

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA
CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL MECÂNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECATRÔNICA**

MATEUS WEISS LOPES

**AVALIAÇÃO DE ALGORITMOS DE *MACHINE LEARNING* PARA
DETECÇÃO DE FALHAS EM REFRIGERADORES**

FLORIANÓPOLIS - SC, 2022.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA
CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL MECÂNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECATRÔNICA**

MATEUS WEISS LOPES

**AVALIAÇÃO DE ALGORITMOS DE *MACHINE LEARNING* PARA
DETECÇÃO DE FALHAS EM REFRIGERADORES**

Trabalho de conclusão de curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Bacharel em Engenharia Mecatrônica.

Orientador:

Prof. Gregory Chagas da Costa Gomes, me. eng.

FLORIANÓPOLIS-SC, 2022.

Ficha de identificação de obra elaborada pelo autor.

Lopes, Mateus Weiss

**AVALIAÇÃO DE ALGORITMOS DE *MACHINE LEARNING* PARA
DETECÇÃO DE FALHAS EM REFRIGERADORES / Mateus Weiss**

Lopes; orientação de Gregory Chagas Da Costa Gomes. – Florianópolis, SC,
2022. 68 p.

Trabalho de Conclusão de Curso (TCC) – Instituto Federal de Santa Catarina,
Câmpus Florianópolis. Bacharelado em Engenharia Mecatrônica.
Departamento Acadêmico de Metal Mecânica.

1. Refrigerador. 2. Machine learning. 3. Algoritmo. 4. Acurácia.
5. Revocação. 6. Precisão. I. Gomes, Gregory Chagas Da Costa. II. Instituto
Federal de Santa Catarina. Departamento Acadêmico de Metal Mecânica. III.
Título.



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS

**ATA DA BANCA FINAL DO TRABALHO DE CONCLUSÃO DO CURSO DE ENGENHARIA
MECATRÔNICA DO INSTITUTO FEDERAL DE SANTA CATARINA – Nº 115**

Aos 22 dias do mês de Julho de 2022 às 16h, o (a) estudante **Mateus Weiss Lopes** apresentou o seu Trabalho de Conclusão de Curso para julgamento à Banca Examinadora constituída pelos seguintes integrantes: Prof Gregory Chagas da Costa Gomes Me. Eng. (orientador/presidente da banca/IFSC), Prof Roberto Alexandre Dias Dr, Eng. (IFSC) e Prof Maurício Edgar Stivanello Dr. Eng. (IFSC). A sessão pública de defesa foi aberta pelo Presidente da Banca, que apresentou a Banca Examinadora e deu continuidade aos trabalhos, fazendo uma breve referência ao TCC que tem como título **Deteção De Falhas Em Refrigeradores Através De Machining Learning**. Na sequência, o(a) estudante teve até 30 minutos para a exposição de seu trabalho, e cada integrante da Banca Examinadora fez a arguição após a apresentação do mesmo. Finalmente, foi aberto um espaço aos presentes para eventuais perguntas ou comentários sobre o trabalho apresentado. Ouvidas as explicações do(a) estudante, a Banca Examinadora, reunida em caráter sigiloso, para proceder à avaliação final, deliberou pela **nota 9**. Foi dada ciência ao(a) estudante que a versão final do trabalho deverá ser entregue até o dia **22/08/2022**, com as devidas alterações sugeridas pela banca. Nada mais havendo a tratar, a sessão foi encerrada às 17h35min, dela sendo lavrada a presente ata, que, uma vez aprovada, foi assinada por todos os membros da Banca Examinadora e pelo(a) estudante.



Documento assinado digitalmente
GREGORY CHAGAS DA COSTA GOMES\gregory.c
Data: 25/08/2022 08:45:35-0300
CPF: 073.008.969-01
Verifique as assinaturas em <https://v.ifsc.edu.br>

Prof.(a) Orientador(a) / Presidente: Gregory Chagas da Costa Gomes Me. Eng.



Documento assinado digitalmente
ROBERTO ALEXANDRE DIAS
Data: 23/08/2022 15:47:52-0300
CPF: 579.134.399-04
Verifique as assinaturas em <https://v.ifsc.edu.br>

Prof.(a) Avaliador 1: Roberto Alexandre Dias Dr, Eng.



Documento assinado digitalmente
Mauricio Edgar Stivanello
Data: 23/08/2022 20:49:16-0300
Verifique em <https://verificador.iti.br>

Prof.(a) Avaliador 2: Maurício Edgar Stivanello Dr. Eng.

Mateus W Lopes

Acadêmico(a): Mateus Weiss Lopes

“O homem é a medida de todas as coisas.”
Protágoras

RESUMO

Nos dias atuais, a conservação de alimentos e outros materiais perecíveis em nossos lares só é possível com a utilização dos refrigeradores domésticos. O mau funcionamento de um refrigerador leva a deterioração dos produtos em seu interior, gerando transtorno ao ambiente familiar. Uma solução encontrada para prevenção e detecção de falhas em refrigeradores foi a utilização do *machine learning* como ferramenta para monitorar certos dados em busca de detectar possíveis falhas antes que elas ocorram. A metodologia usada é do processo unificado, processo esse centrado na arquitetura, iterativo e incremental. A obtenção dos dados e sua organização se deu através de pontos de medições importantes relacionados a defeitos em refrigeradores. O algoritmo utilizado neste trabalho foi obtido da avaliação de três modelos diferentes da biblioteca *skitlearn*. Para escolher o melhor modelo usou-se a métrica de acurácia, revocação e precisão. O melhor resultado foi obtido com o modelo *RandomForestClassifier*, que apresentou acurácia de 94%, revocação de 100% e uma precisão de 90%. Conclui-se que o modelo satisfaz o projeto e serve como indício do potencial do uso destas tecnologias na indústria de refrigeração, com a utilização de sensores de temperatura na entrada do termostato e um transdutor de corrente em série com a alimentação do refrigerador.

Palavras-chave: Refrigerador, Machine learning, algoritmo, acurácia, revocação e precisão.

ABSTRACT

Nowadays, the conservation of food and other perishables products in our home is only possible with the utilization of domestic refrigerators. Malfaction of a refrigerator leads to the deterioration of products inside of it, generating inconvenience to the family enviroment. One solution that was found for prevention and dection of defects in refrigerators was the machine learning as a tool to monitor certain data in search of detecting possible defects before they happen. The methodology used was the unified process, which is centered in the arquitecture, iterative and incremental. The acquirement of data and its organization is through important points of measurement related to the defects in refrigerators. The algorithm used left the evaluation of three different models of the sckit-learn's library. To choose the best model it was used the metric of accuracy, revocation and precision. The best result obtained was with the model RandomForestClassifer, that presented 94% of accuracy, revocation of 100% and a precision of 90%. It was concluded that the model satisfies the project and serves as a potencial evidence of the use of such technologies in the refrigeration industry, with the utilization of temperature sensors in the entrance of the thermostat and a transductor of in series chain with the supply power in the refrigerator.

Keywords: Refrigerator, Machine learning, algorithm, accuracy, revocation and precision.

LISTA DE FIGURAS

Figura 1 – Processo unificado.	17
Figura 2 – Circuito de refrigeração por compressão mecânica.	22
Figura 3 – Diagrama P x h do Isobutano.	23
Figura 4 – Tecnologias específicas.	31
Figura 5 – Linguagem de programação mais usadas em banco de dados.	32
Figura 6 – Crescimento do python no intervalo de 2012 até 2018.	33
Figura 7 – Crescimento de pesquisas utilizando as tags pandas, numpy e matplotlib.	33
Figura 8 – Posição dos pontos de medições utilizados.	36
Figura 9 – Arquivos escolhidos para o tratamento dos dados.	38
Figura 10 – Dados de Temperaturas retirados no dia 14 de setembro de 2014.	38
Figura 11 – Dados de Pressões retirados no dia 14 de setembro de 2014.	39
Figura 12 – Dados de Grandezas Elétricas retirados no dia 14 de setembro de 2014.	39
Figura 13 – Adequação dos arquivos para mesma quantidade de linhas.	40
Figura 14 – Lógica para eliminar o erro da coluna 0x0 == 1.	40
Figura 15 – Gráfico de Potência X Tempo.	41
Figura 16 – Gráfico de Corrente X Tempo.	41
Figura 17 – Gráfico de Tensão X Tempo.	42
Figura 18 – Gráfico de pressão.	42
Figura 19 – Gráfico de temperaturas.	43
Figura 20 – Gráfico de temperaturas específicas.	44
Figura 21 – Função <i>media_inter</i>	45
Figura 22 – Atributos de temperaturas finais.	45
Figura 23 – Atributos de temperaturas máximas e mínimas.	46
Figura 24 – Atributos de temperaturas em pontos específicos.	46
Figura 25 – Primeiro pico de corrente.	47
Figura 26 – Segundo pico de corrente.	47
Figura 27 – Atributos importantes para o modelo.	48
Figura 28 – Identificação de ensaio com falha.	48
Figura 29 – Identificação de ensaio com falha.	49

Figura 30 – Exemplo de função criada para ler os arquivos.....	49
Figura 31 – Criando a tabela Ensaios x Atributos.	50
Figura 32 – Tabela Ensaios x Atributos.....	50
Figura 33 – Exemplo de árvore de decisão para jogar tennis.	51
Figura 34 – Como é feito a otimização por meio do gradient boosting.....	52
Figura 35 – Modelos com os parâmetros padrões da biblioteca.	52
Figura 36 – Fluxo de funcionamento de um cross_validation.	53
Figura 37 – Representação gráfica dos dados com e sem falha.....	54
Figura 38 – Preparando para treinamento dos algoritmos.	56
Figura 39 – Resultados utilizando a ferramenta cross_validacion e o algoritmo RandomForestClassifier.	56
Figura 40 – Resultados utilizando a ferramenta cross_validacion e o algoritmo DecisionTreeClassifier.	56
Figura 41 – Resultados utilizando a ferramenta cross_validacion e o algoritmo GradientBoostingClassifier.	57
Figura 42 – Definição do percentual de treino e de teste.	58
Figura 43 – Matriz de confusão.	59

LISTA DE TABELAS

Tabela 1 - Defeitos comuns em sistemas de refrigeração.....	24
Tabela 2 – Características técnicas dos refrigeradores utilizado.	35
Tabela 3 – Pontos de medição.....	36
Tabela 4 – Disposição de dados.	37
Tabela 5 – Testes de modelos de machine learning.	58
Tabela 6 – Resultados do modelo sem alteração de hiperparâmetro.	60
Tabela 7– Comparação de Threshold para n_estimators = 3.	60
Tabela 8 – Comparação de Threshold para n_estimators = 4.	60
Tabela 9 – Comparação de Threshold para n_estimators =10.	61
Tabela 10 – Comparação de Threshold para n_estimators =100.	61

LISTA DE ABREVIATURAS E SIGLAS

ML	<i>Machine Learning</i>
IA	Inteligência artificial
K	Ferramenta de validação cruzada
IoT	<i>Internet of things (internet das coisas)</i>
MIT	Instituto de tecnologia do <i>massachusetts</i>
TTS	Temperatura do termômetro seco
TTU	Temperatura do termômetro úmido
Qcd	Taxa de calor cedida ao ambiente
Qev	Taxa de calor absorvida ao ambiente
W	Potência entregue ao compressor
P	Pressão
h	Entalpia
ROC	<i>Receiver operating characteristic curve</i>
SQL	Linguagem de programação
R	Linguagem de programação
Tc	Temperatura no termostato do congelador
Ts	Temperatura na sucção do compressor
Te	Temperatura na entrada do evaporador
Ps	Pressão na sucção do compressor
Pd	Pressão na descarga do compressor
rf	<i>RandomForestClassifier</i>
VP	Verdadeiro positivo
VN	Verdadeiro negativo
FN	Falso negativo
FP	Falso positivo

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Definição do Problema.....	15
2	OBJETIVOS.....	16
2.1	Objetivo Geral	16
2.2	Objetivos Específicos	16
3	METODOLOGIA	17
3.1	Fase de concepção.....	18
3.1.1	Definição do modelo principal	18
3.1.2	Planejamento preliminar	18
3.2	Fase de elaboração	18
3.2.1	Aprimoramento	18
3.2.2	Revisar	19
3.3	Fase de construção	19
3.3.1	Organização dos requisitos.....	19
3.3.2	Desenvolvimento, análise e implementação	19
3.4	Fase de transição	19
3.4.1	Validação	20
3.4.2	Documentação.....	20
4	FUNDAMENTAÇÃO DE REFRIGERADORES	21
4.1	Obstrução na linha de líquido	25
4.2	Carga de fluido refrigerante fora do valor nominal e vazamentos do fluido para o ambiente	26
4.3	Válvulas defeituosas	26

4.4	Defeitos elétricos.....	27
5	FUNDAMENTOS DO MACHINE LEARNING.....	28
5.1	Tipos de aprendizagem.....	28
5.1.1	Supervisionada	29
5.1.2	Não supervisionada	29
5.1.3	Por reforço	29
6	TECNOLOGIAS UTILIZADAS.....	31
6.1	Linguagem de programação python.....	32
6.1.1	Biblioteca pandas.....	34
6.1.2	Biblioteca <i>numpy</i>	34
6.1.3	Biblioteca <i>matplotlib</i>	34
7	DESENVOLVIMENTO E APLICAÇÃO DO ML	35
7.1	Conjunto de dados	35
7.2	Tratamento dos dados tabulares	37
7.2.1	Escolhas de atributos.....	44
7.2.2	Confecção da tabela Ensaios x Atributos.....	49
7.3	Algoritmos de <i>machine learning</i>	51
7.4	Avaliação dos algoritmos através de métricas.....	53
8	RESULTADOS ALCANÇADOS	58
9	CONCLUSÃO	63
	REFERÊNCIAS.....	65

1 INTRODUÇÃO

A expressão *Machine Learning*, no português aprendizado de máquina, foi introduzida nos anos 50 por Arthur Samuel. O autor afirma que: “aprendizado de máquina é um campo de estudo que dá a computadores a capacidade de aprender sem ser programado de forma explícita” (SAMUEL, 1959). Com origem na área da robótica, esta tecnologia é caracterizada pela construção de algoritmos e tecnologias que consigam “aprender” com uma certa quantidade de dados. Estes algoritmos conseguem ser treinados através de dados, permitindo encontrar padrões, com o objetivo de auxiliar na tomada de decisões com a mínima intervenção humana e por fim “prever” comportamentos e padrões de certa forma complexos.

Os algoritmos analisam uma grande quantidade de dados, que são interpretados e validados pelo sistema, definindo e encontrando padrões através de um aprendizado único da própria operação. Assim os algoritmos são capazes de determinar razões, classificar elementos bem como determinar porcentagens a respeito de uma classificação desejada.

O modelo *machine learning* é totalmente adaptável e pode ser utilizado em diversas abordagens. Na avaliação de defeitos em refrigeradores, a tecnologia de *machine learning* está sendo utilizada como ferramenta para monitorar dados em busca de detectar possíveis falhas antes que elas ocorram, permitindo gerar um alerta para o usuário possibilitando uma manutenção prévia do refrigerador.

As vantagens são muitas, destacando-se a agilidade com que uma falha pode ser detectada, bem como, pelo fato de a máquina detectar padrões muitas vezes complexos para um humano, o erro na detecção de defeitos tende a diminuir consideravelmente. Outra vantagem competitiva do uso de inteligências artificiais, *machine learning* é a diminuição da manutenção corretiva, gerando uma vida maior para o produto.

1.1 Definição do Problema

O problema deste trabalho de conclusão de curso está no estudo da utilização da tecnologia *machine learning* na detecção de falhas de refrigeradores. O aprendizado de máquina é um método de análise de dados que automatiza a construção de modelos analíticos. É um ramo da inteligência artificial baseada na ideia de que sistemas podem aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção humana. O machine learning pode ajudar a identificar gargalos ocultos no processo de produção industrial, apresentando a oportunidade de criar mais eficiências dos recursos e automações, seja em espaço, mercadorias, equipe ou tempo.

Este projeto tem como foco contribuir na parte da pesquisa e aplicações sobre detecção de falhas em refrigeradores através do *machine learning*. Para demonstrar a aplicação, serão utilizados diversos dados retirados de três refrigeradores em diferentes tipos de situações.

A partir de uma base de dados de falhas detectadas no funcionamento de três refrigeradores, serão organizados os dados de falhas gerando uma biblioteca para tratamento dos dados.

Os algoritmos de *machine learning* serão então aplicados a base de dados possibilitando a obtenção de alerta para manutenção.

Em seguida são realizadas as análises dos resultados, feitas adequações e documentado o processo.

2 OBJETIVOS

Para alcançar os objetivos do trabalho, foram definidos o objetivo geral e os objetivos específicos. Os objetivos específicos atingidos, levam a concretização do objetivo geral. A seguir apresento o objetivo geral e seus objetivos específicos.

2.1 Objetivo Geral

Estudar, selecionar e utilizar algoritmos de *machine learning* para detectar falhas em refrigeradores utilizando dados de medições de ensaios específicos já existentes.

2.2 Objetivos Específicos

Visando atingir o objetivo geral traçado para este trabalho de forma mais organizada, foram estipulados objetivos específicos, sendo estes:

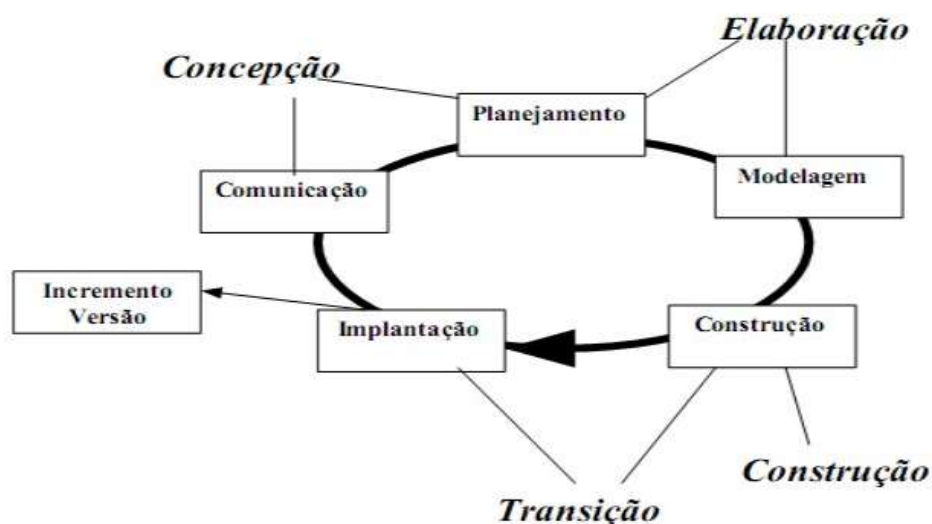
- Organizar dados de falhas de refrigeradores existentes numa base de dados;
- Selecionar uma biblioteca para tratamento de dados;
- Aplicar algoritmos de *machine learning* na base de dados;
- Gerar alerta para manutenção;
- Documentar os resultados obtidos.

3 METODOLOGIA

Segundo Sommerville (2007), existem quatro atividades fundamentais comuns a todos os processos de *software*, que são: especificação de *software*, desenvolvimento (projeto e implementação), validação de *software* e evolução.

No desenvolvimento do projeto vamos utilizar a metodologia do processo unificado, um processo centrado na arquitetura, iterativo e incremental. O processo unificado é um modelo incremental representado por cinco fases (concepção, elaboração, construção, transição e produção) (PRESSMAN, 2010). Nesse projeto não será utilizada a quinta fase, fase de produção, por não possuir um acompanhamento do projeto após o término do mesmo. A principal diferença entre o modelo incremental e o processo unificado é que no processo unificado, é possível o recomeço de ciclo ao fim da implantação. Suas fases serão organizadas por atividades nas quais são geradas através de um planejamento, o término do ciclo não significa que a atividade está completa. É possível fazer incremento de versão renovando a atividade e refinando o projeto como mostrado na Figura 1.

Figura 1 – Processo unificado.



Fonte: PRESSMAN (2010).

3.1 Fase de concepção

Nesta fase, atividades de planejamento, se enfatiza o desenvolvimento e refinamento de casos de uso como o modelo principal. Os requisitos de negócio para o *software* devem ser identificados, um rascunho de arquitetura do sistema (principais subsistemas e interações) deve ser proposto e o plano (considerando um desenvolvimento iterativo e incremental) que vai ser seguido deve ser elaborado. O planejamento identifica recursos, avalia os riscos e elabora um cronograma (PRESSMAN, 2010).

3.1.1 Definição do modelo principal

Levantamento dos possíveis benefícios do trabalho e sua viabilidade técnica.

3.1.2 Planejamento preliminar

Definições de parâmetros do sistema e elaboração de um cronograma para o projeto.

3.2 Fase de elaboração

A fase de elaboração tem como objetivo a modelagem com foco na elaboração de modelos de análise e projeto com ênfase no aprimoramento da arquitetura, subsistemas, identificação e definição de classes. A fase de elaboração refina e expande os casos de uso preliminares que foram desenvolvidos como parte da fase de concepção e expande a representação arquitetural para incluir cinco visões diferentes de *software*. São elas: o modelo de casos de uso, o modelo de análise, de projeto, de implementação e de implantação (PRESSMAN, 2010).

3.2.1 Aprimoramento

Detalhamento dos requisitos e melhoria na arquitetura inicial, nos subsistemas e nas definições para que o começo do projeto inicie com os melhores

parâmetros possíveis.

3.2.2 Revisar

Verificar os benefícios do trabalho e sua viabilidade técnica.

3.3 Fase de construção

Nesta fase é realizada a conversão do modelo de projeto para componentes de *software* implementados. Usando o modelo arquitetural como entrada, na fase de construção, os componentes de *software* são desenvolvidos ou adquiridos para tornar cada caso de uso operacional. As funções requeridas do incremento (versão) serão implementadas. Desta forma, os componentes são implementados, testes unitários são projetados e executados para cada um deles. As atividades de integração (montagem de componentes e testes de integração) são realizadas (PRESSMAN, 2010).

3.3.1 Organização dos requisitos

Os requisitos antes detalhados, são tratados e organizados para serem usados na próxima etapa.

3.3.2 Desenvolvimento, análise e implementação

Com os requisitos organizados e tratados, utiliza técnicas de aprendizado de máquina, análises e implementações de melhorias.

3.4 Fase de transição

Essa fase, está relacionada a estabilidade do protótipo, é feito validações nos resultados. Na conclusão da fase de transição, o incremento de *software* se torna uma versão utilizável do *software* (PRESSMAN, 2010).

3.4.1 Validação

Utiliza os dados de testes para validação dos resultados.

3.4.2 Documentação

Realizado todas as documentações do projeto e disponibilizada para a comunidade.

4 FUNDAMENTAÇÃO DE REFRIGERADORES

Os refrigeradores domésticos são uma necessidade da sociedade moderna, fazem parte do cotidiano das pessoas e são extremamente importantes para a conservação dos alimentos. O refrigerador é responsável por retirar o calor de um ambiente, embora conforme DA COSTA (1982, p.18), a retirada de calor de um corpo seja designada, de uma maneira geral, de refrigeração, podemos utilizar o mesmo conceito e mesmas distinções para um ambiente, visto que o ar que está no ambiente é considerado um corpo.

A refrigeração pode ser obtida de forma distinta pelos processos de arrefecimento, resfriamento e congelamento. O arrefecimento é o abaixamento de temperatura de um corpo até a temperatura ambiente (TTS ou eventualmente TTU). O resfriamento é o abaixamento da temperatura de um corpo da temperatura ambiente até a temperatura de congelamento ($\approx 0\text{ }^{\circ}\text{C}$) e o congelamento é o abaixamento da temperatura de um corpo aquém da sua temperatura de congelamento (DA COSTA, 1982).

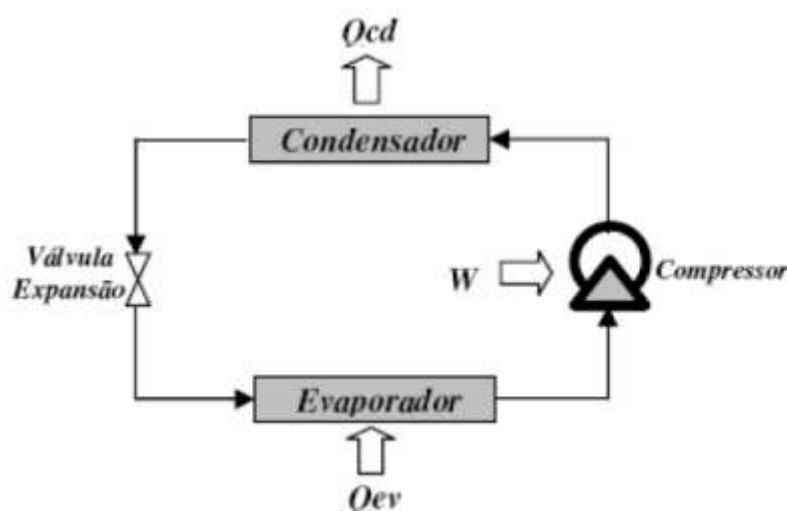
Muitos autores gostam de utilizar somente os termos resfriamento e congelamento quando tratam de refrigeração, pelo simples motivo que o arrefecimento ocorre naturalmente (passagem de calor do corpo quente para o corpo frio). No resfriamento e congelamento é necessário um fluxo de calor no sentido contrário, exigindo dispêndio de energia utilizável. Então para o calor passar do corpo com menor temperatura para o de maior temperatura, é necessária a entrega de energia para que algum componente do circuito realize trabalho sobre o fluido refrigerante (ÇENGEL; BOLES, 2005).

Os primeiros refrigeradores para uso doméstico foram criados na década de 1910. Até meados do século XIX os processos de refrigeração utilizavam gelo como meio de refrigeração, ou também chamado de refrigerante. Com o avanço da industrialização mundial tornou-se necessária a evolução dos sistemas de refrigeração. Métodos que se baseiam na troca de fase líquido-gasosa de refrigerantes foram desenvolvidos para suprir as necessidades dessa nova indústria. Esses novos sistemas muito se pareciam com os utilizados hoje em dia em aplicações domésticas e comerciais, entretanto, não eram de fácil aplicabilidade por

necessitarem de mão de obra qualificada de plantão para lidar com os constantes problemas (DOSSAT, 1963).

O princípio de funcionamento de tais sistemas de refrigeração baseia-se também na entrega de energia ao sistema para que esse, através de um processo de compressão e expansão do fluido, tornasse possível sua mudança de fase. Esse processo ficou conhecido como refrigeração por compressão mecânica (DOSSAT, 1963).

Figura 2 – Circuito de refrigeração por compressão mecânica.



Legenda:

Q_{cd} – taxa de calor cedida ao ambiente;
 Q_{ev} – taxa de calor absorvida do ambiente;
 W – potência entregue ao compressor.

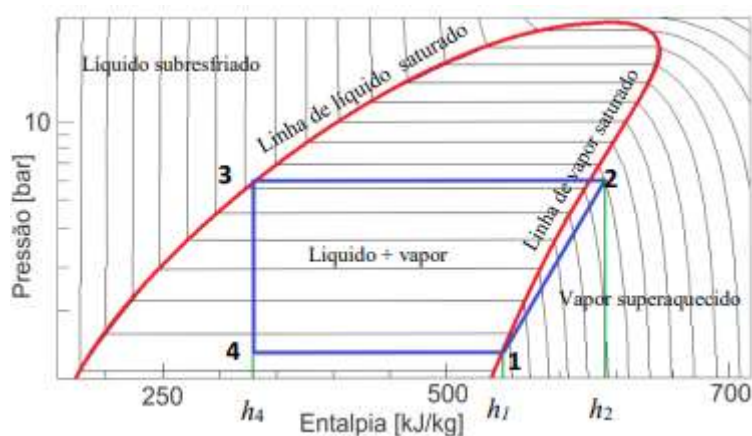
Fonte: GARCIA et al. (2007).

O circuito de refrigeração por compressão mecânica demonstrado na figura 2 tem o seguinte funcionamento: o fluido refrigerante entra no evaporador a baixa pressão, na forma de mistura de líquido mais vapor, e retira energia do meio interno enquanto passa para o estado de vapor. O vapor entra no compressor onde é comprimido e bombeado, tornando-se vapor superaquecido e deslocando-se para o condensador, que tem a função de liberar a energia retirada do ambiente. O fluido, ao liberar energia, passa do estado de vapor superaquecido para líquido (condensação) e finalmente entra no dispositivo de expansão, onde tem sua pressão reduzida, para novamente ingressar no evaporador e repetir-se assim o ciclo (FERRAZ, 2008).

O circuito da Figura 2 trata-se de um circuito ideal, onde não existem perdas da energia entregue ao sistema. Na prática, entretanto, o processo de compressão não é isentrópico, pois, parte da energia usada para elevar a pressão e a temperatura do fluido é cedida ao corpo do compressor e ao ambiente. Além disso, essa perda de energia não é constante, pois, as temperaturas do corpo do compressor e do interior do cilindro, onde ocorre a compressão, variam durante o funcionamento do mesmo. Essa variação da perda de calor é de difícil quantificação (GOMES, 2015).

Na figura 3 está mostrando um diagrama de entalpia e de pressão do fluido refrigerante, os números mostrados nela podem ser localizados em intervalos na figura 2, o número 1 seria o intervalo entre o evaporador e o compressor. O 2 entre o compressor e o condensador. O número 3 entre o compressor e a válvula de expansão e o número 4 entre a válvula de expansão e o evaporador. Na figura 3 observa-se três regiões distintas: região de fluido saturado (líquido + vapor); zona de líquido subresfriado (à esquerda da região de saturação); zona de vapor superaquecido (à direita da região de saturação). No interior da região de saturação, o fluido encontra-se saturado em uma mistura de líquido e vapor (GOMES, 2015).

Figura 3 – Diagrama P x h do Isobutano.



Legenda:

- | | |
|--------------------------|--------------------------------------|
| 1 Vapor saturado; | h1 entalpia do isobutano no ponto 1; |
| 2 Vapor superaquecido; | h2 entalpia do isobutano no ponto 2; |
| 3 Líquido saturado; | h4 entalpia do isobutano no ponto 4; |
| 4 Mistura líquido-vapor. | |

Fonte: GOMES (2015).

Os circuitos de refrigeração e seus componentes são projetados com objetivo de minimizar a quantidade de energia entregue ao sistema e, ao mesmo tempo, com isso maximizar a troca de calor no interior da câmara de refrigeração.

Segundo GOMES (2015), diversos fatores podem alterar o rendimento de um sistema de refrigeração ou de um compressor; alguns deles ambientais, sendo a temperatura ambiente uma das principais variáveis, e outros internos ao sistema. Os fatores ambientais não podem ser controlados, entretanto, as questões internas são dependentes do projeto e da correta fabricação, instalação e funcionamento dos componentes que compõem o sistema. O incorreto controle de variáveis nos processos de fabricação pode tornar um sistema potencialmente eficiente, ou seja, um sistema corretamente projetado e dimensionado, em um sistema com baixa eficiência. Um exemplo comum de defeito de fabricação é a incorreta carga de fluido refrigerante, a qual afeta diretamente a capacidade de refrigeração do sistema. Outros fatores como: qualidade da vedação das válvulas de controle da sucção e de descarga do compressor; vazamentos entre o pistão e o cilindro; rendimento do motor elétrico, podem afetar também o rendimento do compressor e, conseqüentemente, do sistema de refrigeração.

Tabela 1 - Defeitos comuns em sistemas de refrigeração.

<i>Defeitos</i>	<i>Efeito esperado sobre o sistema de refrigeração</i>
<i>Carga de fluido refrigerante fora do valor nominal</i>	Alteração da vazão mássica do sistema, do consumo energético e das pressões e temperaturas de operação.
<i>Carga de óleo fora do valor nominal</i>	Alterações no consumo energético (tendendo ao aumento), nas pressões e nas temperaturas de operação.
<i>Obstrução na linha de líquido do circuito</i>	Diminuição da pressão de sucção do sistema, resultando em menor consumo energético. Diminuição da capacidade de refrigeração do sistema.
<i>Vazamentos – válvulas ou interface pistão-cilindro</i>	Diminuição da vazão mássica do sistema, resultando em uma menor capacidade de refrigeração. Pouco efeito sobre o consumo energético. Ruído excessivo.
<i>Problemas elétricos</i>	Não partida do compressor, consumo energético elevado, ruído excessivo.

Fonte: GOMES (2015).

Na tabela 1, foram sistematizados os defeitos e seus efeitos. Esta tabela prioriza a frequência de ocorrência dos defeitos, relacionando estes com os seus respectivos efeitos sob o funcionamento do sistema de refrigeração.

4.1 Obstrução na linha de líquido

Para que o fluido refrigerante existente em sistemas de refrigeração efetue o seu papel de retirar calor do sistema, ele deve passar de um estado de alta pressão para um estado de baixa pressão.

O dispositivo responsável por essa queda de pressão é o dispositivo de expansão. Ele impõe uma restrição ao fluxo que resulta em uma perda de carga conhecida.

Para que ocorra a perda de carga necessária ao fluido existem duas principais alternativas utilizadas em sistemas de refrigeração: válvulas de expansão e tubos capilares.

Em sistemas de refrigeração domésticos é mais comum o uso de tubo capilar (ÇENGEL; BOLES, 2005; DOSSAT, 1963). Se ocorre uma obstrução no tubo capilar o fluido tenderá a sofrer uma queda de pressão maior do que esperada entre as zonas de alta e baixa pressão.

A maneira com que esse defeito se manifesta no sistema pode ser observada no aumento das pressões e temperatura interna do sistema. Outra falha causada por esse defeito será o não alcance da temperatura esperada no compartimento do congelador dentro do tempo esperado. Todavia, esse efeito da temperatura pode não ser detectado tão rapidamente quanto o efeito da diferença das pressões na entrada e na saída do compressor, dada à inércia térmica do sistema (EMBRACO, 2009).

4.2 Carga de fluido refrigerante fora do valor nominal e vazamentos do fluido para o ambiente

Tanto a carga de fluido fora do valor nominal, quanto o vazamento de fluido, geram consequências relacionadas a rendimento, consumo e vida útil do sistema (GOMES, 2015).

O vazamento de fluido pode ocorrer pelo mal processo de brasagem em pontos da tubulação do circuito de refrigeração, já a quantidade de fluido refrigerante existente nos sistemas de refrigeração tende a afetar a sua capacidade de refrigeração. A capacidade de sistemas de refrigeração é medida com relação à quantidade de calor que esse consegue remover da câmara de refrigeração por unidade de tempo (GOMES, 2015).

Esse defeito fará com que a temperatura do compartimento do congelador e de outras partes do refrigerador não atinjam as temperaturas projetadas dentro do tempo determinado ou fará com que elas não se estabilizem tão facilmente. As pressões de sucção e de descarga também tenderão a estar fora dos valores projetados. O consumo elétrico tende a sofrer alterações, uma vez que a potência consumida pelo compressor tem correlação com as condições termodinâmicas do fluido que passa pelo mesmo (EMBRACO, 2014; LI, HAORONG, 2006).

4.3 Válvulas defeituosas

Refrigeradores de pequeno porte possuem como elemento responsável pela entrega de energia mecânica ao sistema, compressores do tipo hermético de movimento alternativo acionado por mecanismo do tipo, biela manivela (PACHECO, 2007).

O incorreto funcionamento das válvulas internas ao compressor, mais especificamente vazamentos decorrentes de má vedação, podem afetar significativamente as eficiências volumétrica e isentrópica dos compressores (MORRIESEN; DESCHAMPS, 2011; SILVA, R.; DESCHAMPS, 2012).

Os efeitos desses defeitos se refletem, principalmente, na capacidade de refrigeração do sistema, devido à redução da vazão mássica (BALLMANN, 2010). A vazão mássica pode ser medida a partir de um balanço de energia onde se tem como

entrada a potência elétrica consumida pelo compressor e como saída a soma entre o produto do trabalho realizado (diferença da entalpia de saída e da entalpia de entrada) pela vazão mássica e a perda de calor (LI, HAORONG, 2006).

4.4 Defeitos elétricos

Um levantamento de 1989, sobre as origens das falhas em sistemas de refrigeração comerciais apontou uma grande superioridade na frequência de ocorrência de falhas elétricas (76%) sobre outras origens, como falhas mecânicas ou em componentes do sistema (STOUPPE; LAU, 1989). Por tratar-se de um trabalho com mais de três décadas pode-se apenas seguir tais valores como um indicador. Em um estudo mais atual, a parcela de defeitos elétricos no total de todos os defeitos apresentados em circuitos de refrigeração é de apenas 20% (BREUKER; ROSSI; BRAUN, 2000).

Os refrigeradores utilizados no presente trabalho utilizam a tecnologia de motores de corrente contínua do tipo *brushless* de ímãs permanentes com controle de frequência rotacional. Esse tipo de motor não possui escovas para alimentar o rotor, ao contrário de outros motores alimentados por corrente contínua. Ao invés de bobinas de cobre ou alumínio no rotor, como em motores de corrente contínua comuns e de indução, ele possui ímãs (HANSELMAN, 1994). Dessa forma, alguns defeitos como o curto-circuito entre fios do rotor e o estator não ocorrem, diminuindo a frequência de eventos de falhas elétricas nos compressores.

Por ser uma tecnologia que por si própria diminui a ocorrência de defeitos, e, pelo presente trabalho focar em falhas no ciclo de refrigeração, tais defeitos não serão mais abordados neste trabalho.

5 FUNDAMENTOS DO MACHINE LEARNING

Machine Learning pode ser definido como “o campo de estudo que dá ao computador a habilidade de aprender sem ser explicitamente programado” (Géron, 2017), pode-se dizer que os algoritmos de *machine learning* não se baseiam em equações pré-definidas como modelos e sim usam métodos para aprender diretamente dos dados. O *machine learning* é definido como um campo representado por porções dos algoritmos computacionais (responsáveis por entender e aprender padrões a serem previstos) dentro da área da Inteligência Artificial. A Inteligência Artificial é uma “área da ciência da computação que procura a criação de máquinas inteligentes que trabalham e reagem como os seres humanos” (Géron, 2017). É um sistema que consegue ser treinado através de dados, permitindo encontrar padrões, com o objetivo de auxiliar na tomada de decisões com a mínima intervenção humana.

5.1 Tipos de aprendizagem

A base de ML (machine learning) consiste em permitir que um algoritmo aprenda com base em um conjunto de dados. O algoritmo, com base nos dados fornecidos, tende a gerar e criar um conjunto de regras, com decisões e resultados fiáveis, que incluem previsões, com base na inferência de dados, a partir dos exemplos “aprendidos”. (Santos, 2019). Os algoritmos de ML não são novos, pois foram desenvolvidos há décadas, quando em 1950 Arthur Samuel da IBM desenvolveu um programa de computador para jogar damas. É importante referir que, quando estes algoritmos foram desenvolvidos, havia um grande problema: não havia a facilidade de aceder a uma grande quantidade de dados, o que é possível hoje em dia, e é esta mudança que permite a ML ter um grande crescimento e impacto. Desta forma, ML não só pode melhorar a previsão da procura, como também automatizar grande quantidade do trabalho de planeamento, devido à capacidade de processamento de uma grande quantidade de dados, algo incapaz de ser feito pelo ser humano comum (Dickey et al., 2019; Marr, 2016; Santos, 2019). O modelo de *machine learning* é um arquivo que reconhece determinado tipo de padrões, ele utiliza um conjunto de dados para alimentar esse arquivo, onde define parâmetros para refinamento do modelo e métricas estatísticas que avaliam o seu desempenho. Após a criação do modelo, pode-se usar dados de teste para realizar a validação do acerto.

Segundo Santos (2019) e SAS (2021) os métodos de ML são tipicamente categorizados em três tipos de aprendizagem:

5.1.1 Supervisionada

Os algoritmos com base na aprendizagem supervisionada são treinados com base em dados de entrada e de saída que são inicialmente fornecidos ao algoritmo. Existe uma aprendizagem com base em saídas conhecidas, o que permite ao modelo adaptar-se e minimizando o seu erro, de modo, a aprender a melhor forma de obter o resultado conhecido com base em dados de entrada. Este tipo de aprendizagem é amplamente utilizado em aplicações onde os dados históricos podem prever eventos futuros.

5.1.2 Não supervisionada

Os dados de entrada são fornecidos ao algoritmo. O objetivo é que o algoritmo aprenda a reconhecer padrões nos dados fornecidos. É utilizado quando se pretende relacionar e agrupar dados sem que o algoritmo dê uma saída. Exemplos são o seu uso na identificação de segmentos de clientes que têm determinados atributos similares

5.1.3 Por reforço

Um ambiente dinâmico é configurado de modo a interagir com o algoritmo de aprendizagem com o objetivo de fornecer uma avaliação sobre a resposta do sistema, isto é, baseia-se no aprendizado por tentativa — erro, percebendo quais as possíveis ações que trazem mais benefícios. Este tipo de aprendizagem é normalmente utilizado em robótica, jogos e navegação.

Ao se criar um modelo, deve-se definir os parâmetros (GOOGLE CLOUDS..., 2021) a serem utilizados, esses parâmetros controlam o processo de treinamento do modelo, definindo a complexidade das relações encontradas no modelo. Estes parâmetros são otimizados durante o processo de criação do modelo, comparando com os valores reais e por fim ajustando até encontrar os melhores

valores. Atualmente existem diversos *frameworks* disponíveis para o estudo de dados e desenvolvimento de modelos classificatórios para as mais diversas linguagens de programação, desenvolvidos pelas mais diversas companhias como Microsoft (Desenvolvedora do LGBM), *XGBoost* e o *Catboost* (desenvolvido pela Yandex), são alguns dos mais utilizados. Junto com os diversos *frameworks* disponíveis existem também diversos tipos de classificadores como: Naïve Bayes (Rish, I., 2001), regressão linear dos mínimos quadrados, regressão logística e árvore de decisões, apenas para citar alguns.

A avaliação do desempenho de um algoritmo de ML pode ser realizada por inúmeras métricas estatísticas, sendo que a avaliação fidedigna de qual é mais apropriada para a situação depende de cada caso, pois, cada tipo de dado apresenta um episódio único. As métricas vão desde as estatísticas mais simples como a acurácia (a taxa de acerto geral das previsões), precisão (acertos das classes classificadas como positivas) e a matriz de confusão (matriz contendo todas as previsões para as classes) até métricas mais avançadas como o coeficiente de correlação de Matthews (Chicco, D., et al., 2020) e a área sob a curva ROC.

6 TECNOLOGIAS UTILIZADAS

A interação da sociedade com as tecnologias atuais, são fundamentais para a aprendizagem. O usuário deixa de ser passivo e começa a se tornar um ser ativo, chegando ao ponto máximo que seria utilizar as melhores tecnologias para resolver os problemas, contudo, sabemos que a tecnologia avança rapidamente e acompanhar essas mudanças é praticamente impossível. Na confecção de projetos que precisão da utilização de tecnologias específicas, é essencial que o desenvolvedor pesquise sobre essas tecnologias visando uma boa execução do projeto. A área de banco de dados, possui diversas tecnologias específicas para o tratamento e o aprendizado de modelos. Na figura 4 encontramos uma série de ferramentas essenciais para a confecção desse projeto.

Figura 4 – Tecnologias específicas.

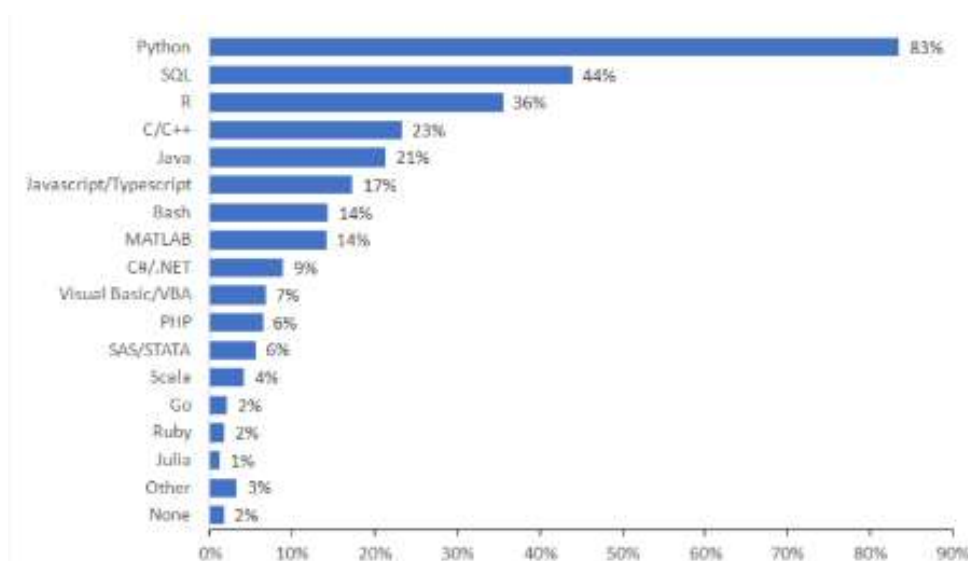


Fonte: do autor.

A prática de ciência de dados requer o uso de ferramentas de análise, tecnologia e linguagens de programação. Uma pesquisa com quase 24000

profissionais de dados da *Kaggle* (comunidade virtual voltada a cientistas de dados) revelou que *Python*, *SQL* e *R* são as linguagens de programação mais populares. O mais popular, de longe foi o *Python*, como mostrado na figura 5. Além disso, 3 a cada 4 profissionais de dados recomendaram que os aspirantes a cientistas da área aprendam primeiramente *Python* (HAYES, 2019).

Figura 5 – Linguagem de programação mais usadas em banco de dados.



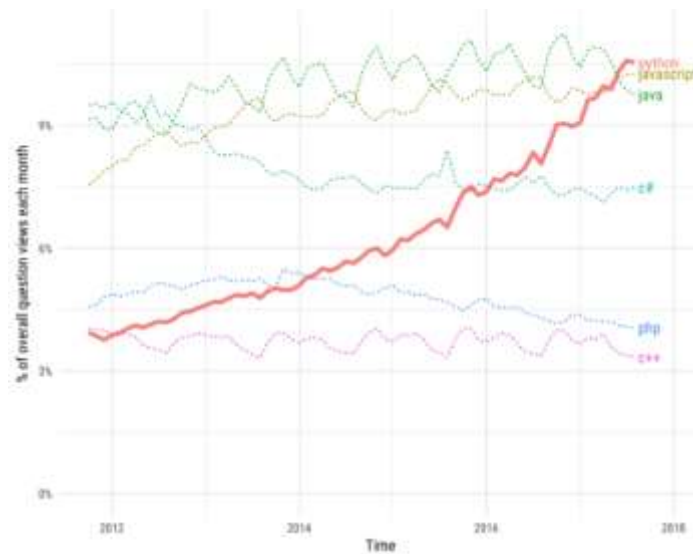
Fonte: HAYES (2019)

6.1 Linguagem de programação *python*

Python é uma linguagem de programação ágil, fácil e objetiva, o que democratiza seu ensino e a faz procurada cada vez mais. É uma linguagem orientada a objetivos criada em 1991 e a sua simplicidade é sua maior característica. Conta com um modelo comunitário de desenvolvimento que é aberto e sem fins lucrativos. A figura 6 mostra o crescimento do *python* nos últimos anos, considerando pesquisas por *tags*. Quando falamos em programação *python* é indispensável comentar sobre as suas bibliotecas, que auxiliam os profissionais a resolverem os problemas e facilitam na programação.

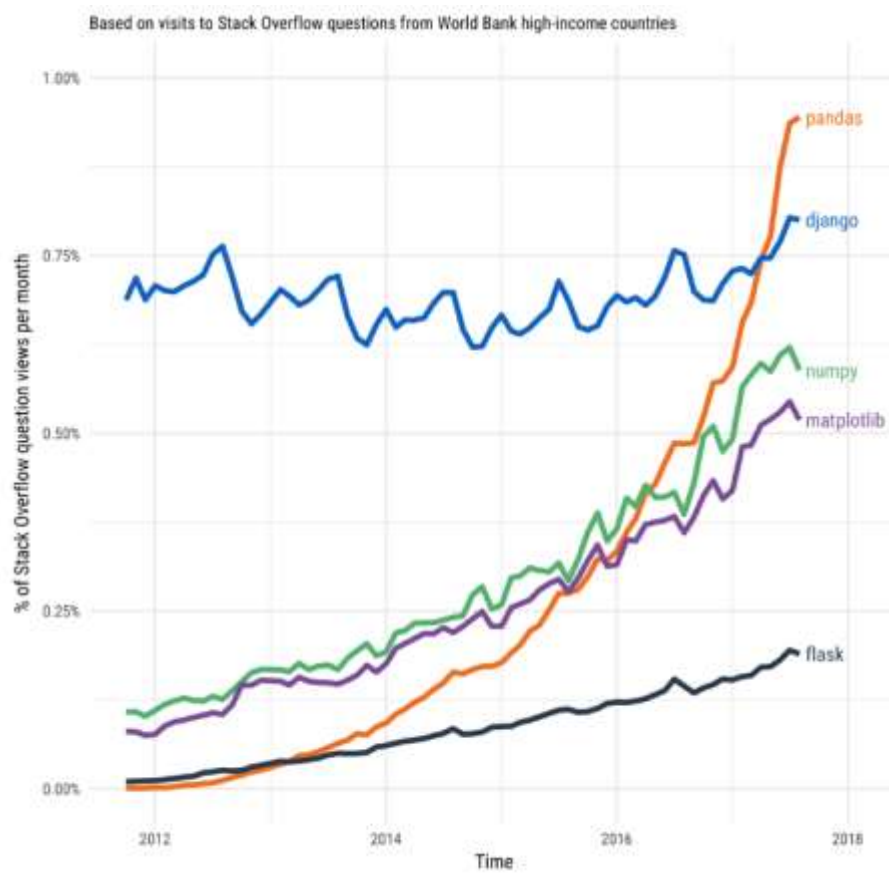
Na parte de tratamento de dados é nítido o uso de três bibliotecas específicas: *pandas*, *numpy* e *matplotlib*. Que possuem um aumento significativo no uso conforme mostrado na figura 7.

Figura 6 – Crescimento do *python* no intervalo de 2012 até 2018.



Fonte: Robinson (2017).

Figura 7 – Crescimento de pesquisas utilizando as tags *pandas*, *numpy* e *matplotlib*.



Fonte: Robinson (2017).

6.1.1 Biblioteca *pandas*

Pandas é um pacote que oferece estruturas de dados rápidas, flexíveis e expressivas, tornando o trabalho com dados fácil e intuitivo, é adequado para diferentes tipos de dados, possibilitando um fácil manuseio dos dados ausentes, mutabilidade de tamanho, alinhamento de dados automáticos, agrupamento flexível, etc (MCKINNEY, 2022).

6.1.2 Biblioteca *numpy*

Numpy traz o poder computacional de linguagens como C e *Fortran* para o *python*, uma linguagem muito mais simples de aprender e usar. Oferecendo funções matemáticas abrangentes, geradores de números aleatórios, rotinas de álgebra linear, transformadas de Fourier, matrizes de N dimensões e muito mais (OLIPHANT, 2022).

6.1.3 Biblioteca *matplotlib*

Matplotlib é uma biblioteca para criação e visualização estáticas, podendo ampliar, deslocar, atualizar, exporta para vários formatos de arquivos... (ROSSUM, 2022).

Com isso, foi analisado que a utilização da linguagem de programação *python* e suas bibliotecas seriam de grande utilidade para o tratamento dos dados e também para criação do modelo de aprendizagem de máquina. Além das bibliotecas acima é importante citar a utilização do módulo “os” (diversas interfaces de sistema operacional), para usar facilmente as funcionalidades que são dependentes de sistema operacional, a biblioteca *scikit-learn* (*Machine learning in python*) com seus algoritmos e a plataforma *jupyter notebook*, onde foi desenvolvido o *software*.

7 DESENVOLVIMENTO E APLICAÇÃO DO ML

Neste capítulo são apresentados os tipos de dados de refrigeradores utilizados, como foram obtidos esses dados, as ferramentas utilizadas durante o processo de tratamento dos dados, o modelo de ML e suas seguintes aplicações. Além disso, são realizadas análises de desempenho por métricas estatísticas, resultando no escopo da aplicação, encontrar uma possível falha, emitindo um alerta para o usuário para que seja possível uma manutenção prévia do equipamento de refrigerador, em busca de prolongar o tempo de utilização do mesmo.

7.1 Conjunto de dados

Em vista da confiabilidade do estudo elaborado, a obtenção dos dados é uma das etapas mais importantes para esse processo. Quando procuramos por dados de refrigeradores para fins de estudos de algoritmos de ML nos repositórios de bases de dados disponíveis e abertos, encontramos quase sempre dados com o objetivo de detectar alimentos e seus tipos no interior do refrigerador e, em pouquíssimos casos, dados relacionados a falhas, pelo simples fato de ser extremamente cansativo e demorado a obtenção dos mesmos, fato descrito por Gomes (2015): “com relação ao tempo mínimo entre ensaios, foi constatado que apenas ensaios realizados com intervalo de 24 horas entre si, apresentam características de funcionamento semelhantes durante os transitórios.”. No presente trabalho são utilizados dados retirados desta dissertação, os quais foram disponibilizados pelo autor. Para a coleta dos dados foram utilizados três refrigeradores equipados com compressor de capacidade variável no qual suas características estão na tabela 2.

Tabela 2 – Características técnicas dos refrigeradores utilizado.

<i>Carga de refrigerante</i>	40 g
<i>Tipo de refrigerante</i>	R600a
<i>Tensão de alimentação</i>	220 V
<i>Potência elétrica</i>	125 W
<i>Frequência</i>	60 Hz
<i>Volume total</i>	428 l

Fonte: GOMES (2015).

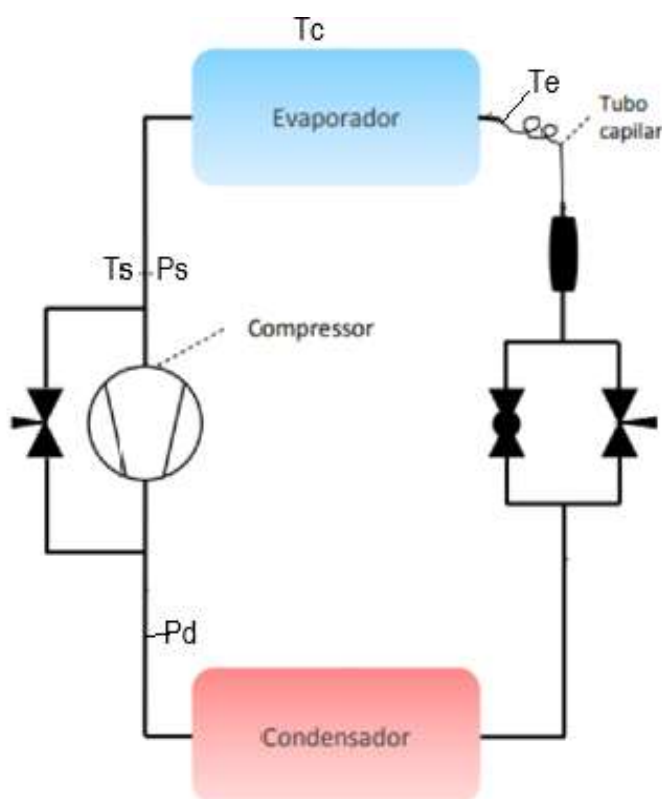
Medições diretas, indiretas e também estimativa de grandezas foram realizadas, para obtenção dos dados e, no seguinte trabalho, iremos priorizar algumas medições pelo fato de serem pontos considerados mais importantes para identificar as falhas ocorridas pelos defeitos provocados e já citados no capítulo 3. Os pontos de medições utilizadas estão descritos pela tabela 3 e expostos na figura 8.

Tabela 3 – Pontos de medição.

Grandezas	Pontos de medição	Identificação
Temperatura	Termostato do congelador	Tc
	Sucção do compressor	Ts
	Entrada do evaporador	Te
Pressão	Sucção do compressor	Ps
	Descarga do compressor	Pd
Corrente elétrica	Alimentação elétrica do refrigerador	-

Fonte: do autor.

Figura 8 – Posição dos pontos de medições utilizados.



Fonte: Adaptado de (GOMES, 2015).

Os dados dispostos constituem-se em aproximadamente 100 ensaios de 3 tipos de refrigeradores, com um período de 30 minutos. Cada ensaio possui um tempo mínimo de espera de 24 horas para o ensaio subsequente, totalizando mais de 2400 horas para aquisição desses dados, que foram dispostos em arquivos no formato “CSV”. Cada ensaio possui de 3 a 9 arquivos, pois foram realizados alguns ensaios para obter dados dos refrigeradores em perfeito estado (sem nenhuma alteração). Na tabela 4 mostra a quantidade de medições para cada grandeza que será utilizada nesse trabalho.

Tabela 4 – Disposição de dados.

Grandezas	Pontos de medição	Número de amostra p/ensaio
Temperatura	Termostato do congelador	902
	Sucção do compressor	902
	Entrada do evaporador	902
Pressão	Sucção do compressor	8993
	Descarga do compressor	8993
Corrente elétrica	Alimentação elétrica do refrigerador	18001

Fonte: do autor.

7.2 Tratamento dos dados tabulares

Visando a organização dos dados fornecidos pela dissertação de mestrado, “Avaliação do comportamento de refrigeradores domésticos frente a defeitos provocados e emulados”, Gomes (2015), decidiu-se escolher de todos os arquivos de ensaio somente aqueles marcados na figura 9, pela importância citada no início deste capítulo.

Após a escolha dos arquivos, utilizou-se a biblioteca *pandas* para abrir os dados no *jupyter notebook*, com o objetivo de analisar como os dados estavam organizados. Nas figuras 10, 11 e 12, é mostrado como os dados estavam distribuídos e suas quantidades.

Figura 9 – Arquivos escolhidos para o tratamento dos dados.

 cabeçalho	08/02/2022 00:22	Planilha do Micro...	1 KB
 Frequência Rotacional	08/02/2022 00:22	Planilha do Micro...	159 KB
 Grandezas Elétricas	08/02/2022 00:22	Planilha do Micro...	399 KB
 Pressões	08/02/2022 00:22	Planilha do Micro...	334 KB
 Temperaturas Realinhadas	08/02/2022 00:22	Planilha do Micro...	49 KB
 Temperaturas	08/02/2022 00:22	Planilha do Micro...	54 KB
 Vazão Mássica	08/02/2022 00:22	Planilha do Micro...	19 KB

Fonte: do autor.

Como analisado (nas figuras 10, 11, 12) as tabelas possuem diferentes quantidades de linhas, com a informação que cada ensaio levou 30 minutos de execução é essencial deixar a mesma quantidade de linhas para todos os arquivos, tendo em vista a divisão igualitária no tempo de 30 minutos de execução do ensaio. Para deixar todos os arquivos com a mesma quantidade de linha foram criados vários grupos de denominação 0,1,2...900,901. Realizou-se a média aritmética para cada grupo, conforme mostrado na figura 13. Observou-se, também, que nos dados de pressão a posição 0x0 = 1 não possuía nenhum significado de medida, realizou-se uma lógica que quando a posição 0x0 == 1 excluiria a primeira coluna como mostrado na figura 14.

Figura 10 – Dados de Temperaturas retirados no dia 14 de setembro de 2014.

	a	b	c	d	e	f	g	h
0	23.856	23.728	23.805	25.757	23.967	24.014	24.098	23.853
1	23.856	23.728	23.805	25.757	23.967	24.014	24.098	23.853
2	23.857	23.726	23.805	25.758	23.968	24.013	24.099	23.857
3	23.858	23.728	23.806	25.758	23.968	24.013	24.099	23.856
4	23.857	23.728	23.808	25.756	23.968	24.013	24.100	23.857
...	...	...	...	...	...	...	...	...
897	27.367	27.458	43.859	35.063	-33.858	9.506	10.770	24.537
898	27.365	27.456	43.835	34.920	-32.236	9.514	10.775	24.537
899	27.364	27.455	43.813	34.794	-31.395	9.510	10.775	24.534
900	27.363	27.451	43.799	35.064	-31.224	9.497	10.772	24.529
901	27.362	27.452	43.793	35.332	-30.876	9.478	10.765	24.525

Fonte: do autor.

Figura 11 – Dados de Pressões retirados no dia 14 de setembro de 2014.

	i	j	k
0	1.0	1.842415	1.860448
1	NaN	1.842482	1.860393
2	NaN	1.842482	1.860393
3	NaN	1.842695	1.860296
4	NaN	1.842580	1.860439
...	...	...	...
8988	NaN	4.803410	0.319914
8989	NaN	4.803760	0.320987
8990	NaN	4.804008	0.321756
8991	NaN	4.804632	0.321734
8992	NaN	4.804580	0.321354

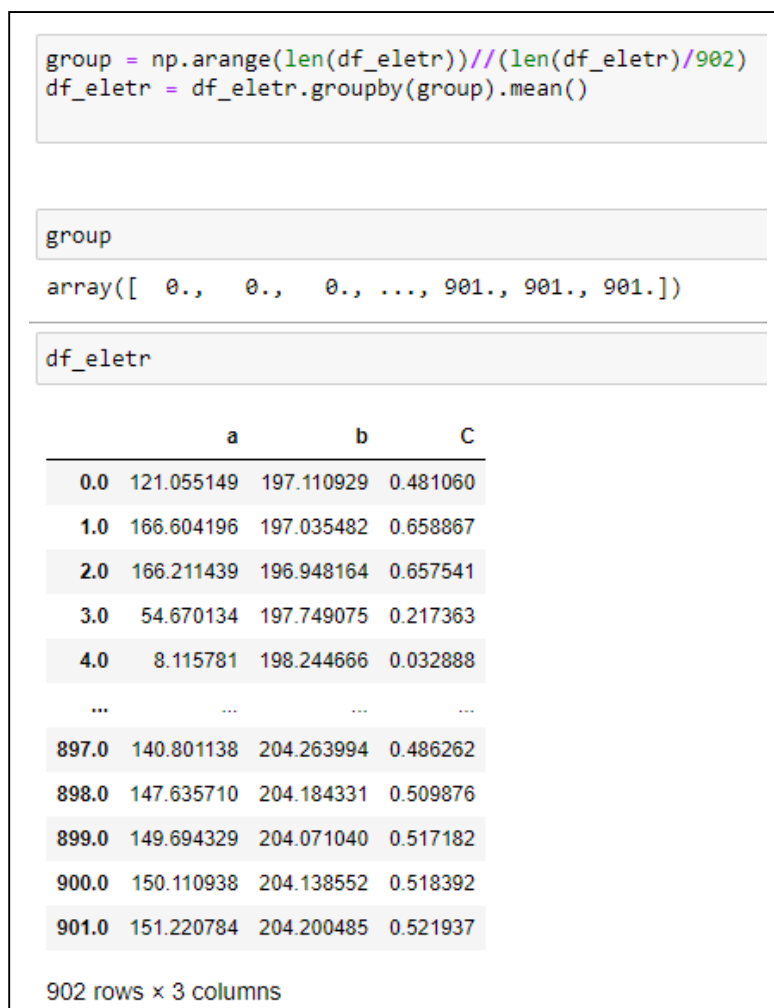
Fonte: do autor.

Figura 12 – Dados de Grandezas Elétricas retirados no dia 14 de setembro de 2014.

	a	b	C
0	6.875451	197.410101	0.037542
1	6.875451	197.410101	0.037542
2	7.264526	197.605238	0.038946
3	7.581506	197.237454	0.039728
4	7.266561	197.358906	0.039297
...	...	...	...
17998	151.665581	204.207510	0.523478
17999	151.754662	204.155568	0.523680
18000	151.690932	204.205342	0.523296
18001	151.945005	204.150489	0.524746
18002	151.790307	204.239895	0.523858

Fonte: do autor.

Figura 13 – Adequação dos arquivos para mesma quantidade de linhas.



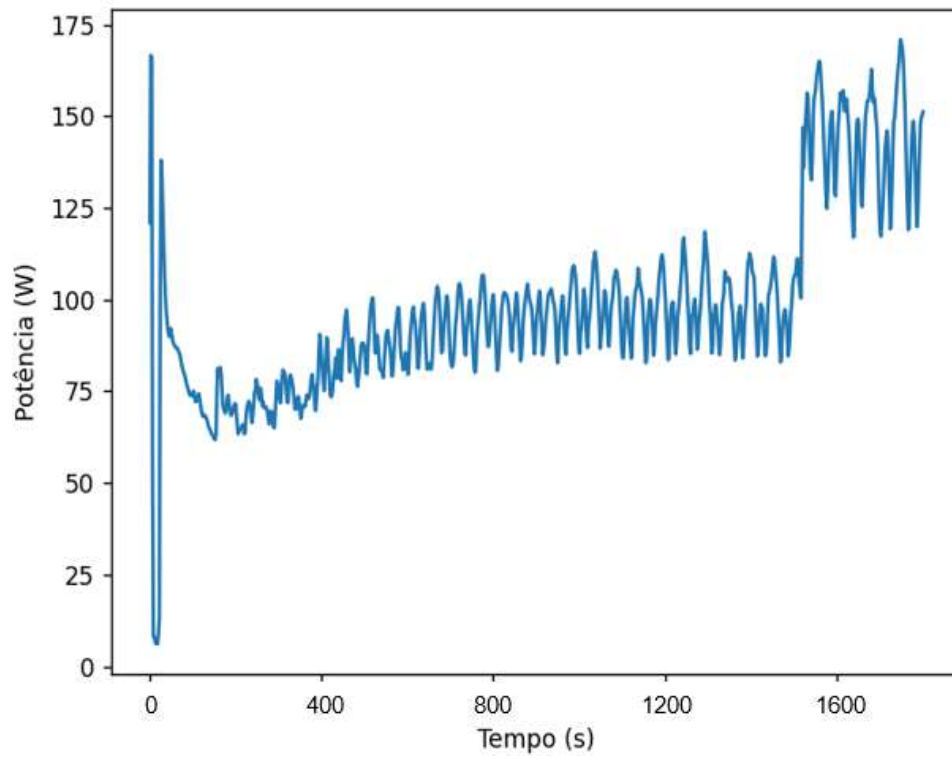
Fonte: do autor.

Figura 14 – Lógica para eliminar o erro da coluna 0x0 == 1.

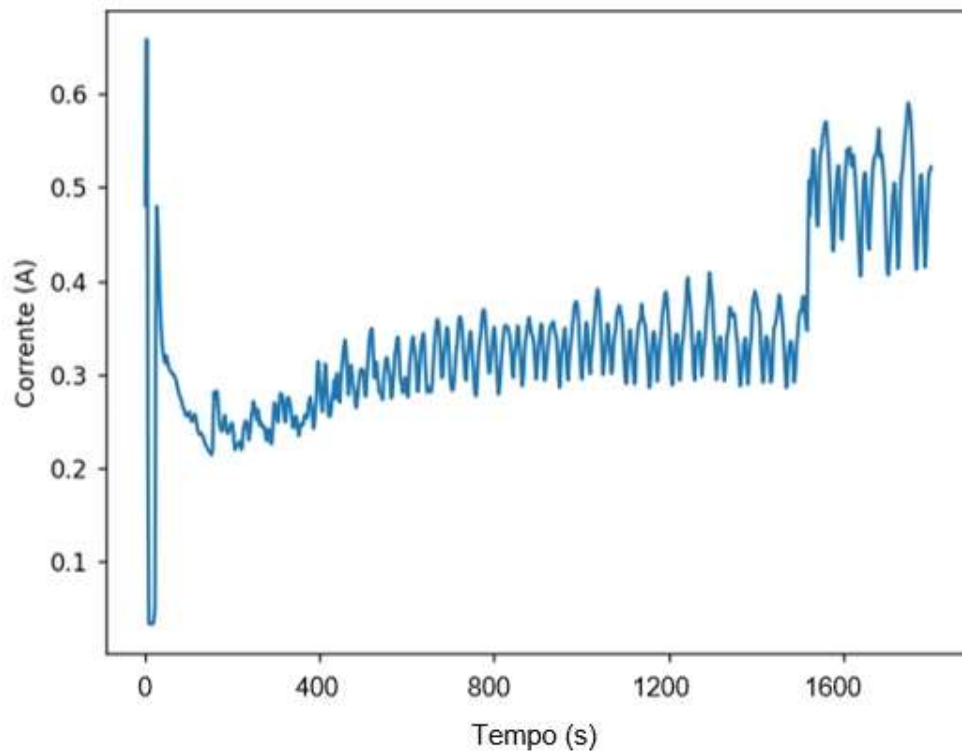
```
df_pres= pd.read_csv(path+'/Pressões.xls', delimiter='\\t',header = None, decimal = ",")
df_pres
if (df_pres.iloc[0,0] == 1):
    usecols = [1,2]
else:
    usecols = [0,1]
df_pres= pd.read_csv(path+'/Pressões.xls', delimiter='\\t',names = ["a","b"], decimal = ",", usecols = usecols)
```

Fonte: do autor.

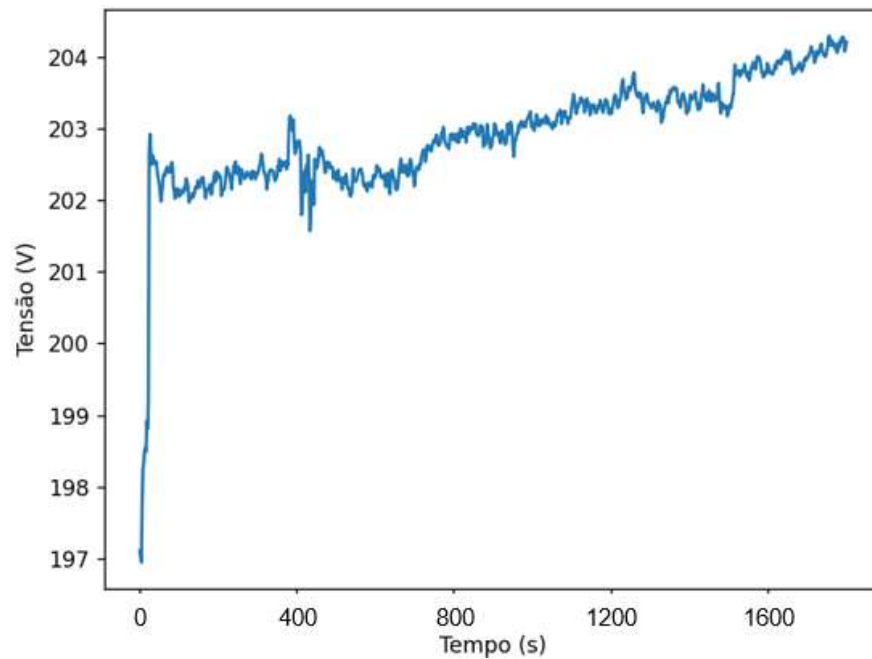
Os gráficos obtidos a partir dos arquivos de grandezas elétricas, pressões e temperaturas, foram exibidos para conseguir nomear as colunas e definir os atributos, no qual serão utilizados no futuro treinamento dos dados. Conforme as figuras 15, 16, 17, 18, 19 e 20.

Figura 15 – Gráfico de Potência X Tempo.

Fonte: do autor.

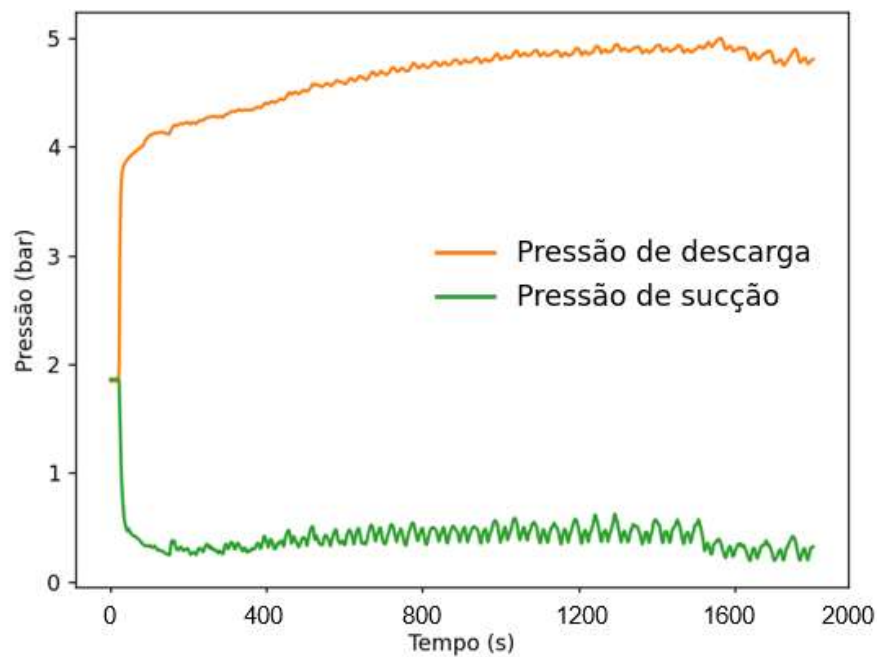
Figura 16 – Gráfico de Corrente X Tempo.

Fonte:do autor.

Figura 17 – Gráfico de Tensão X Tempo.

Fonte: do autor.

Com os dados obtidos, foi possível separar os gráficos de grandezas elétricas nas figuras 15 (curvas de potência elétrica), figuras 16 (curvas de corrente), figuras 17 (curvas de tensão).

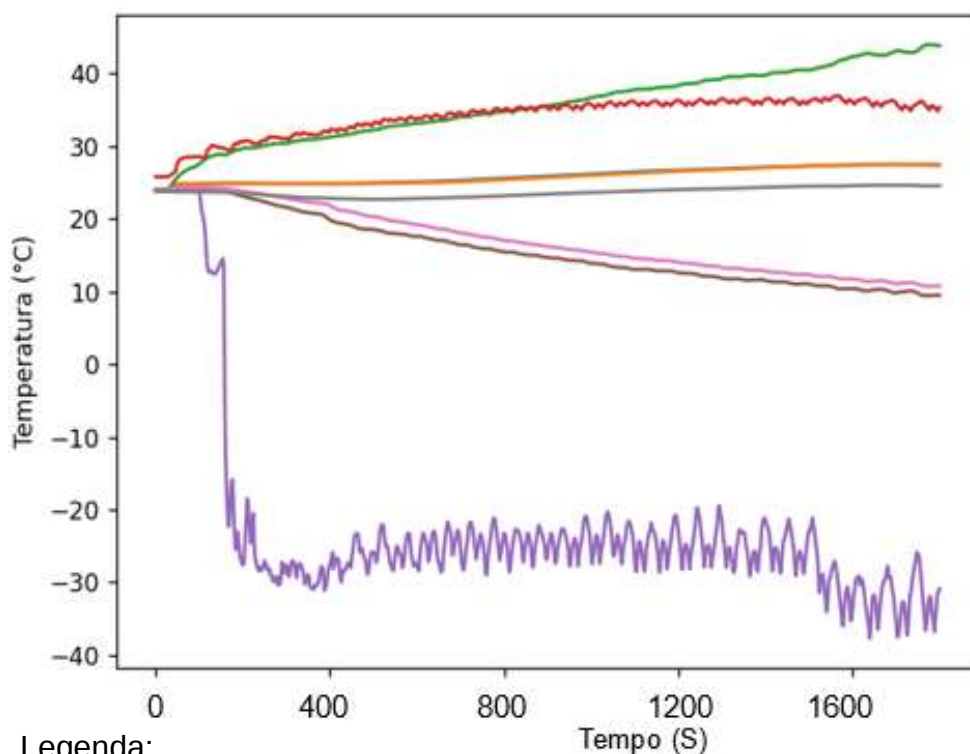
Figura 18 – Gráfico de pressão.

Fonte: do autor.

No gráfico de pressão (figura 18) temos as curvas de pressão de descarga e a curva de pressão de sucção.

No gráfico de temperaturas (figura 19), retirou-se as curvas de temperatura que não são importantes para o projeto, resultando no gráfico de temperaturas específicas (figura 20). As curvas escolhidas foram: curva de temperatura na entrada do evaporador (T_e), curva de temperatura na sucção do compressor (T_s) e a curva de temperatura no termostato do congelador (T_c). Para retirar as curvas que não foram usadas, foi realizado uma lógica onde comparava o último e o primeiro valor, se o último fosse menor que o primeiro escolhia a curva.

Figura 19 – Gráfico de temperaturas.

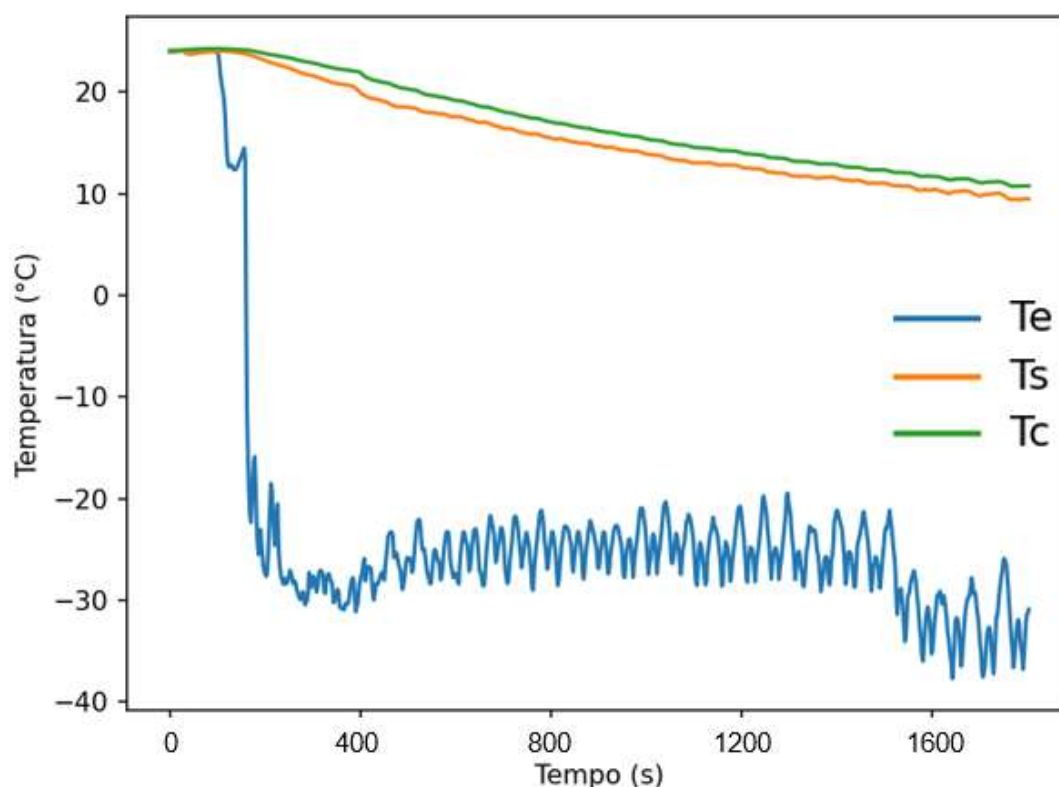


Legenda:

- | | | | |
|---|---|---|---|
| — T_d | – temperatura na descarga do compressor | — T_e | – temperatura na entrada do evaporador |
| — T_h | – temperatura no termostato do congelador | — T_s | – temperatura na sucção do compressor |
| — T_{aa} | – temperatura na entrada da válvula de agulha | — T_c | – temperatura no termostato do congelador |
| — T_{ab} | – temperatura na saída da válvula de agulha | | |

Fonte: do autor.

Figura 20 – Gráfico de temperaturas específicas.



Fonte: do autor.

7.2.1 Escolhas de atributos.

Com os gráficos já plotados e com a documentação mostrando as curvas de medições mais importantes, foi definido alguns parâmetros dessas curvas para utilizar como atributos no tratamento de dados. A escolha destes atributos baseou-se as variáveis de maior relevância para a detecção de falhas em refrigeradores apresentada por Gomes (2015). No gráfico de temperatura, foram obtidos os seguintes atributos:

$m_t_evaporador$ — média da temperatura no evaporador no intervalo de 10 e 20 minutos;

m_t_ts — média da temperatura no compressor no intervalo 10 e 20 minutos;

m_t_tc — média da temperatura no congelador entre o intervalo 10 e 20 minutos.

Esses três primeiros atributos e outros atributos de médias foram retirados de uma função, onde separa a quantidade de dados em partes e faz a média entre 1/3 e 2/3 dos dados sendo aproximadamente entre 10 e 20 minutos de ensaio, lógica mostrada na figura 21.

Figura 21 – Função *media_inter*.

```
def media_inter(col,tempin = 10,tempend = 20,tamint = 30):
    inicio = int(len(col)*tempin/tamint)
    end = int(len(col)*tempend/tamint)
    return col[inicio:end].mean()
```

Fonte: do autor.

t_fim_evaporador — temperatura final no evaporador;

t_fim_ts — temperatura final no compressor;

t_fim_tc — temperatura final no congelador.

Os dados de temperaturas finais, foram retirados pegando os últimos valores como mostrado na figura 22.

Figura 22 – Atributos de temperaturas finais.

```
t = "fim"
dados[f"t_{t}_evaporador"].append(df_temp["temp_evaporador"].values[-1])
dados[f"t_{t}_ts"].append(df_temp["temp_ts_refrigerador"].values[-1])
dados[f"t_{t}_tc"].append(df_temp["tem_tc_refrigerador"].values[-1])
```

Fonte: do autor.

t_maior_10/20_evaporador — Ponto de maior temperatura no intervalo de 10 a 20 minutos (evaporador);

t_maior_10/20_ts — Ponto de maior temperatura no intervalo de 10 a 20 minutos (compressor);

t_maior_10/20_tc — Ponto de maior temperatura no intervalo de 10 a 20 minutos (congelador);

t_menor_10/20_evaporador — Ponto de menor temperatura no intervalo de 10 a 20 minutos (evaporador);

t_menor_10/20_ts — Ponto de menor temperatura no intervalo de 10 a 20 minutos (compressor);

t_menor_10/20_tc — Ponto de menor temperatura no intervalo de 10 a 20 minutos (congelador).

Os maiores e menores valores foram obtidos considerando dos pontos 300 até os 600 como o intervalo de 10 e 20, foi utilizado a função `.max` e `.min`, para retirada desses valores como mostrado na figura 23.

Figura 23 – Atributos de temperaturas máximas e mínimas.

```
t = "maior_10/20"
dados[f"t_{t}_evaporador"].append(df_temp["temp_evaporador"][300:600].max())
dados[f"t_{t}_ts"].append(df_temp["temp_ts_refrigerador"][300:600].max())
dados[f"t_{t}_tc"].append(df_temp["tem_tc_refrigerador"][300:600].max())
t = "menor_10/20"
dados[f"t_{t}_evaporador"].append(df_temp["temp_evaporador"][300:600].min())
dados[f"t_{t}_ts"].append(df_temp["temp_ts_refrigerador"][300:600].min())
dados[f"t_{t}_tc"].append(df_temp["tem_tc_refrigerador"][300:600].min())
```

Fonte: do autor.

Os valores de temperatura foram adquiridos nos tempos de 5, 10, 15, 20 e 25 minutos, para criar 15 atributos onde o nome se difere somente pela troca do *t* pelos valores dos tempos (*t_{t}_evaporador*, *t_{t}_ts*, *t_{t}_início*), foram retirados através da lógica demonstrada na figura 24, considerando que possuímos aproximadamente 900 linhas na tabela sendo 30 minutos de ensaios.

Figura 24 – Atributos de temperaturas em pontos específicos.

```
for t in [5,10,15,20,25]:
    tempo = t*30
    dados[f"t_{t}_evaporador"].append(df_temp["temp_evaporador"][tempo])
    dados[f"t_{t}_ts"].append(df_temp["temp_ts_refrigerador"][tempo])
    dados[f"t_{t}_tc"].append(df_temp["tem_tc_refrigerador"][tempo])
```

Fonte: do autor.

Já nos gráficos de grandezas elétricas (figura 15, 16 e 17) foram obtidos os seguintes atributos:

corrente_média — Média da corrente no intervalo 10 a 20 minutos;

pico1_corrente — Valor de corrente quando ocorre o primeiro pico no gráfico;

pico2_corrente — Valor de corrente quando ocorre o segundo pico no gráfico.

A obtenção da corrente média é igual à da temperatura média (figura 21), o primeiro pico de corrente foi retirado pegando o valor máximo do intervalo de 1 até 4 minutos como mostrado na figura 25 e o segundo pico foi através de da taxa de crescimento dos últimos 5 pontos, começando de trás para frente quando a taxa for maior que 0,15 por cento ele encontra o segundo pico (figura 26).

Figura 25 – Primeiro pico de corrente.

```
dados["pico1_corrente"].append(corrente[30:120].max())
```

Fonte: do autor.

Figura 26 – Segundo pico de corrente.

```
c = df_eletr.iloc[4].argmin()
corrente = df_eletr.iloc[:,c]
s = (corrente.diff().iloc[1:]/corrente.iloc[:-1].values > 0.15)
pos = s[s].index[-1]
dados["pico2_corrente"].append(corrente[pos])
```

Fonte: do autor.

No gráfico de pressões (figura 18), foi retirado somente a média no intervalo de 10 a 20 minutos, utilizando a função *media_inter* mostrada na figura 21:

media_pres_des — média da pressão de descarga;

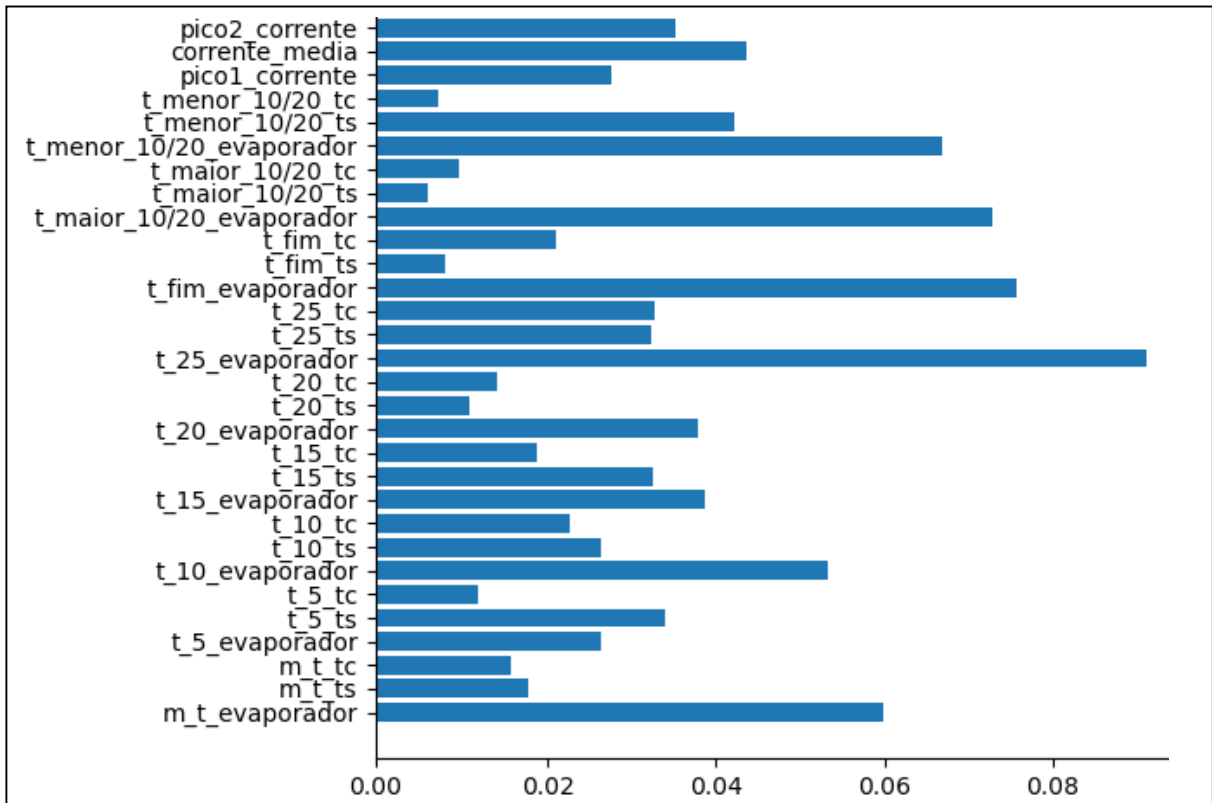
media_pres_suc — média da pressão de sucção.

Por fim foi plotado o gráfico dos atributos importantes para o modelo, mostrado na figura 27.

Os dados fornecidos das medições estavam organizados em pastas e um dos objetivos deste trabalho é a organização destes arquivos em uma grande e

unificada base de dados. Um exemplo de como estas pastas são organizadas é apresentado na figura 28.

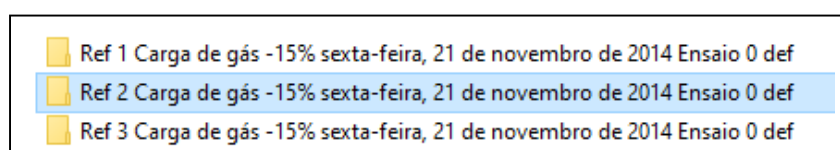
Figura 27 – Atributos importantes para o modelo.



Fonte: do autor.

Além de extrair os dados das pastas e organizá-los por variável conforme já apresentado, foi necessário separar os dados por tipo de ensaio e rotulá-los. Foi analisado se o refrigerador estava com falha ou não através da informação “def” no final do nome da pasta, quando possuía essa informação significava defeito no refrigerador como mostrado na figura 28.

Figura 28 – Identificação de ensaio com falha.



Fonte: do autor.

Foi utilizada a biblioteca “os”, para facilitar o uso do sistema operacional e ser possível o armazenamento do nome da pasta, para em sequência fazer a comparação se possuía ou não defeito e qual refrigerador estava sendo testado, como mostrado na figura 29.

Figura 29 – Identificação de ensaio com falha.

```
def ler_ref(path,dados):
    folder = os.path.basename(path).lower()
    dados["ref1"].append(1 if folder.startswith("ref 1") else 0)
    dados["ref2"].append(1 if folder.startswith("ref 2") else 0)
    dados["ref3"].append(1 if folder.startswith("ref 3") else 0)
    dados["Falha"].append(1 if folder.endswith("def") else 0)
```

Fonte: do autor.

Foi necessário a troca os valores de “true” e “false” para 1 e 0, visando atender as especificações dos modelos de algoritmos (logica “if” utilizada na figura 29).

7.2.2 Confeção da tabela Ensaios x Atributos.

Para a confecção dessa tabela foram criadas 4 funções que leem os dados e recebem os atributos já organizados, conforme a figura 30 abaixo.

Figura 30 – Exemplo de função criada para ler os arquivos.

```
dados["pico1_corrente"]=[]
dados["corrente_media"]=[]
dados["pico2_corrente"]=[]

def lerGrandezasEletricas_tabela_analise(path,dados):
    df_eletr = pd.read_csv(path+'/Grandezas Elétricas.xls', delimiter='\t'
    group = np.arange(len(df_eletr))/(len(df_eletr)/902)
    df_eletr = df_eletr.groupby(group).mean()
```

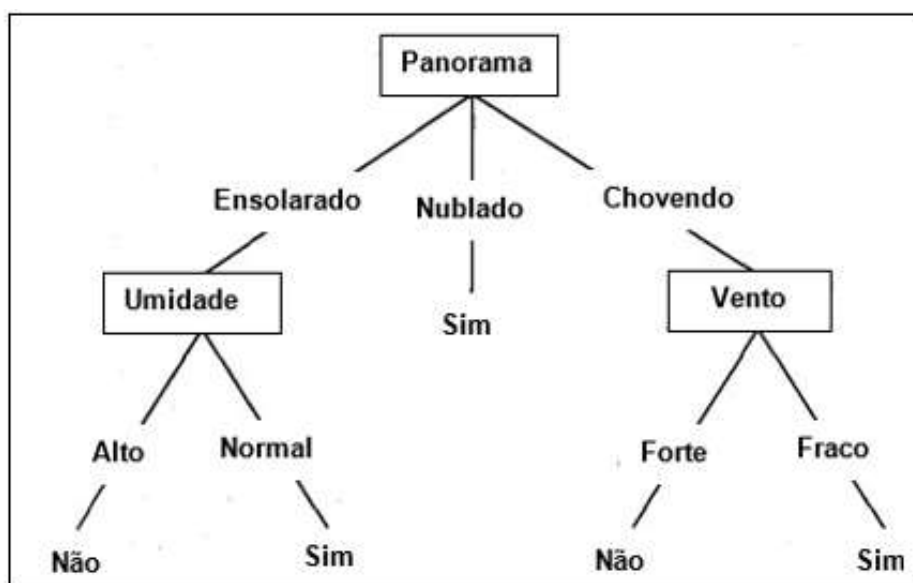
Fonte: do autor.

7.3 Algoritmos de *machine learning*

A escolha de um algoritmo confiável e eficiente é o ponto de partida para a construção do modelo desejado no projeto, por conta disso e tendo conhecimento que o projeto visa classificar se o refrigerador possui ou não falha, foi decidido testar três diferentes algoritmos de classificação da biblioteca *sckitlearn*.

DecisionTreeClassifier: uma árvore é uma coleção de elementos chamados nós, dentre os quais um é distinguido como nó raiz. Cada nó da árvore define um teste de algum atributo e cada ramo descendente daquele nó corresponde a um dos valores possíveis para este atributo, esse processo é repetido para as sub árvores enraizadas no novo nó (MITCHELL, 1997) como mostrado na figura 33.

Figura 33 – Exemplo de árvore de decisão para jogar *tennis*.



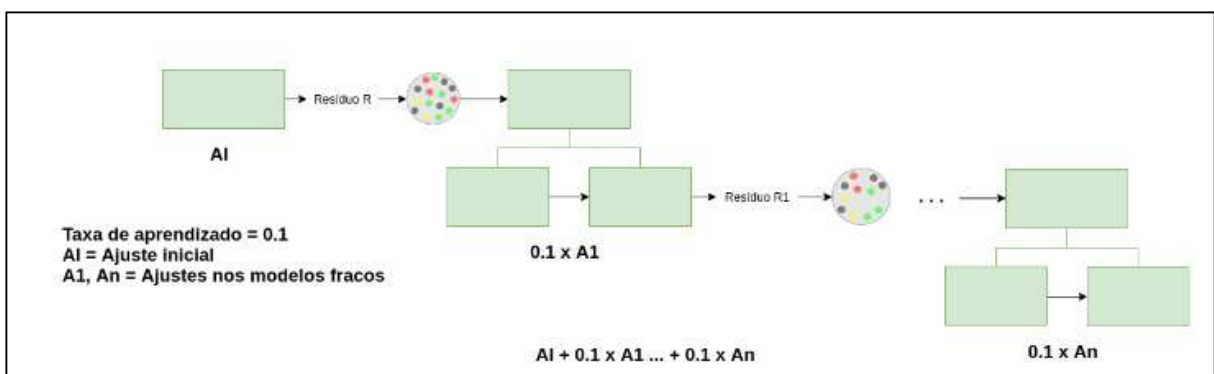
Fonte: adaptada de Mitchell, 1997.

RandomForestClassifier: uma floresta aleatória é um conjunto de árvores de classificação, no entanto, as árvores de decisão individuais normalmente exibem alta variância e tende a se ajustar demais. A aleatoriedade injetada nas florestas produz árvores de decisão com erros de previsão um tanto desacoplados. Ao fazer uma média dessas previsões, alguns erros podem ser cancelados. A floresta aleatória atinge uma variação reduzida combinando diversas árvores, às vezes ao custo de um

leve aumento no tempo. Na prática, a redução da variância é muito mais significativa, resultando em um modelo globalmente melhor (BREIMAN, 1998).

GradientBoostingClassifier: é um algoritmo criado por uma corrente de modelos fracos, onde o primeiro modelo é uma aproximação simples do resultado, sobrando um resíduo. A ideia é criar o próximo modelo em cima do resíduo, fazendo isso várias vezes buscando minimizar o resíduo gerado e o resultado ser o mais próximo possível do valor real (SILVA, 2020), como mostrado na figura 34.

Figura 34 – Como é feito a otimização por meio do *gradient boosting*.



Fonte: Silva, Jonhy (2020)

A figura 35 mostra a utilização dos modelos com os parâmetros padrões da biblioteca, mostrando somente a quantidade de árvores (*n_estimators*), a profundidade (*max_depth*) e um parâmetro para que os dados toda vez que forem treinados continuem os mesmos (*random_state*).

Figura 35 – Modelos com os parâmetros padrões da biblioteca.

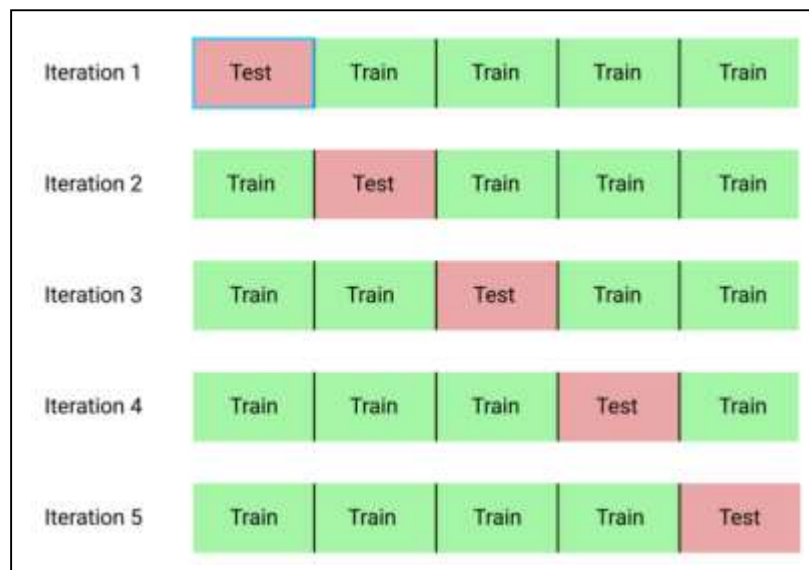
```
gb = GradientBoostingClassifier(n_estimators=100, learning_rate=0.9,
                               max_depth=20, random_state=0)
dt = DecisionTreeClassifier(max_depth=20, random_state=0)
rf = RandomForestClassifier(n_estimators=100, max_depth=20, random_state=0)
```

Fonte: do autor.

Com o objetivo de diminuir o risco de *overfitting* (sobreajuste ou superajuste do modelo), foi utilizada uma ferramenta de validação cruzada (*K-fold* ou *cross validate*) da biblioteca *sckitlearn*, onde tem como objetivo fragmentar os dados de treino em k parcelas, onde usa as k-1 primeiras para treino e a última para teste,

como mostrado na figura de exemplo 36. O *overfitting* é um termo usado em estatística para descrever quando um modelo estatístico se ajusta muito bem ao conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados, ocorre quando o modelo para de “aprender” e começa a “decorar”. Principal sintoma de um modelo que tem um excelente desempenho nos dados de treino, mas que vai muito mal na hora de analisar dados que nunca viu antes.

Figura 36 – Fluxo de funcionamento de um *cross_validation*.



Fonte: Shaikh, Raheel (2018).

7.4 Avaliação dos algoritmos através de métricas

De modo a ter um bom resultado foram analisadas as métricas existente, definido inicialmente duas métricas para analisar qual o melhor modelo a ser usado. As métricas utilizadas foram acurácia e revocação (*recall*).

Acurácia: a acurácia refere-se a uma medida do grau em que as previsões de um modelo correspondem à realidade que está sendo modelada (TING, 2011). Por exemplo, se temos 100 (cem) dados de refrigeradores e 90 (noventa) foram classificados corretamente quanto a possuir ou não falha, nosso modelo possui uma acurácia de 90 por cento. A acurácia é definida pela equação (1) abaixo.

Uma das maiores desvantagens desta métrica é que em alguns problemas a acurácia pode ser elevada e o modelo ainda ter desempenho inadequado. Por

exemplo, considerando que nosso projeto tivesse 100 (cem) dados, sendo 99 (noventa e nove) de refrigeradores com falhas e 1 (um) de refrigerador sem falha. Caso nosso modelo seja “ingênuo” e sempre classificar todos como defeituosos, ele obterá uma acurácia de 99%. O que parece uma excelente marca, mas, na verdade, não estamos avaliando nosso modelo adequadamente.

$$\text{Acurácia} = \frac{(VP + VN)}{(VP + FN + VN + FP)} \quad (1)$$

Legenda:

VP – Verdadeiro positivos (Aquele que falhou e foi classificado como falha).

VN – Verdadeiro negativos (Aquele que não falhou e foi classificado como não falhou).

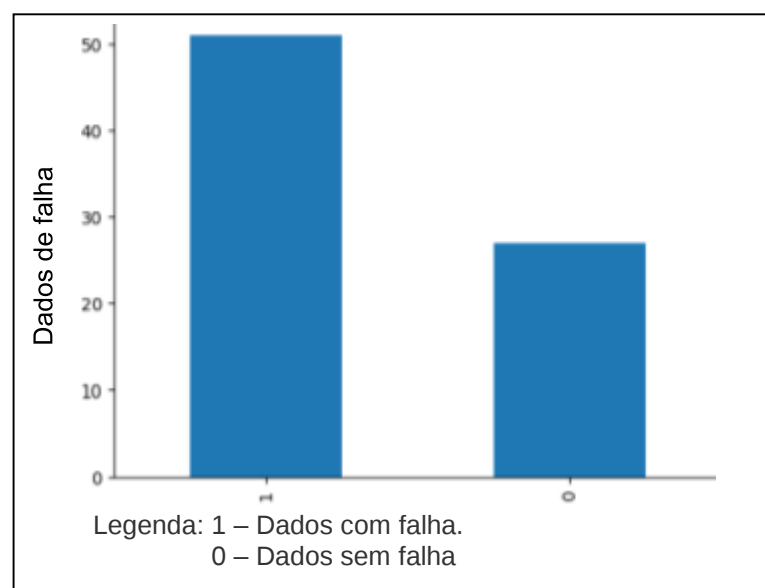
FP – Falso positivos (Aquele que não falhou e foi classificado como falha).

FN – Falso negativo (Aquele que falhou e foi classificado como não falhou).

Fonte: Scudilio, 2020.

Neste projeto, os dados utilizados estão representados no gráfico abaixo. Podemos observar a quantidade de dados com e sem falhas. Foi avaliado que poderia utilizar a métrica acurácia, devido uma boa distribuição dos dados entre com e sem falhas como mostrado na figura 37.

Figura 37 – Representação gráfica dos dados com e sem falha.



Fonte: do autor.

Revocação (*recall*): Também conhecida como taxa de verdadeiro positivo, da maior ênfase para os erros por falso negativo (TING, 2011). A métrica busca responder a seguinte pergunta: de todos os exemplos que são positivos, quantos foram classificados corretamente como positivos? Por exemplo, se o valor pra revocação fosse de 95%, isso indicaria que a cada 100 refrigeradores que estão com falha, é esperado que apenas 95 sejam corretamente identificados como com falha. A métrica é definida pela equação (2) abaixo.

$$\textbf{Revocação} = \frac{VP}{VP + FN} \quad (2)$$

Legenda:

VP – Dados verdadeiro positivo

FN – Dados falso negativo

Fonte: Scudilio, 2020.

A utilização da métrica “*recall*” se deu pelo fato de ser um problema maior uma falha não notificada do que uma notificação de falha fictícia.

A precisão (equação 3) é o grau de variação gerado por diferentes medições. Dessa forma, quanto mais preciso for um processo, menor será a variação entre os valores obtidos. (ClearSale, 2022)

$$\textbf{Precisão} = \frac{VP}{VP + FP} \quad (3)$$

Legenda:

VP – Dados verdadeiro positivo

FP – Dados falso positivo

Fonte: Scudilio, 2020.

Afim de testar os modelos escolhidos, foram retiradas algumas colunas que não possuíam valores significativos para o treinamento (figura 38) e utilizado o *Cross validation* separando em 20 fragmentos.

Figura 38 – Preparando para treinamento dos algoritmos.

```
X = df.drop(columns=["Falha","Unnamed: 0","ref1","ref2","ref3", "random" ]);
y = df["Falha"]
```

Fonte: do autor.

Os resultados dos testes nos diferentes modelos podem ser observados pelas figuras 39, 40, 41.

Figura 39 – Resultados utilizando a ferramenta cross_validacion e o algoritmo RandomForestClassifier.

```
resultado_rf = cross_validate(rf, X, y, cv=20, scoring=["accuracy","recall"])

print("time",np.mean(resultado_rf["fit_time"]))
print("mean acc",np.mean(resultado_rf["test_accuracy"]))
print("std acc",np.std(resultado_rf["test_accuracy"]))
print("mean recall",np.mean(resultado_rf["test_recall"]))
print("std recall",np.std(resultado_rf["test_recall"]))

time 0.134180748462677
mean acc 0.9375
std acc 0.1340475661845451
mean recall 0.9333333333333332
std recall 0.16996731711975951
```

Fonte: do autor.

Figura 40 – Resultados utilizando a ferramenta cross_validacion e o algoritmo DecisionTreeClassifier.

```
resultado_dt = cross_validate(dt, X, y, cv=20, scoring=["accuracy","recall"])

print("time",np.mean(resultado_dt["fit_time"]))
print("mean acc",np.mean(resultado_dt["test_accuracy"]))
print("std acc",np.std(resultado_dt["test_accuracy"]))
print("mean recall",np.mean(resultado_dt["test_recall"]))
print("std recall",np.std(resultado_dt["test_recall"]))

time 0.0025505781173706054
mean acc 0.9
std acc 0.16583123951777
mean recall 0.9083333333333332
std recall 0.18615256586407228
```

Fonte: do autor.

Figura 41 – Resultados utilizando a ferramenta cross_validacion e o algoritmo GradientBoostingClassifier.

```
resultado_gb = cross_validate(gb, X, y, cv=20, scoring=["accuracy","recall"])
```

```
print("time",np.mean(resultado_gb["fit_time"]))  
print("mean acc",np.mean(resultado_gb["test_accuracy"]))  
print("std acc",np.std(resultado_gb["test_accuracy"]))  
print("mean recall",np.mean(resultado_gb["test_recall"]))  
print("std recall",np.std(resultado_gb["test_recall"]))
```

```
time 0.0372584342956543  
mean acc 0.9125  
std acc 0.16345871038277526  
mean recall 0.9083333333333332  
std recall 0.18615256586407228
```

Fonte: do autor.

8 RESULTADOS ALCANÇADOS

A tabela 5 mostra os resultados alcançados no capítulo anterior, utilizando os modelos de algoritmos *DecisionTreeClassifier*, *RandomForestClassifier*, *GradientBoostingClassifier*. Para cada modelo analisado temos os valores de tempo (*time*), acurácia (*accuracy*) e revocação (*recall*). Em destaque estão os melhores valores obtidos para cada métrica avaliada. A métrica “time” mostra o tempo necessário para execução de cada modelo de algoritmo, podendo ser descartada do nosso projeto. Os tempos encontrados foram baixos em comparação com a relação: dados por tempo (os modelos teriam sua execução mais rápido em comparação com a retirada dos dados do sistema). O algoritmo de *machine learning* *RandomForestClassifier* (“rf”) (ClearSale, 2022) foi para ser utilizado neste projeto por possuir maior acurácia (“*accuracy*”) e maior taxa de verdadeiro positivo (“*recall*”), como mostrado na tabela 5.

Tabela 5 – Testes de modelos de *machine learning*.

	<i>RadomForestClassifier</i>	<i>DecisionTreeClassifier</i>	<i>GradientBoostingClassifier</i>
<i>Time</i>	0.1341807 segundos	0.0025505 segundos	0.03725843 segundos
<i>Accuracy</i>	93.75%	90.00%	91.25%
<i>Recall</i>	93.33%	90.83%	90.83%

Fonte: do autor.

Após a determinação do algoritmo, foi separado os dados em *X_test*, *X_train*, *y_test*, *y_train*. Definindo 80 por cento para treino e 20 por cento para teste (figura 42), para ser possível a análise dos resultados do modelo com diferentes hiperparâmetros, utilizamos a métrica matriz de confusão para determina o melhor o modelo.

Figura 42 – Definição do percentual de treino e de teste.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
```

Fonte: do autor.

Para avaliar os resultados foram utilizadas a matriz de confusão (figura 43) que é uma das métricas mais usadas nos modelos atuais de classificação, sendo um dos termos mais utilizados na área de *machine learning* atualmente e tendo sua origem atribuída ao termo tabela de contingência (PEARSON, K., 1897).

Isso se deve ao fato da facilidade em identificarmos o comportamento do modelo em diferentes frentes da predição, pois, a matriz demonstra quatro casos possíveis de ocorrer em uma predição binária, sendo eles:

- **Verdadeiros negativos:** aquele que não falhou e foi classificado como não falhou.
- **Verdadeiros positivos:** aquele que falhou e foi classificado como falha.
- **Falsos negativos:** aquele que falhou e foi classificado como não falhou.
- **Falsos positivos:** aquele que não falhou e foi classificado como falha.

Figura 43 – Matriz de confusão.

Verdadeiro Negativo (VN)	Falso Positivo (FP)
Falso Negativo (FN)	Verdadeiro Positivo (VP)

Fonte: do autor.

Foi utilizada a métrica precisão para ser analisada após as outras métricas, garantindo a melhor escolha do modelo para o projeto. Precisão é uma métrica que avalia a quantidade de verdadeiros positivos sobre a soma de todos valores positivos (SCUDILIO, 2020).

Para melhorar o modelo utilizado foi decidido ajustar o principal parâmetro: *n_estimators*. É o número de árvores na floresta, quanto maior, melhor, mas também levará mais tempo para calcular. (GEURTS et al, 2006). Visto isso foi testado o modelo para os seguintes valores de *n_estimators* = [3; 4; 10; 100] variando o *threshold* nos valores de [0,20; 0,40; 0,60; 0,70]. O *threshold* é o valor limiar de decisão na curva ROC, ou seja, valores abaixo desse número serão classificados como uma classe e valores acima deste valor serão classificados como outra classe. O valor comum do *threshold* é o de 0,5, associado ao valor de 50% de um intervalo probabilístico de 0 à 100%, porém, ao contrário do senso empírico, o *threshold* é um

valor dependente do problema (GOOGLE DEVELOPERS, 2020). Dependendo do tipo de modelo criado para um conjunto de dados, podemos ter resultados melhores se utilizarmos um valor de 0,3 ao invés do comum 0,5. A tabela 6 mostra os resultados do modelo sem alteração de hiperparâmetro, já as tabelas 7, 8, 9 e 10 mostram os resultados com as alterações feitas.

Tabela 6 – Resultados do modelo sem alteração de hiperparâmetro.

Abordagem	Accuracy	Matriz de confusão		Precisão	Recall
Parâmetros iniciais	94%	8	1	88%	100%
		0	7		

Fonte: do autor.

Tabela 7– Comparação de *Threshold* para $n_estimators = 3$.

Abordagem	Accuracy	Matriz de confusão		Precisão	Recall
<i>Threshold</i> de 0.80	69%	5	0	100%	55%
		5	6		
<i>Threshold</i> de 0.60	94%	5	0	100%	91%
		1	10		
<i>Threshold</i> de 0.40	94%	5	0	100%	91%
		1	10		
<i>Threshold</i> de 0.20	69%	0	5	69%	100%
		0	11		

Fonte: do autor.

Tabela 8 – Comparação de *Threshold* para $n_estimators = 4$.

Abordagem	Accuracy	Matriz de confusão		Precisão	Recall
<i>Threshold</i> de 0.80	69%	5	0	100%	55%
		5	6		
<i>Threshold</i> de 0.60	88%	5	0	100%	82%
		2	9		
<i>Threshold</i> de 0.40	88%	3	2	85%	100%
		0	11		
<i>Threshold</i> de 0.20	69%	0	5	69%	100%
		0	11		

Fonte: do autor.

Tabela 9 – Comparação de *Threshold* para *n_estimators* =10.

Abordagem	<i>Accuracy</i>	Matriz de confusão		Precisão	<i>Recall</i>
<i>Threshold</i> de 0.80	69%	5	0	100%	55%
		5	6		
<i>Threshold</i> de 0.60	94%	5	0	100%	91%
		1	10		
<i>Threshold</i> de 0.40	81%	3	2	83%	91%
		1	10		
<i>Threshold</i> de 0.20	75%	1	4	73%	100%
		0	11		

Fonte: do autor.

Tabela 10 – Comparação de *Threshold* para *n_estimators* =100.

Abordagem	<i>Accuracy</i>	Matriz de confusão		Precisão	<i>Recall</i>
<i>Threshold</i> de 0.80	69%	5	0	100%	55%
		5	6		
<i>Threshold</i> de 0.60	81%	5	0	100%	73%
		3	8		
<i>Threshold</i> de 0.40	94%	6	1	90%	100%
		0	9		
<i>Threshold</i> de 0.20	81%	2	3	79%	100%
		0	11		

Fonte: do autor.

Aqui vamos reforçar a diferença entre *acurácia* e *precisão*. Embora possam ter significados parecidos e até mesmo serem utilizados muitas vezes como sinônimos, é muito importante deixar claro que são conceitos diferentes. Para evitar equívocos, é preciso ter clareza sobre as especificidades de cada um. A *precisão* é o grau de variação gerado por diferentes medições. Dessa forma, quanto mais preciso for um processo, menor será a variação entre os valores obtidos. Já a *acurácia* é uma espécie de soma entre exatidão e *precisão*. Ou seja, nela, os resultados obtidos por uma experiência não apenas devem ser precisos, mas também precisam estar perto do valor de referência ou real usado como base (ClearSale, 2022).

O modelo que apresentou melhor resultado foi o encontrado na tabela 10, onde os valores dos hiperparâmetros “*Threshold*” e “*n_estimators*”, foram respectivamente 0.40 e 100. O resultado das métricas do modelo escolhido

apresentou uma acurácia (taxa de acerto geral do modelo) de 0,94 (94%), com uma taxa de verdadeiro positivo (“*recall*”) de 1 (100%) e uma precisão 0,90 (90%).

Ao verificarmos alguns valores referente a métricas nas tabelas 7, 8, 9, e 10, detectamos mais de um modelo com todas as métricas maiores que 90 por cento, porem só uma que possui taxa de recall = 100 por cento, sendo uma métrica importante como comentado no capítulo 7.

Diante deste cenário, verifica-se que o objetivo geral da presente pesquisa foi alcançado. O estudo de aprendizado de máquina, além de proporcionar o conhecimento necessário para o desenvolvimento dos modelos que demonstraram os resultados citados acima, gerou um banco de dados de falhas de refrigeradores, para possíveis testes futuros.

9 CONCLUSÃO

Durante o trabalho, apresentaram-se os passos realizados para o desenvolvimento de um banco de dados e um modelo de predição para falhas de refrigeradores. Utilizando de metodologia do processo unificado para o gerenciamento do trabalho, conseguindo um detalhamento específico de todas as técnicas e tecnologias utilizadas durante o processo. Em relação ao banco de dados, conclui-se que demanda muito tempo para retirada desses valores, como consequência obteve-se uma menor quantidade de dados. Porém já que a produção do trabalho foi toda com soluções de código aberto, é possível no futuro ser utilizado o mesmo banco de dados para incrementação de mais dados. Contudo, é importante citar, que além de ser demorado a obtenção desses dados é difícil medir todos os pontos definidos nesse trabalho, uma solução para futuras medições é além de utilizar um banco de dados na nuvem, medir somente os pontos mostrados na figura 27 como atributos importantes.

Em relação ao modelo desenvolvido, conseguiu-se uma taxa de acerto geral do modelo de 94% para predição de falhas em refrigeradores. Um fator importante foi a escolha entre os três diferentes modelos de classificação e a utilização da ferramenta *cross_validation* para obter, mesmo com poucos dados, um modelo eficaz. O modelo escolhido foi o *RandomForestClassifier*, por possuir melhores métricas como mostrado no capítulo 8. Foram utilizadas várias outras maneiras para tentar garantir o funcionamento do algoritmo, como a visualização da quantidade de refrigeradores com e sem falha, descartando a ocorrência de um possível *overfitting*, o ponto de utilização das matrizes de confusão para detectar o melhor *threshold*, torna visível a direção do modelo. Definiu-se uma ordem de avaliação das métricas, levando em conta o tipo de projeto desenvolvido. A métrica “*Recall*” foi definida como a mais importante, pois tem um maior peso identificar quando um refrigerador está em falha mesmo que isso não ocorra. A métrica “*Accuracy*” foi definida como a segunda mais importante, porque mostra uma indicação geral de como o modelo performou, quanto maior a acurácia maior o acerto geral do modelo. E por último a precisão, pois tendo um “*recall*” e uma acurácia elevados, o valor da precisão sendo o maior encontrado garante uma menor chance de se ter falsos positivos.

O melhor resultado foi obtido com o modelo *RandomForestClassifier*, que apresentou acurácia de 94%, revocação de 100% e uma precisão de 90%.

Conclui-se que o modelo satisfaz o projeto e poderá ser de grande uso na indústria de refrigeração, com a utilização de sensores de temperatura na entrada do termostato e um transdutor de corrente em série com a alimentação do refrigerador, gerando as principais medições para obtenção dos *feature_importances* como visto na figura 27. Com os principais atributos e dados medidos em tempo real, é possível utilizar um modelo parecido com o do projeto para prever a falha do refrigerador.

REFERÊNCIAS

ACCURACY_SCORE: **Accuracy classification score**. [S. l.], 2022. Disponível em:< https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html?highlight=accuracy_score> Acesso em: 18 de jun. de 2022.

BALLMANN, Edinei. **Aprimoramento de Processo de Medição de Deslocamento de Válvulas em Compressores de Refrigeração** - Dissertação (Mestrado) - Curso de Engenharia Mecânica, Centro Tecnológico. [S.l: s.n.], 2010.

BREUKER, M.; ROSSI, T.; BRAUN, J. Smart maintenance for rooftop units. **ASHRAE Journal** v. 42, n. November, p. 41–42+44 , 2000.

BREIMAN, “Arcing Classifiers”, *Annals of Statics* 1998.

CONFUSION_MATRIX: **Compute confusion matrix to evaluate the accuracy of a classification**. [S. l.], 2022. Disponível em:< https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html?highlight=matrix%20confusion#sklearn.metrics.confusion_matrix> Acesso em: 16 de jun. de 2022.

ClearSale: **O que é acurácia: o conceito, a importância e como aplicar**, 2022. Disponível em:< <https://blogbr.clear.sale/conheca-e-saiba-como-aplicar-a-acuracia>>. Acesso em: 02 de agosto de 2022

CHICCO, D., & JURMAN, G. (2020). **The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation**. *BMC genomics*, 21(1), 1-13.

CROSS-VALIDATION: **Cross-validation: evaluating estimator performance**. [S. l.], 2022. Disponível em: < https://scikit-learn.org/stable/modules/cross_validation.html >. Acesso em: 10 de jun. de 2022.

DA COSTA, Ennio Cruz. **Refrigeração**. Editora Blucher, 1982. 9788521217527. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788521217527/>. Acesso em: 02 jul. 2022.

DECISIONTREES: **Decision Trees (DTs)**. [S. l.], 2022. Disponível em:< <https://scikit-learn.org/stable/modules/tree.html#decision-trees>> Acesso em: 14 de jun. de 2022.

DOSSAT, Roy J. **Princípios de refrigeração**. 1. ed. [S.l.]: Hemus, 1963. 884 p.

EMBRACO. **Informativo técnico 94010**. Joinville: [s.n.], 2009

FERRAZ, F.; **Apostila de refrigeração**. Santo Amaro, 2008. Centro federal de educação tecnológica da Bahia.

GARCIA, F. E. M.; BANDARRA FILHO, E. P.; MENDOZA, O. S. H. **Avaliação do desempenho de um sistema refrigeração por compressão de vapor a velocidade variável**. IN: 17º Simpósio do programa de pós-graduação em engenharia mecânica, 2007, Uberlândia, p.1 - 6.

GEURTS P., ERNST D., and WEHENKEL L., “**Extremely Randomized Trees**”, **Machine Learning**, 63(1), 3-42, 2006.

ÇENGEL, Yunus A.; BOLES, Michael A. **Termodinâmica**. [S.l.]: McGraw Hill, 2005. 675p.

GÉRON, A. **Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. O'Reilly Media, Incorporated, 2017. ISBN 9781491962282.

GOOGLE DEVELOPERS. **Classification: Thresholding**. [S.l.]. 2020. Disponível em: <<https://developers.google.com/machine-learning/crash-course/classification/thresholding>> Acesso em: 7 de jul. 2022.

GOMES, Gregory Chagas. **Avaliação do comportamento de refrigeradores domésticos frente a defeitos provocados e emulados**. Florianópolis, 2015. 161 p. Tese (Mestre em Engenharia Mecânica) Pós-Graduação em Engenharia Mecânica, Universidade Federal de Santa Catarina.

GRADIENTBOOSTING: **Gradient Boosting for classification**. [S. l.], 2022. Disponível em: <<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html>> Acesso em: 14 de jun. de 2022.

HAYES, Bob. Programming Languages Most Used and Recommended by Data Scientists. **Business Broadway**, 2019. Disponível em: <<https://businessoverbroadway.com/2019/01/13/programming-languages-most-used-and-recommended-by-data-scientists/>> Acesso em: 7 de jul. 2022.

LI, Haorong. Evaluation of a decoupling-based fault detection and diagnostic technique -- part I: field emulation evaluation.pdf. **ICEBO** v. 5-4, n. Control Systems for Energy Afficiency and Comfort, p. 9 , 2006.

MICHELL, Tom. Aprendizado de máquina: árvore de decisão indutiva. **Machine Learning**. 1997.

MORRIESEN, A.; DESCHAMPS, C. J. Experimental investigation of transient fluid flow and superheating in the suction chamber of a refrigeration reciprocating compressor. **Applied Thermal Engineering** v. 41, p. 61–70, 2011. Disponível em: <<http://dx.doi.org/10.1016/j.applthermaleng.2022.07.003>> Acesso em: 7 de jul. 2022.

PACHECO. Luis Antônio Schalata. **Desenvolvimento de um sistema para medir a resistência de enrolamento em motores de compressores energizados**. [S.l.: s.n.], 2007.

PEARSON, K., 1897. Mathematical contributions to the theory of evolution.—on a form of spurious correlation which may arise when indices are used in the 73 measurement of organs. **Proceedings of the royal society of london**, 60(359-367), pp.489-498.

PRESSMAN R. **Engenharia de Software**, 6a edição, AMGH Editora. 2010.
RECALL_SCORE: **Compute the recall**. [S. I.], 2022. Disponível em:< https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html?highlight=recall#sklearn.metrics.recall_score> Acesso em: 18 de jun. de 2022.

RISH, I., 2001, August. **An empirical study of the naive Bayes classifier**. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, No. 22, pp. 41-46).

ROC_AUC_SCORE: **Compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores**. [S. I.], 2022. Disponível em: <sklearn.metrics.roc_auc_score — [scikit-learn 1.1.1 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html)>. Acesso em: 10 de jun. de 2022.

ROBINSON, David. Why is Python Growing So Quickly? **Stack Overflow Blog**, 2017. Disponível em: <<https://stackoverflow.blog/2017/09/14/python-growing-quickly/>>. Acesso em: 5 de jul. de 2022.

RANDOMFORESTS: **Ensemble methods**. [S. I.], 2022. Disponível em:< <https://scikit-learn.org/stable/modules/ensemble.html#id6>> Acesso em: 11 de jun. de 2022.

ROSSUM, Guido. Matplotlib: “**Visualization with Python**”. [S. I.], 2022. Disponível em:< <https://matplotlib.org/>>Acesso em: 18 de jun. de 2022.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM Journal**, p.211-229, jul. 1959.

SILVA, R.; DESCHAMPS, Cesar J. Modeling of gas leakage through compressor valves. **International Journal of Refrigeration** , 2012.

SILVA, Jonhy. **Uma breve introdução ao algoritmo de Machine Learning Gradient Boosting utilizando a biblioteca Scikit-Learn**. Medium, 2020. Disponível em:< <https://medium.com/equal-lab/uma-breve-introdu%C3%A7%C3%A3o-ao-algoritmo-de-machine-learning-gradient-boosting-utilizando-a-biblioteca-311285783099>> Acesso em: 12 de jul. de 2022.

SHAIKH, Raheel. Cross Validation Explained: Evaluating estimator performance. Towards Data Science, 2018. Disponível em: Acesso em: 12 de jul. de 2022.

SOMMERVILLE, I. **Engenharia de Software**. 8a Edição. Addison Wesley. 2007.

SCUDILIO, Juliana. Como Calcular as Métricas de Validação dos Modelos de *Machine Learning em Python*. **Flai**, 2020. Disponível em:

<https://www.flai.com.br/juscudilio/como-calculas-as-metricas-de-validacao-dos-modelos-de-machine-learning-em-python>. Acesso em: 7 de jul. 2022.

STRATIFIEDKFOLD: **Stratified K-Folds cross-validator**. [S. I.], 2022. Disponível em:< [sklearn.model_selection.StratifiedKFold — scikit-learn 1.1.1 documentation](#)> . Acesso em: 10 de jun. de 2022

STOUPPE, D. E.; LAU, T. Air conditioning and refrigeration equipment failures. **National Engineer** p. 9 , 1989.

TING, Kai Ming. 2011. **Encyclopedia of machine learning**. Springer. ISBN 978–0–387