em hora, pois acredita-se que possivelmente as variáveis do processo podem sofrer mudança pouco tempo antes das falhas acontecerem e pelo intervalo de hora em hora, não é possível ser acompanhadas;

ii. Manter a coleta de dados mesmo com a caldeira em estado de pane até ela ser desligada completamente pode ser um caminho para caracterizar cada tipo de falha;

iii. Analisar apenas ciclos de uma só falha pode ser a chave para compreender as características de cada ciclo. Entretanto, acredita-se fortemente que a maioria das falhas em componentes da caldeira não afetam diretamente nas variáveis do processo da mesma, com exceção das falhas de pressão positiva, que pode ser um ótimo objeto de estudo;

iv. Apesar das falhas de pressão positiva serem um ótimo objeto de estudo, não é possível determinar quando elas irão acontecer ou a partir de quantos meses se terá uma boa quantidade de dados para estudar;

v. Buscar uma ferramenta para selecionar os melhores parâmetros de treinamento dos algoritmos para série temporal pode melhorar o desempenho do algoritmo;

vi. Realizar um estudo com intuito de identificar os motivos pelo qual as falhas ocorrem e validar que elas não impactam nas variáveis de processo da caldeira.

REFERÊNCIAS

AZANK, Felipe. **Como avaliar seu modelo de regressão:** as principais métricas para avaliar seus modelos de regressão. Medium. 2020. Disponível em: https://medium.com/turing-talks/como-avaliar-seu-modelo-de-regress%C3%A3o-c2c8d73dab96#:~:text=Em%20outras%20palavras%2C%20pega%2Dse,esse%20n%C3%BAmero%2C%20pior%20o%20modelo. Acesso em: 25 fev. 2022.

BARRIGOSSI, José Alexandre Freitas *et al*. **Recomendações técnicas para a cultura de arroz irrigado no Mato Grosso do Sul**. 1. ed. Santo Antônio de Goiás: Embrapa Arroz e Feijão, 2009. 148 p. ISSN 1678-9644. Disponível em: https://www.infoteca.cnptia.embrapa.br/handle/doc/578317?locale=en. Acesso em 25 jan. 2023.

BI4ALL. **O que é o PCA**: Principal Component Analysis e como aplicá-lo a um conjunto de dados. bi4all. 2018. Disponível em: https://www.bi4all.pt/noticias/blog/o-que-e-o-pca/#:~:text=O%20PCA%20%C3%A9%20caracterizado%20por,seja%2C%20os%20seus%20componentes%20principais. Acesso em: 17 jan. 2023.

BITTAR, Marcel. **Métricas para avaliação de modelos de *Machine Learning*.** GitHub. 2020. Disponível em: https://mabittar.github.io/Metricas/. Acesso em: 26 fev. 2022.

BOTELHO, Manoel Henrique Campos; BIFANO, Hercules Marcello. **Operação de caldeiras**: gerenciamento, controle e manutenção. 2 ed. São Paulo: Edgard Blücher Ltda, 2015. *E*-book. ISBN: 978-85-212-0944-7. Disponível em: https://books.google.com.br/books?id=jIewDwAAQBAJ&lpg=PA4&hl=pt-BR&authuser=1&pg=PA4#v=onepage&q&f=false. Acesso em: 21 out. 2022.

BORCAN, Marius. **Decision Tree Classifiers Explained**. Medium. 2020. Disponível em: https://medium.com/@borcandumitrumarius/decision-tree-classifiers-explained-e47a5b68477a. Acesso em: 11 fev. 2023.

BRASIL. Ministério da Ciência, Tecnologia e Inovações. **Câmara da Indústria**. Brasília: MCTI, [2022?]. Disponível em: https://www.gov.br/mcti/pt-br/acompanhe-o-mcti/transformacaodigital/camara-industria. Acesso em: 08 dez. 2022.

BRASIL. Ministério do Trabalho. **NR-13**: Manual técnico de caldeiras e vasos de pressão. Brasília: MTE, SIT, DSST, 2006. Edição comemorativa 10 anos da NR-13. 124 p. Disponível em: http://www.segurancanotrabalho.eng.br/manuais_tecnicos/manualcaldeiras.pdf. Acesso em: 21 out. 2022.

CHEN, Daniel Y. **Análise de dados com Python e Pandas**. São Paulo: Novatec Editora, 2018. *E-book.* 432 p. ISBN: 978-85-7522-699-5. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr&id=ILFwDwAAQBAJ&oi=fnd&pg=PA31&dq=an%C3%A1lise+de+dados+python+e+pandas+chen&ots=sP1hUanhhm&sig=7IdmS9NdH6QqOTBOoJmjHvvwQCY&pli=1#v=onepage&q&f=false. Acesso em: 11 mar. 2022.

CHINELATO, Gressa. **Tudo que você precisa saber sobre armazenagem do arroz.** Aegro. 2020. Disponível em: https://blog.aegro.com.br/armazenagem-do-arroz/#:~:text=O%20gr%C3%A3o%20%C3%A9%20considerado%20seco,recomendado%20%C3%A9%2018%C2%B0C. Acesso em: 12 fev. 2022.

CONFEDERAÇÃO NACIONAL DA INDÚSTRIA. **Sondagem Especial**: Indústria 4.0. [Brasil?]: CNI, 2016. Disponível em: https://static.portaldaindustria.com.br/media/filer_public/e0/aa/e0aabd52-53ee-4fd8-82ba-9a0ffd192db8/sondespecial_industria40_abril2016.pdf. Acesso em: 9 dez. 2022.

COSTA, Kelton A. P.; PRADO, Simone G. D.; SILVA, Marcia A. Z. M.; SILVA, Luis F. B.; ULTIMURA, Luan N. Study on *Machine Learning* techniques for botnet detection. **IEEE Latin America Transactions**, v. 18, n. 5, p. 881-888, 2020. Disponível em: https://ieeexplore.ieee.org/abstract/document/9082916/. Acesso em: 23 fev. 2022.

DA EIRA, Danielle Bendlin. **Utilização do PDCA no Processo de Beneficiamento de Arroz.** Orientador: Daily Morales. 2010. Trabalhos de Conclusão de Curso (Bacharel em Engenharia de Produção) - Universidade Estadual de Maringá, Maringá, 2010. Disponível em: http://www.dep.uem.br/gdct/index.php/dep_tcc/article/view/954. Acesso em: 10 fev. 2022.

D'ANGELO, Pedro. **O que é métrica, indicador e como medir o sucesso das suas estratégias.** Opinion Box. 2020. Disponível em: https://blog.opinionbox.com/o-que-e-metrica/#:~:text=Uma%20m%C3%A9trica%20%C3%A9%20uma%20medida,ou%20n%C3%A3o)%20no%20seu%20trabalho. Acesso em: 25 fev. 2022.

DATA SCIENCE TEAM. **Florestas Aleatórias.** DATA SCIENCE. 2020. Disponível em: https://datascience.eu/pt/programacao/entendendo-os-classificadores-de-florestas-aleatorias-em-python/. Acesso em: 23 fev. 2022.

DATA SCIENCE. **O que é um Z-Score?**. DATA SCIENCE. 2020. Disponível em: https://datascience.eu/pt/matematica-e-estatistica/o-que-e-um-z-score/. Acesso em: 17 jan. 2023.

DE ALMEIDA, Paulo Samuel. **Manutenção Mecânica Industrial**: Conceitos Básicos e Tecnologia Aplicada. 1 ed. São Paulo: Érica, 2015. *E-book*. ISBN 978-85-3651-979-1. Disponível em: https://pergamum.unipampa.edu.br/biblioteca/index.php. Acesso em: 03 dez. 2022.

DE ALMEIDA, Paulo Samuel. **Manutenção Mecânica Industrial**: Princípios Técnicos e Operações. 1 ed. São Paulo: Érica, 2016. *E-book*. ISBN 978-85-3651-980-7. Disponível em: https://pergamum.unipampa.edu.br/biblioteca/index.php. Acesso em: 04 dez. 2022.

DICK, Stephanie. Artificial intelligence. **Jornal Harvard Data Science Review.** jul. 2019. Disponível em: https://hdsr.mitpress.mit.edu/pub/0aytgrau/release/2. Acesso em: 10 jan. 2022.

DIÓRIO, Alexandre. **Geração e Distribuição de Vapor**. Londrina: Editora e Distribuidora Educacional S.A, 2019. 200 p.

DOS SANTOS, Rayssa Fernanda. **Secagem do arroz:** tudo sobre esse processo. Aegro. 2020. Disponível em: https://blog.aegro.com.br/secagem-do-arroz/. Acesso em: 12 fev. 2022.

ESTÚDIO ABC. **O Brasil está pronto para a indústria 4.0?**. Exame. 2017. Disponível em: https://exame.com/tecnologia/o-brasil-esta-pronto-para-a-industria-4-0/. Acesso em: 8 dez. 2022.

GEEKSFORGEEKS. ***Box plot* usando Seaborn em Python**. GeeksforGeeks. 2020. Disponível em: https://www.geeksforgeeks.org/*box plot*-using-seaborn-in-python/. Acesso em: 20 jan. 2023.

GOMES, Pedro César Tebaldi. **Regressão Linear:** entenda como utilizar. Data Geeks. 2019. Disponível em: https://www.datageeks.com.br/regressao-linear/#:~:text=Um%20dos%20conceitos%20estat%C3%ADsticos%20mais,ser%20utilizada%20para%20realizar%20previs%C3%B5es. Acesso em: 18 fev. 2022.

GREGÓRIO, Gabriela Fonseca Parreira; DA SILVEIRA, Aline Morais. **Manutenção industrial**. Porto Alegre: SAGAH EDUCAÇÃO S.A., 2018. *E-book*. ISBN 978-85-9502-697-1. Disponível em: https://pergamum.unipampa.edu.br/biblioteca/index.php. Acesso em: 04 dez. 2022.

HADRI, Walid. **Introduction to *Machine Learning*.** Medium, mar. 2021. Disponível em: https://medium.com/geekculture/introduction-to-machine-learning-428a630417dd.. Acesso em: 5 fev. 2022.

HALL, Tim. **The Role of Data in Industry 4.0**. Industry Today. 2020. Disponível em: https://industrytoday.com/the-role-of-data-in-industry-4-0/. Acesso em: 25 jan. 2023.

HASHTAG TREINAMENTOS. **Python**. Hashtag Treinamentos. 2021. Disponível em: https://www.hashtagtreinamentos.com/o-que-e-python?gclid=Cj0KCQjwuMuRBhCJARIsAHXdnqNfQpR4H_27A2wzxQtfFdbruXFxE02E_TxV69DlYlGCr0sn-Fsl_xQaAs8iEALw_wcB. Acesso em: 7 fev. 2022.

HURST BOILER. **Hurst Boiler Image Gallery**. Hurst Boiler. [2023?]. Disponível em: https://www.hurstboiler.com/boiler-images/solid_fuel_boilers. Acesso em: 20 jan. 2023.

IBM. **Visualização da Matriz de Confusão**. IBM. 2021. Disponível em: https://www.ibm.com/docs/pt-br/db2/10.5?topic=visualizer-confusion-matrix-view. Acesso em: 20 jan. 2023.

INSTITUTO BRASILEIRO DE PETRÓLEO E GÁS (IBP). **Guia de Inspeção**: Inspeção de Caldeiras. 3 ed. Rio de Janeiro: IBP, 2020. 72 p. Disponível em: www.ibp.org.br/biblioteca. Acesso em: 21 out. 2022.

IPNET. **A importância da normalização e padronização dos dados em *Machine Learning***. Medium. 2021. Disponível em: https://medium.com/ipnet-growth-partner/padronizacao-normalizacao-dados-machine-learning-f8f29246c12. Acesso em: 17 jan. 2023.

KARDEC, Alan; NASCIF, Júlio. **Manutenção**: funcao estrategica. Rio de Janeiro, RJ: Qualitymark, 2001. 341 p. *E-book*. ISBN 857-30-3323-1. Disponível em: https://pergamum.unipampa.edu.br/biblioteca/index.php. Acesso em: 06 dez. 2022.

KHANDELWAL, Neetika. **A Brief Introduction to XGBoost**: Extreme Gradient Boosting with XGBoost. Towards Data Science. 2020. Disponível em: https://towardsdatascience.com/a-brief-introduction-to-xgboost-3eaee2e3e5d6#:~:text=XGBoost%20vs%20Gradient%20Boosting,can%20be%20parallelized%20across%20clusters. Acesso em: 24 jan. 2023.

KHO, Julia. **Annotated *Heatmap*s of a Correlation Matrix in 5 Simple Steps**. Kdnuggest. 2019. Disponível em: https://www.kdnuggets.com/2019/07/annotated-*heatmap*s-correlation-matrix.html. Acesso em: 20 jan. 2023.

KUMAR, Ajitesh. **Bagging vs Boosting Machine Learning Methods**. Vitalflux. 2022. Disponível em: https://vitalflux.com/bagging-vs-boosting-machine-learning-methods/. Acesso em: 11 fev. 2023.

LEAL, Renato dos Santos. **Métricas Comuns em *Machine Learning*:** como analisar a qualidade de chat bots inteligentes. Medium. 2017. Disponível em: https://medium.com/as-m%C3%A1quinas-que-pensam/m%C3%A9tricas-comuns-em-machine-learning-como-analisar-a-qualidade-de-chat-bots-inteligentes-m%C3%A9tricas-1ba580d7cc96. Acesso em: 27 fev. 2022.

LEE, Huei Diana. **Seleção e construção de features relevantes para o Aprendizado de Máquina**. Orientadora: Profa. Dra. Maria Carolina Monard. São Carlos. 2000. Dissertação (Mestrado em Ciências – Área de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, São Carlos, 2000. Disponível em: https://www.teses.usp.br/teses/disponiveis/55/55134/tde-15032002-113112/en.php. Acesso em: 17 jan. 2023.

LOPES, Gesiel Rios *et al*. **Introdução à análise exploratória de dados com python.** Minicursos ERCAS ENUCMPI, v. 2019, p. 160-176, 2019. Disponível em: https://www.researchgate.net/profile/Gesiel-Lopes/publication/336778766_Introducao_a_Analise_Exploratoria_de_Dados_com_Python/links/5db225d2a6fdccc99d9426f2/Introducao-a-Analise-Exploratoria-de-Dados-com-Python.pdf. Acesso em: 10 mar. 2022.

LORBERFELD, Audrey. ***Machine Learning* Algorithms In Layman's Terms**. Towards Data Science. 2019. Disponível em: https://towardsdatascience.com/machine-learning-algorithms-in-laymans-terms-part-1-d0368d769a7b. Acesso em: 20 jan. 2023.

LUZ, Carlos A. S. *et al*. Relações granulométricas no processo de brunimento de arroz. **Engenharia Agrícola**, v. 25, n. 1, p. 214-221, 2005. Disponível em: https://www.scielo.br/pdf/eagri/v25n1/24888.pdf. Acesso em: 13 fev. 2022.

MEDEIROS, Soraya Maria de; ROCHA, Semíramis Melani Melo. **Considerações sobre a terceira revolução industrial e a força de trabalho em saúde em Natal.** Ciência & Saúde Coletiva, v. 9, p. 399-409, 2004. Disponível em: https://www.scielo.br/j/csc/a/Cwp5Sxn7vqJWKLdcGqqqJ7D/?format=html&lang=pt. Acesso em: 14 nov. 2021

MINITAB BLOG EDITOR. **Análise de regressão:** Como interpretar o R-quadrado e avaliar a qualidade de ajuste. Minitab. 2019. Disponível em: https://blog.minitab.com/pt/analise-de-regressao-como-interpretar-o-r-quadrado-e-avaliar-a-qualidade-de-ajuste. Acesso em: 25 fev. 2022.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, v. 1, n. 1, p. 32, 2003. Disponível em: http://dcm.ffclrp.usp.br/~augusto/publications/2003-sistemas-inteligentes-cap4.pdf. Acesso em: 13 jan. 2022

MONTIEL,Selene Athenas Espinoza. **Efeito do tratamento térmico convencional e do processo de extrusão sobre a qualidade nutricional de arroz (*Oryza sativa*) e feijão (*Phaseolus vulgaris*)**. Orientadora: Prof. Dr. Ivo Mottin Demiate. Ponta Grossa. 2020. Dissertação (Mestrado em Ciências e Tecnologia de Alimentos) - Universidade Estadual de Ponta Grossa. Programa de Pós-Graduação *Stricto Sensu,* Ponta Grossa, 2020. Disponível em: https://tede2.uepg.br/jspui/bitstream/prefix/3034/1/Selene Athenas Espinoza Montiel.pdf. Acesso em: 25 jan. 2023.

NASCIMENTO, Daniel. ***Machine Learning***: entendendo o processo de aprendizagem. Medium. 2017. Disponível em: https://medium.com/@Daniel_nasci/machine-learning-entendendo-o-processo-de-aprendizagem-80835b3ec2dc. Acesso em: 20 jan. 2023.

NEPOMUCENO, Lauro Xavier. **Técnicas de manutenção preditiva**. 1 ed. São Paulo: Edgard Blücher Ltda., v. 1, 1989. *E-book*. ISBN 978-85-212-0092-5. Disponível em: https://pergamum.unipampa.edu.br/biblioteca/index.php. Acesso em: 06 dez. 2022.

NITZKE, Julio Alberto; BIEDRZYCKI, Aline. **Terra de Arroz**: Processamento. URGS. 2005. Disponível em: https://www.ufrgs.br/alimentus1/terradearroz/grao/gr_apresenta.htm. Acesso em: 18 jan. 2023.

PASQUINI, Nilton Cesar. Revoluções Industriais: uma abordagem conceitual. **Revista Tecnológica da Fatec Americana**, v. 8, n. 01, p. 29-44, 2020. Disponível em: https://www.fatec.edu.br/revista/index.php/RTecFatecAM/article/view/235. Acesso em: 12 nov. 2021.

PEREIRA, Ana C.; ROMERO, Fernando. **A review of the meanings and the implications of the Industry 4.0 concept.** Procedia Manufacturing, v. 13, p. 1206-1214, 2017. Disponível em: https://www.sciencedirect.com/science/article/pii/S2351978917306649. Acesso em: 14 nov. 2021.

PEREIRA, João Victor dos Santos; MUNIZ, Pedro Henrique dos Santos; VARGAS, Alessandra Alves Fonseca. Árvore de Decisão. **PESQUISA & EDUCAÇÃO A DISTÂNCIA**, n. 19, 2020. Disponível em: http://www.revista.universo.edu.br/index.php?journal=2013EAD1&page=article&op=viewArticle&path%5B%5D=8697. Acesso em: 22 fev. 2022.

PYTHON SOFTWARE FOUNDATION. **Python for beginners.** Python. 2022. Disponível em: https://www.python.org/about/gettingstarted/. Acesso em: 7 mar. 2022.

RIBEIRO, Débora. **Indução.** Dicio, Dicionário Online de Português Dicionário Online de Português. 2019. Disponível em: https://www.dicio.com.br/inducao/. Acesso em: 14 jan. 2022.

RODRIGUES, Sandra Cristina Antunes. **Modelo de regressão linear e suas aplicações.** Orientadora: Profa. Dra. Célia Maria Pinto Nunes. 2012. Relatório de Estágio (Mestrado em Ensino de Matemática) - Universidade da Beira Interior, Covilhã, 2012. Disponível em: https://ubibliorum.ubi.pt/handle/10400.6/1869. Acesso em: 18 fev. 2022.

RODRIGUES, Vitor. **Métricas de avaliação:** acurácia, precisão, recall. quais as diferenças. Medium. 2019. Disponível em: https://vitorborbarodrigues.medium.com/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-recall-quais-as-diferen%C3%A7as-c8f05e0a513c. Acesso em: 27 fev. 2022.

ROJAS, Esperanza Manrique. *Machine Learning*: *análisis de lenguajes de programación y herramientas para desarrollo*. **Revista Ibérica de Sistemas e Tecnologias de Informação Iberian Journal of Information Systems and Technologies**. E28. ed. 586–599 p, 2020. ISSN 1646-9895 versão *online*. Disponível em: http://www.risti.xyz/index.php/pt-pt/edicoes. Acesso em: 11 fev. 2023.

ROSA, Vitor Reis. **Estudo de desempenho de algoritmos de *Machine Learning* aplicado à prognósticos e monitoramento de condição**. Orientador: Profa. Dra. Luciana Castanheira. 2019. Trabalhos de Conclusão de Curso (Bacharel em Engenharia de Controle e Automação) - Universidade Federal de Ouro Preto, Curso em Engenharia de Controle e Automação, Ouro Preto, 2019. Disponível em: https://www.monografias.ufop.br/bitstream/35400000/2506/1/MONOGRAFIA_EstudoDesempenhoAlgoritmos.pdf. Acesso em: 07 dez. 2022.

RUSSELL, Stuart; NORVIG, Peter. **Inteligencia artificial.** 3. ed. São Paulo: GEN LTC, set. 2013.

SAIDELLES, Ana Paula Fleig *et al*. Gestão de resíduos sólidos na indústria de beneficiamento de arroz. **Revista Eletrônica em Gestão, Educação e Tecnologia Ambiental**, v. 5, n. 5, p. 904-916, 2012. Disponível em: https://periodicos.ufsm.br/reget/article/view/4314. Acesso em: 11 fev. 2022.

SANTA RITA, Bruno. **Empresas brasileiras ainda não estão preparadas para Indústria 4.0**. Correio Braziliense. 2019. Disponível em: https://www.correiobraziliense.com.br/app/noticia/economia/2019/04/03/internas_economia,747115/empresas-brasileiras-ainda-nao-estao-preparadas-para-industria-4-0.shtml. Acesso em: 8 dez. 2022.

SCHWAB, Klaus. **A quarta revolução industrial.** 1.ed. São Paulo: Edipro, 2016. *E-book*. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=XZSWDwAAQBAJ&oi=fnd&pg=PT161&ots=Y9ag1rMEgb&sig=DlfgsY8PEkZPfVniPx3SAP0_3TY. Acesso em: 13 nov. 2021.

SCIKIT-LEARN. **Cross-validation:** evaluating estimator performance. Scikit-learn. [2022?]. Disponível em: https://Scikit-learn.org/stable/modules/cross_validation.html. Acesso em: 28 fev. 2022.

SCIKIT-LEARN. **sklearn.feature_selection.RFE**. Scikit-learn. 2023. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html. Acesso em: 17 jan. 2023.

SCUDILIO, Juliana. **Qual a melhor métrica para avaliar os modelos de *Machine Learning*.** FLAI - Inteligência Artificial. 2020. Disponível em: https://www.flai.com.br/juscudilio/qual-a-melhor-metrica-para-avaliar-os-modelos-de-machine-learning/. Acesso em: 26 fev. 2022.

SEABORN. **seaborn.*pairplot***. seaborn. [2023?]. Disponível em: https://seaborn.pydata.org/generated/seaborn.*pairplot*.html. Acesso em: 20 jan. 2023.

SICHMAN, Jaime Simão. **Inteligência Artificial e sociedade:** avanços e riscos. Estudos Avançados, v. 35, p. 37-50, 2021. Disponível em: https://www.scielo.br/j/ea/a/c4sqqrthGMS3ngdBhGWtKhh/?format=html. Acesso em: 13 jan. 2022.

SILVEIRA, Guilherme; BULLOCK, Bennett. ***Machine Learning*:** introdução a classificação. São Paulo: Editora Casa do Código, 2017. *E-book*. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=XL46DwAAQBAJ&oi=fnd&pg=PT3&dq=machine+learning+introdu%C3%A7%C3%A3o+a+classifica%C3%A7%C3%A3o+silveira&ots=ZK3_8OOcLa&sig=GHQAfjFLwee_j2KOSCjBiU8ImFg. Acesso em: 16 jan. 2022.

SKA. **SYNECO É O SISTEMA MES DA SKA**. SKA. [2023?]. Disponível em: https://ska.com.br/produtos/syneco/sistema-mes. Acesso em: 20 jan. 2023.

SOUSA, Rafaela. **Segunda Revolução Industrial.** Brasil Escola, 2016. Disponível em: https://brasilescola.uol.com.br/historiag/segunda-revolucao-industrial.htm. Acesso em 12 nov. 2021.

SUGAHARA, José Afonso Santos. ***Machine Learning*** **e Data Science na indústria:** aplicações e desafios. Orientador: José Roberto Dale Luche. 2020. Trabalhos de Conclusão de Curso (Bacharel Engenharia de Produção) - Universidade Estadual Paulista, Curso de Engenharia de Produção, Guaratinguetá, 2020. Disponível em: https://repositorio.unesp.br/bitstream/handle/11449/217075/sugahara_jas_tcc_guara.pdf?sequence=7&isAllowed=y. Acesso em: 15 fev. 2022.

TAURION, Cezar. **Big data.** Rio de Janeiro: Brasport, 2013. *E-book*. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=GAVLAgAAQBAJ&oi=fnd&pg=PT11&dq=big+data+taurion&ots=YSeslTwdtL&sig=DwJyzeXH8ntLvD8Hnczr6LiPh5w#v=onepage&q=big%20data%20taurion&f=false. Acesso em: 25 nov. 2021.

TSAI, Chun-Wei *et al*. **Big data analytics:** a survey. Journal of Big data, v. 2, n. 1, p. 1-32, 2015. Disponível em: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0030-3. Acesso em: 29 nov. 2021.

VARGAS JUNIOR, Edson Cilos. **Medidas de desempenho para regressão.** Arquivo - Universidade Federal de Santa Catarina. 2020. Disponível em: https://geam.ufsc.br/aula-02-03/. Acesso em: 26 fev. 2022.

VISHAL. **Decision Trees:** A Bird's eye viewandan Implementation. Medium. 2018. Disponível em: https://towardsdatascience.com/decision-trees-a-birds-eye-view-and-an-implementation-c91754f0dcd0. Acesso em: 22 fev. 2022.

VIZONÁ, Amanda. **Como interpretar um *Box plot*?**. ICMC Júnior. 2021. Disponível em: https://icmcjunior.com.br/como-interpretar-um-grafico-*box plot*/?gclid=Cj0KCQiA_P6dBhD1ARIsAAGI7HAuUcVtlTK3tQE5L4PW0833VrRbLhUYzqMzLXY1QT2QpUkTwQ4os5waAjtCEALw_wcB. Acesso em: 20 jan. 2023.

WEBDESIGN EM FOCO. **Data Science and *Machine Learning***: #30 Florestas Aleatórias I. Webdesign em Foco. 2021. Disponível em: https://webdesignemfoco.com/cursos/python/data-science-and-machine-learning-30-florestas-aleatorias-i. Acesso em: 20 jan. 2023.

YADIN, Tovi. **Revolutions are only visible in retrospect**. Siemens. 2021. Disponível em: https://blogs.sw.siemens.com/valor/2021/02/15/revolutions-are-only-visible-in-retrospect/. Acesso em: 20 jan. 2023.