UNIVERSIDADE SANTA CECÍLIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DOUTORADO EM ENGENHARIA MECÂNICA

JOSEFFE BARROSO DE OLIVEIRA

UM MODELO FUNDAMENTADO EM ALGORITMOS DE APRENDIZAGEM DE MÁQUINA DEDICADO À PREDIÇÃO DE OCORRÊNCIAS EM LINHA DE MONTAGEM INDUSTRIAL

SANTOS/SP
2023

JOSEFFE BARROSO DE OLIVEIRA

UM MODELO FUNDAMENTADO EM ALGORITMOS DE APRENDIZAGEM DE MÁQUINA DEDICADO À PREDIÇÃO DE OCORRÊNCIAS EM LINHA DE MONTAGEM INDUSTRIAL

Tese apresentada à Universidade Santa Cecília como parte dos requisitos para obtenção de título de Doutor no Programa de Pós-Graduação em Engenharia Mecânica, sob orientação do Prof. Dr. Carlos José de Lima e co-orientador Prof. Dr. João Inácio da Silva Filho.

SANTOS/SP

2023

Autorizo a reprodução parcial ou total deste trabalho, por qualquer que seja o processo, exclusivamente para fins acadêmicos e científicos.

Elaborada pelo SiBi - Sistema Integrado de Bibliotecas - Unisanta

Dedico esse trabalho a minha família, em especial, minha esposa, meu filho, meu pai, minha mãe e meus irmãos.

AGRADECIMENTOS

A Deus, nosso pai todo protetor! Por me abençoar e me dar forças para conseguir concluir mais uma etapa em minha vida.

Aos amores da minha vida, minha esposa Myrian e ao meu filhão Lucas, que estão sempre ao meu lado diariamente e por me impulsionar nessa evolução.

Aos amores da minha vida, meus pais, Jarina e Ivanildo, pela excelente educação que me foi dada e por sempre acreditarem em mim, me apoiando, ajudando e principalmente, por estarem sempre ao meu lado.

Aos meus irmãos, Jair, pelos excelentes conselhos dados, e ao João Marcelo (Joãozinho/Marcelinho) pelo carinho e companheirismo de sempre.

Aos meus tios, Jairo e Jarbas, por sempre estarem prontos para ajudar quando precisei.

Ao Professor Dr. Carlos Lima, pela valiosa orientação e principalmente pelo apoio durante a construção desse trabalho.

Ao Professor Dr. João Inácio, pelos preciosos conselhos e observações.

Aos colegas de trabalho Dr. Vinícius Albert Ferraris e Douglas Ayres, pela amizade, orientação e apoio em todos os momentos em que precisei.

Aos professores e hoje amigos pessoais de Doutorado, Davi Silvestre e Claudio Magalhães, pelo companheirismo e união durante todo o Doutorado.

A todo corpo docente do programa de Doutorado, que certamente contribuíram para minha formação e a toda equipe da secretaria, especialmente a Sandra e Imaculada, pela simpatia, profissionalismo e atenção durante todo o curso.

"Se você quer ser bem-sucedido, precisa ter dedicação total, buscar seu último limite e dar o melhor de si mesmo."

(Ayrton Senna da Silva)

RESUMO

Este trabalho consiste na aplicação de inteligência artificial através de algoritmos de aprendizagem de máquina para redução de ordens de serviço de uma indústria multinacional de bens de consumo. Para isso, foi construída uma configuração especial estruturada em três algoritmos compondo um modelo dedicado à predição de ocorrências em linha de montagem industrial. Na arquitetura computacional construída o processo de separação e organização dos dados é realizada através da linguagem SQL e uma base de dados SQL Server. Os processos mais ligados à Inteligência Artificial são efetuados através da aplicação de algoritmos de aprendizagem de máquina na forma de *Random Forest*, *Light GBM* e *Linear Regression* programados através da linguagem de programação Python com a utilização de técnicas como *Grid Search* e *Cross Validação*. A arquitetura construída foi validada em uma empresa de bens de consumo onde foram utilizados os dados provenientes das falhas da linha de montagem de refrigeradores e ordens de serviço geradas para refrigeradores pós-venda em um intervalo de 6 meses após serem fabricados. Os dados das falhas de montagem de refrigeradores foram obtidos através de arquivos históricos da linha de montagem industrial, que consecutivamente ficaram gravados em bases de dados específicas. Após o acesso aos dados com estas informações, estes são migrados para uma nova base de dados e formatados para a aplicação da IA na estrutura algorítmica construída. Da mesma forma os dados de ordens de serviço que foram gerados, gravados em bases de dados específicas, foram também formatados e migrados para uma nova base de dados e disponibilizados para serem analisados e comparados com padrões aprendidos na estrutura algorítmica. O objetivo principal da combinação dessas tecnologias é destacar as principais falhas que contribuem para geração de ordens de serviço e construir a predição da quantidade de OS que poderão ser geradas, baseado nas ocorrências que aconteceram na linha de montagem industrial de refrigeradores. Mesmo considerando que a estrutura algorítmica ainda está na sua primeira versão, os testes efetuados mostraram resultados de boa eficácia contribuindo para diminuição dos custos na fabricação de refrigeradores.

Palavras-Chave: Falhas em Linha Montagem. Ordem de Serviço. Refrigeradores. SQL. Python. Aprendizagem de Máquina. Inteligência Artificial.

ABSTRACT

This work consists of applying artificial intelligence through machine learning algorithms to reduce work orders in a multinational consumer goods industry. To this end, a special configuration structured around three algorithms was built, composing a model dedicated to predicting occurrences on an industrial assembly line. In the constructed computational architecture, the process of separating and organizing data is carried out using the SQL language and a SQL Server database. The processes most linked to Artificial Intelligence are carried out through the application of machine learning algorithms in the form of Random Forest, Light GBM and Linear Regression programmed through the Python programming language using techniques such as Grid Search and Cross Validation. The constructed architecture was validated in a consumer goods company where data from refrigerator assembly line failures and service orders generated for after-sales refrigerators were used within a 6-month interval after being manufactured. Data on refrigerator assembly failures were obtained through historical files from the industrial assembly line, which were consecutively recorded in specific databases. After accessing the data with this information, it is migrated to a new database and formatted for the application of AI in the constructed algorithmic structure. Likewise, the work order data that was generated, recorded in specific databases, was also formatted and migrated to a new database and made available to be analyzed and compared with patterns learned in the algorithmic structure. The main objective of transferring these technologies is to highlight the main failures that are highlighted for the generation of work orders and to build a prediction of the amount of OS that could be generated, based on the occurrences that occurred on the industrial refrigerator assembly line. Even considering that the algorithmic structure is still in its current version, the tests demonstrated good effectiveness results, contributing to the reduction of costs in the manufacture of refrigerators.

Keywords: Assembly Line Failures. Order of Service. Refrigerators. SQL. Python. Machine Learning. Artificial intelligence.

LISTA DE ILUSTRAÇÕES

Figura 1 – Processo de organização de dados..... 18
Figura 2 – Banco de Dados Microsoft SQL Server.....19
Figura 3 – Linguagem de programação SQL e suas aplicações.....20
Figura 4 – Campos de atuação da inteligência artificial..... 22
Figura 5 – Aprendizado de máquina e seus modelos.....23
Figura 6 – Funcionamento do modelo supervisionado.....24
Figura 7 – Classificação no modelo supervisionado.....24
Figura 8 – Regressão no modelo supervisionado..... 25
Figura 9 – Aplicação do modelo supervisionado..... 25
Figura 10 – Linguagem de programação Python e suas aplicações.....26
Figura 11 – Biblioteca Scikit-Learn e suas aplicações.....28
Figura 12 – Funcionamento do Linear Regression a partir de dados submetidos para análise. 29
Figura 13 – Funcionamento do Random Forest a partir de dados submetidos para análise. 30
Figura 14 – Funcionamento do Light GBM a partir de dados submetidos para análise.....31
Figura 15 – Características da biblioteca Pandas..... 32
Figura 16 – Funcionamento da Cross Validação..... 33
Figura 17 – Aplicação de Hiper parâmetros e seus resultados..... 34
Figura 18 – Interface de desenvolvimento do Google Colab.....35
Figura 19 – Arquitetura de 6 linhas de montagem industrial..... 36
Figura 20 – Processo resumido aplicado na linha industrial de montagem.....37
Figura 21 – Fluxo do processo desenvolvido para esse trabalho.....38
Figura 22 – Origem da extração dos dados.....39
Figura 23 – Etapas desenvolvidas na primeira parte.....39
Figura 24 – Estrutura dos dados antes da organização..... 40
Figura 25 – Estrutura dos dados depois do início da organização.....41
Figura 26 – Estrutura dos dados da DataCorrigida e Modelo.....42
Figura 27 – Estrutura dos motivos das falhas.....42
Figura 28 – Estrutura dos dados depois da organização dos motivos..... 43
Figura 29 – Modelagem da tabela TB_FALHA_MONTAGEM..... 46
Figura 30 – Estrutura da tabela TB_FALHA_MONTAGEM normalizada.....58
Figura 31 – Etapas desenvolvidas na segunda parte.....59
Figura 32 – Estrutura dos dados de ordens de serviço antes da organização.....59
Figura 33 – Estrutura dos dados de ordens de serviço depois do início da organização.....60
Figura 34 – Estrutura dos dados de ordens de serviço durante a organização..... 63
Figura 35 – Estrutura dos dados de ordens de serviço depois da organização completa para os produtos MODELO 01.....64
Figura 36 – Estrutura dos dados de ordens de serviço depois da organização completa para os produtos MODELO 01.....64
Figura 37 – Estrutura dos dados de ordens de serviço depois da organização completa para os produtos MODELO 02.....65
Figura 38 – Estrutura dos dados de ordens de serviço depois da organização completa para os produtos MODELO 02.....65

Figura 39 – Etapas desenvolvidas na terceira parte.....66
Figura 40 – Conjunto de dados para o desenvolvimento do modelo de dados.....66
Figura 41 – Conjunto de dados para o desenvolvimento do modelo de dados.....67
Figura 42 – Arquivo CSV com conteúdo organizado em texto.....67
Figura 43 – Dataset da variável df_sku no intervalo de fevereiro de 2021 até junho de 2023... 69
Figura 44 – Dataset do intervalo de fevereiro de 2021 até junho de 2023.....69
Figura 44 – Dataset do intervalo de fevereiro de 2023 até dezembro de 2023 para o produto MODELO 01..... 73
Figura 45 – Análise das falhas de forma independente para o produto MODELO 01.....74
Figura 46 – Análise das falhas de forma independente para o produto MODELO 01.....74
Figura 47 – Quantidade de OS criadas e quantidade de falhas geradas para o produto MODELO 01..... 75
Figura 48 – As dez falhas mais significativas para a análise do produto MODELO 01.....80
Figura 49 – Dataset do intervalo de fevereiro de 2023 até dezembro de 2023 para o produto MODELO 02..... 81
Figura 50 – Análise das falhas de forma independente para o produto MODELO 02.....82
Figura 51 – Análise das falhas de forma independente para o produto MODELO 02.....82
Figura 52 – Quantidade de OS criadas e quantidade de falhas geradas para o produto MODELO 02..... 83
Figura 53 – As dez falhas mais significativas para a análise do produto MODELO 02.....88

LISTA DE GRÁFICOS

Gráfico 1 – Relação entre quantidade de OS e data de cada falha para o produto MODELO 01 76
Gráfico 2 – Etapa de treinamento no período de fevereiro de 2021 e setembro de 2022 para o produto MODELO 01 78
Gráfico 3 – Etapa de teste entre o período de outubro de 2022 e dezembro de 2022 para o produto MODELO 01 79
Gráfico 4 – Relação de favorecimento de cada falha para o resultado do produto MODELO 01 80
Gráfico 5 – Relação entre quantidade de OS e data de cada falha para o produto MODELO 02 84
Gráfico 6 – Etapa de treinamento no período de fevereiro de 2021 e setembro de 2022 para o produto MODELO 02 86
Gráfico 7 – Etapa de teste entre o período de outubro de 2022 e dezembro de 2022 para o produto MODELO 02 87
Gráfico 8 – Relação de favorecimento de cada falha para o resultado do produto MODELO 02 88

LISTA DE TABELAS

Tabela 1 – Descrição dos grupos de falhas..... 45
Tabela 2 – Descrição das categorias de falhas..... 46

LISTA DE SIGLAS

CSV - Comma Separated Values
IA - Inteligência Artificial
OS - Ordem de Serviço
PU - Poliuretano
SKU - Stock Keeping Unit
SQL - Structured Query Language

SUMÁRIO

1. INTRODUÇÃO.....14
1.1 JUSTIFICATIVA..... 15
1.2 OBJETIVOS.....15
1.2.1 OBJETIVOS SECUNDÁRIOS..... 16
1.3 BANCO DE DADOS E ORGANIZAÇÃO DOS DADOS.....17
1.4 SQL SERVER..... 18
1.5 LINGUAGEM SQL..... 19
1.6 INTELIGÊNCIA ARTIFICIAL.....21
1.7 APRENDIZAGEM DE MÁQUINA..... 22
1.8 PYTHON.....25
1.9 SCIKIT-LEARN..... 27
1.10 LINEAR REGRESSION.....28
1.11 RANDOM FOREST.....29
1.12 LIGHT GBM..... 31
1.13 PANDAS..... 32
1.14 CROSS VALIDAÇÃO.....33
1.15 GRID SEARCH E HIPER PARÂMETROS.....34
1.16 GOOGLE COLAB..... 35

2. MATERIAIS E MÉTODOS.....36
2.1 ANÁLISE E ORGANIZAÇÃO DOS DADOS DE FALHAS EM LINHA DE MONTAGEM DE REFRIGERADORES..... 38
2.2 ANÁLISE E ORGANIZAÇÃO DOS DADOS DE ORDENS DE SERVIÇO DE REFRIGERADORES..... 58
2.3 DESENVOLVIMENTO DO MODELO DE DADOS UTILIZANDO IA E APRENDIZAGEM DE MÁQUINA..... 65

3. RESULTADOS E DISCUSSÕES.....73
3.1 REFRIGERADOR MODELO 01..... 73
3.2 REFRIGERADOR MODELO 02..... 81

4. CONCLUSÃO.....90

REFERÊNCIAS..... 91

1. INTRODUÇÃO

A fabricação de bens de consumo, como geladeiras e eletrodomésticos em geral, desempenha um papel fundamental em nossa sociedade moderna. Esses produtos são essenciais para o nosso dia a dia, tornando nossas vidas mais confortáveis e convenientes. A importância desses bens vai além da mera comodidade, estendendo-se à preservação de alimentos, à eficiência energética e ao bem-estar geral das famílias em todo o mundo. A demanda por esses produtos tem sido crescente, o que coloca uma pressão significativa sobre as indústrias que os produzem para manter altos padrões de qualidade e eficiência.

Segundo Thompson (2007), o processo industrial de fabricação e montagem de eletrodomésticos é complexo e envolve diversas etapas, desde a seleção e preparação de materiais até a montagem final. Nesse contexto, uma das fases críticas é a linha de montagem, onde ocorrem muitas operações manuais e automatizadas. Mesmo com rigorosos controles de qualidade, não é incomum que ocorram falhas durante o processo de montagem. Essas falhas podem se manifestar de várias formas, desde defeitos estéticos, como arranhões e marcas, problemas de ingestão de plástico em portas, gabinete e interior do produto, como também problemas funcionais, desde elétricos, eletrônicos e até mesmo mecânicos.

As falhas que ocorrem na linha de montagem de refrigeradores podem ter um impacto significativo nas finanças e na produção de uma indústria multinacional de bens de consumo. Além dos custos diretos associados ao retrabalho e substituição de componentes defeituosos, as falhas também podem resultar em atrasos na produção, o que afeta a capacidade da empresa de atender à demanda do mercado (MORALES; MELO; MIYAGI, 2007; RIASCOS, 2002). Além disso, o retrabalho e as devoluções de produtos defeituosos podem prejudicar a reputação da marca e a confiança do cliente, o que tem implicações a longo prazo no sucesso do negócio.

Uma vez que os produtos produzidos deixam a linha de montagem e são vendidos aos clientes, podem surgir problemas adicionais na forma de Ordens de Serviço (OS), visto que falhas são inevitáveis em sistemas concebidos e operados pelo homem (RIASCOS; MIYAGI, 2006). Essas OS são geradas quando os consumidores encontram problemas com os produtos que adquiriram, problemas que podem estar diretamente relacionados às falhas encontradas no processo industrial de montagem. Essas reclamações pós-venda têm um impacto negativo na

satisfação do cliente e exigem recursos significativos para serem resolvidas, incluindo reparos, substituições e reembolsos. Com isso, não há forma de definir qualidade sem atentar para o atendimento integral ao cliente (PALADINI, 1990). Portanto, a identificação e a prevenção precoce de falhas na linha de montagem industrial são cruciais para evitar custos excessivos e manter a satisfação do cliente em alta.

Atualmente com o avanço da tecnologia, a aplicação de inteligência artificial, lógicas não clássicas e algoritmos de aprendizagem de máquina desempenham um papel vital, oferecendo a oportunidade de identificar padrões (DA SILVA FILHO, J. I. da S.; ABE J. M.; TORRES, G. L., 2008), inclusive entre dados de linha de produção industrial (EDDY, KRISHNAMURTY, GROSSE, e WHITE, 2020). Permitindo assim o aprimoramento, eficiência e a precisão dos processos de controle de qualidade, contribuindo na redução do número de ordens de serviço e falhas industriais, além de melhorar o desempenho global da indústria de bens de consumo.

1.1 JUSTIFICATIVA

Os problemas em refrigeradores podem ser reduzidos em larga escala dependendo dos componentes que o compõem e da eficiência do processo industrial praticado. Um fator importante que justifica essa pesquisa é que as linhas de produção e montagem possuem um número grande de informações resultante de cada fase do processo industrial (MORALES; MELO; MIYAGI, 2007; RIASCOS, 2002), principalmente na fase final, que é a fase de montagem.

Esses dados podem gerar diversas informações, dentre elas número de falhas ocorridas, tipo de falha, tipo de produto e número de OS criadas pelos clientes, através de problemas encontrados pós venda. Todo esse volume de dados pode ser muito mais produtivo para a indústria desde que exista um tratamento, análise e classificação dessas informações.

Com a aplicação de algoritmos de aprendizagem de máquina, é proposto neste trabalho uma nova forma de manipulação e análise desses dados, podendo gerar tomadas de decisão a partir das informações normalizadas e classificadas em um modelo desenvolvido utilizando algoritmos de inteligência artificial (LEBERRUYER, BRUCH, AHLSKOG e AFSHAR, 2023).

1.2 OBJETIVOS

O objetivo deste trabalho consiste em construir uma arquitetura computacional configurada com três algoritmos especiais capazes de atuarem em conjunto para analisar dados de falhas no processo de montagem industrial de uma indústria de bens de consumo, mais especificamente refrigeradores, identificando potenciais de geração de ordens de serviço para predizer as principais falhas industriais.

1.2.1 OBJETIVOS SECUNDÁRIOS

Validar a arquitetura construída através de análises de montagem de refrigeradores fundamentados em banco de dados reais originados da produção de uma indústria de bens de consumo.

Os passos para conclusão deste objetivo foram:

1. Analisar dados de falhas encontradas na linha de montagem de refrigeradores dos últimos 3 anos, conforme disponibilizado pela empresa;

2. Organizar esses dados categorizando-os em 20 grupos de falhas, sendo divididos em 3 grandes categorias;

3. Analisar dados de OS de refrigeradores que foram abertas por clientes nos últimos 3 anos, conforme disponibilizado pela empresa;

4. Criar e agrupar os 20 grupos de falhas encontradas durante o processo industrial de montagem com as OS criadas por clientes finais;

5. Organizar os dados de falhas com os dados de OS, contabilizando apenas as OS abertas em um intervalo de 6 meses após a venda do refrigerador;

6. Utilização de bibliotecas de inteligência artificial para desenvolvimento de um modelo de dados utilizando os dados normalizados;

7. Criação de *pipeline* para submissão do conjunto de dados no algoritmo *Linear Regression*;

8. Criação de *pipeline* para submissão do conjunto de dados no algoritmo *Random Forest*;

9. Criação de *pipeline* para submissão do conjunto de dados no algoritmo *Light GBM*;

10. Utilização do *Grid Search* com *Cross Validação* para aplicação dos 3 algoritmos e descoberta de quais deles obtiveram melhor resultado;

11. Identificação das principais falhas industriais que são potenciais de geração de ordens de serviço por clientes;

12. Predição da quantidade de OS que poderão ser geradas, baseado nas falhas que acontecem na linha de montagem industrial de refrigeradores.

1.3 BANCO DE DADOS E ORGANIZAÇÃO DOS DADOS

Os bancos de dados desempenham um papel central em inúmeras aplicações e sistemas de informação em todo o mundo, armazenando e gerenciando uma vasta quantidade de dados de maneira estruturada. Para garantir a eficiência, integridade e flexibilidade desses sistemas, a organização de bancos de dados desempenha um papel fundamental. A organização é um conjunto de técnicas que visa organizar os dados em tabelas relacionais, minimizando a redundância e mantendo a consistência dos dados. Ela é essencial para otimizar consultas, economizar espaço de armazenamento e evitar problemas de integridade dos dados (SILBERSCHATZ et al., 2020).

A organização de bancos de dados é baseada em um conjunto de regras normalmente referidas como as Formas Normais. A teoria das Formas Normais, introduzida por Edgar F. Codd na década de 1970, oferece diretrizes precisas para a organização eficiente de dados em um banco de dados relacional. A primeira forma normal (1NF) exige que cada atributo de uma tabela contenha apenas valores atômicos, enquanto a segunda forma normal (2NF) exige que todos os atributos não-chave sejam totalmente funcionais em relação à chave primária. Essas regras são refinadas nas Formas Normais subsequentes (3NF, BCNF, 4NF, 5NF) para eliminar a redundância e as dependências funcionais indesejadas (ELMASRI & NAVATHE, 2019).

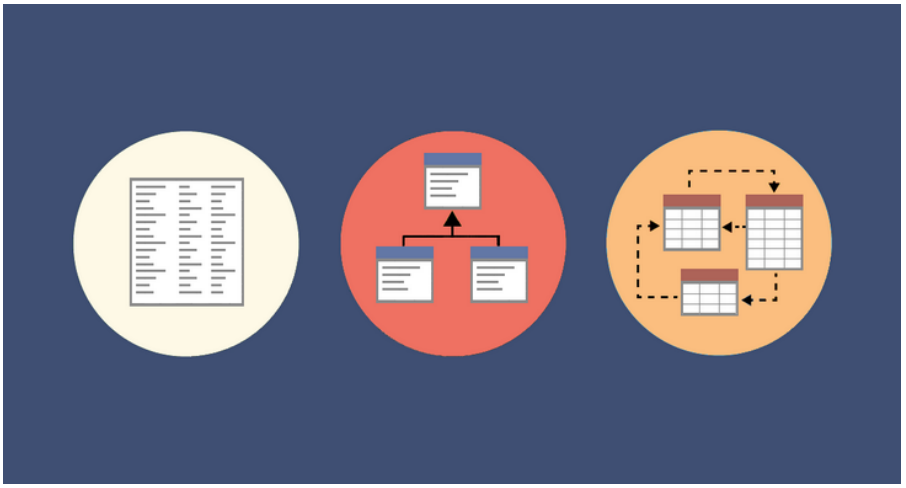

Figura 1 – Processo de organização de dados.
Fonte: KONONOW, 2023.

A organização não é apenas uma técnica teórica; ela tem implicações práticas significativas, conforme figura 1. Reduzir a redundância de dados economiza espaço de armazenamento e melhora a consistência dos dados, uma vez que as atualizações precisam ser feitas em apenas um local. Além disso, facilita a manutenção e a escalabilidade do banco de dados, tornando-o mais flexível para futuras mudanças nos requisitos do sistema. A organização também contribui para a otimização de consultas, uma vez que as tabelas mais organizadas facilitam a recuperação eficiente de informações (RAMAKRISHNAN & GEHRKE, 2003).

Em resumo, a organização de bancos de dados desempenha um papel crucial na criação e manutenção de sistemas de gerenciamento de dados eficazes e confiáveis. Ela fornece um conjunto de diretrizes sólidas para a organização de dados em tabelas relacionais, minimizando a redundância e maximizando a eficiência e integridade dos dados. A compreensão desses princípios é essencial para projetistas e administradores de banco de dados, garantindo a entrega de sistemas de alta qualidade e desempenho.

1.4 SQL SERVER

O Microsoft SQL Server, desenvolvido pela Microsoft Corporation, é um software que disponibiliza uma solução robusta para o gerenciamento de dados em empresas de todos os portes. Conhecido por sua eficiência no tratamento de

grandes volumes de dados, o SQL Server é amplamente utilizado em todo o mundo (LAROCK, 2020).

Uma das principais vantagens do SQL Server é sua compatibilidade com a linguagem de programação SQL, tornando-o acessível para desenvolvedores e administradores de banco de dados (NIELSEN, 2019). Além disso, sua estreita integração com tecnologias Microsoft, como os sistemas Windows e o .NET, conforme figura 2, facilita o desenvolvimento e a implantação de aplicativos (MISTRY & MISNER, 2019).

Figura 2 – Banco de Dados Microsoft SQL Server.
Fonte: MICROSOFT, 2014.

O SQL Server também oferece recursos avançados de segurança, como o Transparent Data Encryption (TDE) e auditoria, garantindo a proteção de dados sensíveis e conformidade com regulamentações de privacidade (LaRock, 2020).

Isso torna o SQL Server uma escolha confiável para empresas que buscam uma solução de gerenciamento de banco de dados escalável, segura e compatível com padrões da indústria.

1.5 LINGUAGEM SQL

A linguagem SQL (Structured Query Language) é uma poderosa linguagem de programação utilizada para gerenciar, consultar e manipular bancos de dados relacionais. Criada originalmente nos laboratórios da IBM na década de 1970, a SQL se tornou a linguagem padrão para interação com sistemas de gerenciamento de

banco de dados (SGBDs) e é amplamente adotada em todo o mundo. Seu amplo uso se deve à sua simplicidade e eficácia na recuperação e manipulação de dados em bancos de dados relacionais, permitindo aos desenvolvedores e administradores de banco de dados executar tarefas complexas de forma estruturada e eficiente (CARDOSO, 2013).

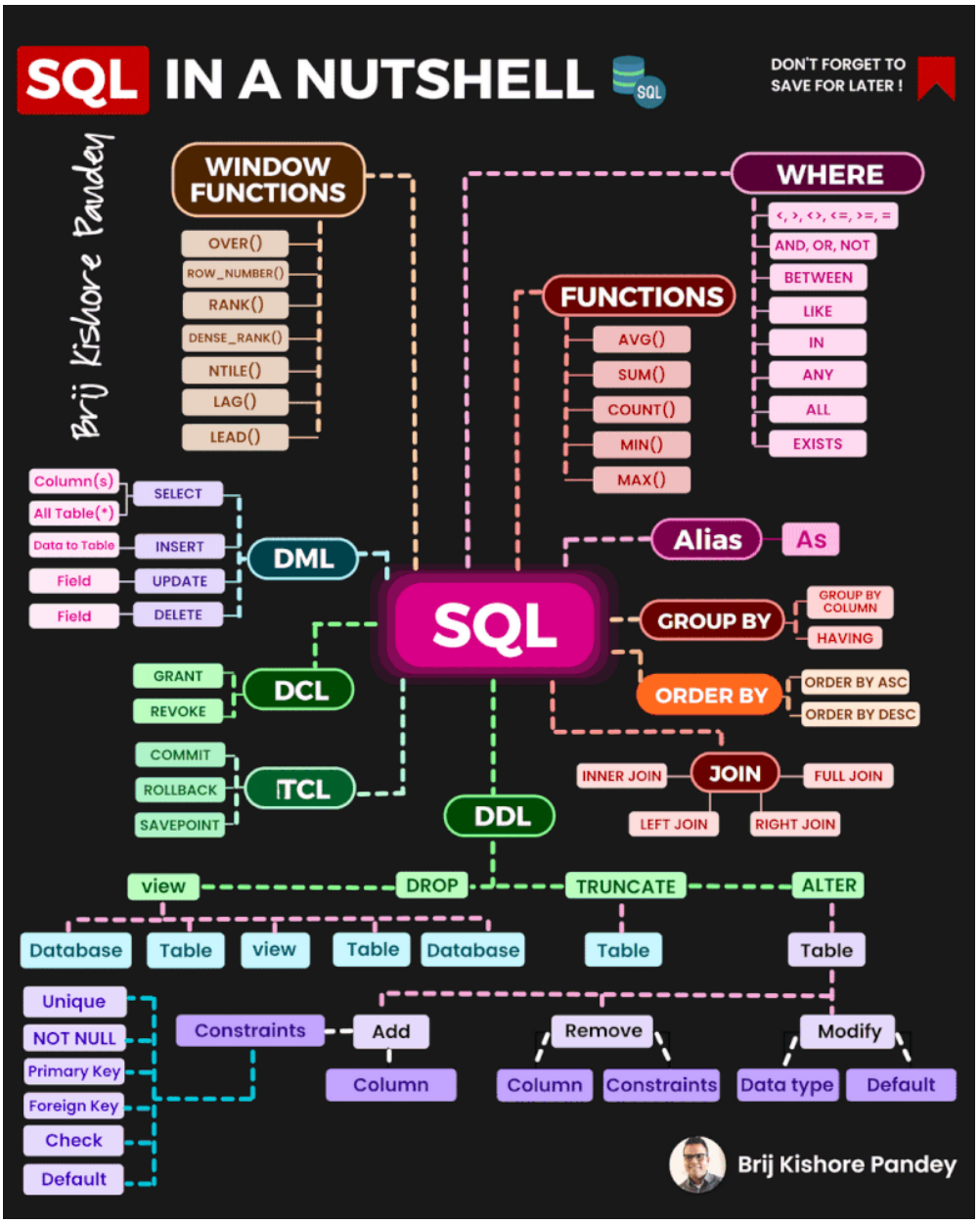

Figura 3 – Linguagem de programação SQL e suas aplicações.
Fonte: PANDEY, 2023.

De acordo com a figura 3, SQL é uma linguagem declarativa, o que significa que os usuários especificam o que desejam fazer com os dados, em vez de como

fazer. Ela é composta por diversos comandos, como SELECT (para consultar dados), INSERT (para adicionar novos registros), UPDATE (para modificar registros existentes) e DELETE (para remover registros). Além disso, a SQL suporta funções agregadas, operadores lógicos e aritméticos, o que a torna uma linguagem versátil para análise e manipulação de dados. A sua sintaxe e estrutura seguem um conjunto de regras rigorosas e consistentes, tornando-a facilmente compreensível e aplicável em diversas plataformas de bancos de dados.

1.6 INTELIGÊNCIA ARTIFICIAL

A Inteligência Artificial (IA) tem emergido como uma das tecnologias mais influentes e transformadoras da era digital. Ela capacita máquinas e sistemas a realizar tarefas que anteriormente eram exclusivas da capacidade humana, como aprendizado, raciocínio e tomada de decisões. A IA é uma área multidisciplinar que combina conceitos da ciência da computação, matemática, estatísticas e até mesmo neurociência para criar sistemas capazes de aprender e adaptar-se a partir de dados (RUSSELL & NORVIG, 2021).

Segundo Domingos, 2018, a IA é uma revolução que rivaliza com a eletricidade e a internet em termos de seu impacto na sociedade. Ela está sendo aplicada em diversos setores, desde assistentes virtuais em dispositivos móveis até carros autônomos, diagnóstico médico, análise financeira e muito mais.

Uma das áreas mais promissoras da IA é o Aprendizado de Máquina (Machine Learning), conforme figura 4, que permite que sistemas aprendam com dados e melhorem seu desempenho ao longo do tempo. O aprendizado de máquina tem sido fundamental em aplicações como reconhecimento de padrões, recomendação de conteúdo, previsão de demanda e até mesmo na detecção de fraudes financeiras (GOODFELLOW et al., 2016).

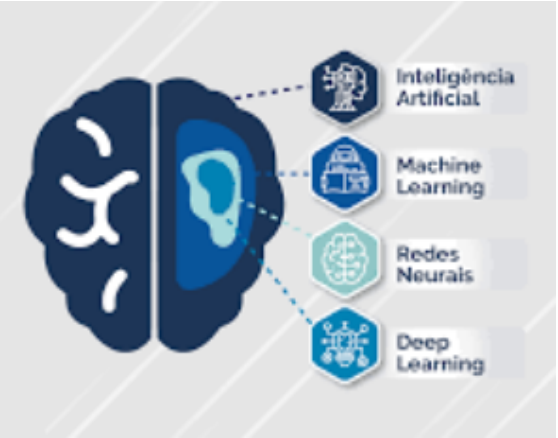

Figura 4 – Campos de atuação da inteligência artificial.
Fonte: INSIGHT, 2023.

A IA também está impulsionando a pesquisa em Redes Neurais Artificiais (RNAs), que são modelos computacionais inspirados no funcionamento do cérebro humano. As RNAs têm demonstrado sucesso em tarefas de visão computacional, processamento de linguagem natural e jogos complexos, como o xadrez (LeCun, Bengio & Hinton, 2015).

No entanto, a crescente adoção da IA levanta questões éticas e sociais importantes. É fundamental abordar preocupações relacionadas à privacidade, preconceito algorítmico e o impacto potencial no mercado de trabalho à medida que a IA se expande (Bostrom, 2014).

A Inteligência Artificial está moldando nosso mundo de maneiras profundas e revolucionárias. Ela não apenas automatiza tarefas, mas também transforma a maneira como trabalhamos, aprendemos, cuidamos da saúde e interagimos com a tecnologia. À medida que a IA continua a evoluir, é crucial considerar os desafios éticos e sociais que ela apresenta para garantir que seu impacto seja positivo e inclusivo.

1.7 APRENDIZAGEM DE MÁQUINA

O Aprendizado de Máquina (Machine Learning, em inglês) é uma disciplina que capacita sistemas computacionais a aprenderem padrões a partir de dados e aperfeiçoarem seu desempenho com base nessa aprendizagem. Kevin P. Murphy destaca que "o aprendizado de máquina é uma das tecnologias mais transformadoras do século 21" (Murphy, 2012).

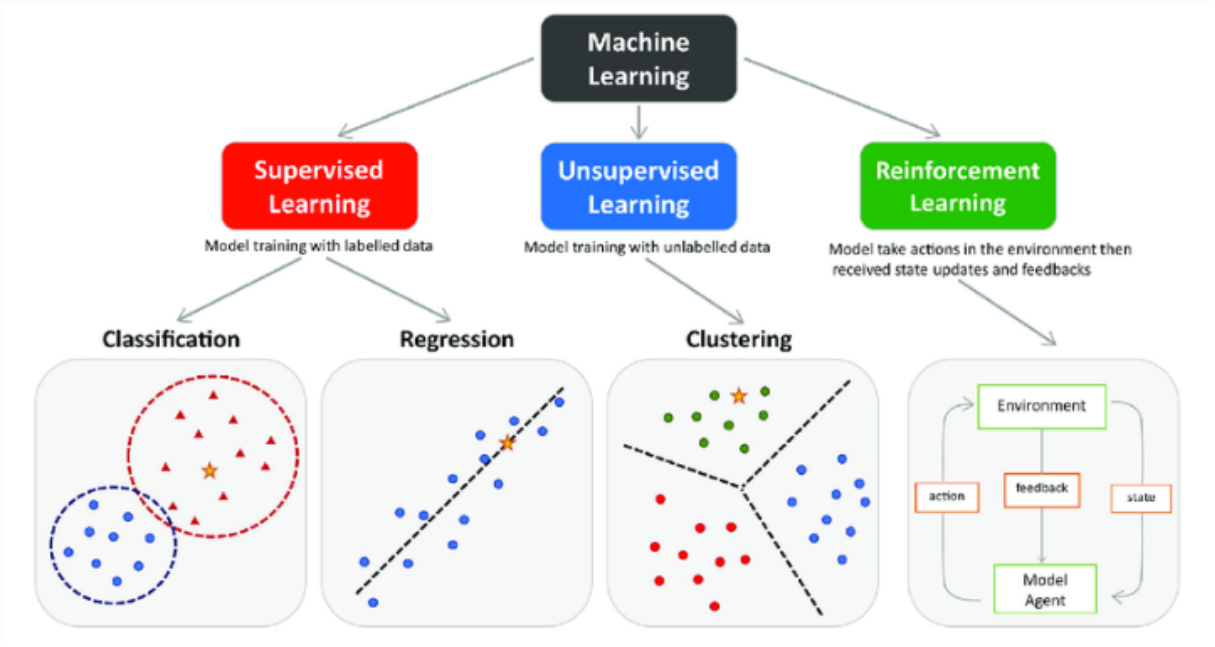

Figura 5 – Aprendizado de máquina e seus modelos.
Fonte: MANOJ, 2023.

Sua aplicação é ubíqua, desde carros autônomos até diagnósticos médicos. O Aprendizado Profundo (Deep Learning), uma subárea, é reconhecido por sua eficácia em tarefas como visão computacional e processamento de linguagem natural (Goodfellow et al., 2016), conforme figura 5.

No entanto, o Aprendizado de Máquina enfrenta desafios, incluindo a escolha do algoritmo correto para um problema específico (Domingos, 2018). Questões éticas e de privacidade também são relevantes.

Nesse estudo será aplicado o modelo supervisionado que funciona da seguinte maneira conforme figura 6:

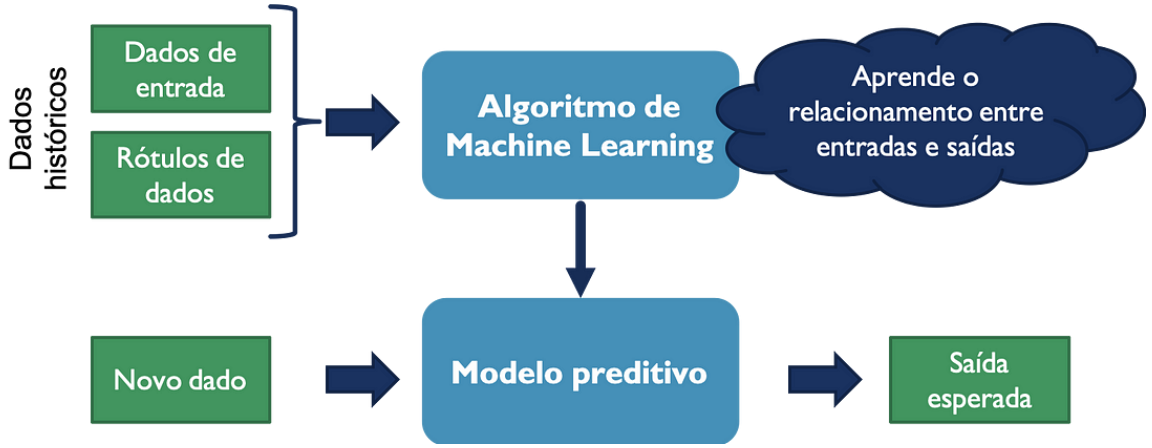

Figura 6 – Funcionamento do modelo supervisionado.
Fonte: ESCOVEDO & KOSHIYAMA, 2020.

O modelo supervisionado, possui basicamente 2 tipos: Classificação e Regressão. A Classificação é uma das categorias de problemas de Machine Learning mais importantes e mais populares e o objetivo do algoritmo é aprender uma regra geral que mapeie as entradas nas saídas corretamente, conforme ilustrado na figura 7.

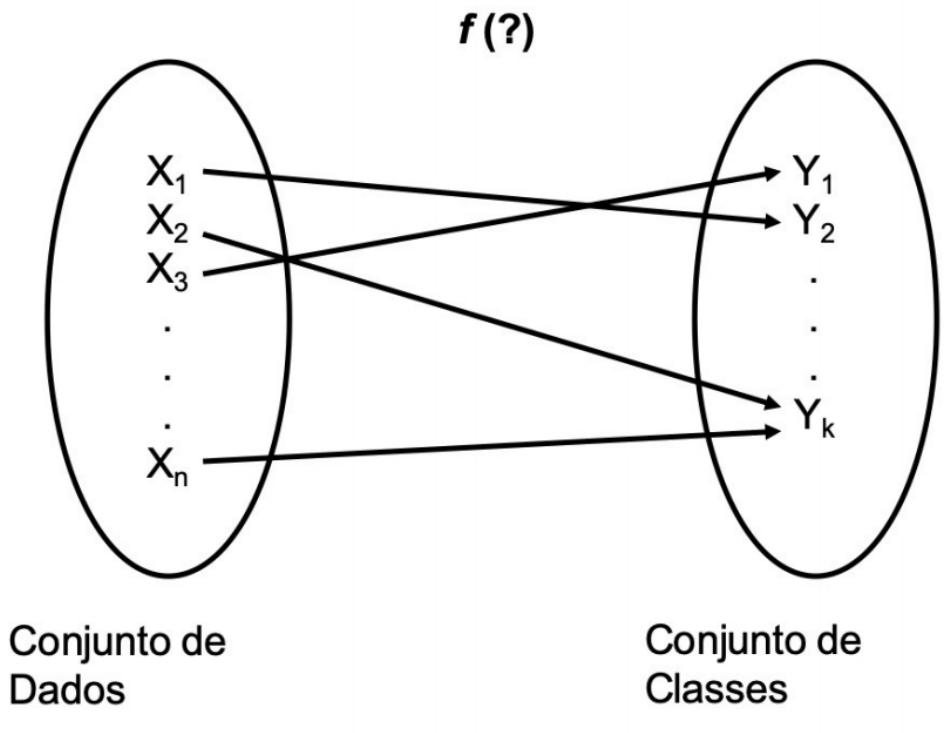

Figura 7 – Classificação no modelo supervisionado.
Fonte: ESCOVEDO & KOSHIYAMA, 2020.

Já o tipo Regressão tem como principal diferença a saída do modelo: na Classificação, o resultado é categórico e na Regressão, o resultado é numérico (contínuo ou discreto), conforme figura 8.

Size of house(10 sq mt)	Price of house(10 L)
1	2.5
3	3
3.5	4.5
5	5
5.5	6
6.5	6
8	7

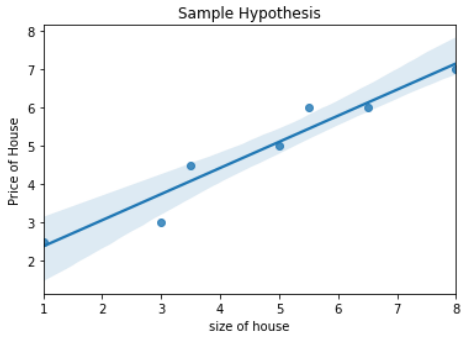

Figura 8 – Regressão no modelo supervisionado.
Fonte: CHANDEKAR, 2021.

De acordo com a figura 9, a aplicação básica do modelo supervisionado utilizando classificação ou regressão seria dessa forma:

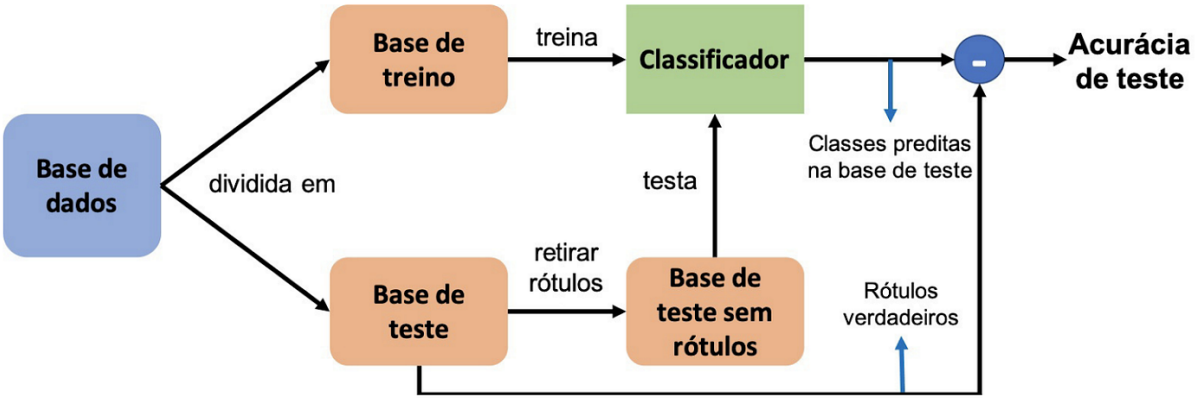

Figura 9 – Aplicação do modelo supervisionado.
Fonte: ESCOVEDO & KOSHIYAMA, 2020.

1.8 PYTHON

Python é uma linguagem de programação de alto nível conhecida por sua simplicidade e versatilidade. Guido van Rossum, o criador de Python, definiu a filosofia da linguagem como "legibilidade conta" (Python.org). Isso a torna uma escolha popular para programadores de todos os níveis de experiência.

Como descrito por Charles Severance em seu livro "Python for Everybody," Python é uma linguagem de fácil aprendizado, ideal para iniciantes (Severance, 2016). Sua sintaxe limpa e clara permite que os desenvolvedores se concentrem na lógica do programa em vez de se perderem em detalhes de formatação.

Além disso, Python possui uma vasta biblioteca padrão e uma comunidade ativa de desenvolvedores que contribuem com bibliotecas de terceiros. Isso amplia sua utilidade em diversas áreas, como desenvolvimento web, ciência de dados, automação e aprendizado de máquina, conforme ilustrado na figura 10.

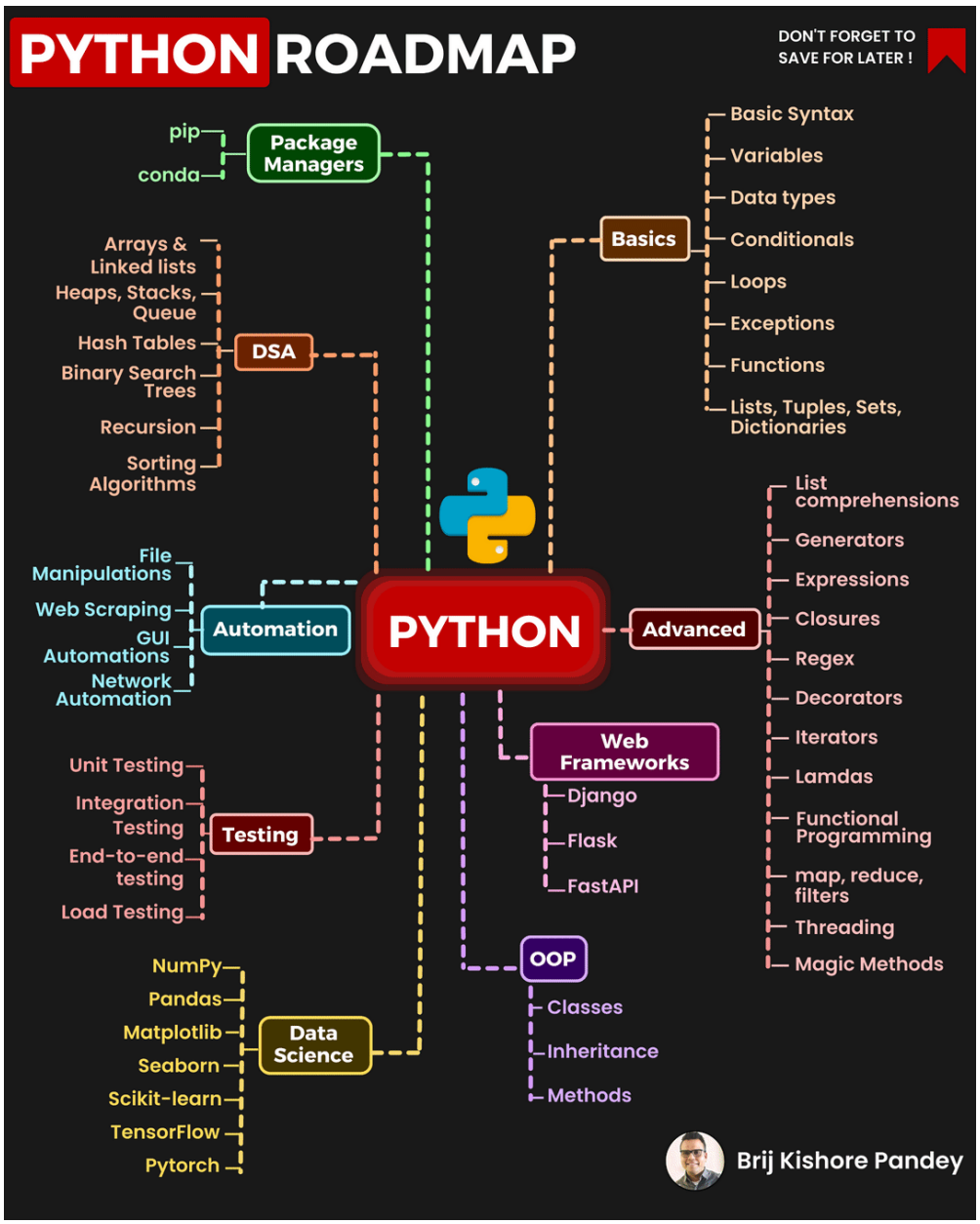

Figura 10 – Linguagem de programação Python e suas aplicações.
Fonte: PANDEY, 2023.

Em relação à ciência de dados, Jake VanderPlas, autor de "Python Data Science Handbook," destaca que "Python se tornou uma das linguagens mais

populares para análise de dados e é uma escolha essencial para qualquer cientista de dados" (VANDERPLAS, 2016).

Sua versatilidade também se estende ao aprendizado de máquina, onde bibliotecas como scikit-learn e TensorFlow o tornam uma ferramenta poderosa para a criação de modelos de IA (GÉRON, 2019).

Em resumo, Python é uma linguagem de programação versátil e de fácil aprendizado, amplamente utilizada em várias disciplinas, desde desenvolvimento web até ciência de dados e aprendizado de máquina.

1.9 SCIKIT-LEARN

O Scikit-Learn é uma biblioteca de Aprendizado de Máquina em Python que se destaca pela simplicidade e eficácia. Segundo Andreas C. Müller e Sarah Guido, autores de "Introduction to Machine Learning with Python," o Scikit-Learn oferece "ferramentas simples e eficazes para análise de dados e modelagem estatística" (Müller & Guido, 2016).

Uma das principais vantagens do Scikit-Learn é sua vasta documentação. Ele fornece uma interface consistente e intuitiva para criar, treinar e avaliar modelos de Aprendizado de Máquina. Isso o torna uma escolha popular para desenvolvedores e cientistas de dados, como destaca Jake VanderPlas em "Python Data Science Handbook" (VANDERPLAS, 2016).

O Scikit-Learn também oferece suporte para uma ampla variedade de algoritmos de Aprendizado de Máquina, desde regressão linear até árvores de decisão e algoritmos de clustering. Essa flexibilidade permite que os desenvolvedores escolham a melhor abordagem para seus problemas específicos (PEDREGOSA et al., 2011), conforme ilustrado na figura 11.

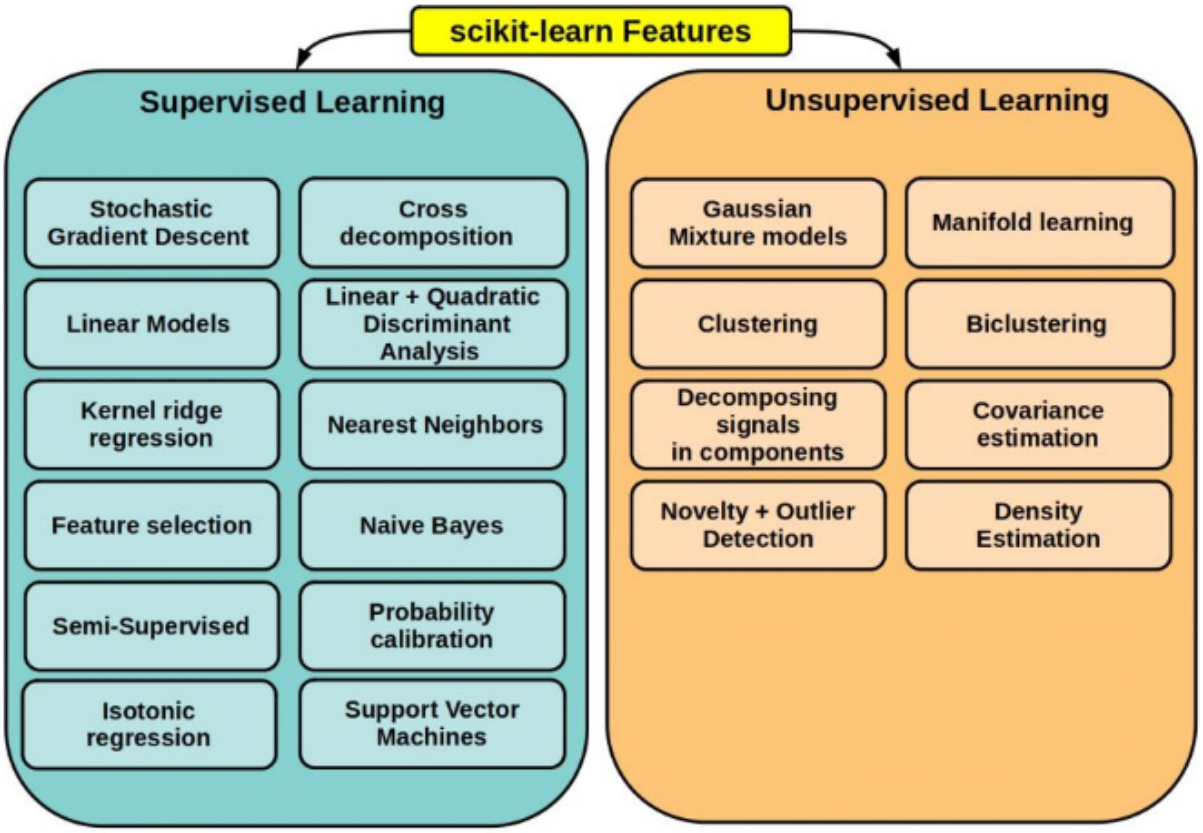

Figura 11 – Biblioteca Scikit-Learn e suas aplicações.
Fonte: RAINERGEWALT, 2020.

Além disso, o Scikit-Learn é uma biblioteca de código aberto amplamente utilizada em projetos de pesquisa e aplicações do mundo real. Seu compromisso com a qualidade e documentação detalhada o tornam uma ferramenta confiável para a construção de soluções baseadas em Aprendizado de Máquina (PEDREGOSA et al., 2011).

Sendo assim, o Scikit-Learn é uma biblioteca essencial para aqueles que desejam explorar e aplicar técnicas de Aprendizado de Máquina em Python. Sua simplicidade, eficácia e ampla aceitação o tornam uma escolha valiosa para desenvolvedores e cientistas de dados.

1.10 LINEAR REGRESSION

A Regressão Linear (*Linear Regression*) é um dos algoritmos mais simples e amplamente utilizados no campo de Aprendizado de Máquina. Segundo Andrew Ng, em seu curso "*Machine Learning*" na plataforma Coursera, a Regressão Linear é

"um método estatístico que modela a relação entre uma variável dependente e uma ou mais variáveis independentes" (NG, 2021), conforme ilustrado na figura 12.

Sua simplicidade é uma de suas principais vantagens. Através da determinação de coeficientes que ponderam as variáveis independentes, a Regressão Linear pode ser aplicada para prever valores futuros com base em dados históricos. Como destacado por Trevor Hastie, Robert Tibshirani e Jerome Friedman em "The Elements of Statistical Learning," "a Regressão Linear é um ótimo ponto de partida para muitos problemas de regressão" (HASTIE, TIBSHIRANI & FRIEDMAN, 2009).

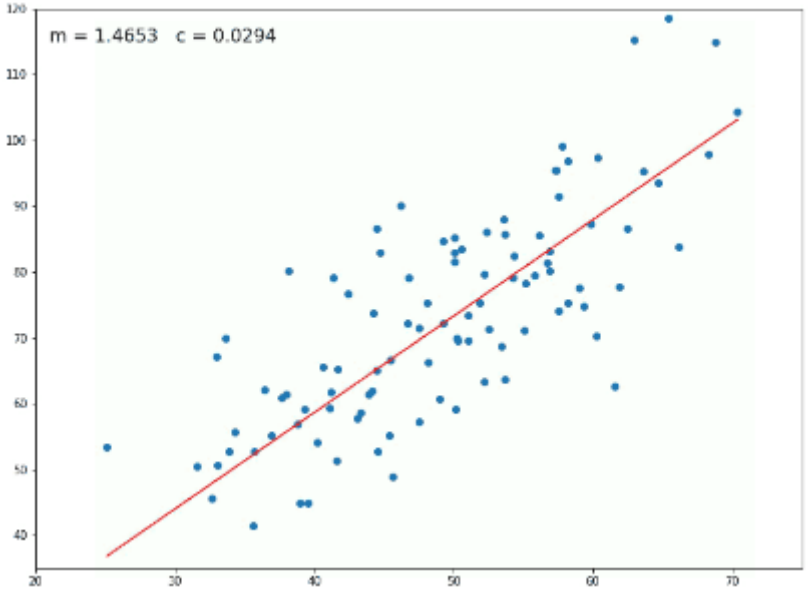

Figura 12 – Funcionamento do Linear Regression a partir de dados submetidos para análise.
Fonte: SANTANA, 2020.

A Regressão Linear é amplamente utilizada em áreas como economia, finanças, ciências sociais e engenharia. No entanto, sua simplicidade não deve ser subestimada. Com as técnicas adequadas, é possível estender a Regressão Linear para lidar com problemas complexos, incluindo múltiplas variáveis independentes e não-lineares.

1.11 RANDOM FOREST

O algoritmo Random Forest é uma técnica poderosa e versátil que se destaca no campo de Aprendizado de Máquina. Leo Breiman, em seu artigo seminal "Random Forests," descreve o Random Forest como "uma combinação de árvores

de decisão, criadas em um processo de bootstrap, que se combinam por meio de voto majoritário para classificação" (BREIMAN, 2001).

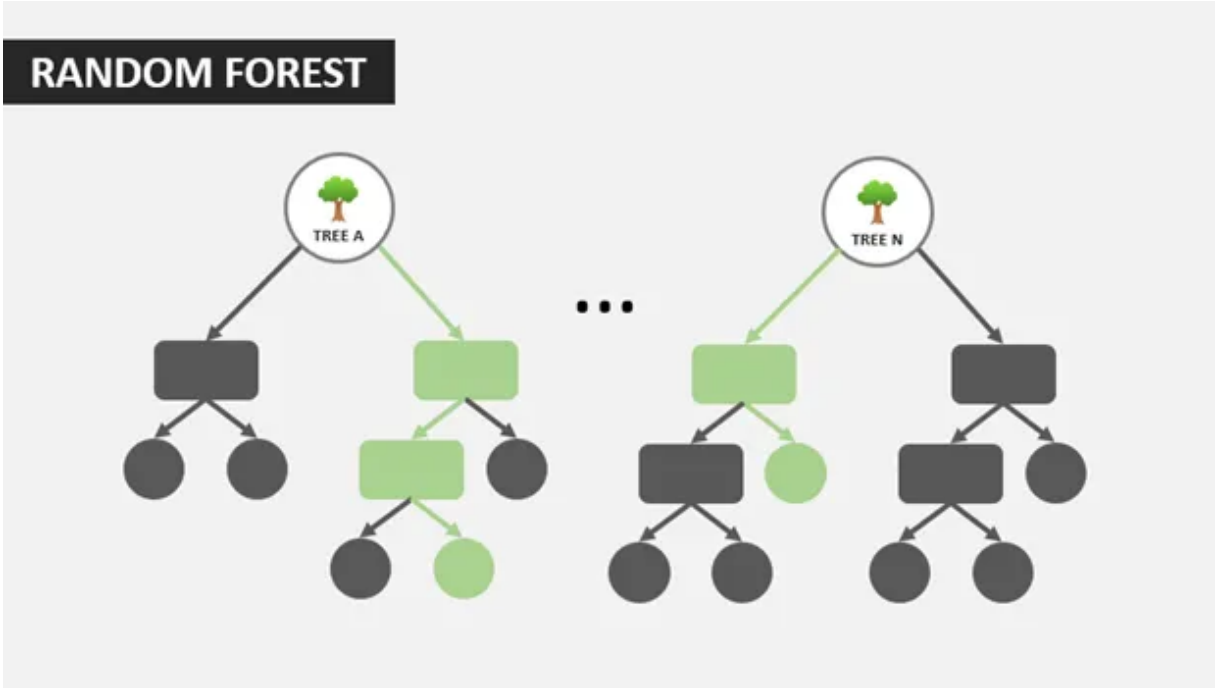

Figura 13 – Funcionamento do Random Forest a partir de dados submetidos para análise.
Fonte: PESSANHA, 2019.

A principal característica que diferencia o Random Forest é sua capacidade de lidar com problemas de classificação e regressão de maneira eficaz. Conforme ilustrado na figura 13, através da criação de um conjunto de árvores de decisão independentes, o Random Forest reduz o risco de *overfitting* e melhora a precisão das previsões. Como aponta Aurélien Géron em "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow," "Random Forest é uma das técnicas mais poderosas para classificação e regressão" (GÉRON, 2019).

Outra vantagem é sua capacidade de lidar com características importantes em um conjunto de dados, permitindo que se avalie a importância de cada variável na classificação. Além disso, o Random Forest é resistente a valores atípicos e não exige um pré-processamento extensivo dos dados.

O Random Forest tem aplicações em uma ampla variedade de áreas, incluindo medicina, finanças e análise de dados. Sua eficácia e versatilidade o tornam uma escolha valiosa para problemas complexos de aprendizado de máquina.

1.12 LIGHT GBM

O LightGBM, ou Gradient Boosting Machine, é um algoritmo de Aprendizado de Máquina que tem se destacado por sua eficiência e desempenho. De acordo com Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang e Wei Chen, autores do artigo "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," o LightGBM é "um método de classificação baseado em árvore que se concentra em eficiência e precisão" (KE et al., 2017).

Uma das principais características do LightGBM é sua velocidade de treinamento e inferência. Ao usar um algoritmo de otimização de histograma, ele reduz significativamente o tempo necessário para construir modelos de árvore, tornando-o adequado para conjuntos de dados grandes. Isso é enfatizado por Aurélien Géron em "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow," onde ele destaca que "o LightGBM é conhecido por sua velocidade excepcionalmente rápida de treinamento" (GÉRON, 2019).

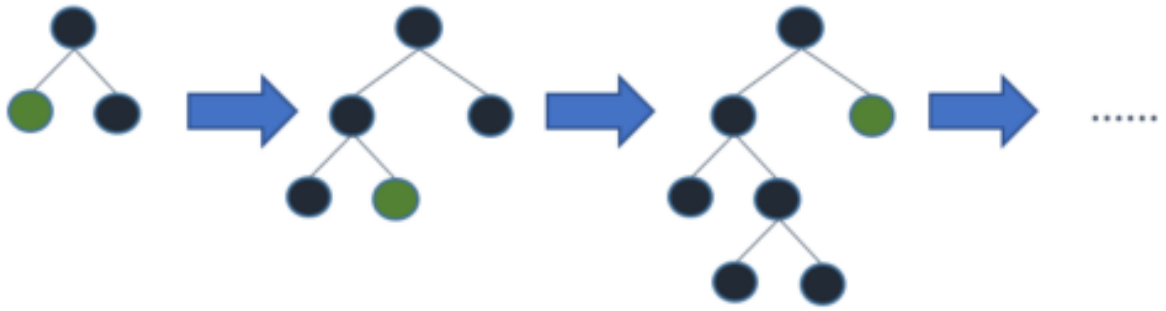

Figura 14 – Funcionamento do Light GBM a partir de dados submetidos para análise.
Fonte: RENNER, 2019.

Além disso, o LightGBM é altamente escalável e pode lidar com grandes volumes de dados e características. Ele também oferece suporte para tarefas de classificação, regressão e ranking, conforme figura 14.

Devido à sua eficiência e precisão, o LightGBM é amplamente utilizado em competições de ciência de dados, como o Kaggle, e em aplicativos do mundo real, como detecção de fraudes, recomendações e previsões financeiras.

1.13 PANDAS

A biblioteca Pandas é uma ferramenta essencial no arsenal de qualquer cientista de dados ou analista. Segundo Wes McKinney, autor do livro "Python for Data Analysis," o Pandas é "uma biblioteca que fornece estruturas de dados de alto desempenho e fáceis de usar, além de ferramentas de análise de dados" (McKinney, 2017).

Uma das principais estruturas de dados oferecidas pelo Pandas é o DataFrame, que permite armazenar e manipular dados tabulares de forma eficiente. Isso é particularmente útil ao lidar com conjuntos de dados complexos, como observa Joel Grus em "Data Science from Scratch," onde ele afirma que o Pandas "torna muito mais fácil trabalhar com dados tabulares" (GRUS, 2015).

Figura 15 – Características da biblioteca Pandas.
Fonte: TESTGORILLA, 2023.

Além disso, o Pandas oferece uma ampla gama de funcionalidades para limpeza, transformação e análise de dados. Com métodos simples e intuitivos, os usuários podem realizar tarefas como filtrar, agrupar e calcular estatísticas com facilidade, conforme ilustrado na figura 15.

O Pandas é uma parte fundamental do ecossistema de Python para ciência de dados e é frequentemente usado em conjunto com outras bibliotecas, como

NumPy, Matplotlib e Scikit-Learn. Sua flexibilidade e capacidade de integração o tornam uma escolha valiosa em uma variedade de projetos de análise de dados.

1.14 CROSS VALIDAÇÃO

A Cross-Validação é uma técnica fundamental no mundo do Aprendizado de Máquina que visa avaliar a capacidade de generalização de um modelo de forma robusta. Segundo James, Witten, Hastie e Tibshirani em "An Introduction to Statistical Learning," a Cross-Validação "avalia como um modelo se comportará em dados não vistos" (JAMES et al., 2013).

A abordagem mais comum de Cross-Validação é a validação cruzada k-fold, onde os dados são divididos em k subconjuntos (ou "dobras"). O modelo é treinado em k-1 dessas dobras e testado na dobra restante. Isso é repetido k vezes, garantindo que cada dobra seja usada como conjunto de teste uma vez. A média das métricas de desempenho obtidas nas k iterações fornece uma avaliação geral do modelo, conforme ilustrado na figura 16.

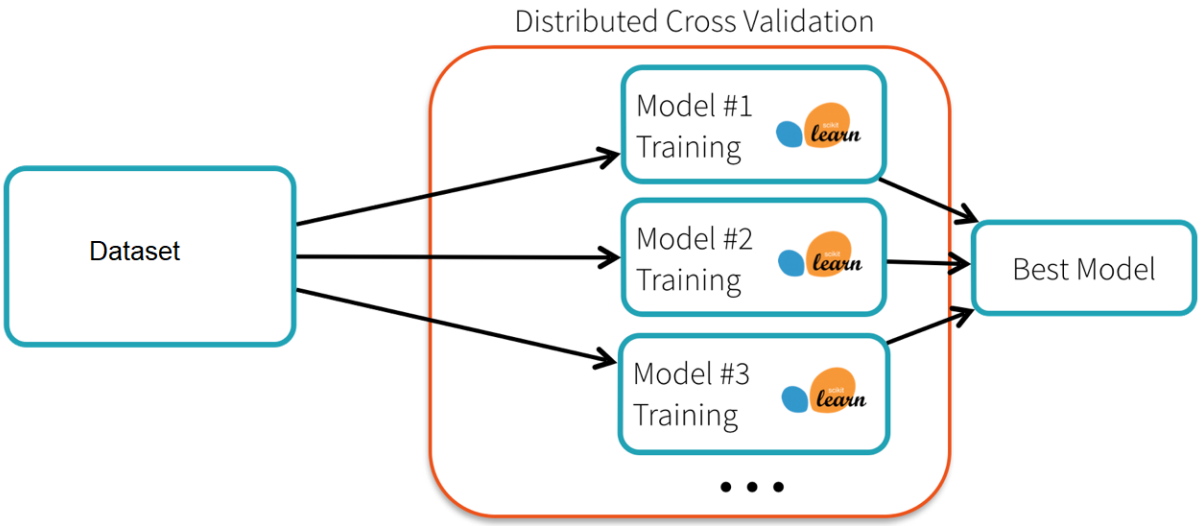

Figura 16 – Funcionamento da Cross Validação.
Fonte: BROOKEWENIG, 2023.

A Cross-Validação é crucial para evitar problemas como overfitting, onde um modelo se ajusta demasiadamente aos dados de treinamento, mas não generaliza bem para novos dados. Ao avaliar um modelo em várias divisões dos dados, a Cross-Validação ajuda a identificar se o desempenho observado é consistente e confiável.

Além disso, a Cross-Validação auxilia na seleção de hiper parâmetros do modelo. Ao testar diferentes configurações em várias dobras, é possível escolher os valores que resultam no melhor desempenho médio.

1.15 GRID SEARCH E HIPER PARÂMETROS

Em Aprendizado de Máquina, a escolha dos hiper parâmetros corretos é essencial para construir modelos de alta qualidade. Os hiper parâmetros são configurações ajustáveis que não são aprendidos pelo modelo, como a taxa de aprendizado em redes neurais ou a profundidade de uma árvore de decisão.

O Grid Search é uma técnica que auxilia na busca pelos melhores hiper parâmetros para um modelo. Segundo Andreas C. Müller e Sarah Guido em "Introduction to Machine Learning with Python," o Grid Search "testa todas as combinações possíveis de hiper parâmetros em um espaço de busca pré-definido" (MULLER & GUIDO, 2016). Isso permite que você avalie o desempenho do modelo em várias configurações e escolha aquela que produz os melhores resultados.

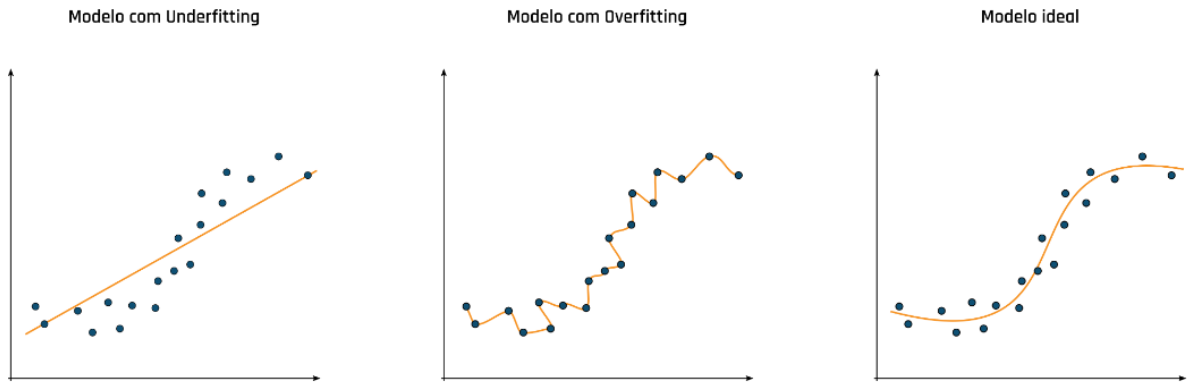

Figura 17 – Aplicação de Hiper parâmetros e seus resultados.
Fonte: ALMEIDA, CARVALHO e MENINO, 2023.

A importância do Grid Search reside na otimização do desempenho do modelo. Por exemplo, em algoritmos de Aprendizado Profundo, ajustar a taxa de aprendizado e o tamanho do lote pode ter um impacto significativo na convergência e na precisão do modelo. Um Grid Search bem executado ajuda a encontrar a combinação ideal de hiper parâmetros, conforme ilustrado na figura 17.

Além disso, o Grid Search é frequentemente combinado com cross validação (cross-validation), garantindo que a seleção dos hiper parâmetros seja robusta e confiável.

1.16 GOOGLE COLAB

O Google Colab, abreviação de Google Colaboratory, é uma plataforma baseada na nuvem que se tornou um recurso essencial no campo do Aprendizado de Máquina. De acordo com Aurélien Géron em "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow," o Google Colab oferece "um ambiente de desenvolvimento Jupyter Notebook acessível gratuitamente" (GÉRON, 2019).

Uma das principais vantagens do Google Colab é a facilidade de uso on line. Os usuários podem criar e executar Notebooks interativos diretamente no navegador, sem a necessidade de configurar uma infraestrutura local. Além disso, ele oferece suporte para várias bibliotecas populares de Aprendizado de Máquina, como TensorFlow e scikit-learn.

Outra característica poderosa é a integração com o Google Drive, permitindo o armazenamento e o compartilhamento de Notebooks e conjuntos de dados de forma conveniente, conforme figura 18.

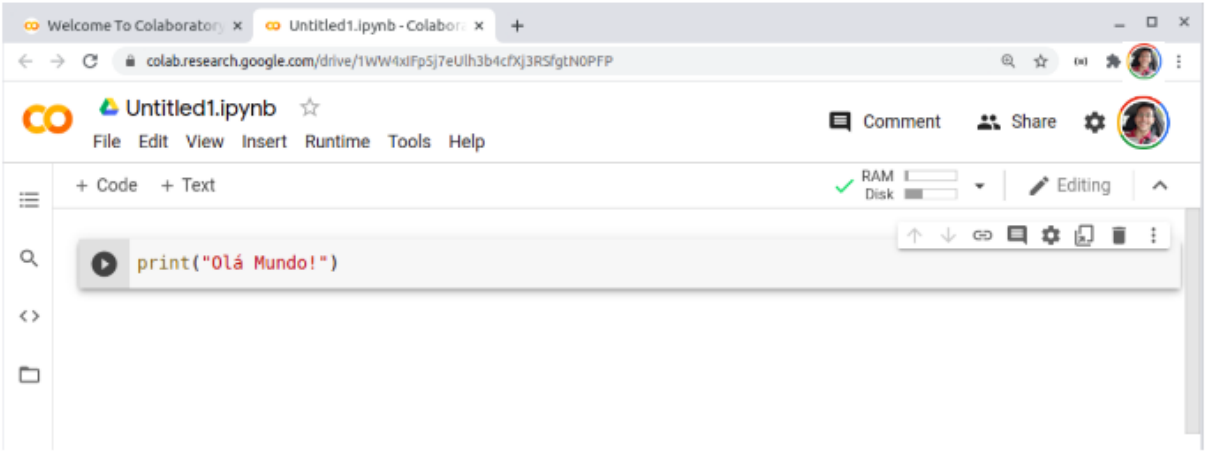

Figura 18 – Interface de desenvolvimento do Google Colab.
Fonte: Elaborado pelo autor.

Além disso, o Google Colab oferece acesso gratuito a GPUs (Unidades de Processamento Gráfico), acelerando o treinamento de modelos de Aprendizado de

Máquina. Isso é particularmente útil para tarefas intensivas em computação, como treinamento de redes neurais profundas.

No entanto, é importante observar que o uso gratuito do Google Colab tem limitações de recursos e duração das sessões, mas é uma ótima opção para experimentação e aprendizado.

Em resumo, o Google Colab é uma ferramenta valiosa para desenvolvedores e cientistas de dados que desejam experimentar, prototipar e treinar modelos de Aprendizado de Máquina de forma acessível e conveniente.

2. MATERIAIS E MÉTODOS

Para o desenvolvimento da pesquisa foi considerada a arquitetura de 6 linhas de montagem industrial de refrigeradores, divididas em 2 fábricas de uma multinacional localizada no Brasil, conforme figura 19:

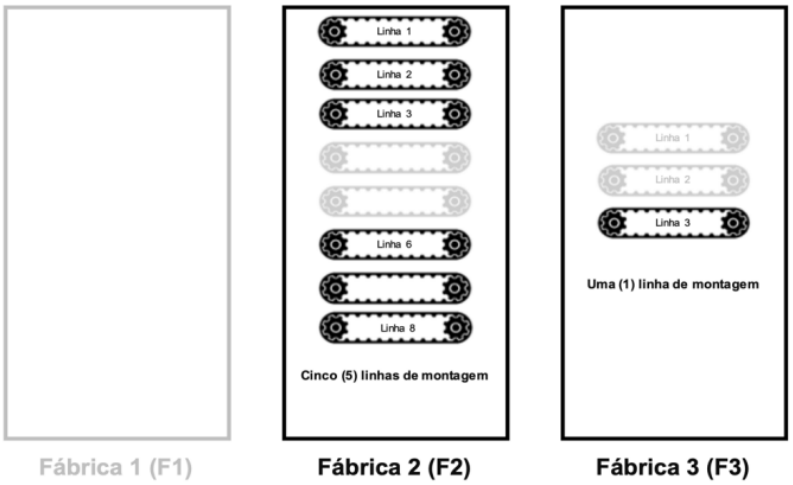

Figura 19 – Arquitetura de 6 linhas de montagem industrial.
Fonte: Elaborado pelo autor.

O processo resumido de uma linha de montagem pode ser visto na figura 20::

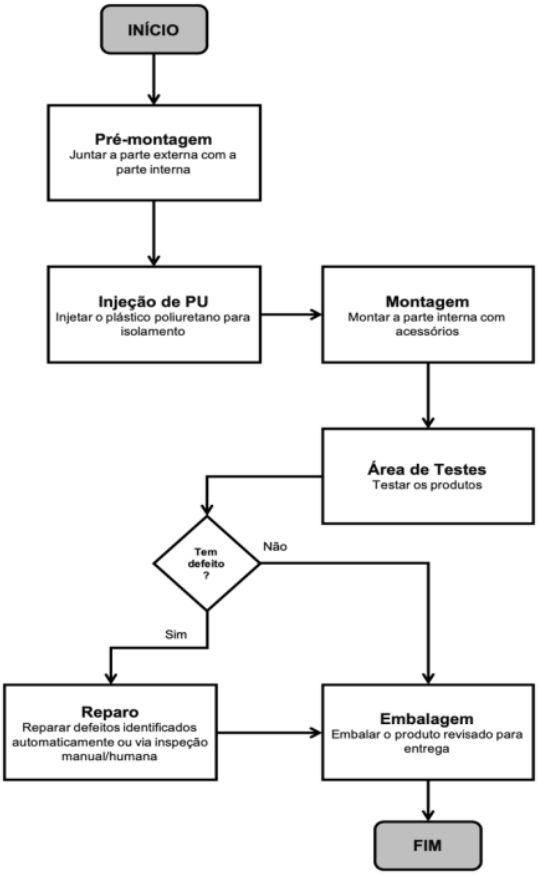

Figura 20 – Processo resumido aplicado na linha industrial de montagem.
Fonte: Elaborado pelo autor.

Por meio do tratamento e organização das informações de falhas que ocorreram na linha de montagem de refrigeradores e nas ordens de serviço (OS) que foram criadas, pretende-se demonstrar uma correlação das falhas com a geração de OS através de aplicação de inteligência artificial.

Entretanto, primeiramente foi necessário obter os dados de forma bruta, analisá-los e classificá-los de uma forma organizada e estruturada para aplicação posterior dos algoritmos de IA.

Além disso, foram utilizados 3 diferentes algoritmos de IA, além de técnicas avançadas para identificação de padrões, tornando o modelo mais confiável. Tendo como resultado um percentual considerável de acerto de OS que poderão ser geradas, diante das falhas encontradas no processo industrial da linha de montagem de refrigeradores.

Sendo assim, temos o resumo do processo desenvolvido, conforme figura 21:

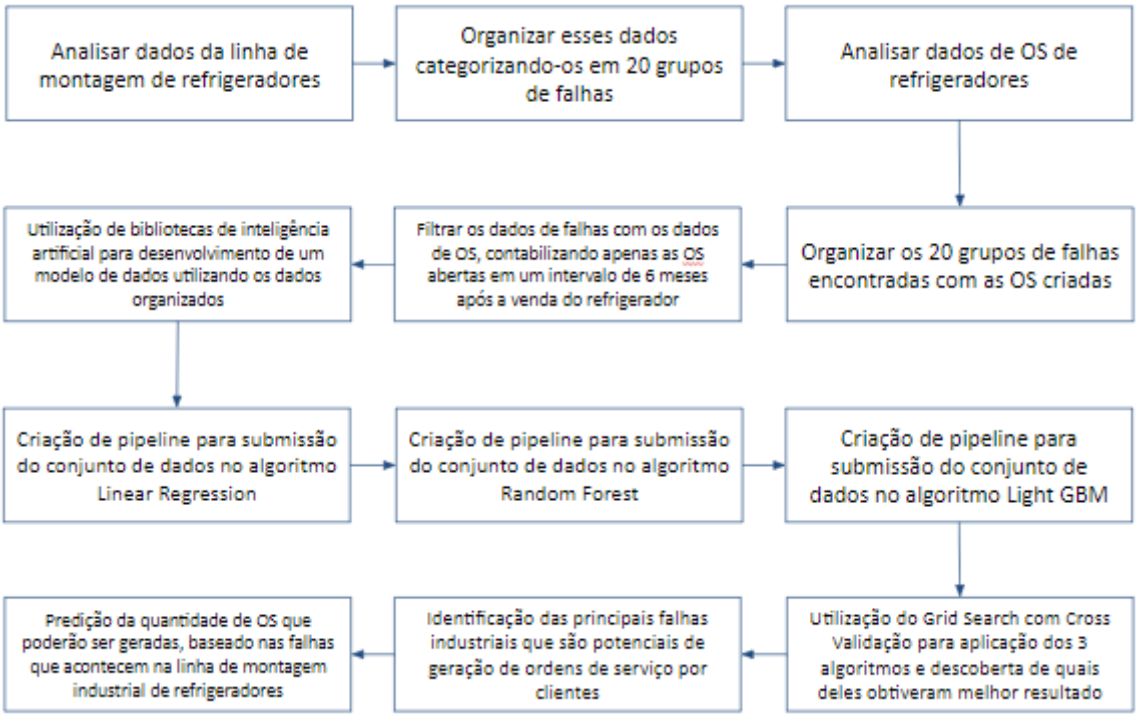

Figura 21 – Fluxo do processo desenvolvido para esse trabalho.
Fonte: Elaborado pelo autor.

2.1 ANÁLISE E ORGANIZAÇÃO DOS DADOS DE FALHAS EM LINHA DE MONTAGEM DE REFRIGERADORES

Para o desenvolvimento dessa etapa, os dados foram extraídos do Data Warehouse da empresa, onde ele é integrado com a malha e os sistemas industriais, conforme figura 22.

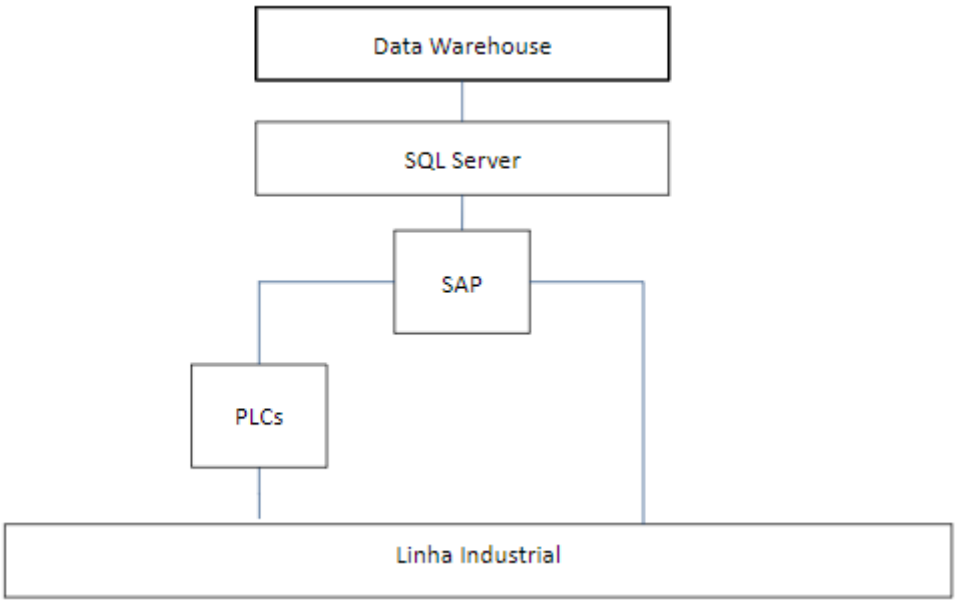

Figura 22 – Origem da extração dos dados.
Fonte: Elaborado pelo autor.

Foi extraído toda a base de dados de falhas encontradas por humanos ou máquinas no período de fevereiro de 2021 até junho de 2023. Esse período compreende em um total de 34.011 registros, divididos em 3 categorias de falhas: Injeção de PU, Funcional e Estético, onde:

Injeção de PU: Compreende todo plástico utilizado para compor peças e partes de um refrigerador;
Funcional: Compreende todos os materiais ou acessórios que impactam no funcionamento do refrigerador;
Estético: Compreende toda avaria no refrigerador, contemplando partes internas e externas.

Essa etapa compreende as seguintes partes do processo (em vermelho), conforme figura 23:

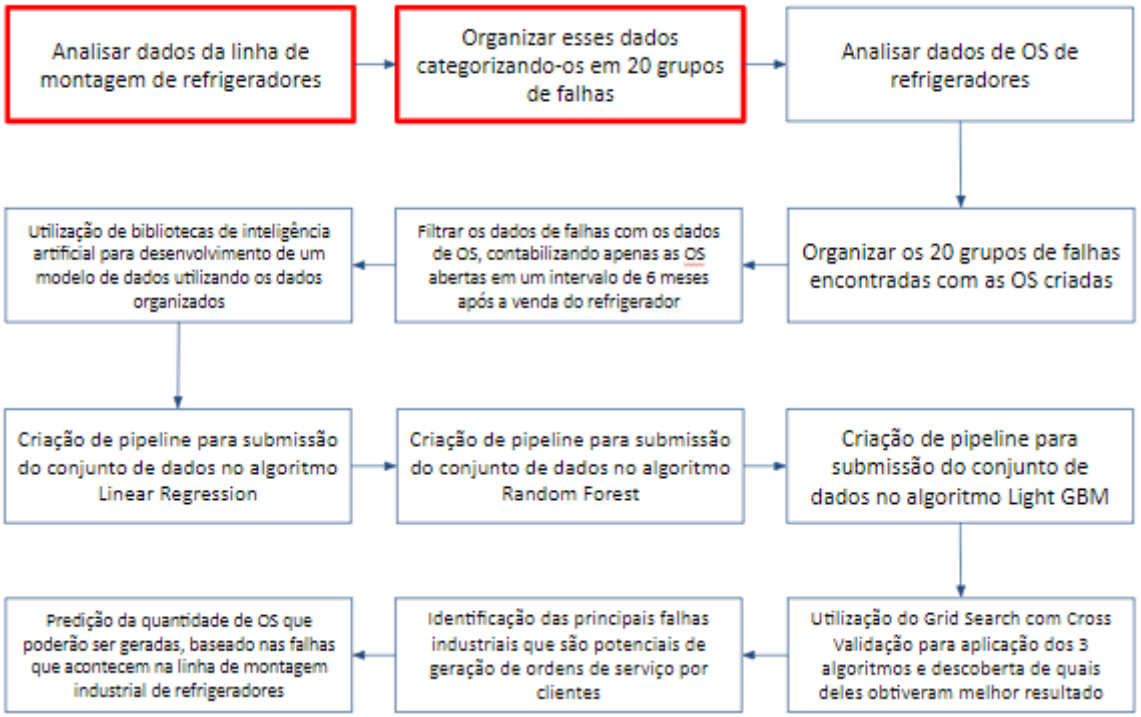

Figura 23 – Etapas desenvolvidas na primeira parte.
Fonte: Elaborado pelo autor.

Todo esse conteúdo de dados foi armazenado em um arquivo CSV, onde podemos observar a estrutura dos dados obtidos antes da organização, utilizando uma planilha Excel, conforme figura 24:

	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	Linha	Conserto	IlhaGeradora	Posto	SerialNumber	Modelo	Motivo1	Motivo2	Motivo3	Motivo4	Data	Dia	Mes	Ano
2	LINHA 2 F2	CONSERTO ESTÉTICO					VAZAMENTO DE PL	FUNDO POLIONDAS	FUNDO POLIONDAS	VAZAMENTO DE PL	2021-11-18 00:33:23	18	11	
3	LINHA 2 F2	CONSERTO FUNCIONAL					FALTANTE (00000)	PORTA (00000)	PORTA FREEZER (0	FALTANTE (00000)	2021-05-04 00:29:44	4	5	
4	LINHA 3 F3	CONSERTO ESTÉTICO					VEDAÇÃO DEFICIE	GAXETA (00000)	GAXETA FREEZER	VEDAÇÃO DEFICIE	2021-11-26 00:08:35	3	11	
5	LINHA 3 F3	CONSERTO ESTÉTICO					MAL FIXADO / SOLT	DOBRADIÇA (00000	PARAFUSO DOBRA	ESPANADO (00000	2022-06-15 00:24:42	15	6	
6	LINHA 6 F2	CONSERTO FUNCIO	Ilha 5	E.29.5			DANIFICADO (00000	INTERFACE (00000)	INTERFACE (00000)	DANIFICADO (00000	2021-05-27 00:30:48	27	5	
7	LINHA 6 F2	CONSERTO FUNCIONAL					MAL FIXADO / SOLT	REDE ELÉTRICA (0	CONECTOR FIAÇÃO	MAL FIXADO / SOLT	2022-11-28 00:20:45	28	11	
8	LINHA 3 F3	CONSERTO ESTÉTICO					AMASSADO (00000	PORTA (00000)	PAINEL FREEZER (0	AMASSADO NO MO	2022-10-31 00:14:18	31	10	
9	LINHA 1 F2	CONSERTO FUNCIONAL					REPROVAÇÃO MAN	CARROSSEL VÁCU	CARGA DE GÁS (00	FALHA SELAGEM (0	2022-02-02 00:33:57	2	2	
10	LINHA 6 F2	CONSERTO FUNCIONAL					DANIFICADO (00000	MÓDULO (00000)	MÓDULO (00000)	ERRO DE REGRAVA	2021-11-16 00:22:28	16	11	
11	LINHA 3 F2	CONSERTO PU					VAZAMENTO DE PL	CANTO PLÁSTICO (	ACABAMENTO CAN	SEM VEDAÇÃO	2021-04-26 00:41:20	26	4	
12	LINHA 3 F2	CONSERTO PU					VAZAMENTO DE PL	CANTO PLÁSTICO	CANTO PLÁSTICO I	SEM VEDAÇÃO	2022-03-28 00:14:04	28	3	
13	LINHA 2 F2	CONSERTO PU					VAZAMENTO DE PL	TRAVESSA (00000)	TRAVESSA INFERIC	VAZAMENTO DE PL	2021-06-18 00:44:26	18	6	
14	LINHA 3 F2	CONSERTO ESTÉTICO					AMASSADO (00000	PORTA (00000)	PORTA FREEZER (0	BATIDO NA LINHA (	2022-07-09 00:57:14	9	7	
15	LINHA 1 F2	CONSERTO FUNCIONAL					FRESTA / FOLGA (0	INTERFACE (00000)	INTERFACE (00000)	MAL FIXADO / SOLT	2022-02-09 00:51:23	9	2	
16	LINHA 3 F3	CONSERTO PU					AMASSADO (00000	CAIXA INTERNA FRI	FLANGE FREEZER I	CAIXA FRACA	2022-11-08 00:38:21	8	11	
17	LINHA 3 F3	CONSERTO FUNCIONAL					REPROVAÇÃO AUT	CARROSSEL VÁCU	PCU (00000)	NÃO ATINGIU NÍVEI	2022-02-04 00:26:37	4	2	
18	LINHA 6 F2	CONSERTO ESTÉTICO					RISCADO / SOVADC	GABINETE (00000)	GABINETE LATERA	RISCADO / SOVADC	2022-01-28 00:55:26	28	1	
19	LINHA 2 F2	CONSERTO PU					DESVIO DE PROCE	DESVIO DE PROCE	PARADA MANUTEN	PRODUTO SEM DEI	2022-02-07 00:21:30	7	2	
20	LINHA 6 F2	CONSERTO FUNCIONAL					DESVIO DE PROCE	ILUMINAÇÃO (0000	INTERRUPTOR (000	DESCONECTADO (0	2022-05-18 00:52:22	18	5	
21	LINHA 3 F2	CONSERTO PU					VAZAMENTO DE PL	CAIXA INTERNA (00	ALOJAMENTO INTE	DESLOCADO (0000	2021-04-24 00:28:19	24	4	
22	LINHA 3 F2	CONSERTO PU					VAZAMENTO DE PL	FUNDO POLIONDAS	FUNDO POLIONDAS	VAZAMENTO DE PL	2021-08-10 00:18:50	10	8	

Figura 24 – Estrutura dos dados antes da organização.
Fonte: Elaborado pelo autor.

Para dar início a etapa de organização, os dados já contidos em um arquivo CSV, foram importados na tabela TB_FPY_SKU_HORA em uma base de dados SQL Server, permitindo maior flexibilidade na análise e manipulação dos dados, obtendo a seguinte estrutura já em uma tabela, conforme figura 25

Resultados | Mensagens

	Name	Owner	Type	Created_datetime
1	TB_FPY_SKU_HORA	dbo	user table	2023-09-07 14:52:34.690

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	Linha	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
2	Conserto	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
3	IlhaGeradora	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
4	Posto	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
5	SerialNumber	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
6	Modelo	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
7	Motivo1	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
8	Motivo2	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
9	Motivo3	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
10	Motivo4	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
11	Data	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
12	Dia	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
13	Mes	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
14	Ano	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
15	DataCorrigida	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
16	HoraCheia	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
17	Turno	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
18	Agora	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
19	prodreal	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
20	Contagem	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
21	NumeracaoH...	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
22	SEMANA	varchar	no	50			yes	no	yes	Latin1_General_CI_AS

Figura 25 – Estrutura dos dados depois do início da organização.
Fonte: Elaborado pelo autor.

Durante a análise de dados foram identificadas algumas informações importantes para o processo de organização. A primeira delas é a data que aconteceu a falha na linha de montagem, essa informação está contida no campo “Data Corrigida”.

Outra informação é o SKU do produto, contido no campo “Modelo”. A sigla SKU vem do termo em inglês Stock Keeping Unit, ou Unidade de Manutenção de Estoque em português. O SKU é formado por um código utilizado pelos varejistas para identificarem as mercadorias armazenadas de modo rápido e simplificado. Ambas informações, podem ser vista em destaque na figura 26.

Resultados | Mensagens

	Name	Owner	Type	Created_datetime
1	TB_FPY_SKU_HORA	dbo	user table	2023-09-07 14:52:34.690

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	Linha	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
2	Conserto	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
3	IlhaGeradora	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
4	Posto	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
5	SerialNumber	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
6	Modelo	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
7	Motivo1	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
8	Motivo2	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
9	Motivo3	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
10	Motivo4	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
11	Data	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
12	Dia	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
13	Mes	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
14	Ano	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
15	DataCorrigida	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
16	HoraCheia	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
17	Turno	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
18	Agora	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
19	prodreal	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
20	Contagem	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
21	NumeracaoH...	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
22	SEMANA	varchar	no	50			yes	no	yes	Latin1_General_CI_AS

Figura 26 – Estrutura dos dados da DataCorrigida e Modelo.
Fonte: Elaborado pelo autor.

E, por último, a informação mais importante é o motivo da falha. Essa informação, vem dividida em 4 campos: Motivo1, Motivo2, Motivo3 e Motivo4. Juntos, esses campos, apresentam os motivos de falha desde o mais genérico até o mais detalhado, conforme figura 27.

Resultados | Mensagens

		Motivo1	Motivo2	Motivo3	Motivo4	Data	Dia	Mes	Ano
1	HKANA	AMASSADO (00000)	CAIXA INTERNA (00000)	CAIXA INTERNA FREEZER (00000)	CAIXA FRACA (00000)	2022-01-24 22:12:05.000000 UTC	25	1	2022
2	HKANA	DESVIO DE PROCESSO (00000)	DESVIO DE PROCESSO (00000)	PARADA MANUTENÇÃO (00000)	PRODUTO SEM DEFEITO (00000)	2022-04-01 22:54:06.000000 UTC	2	4	2022
3	ABANA	FALTANTE (00000)	COMPONENTES INTERNOS (00000)	TRILHO PRATELEIRA REFRIGERADOR	FALTANTE (00000)	2021-11-05 22:22:32.000000 UTC	6	11	2021
4	HBBNA	FALTANTE (00000)	DOBRADIÇA (00000)	CAPA DOBRADIÇA (00000)	FALTANTE (00000)	2021-05-19 22:21:51.433000 UTC	20	5	2021
5	AKBNA	RISCADO / SOVADO (00000)	CAIXA INTERNA (00000)	CAIXA INTERNA REFRIGERADOR SUPERIOR (00000)	RISCADO / SOVADO (00000)	2021-11-30 22:36:34.000000 UTC	1	12	2021
6	HBBNA	VAZAMENTO DE PU (00000)	GABINETE (00000)	BUCHA INJEÇÃO DE PU (00000)	DESALINHADO (00000)	2021-08-18 22:38:17.000000 UTC	19	8	2021
7	AKBNA	DESCONECTADO (00000)	CONTROL BOX (00000)	CONECTOR CONTROL BOX (00000)	DESCONECTADO (00000)	2022-10-11 22:25:52.433000 UTC	12	10	2022
8	JBANA	REPROVAÇÃO AUTOMÁTICA ...	VAZAMENTO EM BAIXA (99997)	PONTO DE SOLDA 13 (00000)	VAZAMENTO DE GÁS (00000)	2022-10-18 22:47:18.497000 UTC	19	10	2022
9	AKBNA	DESVIO DE PROCESSO (00000)	NÃO LÊ NO CARROSSEL	FALHA SENSOR DA BARREIRA	FALHA SENSOR DA BARREIRA	2022-05-19 22:56:36.000000 UTC	20	5	2022
10	AKBNA	AMASSADO (00000)	CAIXA INTERNA (00000)	CAIXA INTERNA FREEZER (00000)	AMASSADO NO MOLDE (00000)	2022-03-28 22:27:00.000000 UTC	29	3	2022
11	HBANA	REPROVAÇÃO MANUAL (999...	CONTROLE BAIXA 12 E 13 (99996)	ETIQUETA (00000)	ERRO DE LEITURA	2021-07-27 22:20:46.370000 UTC	28	7	2021
12	HBBNA	VAZAMENTO DE PU	TRAVESSA	TRAVESSA INTERMEDIÁRIA	FALHA NA VEDAÇÃO	2021-12-01 22:52:28.167000 UTC	2	12	2021
13	HKANA	REPROVAÇÃO AUTOMÁTICA ...	VAZAMENTO EM BAIXA (99997)	DESVIO DE PROCESSO (00000)	PRODUTO SEM STATUS (00000)	2022-12-14 22:30:54.720000 UTC	15	12	2022
14	AKANA	RISCADO / SOVADO (00000)	GABINETE (00000)	FLANGE FRONTAL FREEZER LD (00000)	SUJEIRA NO MOLDE PU (00000)	2022-12-20 22:51:03.000000 UTC	21	12	2022
15	HBBNA	AMASSADO (00000)	TRAVESSA	TRAVESSA INTERMEDIÁRIA (00000)	BATIDO NA PREPARAÇÃO	2021-08-03 22:55:28.000000 UTC	4	8	2021
16	AKANA	MAL FIXADO / SOLTO (00000)	INTERRUPTOR (00000)	INTERRUPTOR (00000)	TRAVADO / TRANCADO	2021-08-13 22:55:00.000000 UTC	14	8	2021
17	HKBNA	REPROVAÇÃO AUTOMÁTICA ...	VAZAMENTO EM BAIXA (99997)	MOTOR VENTILADOR (00000)	BATENTE FALTANTE (00000)	2021-09-24 22:55:33.863000 UTC	25	9	2021
18	HKBNA	RISCADO / SOVADO	TRAVESSA	TRAVESSA INFERIOR	DEFEITO NA MANOVIA	2022-09-28 22:40:29.453000 UTC	29	9	2022

Figura 27 – Estrutura dos motivos das falhas.
Fonte: Elaborado pelo autor.

Com essas 3 informações juntas (Data, Modelo e Motivos) é possível interpretar uma falha ocorrida em determinado dia para um determinado produto em uma linha de montagem industrial de refrigeradores.

Em uma próxima etapa do processo de organização, foi criado uma coluna na tabela de falhas chamada “Motivo”. Essa coluna irá transformar as 4 colunas

(Motivo1, Motivo2, Motivo3 e Motivo4) em uma única coluna, criando assim o grupo de falha, conforme ilustrado na figura 28.

Resultados | Mensagens

	Data	Dia	Mes	Ano	DataCorrigida	HoraCheia	Turno	Agora	prodreal	Contagem	NumeracaoHorario	SEMANA	MOTIVO
1	2022-01-24 22:12:05.000000 UTC	25	1	2022	2022-01-25	22	T6	0	410	1	1	4	AMASSADO PORTA
2	2022-04-01 22:54:06.000000 UTC	2	4	2022	2022-04-02	22	T6	0	506	1	1	13	PRODUTO SEM DEFEITO
3	2021-11-05 22:22:32.000000 UTC	6	11	2021	2021-11-06	22	T6	0		1	1		COMPONENTES INTERNOS DANIFICADOS
4	2021-05-19 22:21:51.433000 UTC	20	5	2021	2021-05-20	22	T6	0		1	1		COMPONENTES INTERNOS DANIFICADOS
5	2021-11-30 22:36:34.000000 UTC	1	12	2021	2021-12-01	22	T6	0		1	1		RISCADO/SOVADO
6	2021-08-18 22:38:17.000000 UTC	19	8	2021	2021-08-19	22	T6	0		1	1		VAZAMENTO DE PU GABINETE
7	2022-10-11 22:25:52.433000 UTC	12	10	2022	2022-10-12	22	T6	0	585	1	1	41	FALHA ELÉTRICA/ELETRÔNICA
8	2022-10-18 22:47:18.497000 UTC	19	10	2022	2022-10-19	22	T6	0	665	1	1	42	VAZAMENTO DE GÁS
9	2022-05-19 22:56:36.000000 UTC	20	5	2022	2022-05-20	22	T6	0	37	1	1	20	COMPONENTES INTERNOS DANIFICADOS
10	2022-03-28 22:27:00.000000 UTC	29	3	2022	2022-03-29	22	T6	0	444	1	1	13	AMASSADO PORTA
11	2021-07-27 22:20:46.370000 UTC	28	7	2021	2021-07-28	22	T6	0		1	1		IDENTIFICAÇÃO AUSENTE
12	2021-12-01 22:52:28.167000 UTC	2	12	2021	2021-12-02	22	T6	0		1	1		VAZAMENTO DE PU TRAVESSA
13	2022-12-14 22:30:54.720000 UTC	15	12	2022	2022-12-15	22	T6	0	672	1	1	50	VAZAMENTO DE PU
14	2022-12-20 22:51:03.000000 UTC	21	12	2022	2022-12-21	22	T6	0	670	1	1	51	RISCADO/SOVADO
15	2021-08-03 22:55:28.000000 UTC	4	8	2021	2021-08-04	22	T6	0		1	1		AMASSADO (00000)TRAVESSATRAVESSA INTERMEDIÁRIA (...
16	2021-08-13 22:55:00.000000 UTC	14	8	2021	2021-08-14	22	T6	0		1	1		COMPONENTE MAL FIXADO / SOLTO
17	2021-09-24 22:55:33.863000 UTC	25	9	2021	2021-09-25	22	T6	0		1	1		FALHA ELÉTRICA/ELETRÔNICA
18	2022-09-28 22:40:29.453000 UTC	29	9	2022	2022-09-29	22	T6	0	665	1	1	39	RISCADO/SOVADO

Figura 28 – Estrutura dos dados depois da organização dos motivos.
Fonte: Elaborado pelo autor.

Para esse processo de organização foram criados 20 grupos de falha, onde podemos entender cada um deles, conforme tabela 1:

Grupo de falha	Descrição da falha
VAZAMENTO_PU_CANTO_PLASTICO	Responsável pela falha de isolamento na região dos cantos superiores da porta do refrigerador
VAZAMENTO_PU_TRAVESSA	Responsável pela falha de isolamento na região que divide o freezer e o refrigerador
VAZAMENTO_PU_GABINETE	Responsável pela falha de isolamento nas regiões laterais do refrigerador
VAZAMENTO_PU_FUNDO_POLIONDAS	Responsável pela falha de isolamento entre a parte metálica e a parte plástica nas laterais e no fundo
VAZAMENTO_PU_CAIXA_INTERNA	Responsável pela falha de isolamento na região de algum canto na parte interna do refrigerador ou freezer
VAZAMENTO_PU	Responsável pela falha de isolamento em alguma região do refrigerador não classificada
AMASSADO_PORTA	Responsável por amassados na parte externa da porta do refrigerador ou freezer

RISCADO_SOVADO	Responsável por riscos ou ondulações nas portas ou gabinete do refrigerador
AMASSADO_GABINETE	Responsável por amassados na parte externa do gabinete do refrigerador
IDENTIFICACAO_AUSENTE	Responsável pela ausência do código do produto na etiqueta colada dentro do refrigerador
AVARIA_LOGISTICA	Responsável por alguma irregularidade no produto ou embalagem na hora do transporte para estocagem
LIMPEZA_INADEQUADA	Responsável por algum problema na limpeza do produto após sua montagem, antes de ser embalado
FALHA_PINTURA	Responsável por algum risco ou falha na pintura do gabinete ou portas do refrigerador
COMPONENTE_MAL_FIXADO_SOLTO	Responsável por algum item dentro do refrigerador mal fixado ou solto
FALHA_ELETRICA_ELETRONICA	Responsável por qualquer falha elétrica ou eletrônica do refrigerador
VAZAMENTO_GAS	Responsável por alguma tubulação de gás do refrigerador estar com vazamento
AUSENCIA_COMPONENTES	Responsável pela falta de peças ou itens internos e externos do refrigerador
COMPONENTES_INTERNOS_DANIFICADOS	Responsável por algum item dentro do refrigerador quebrado ou danificado
FALHA_VACUO	Responsável por alguma falha no processo de aplicação de vácuo do refrigerador, antes da carga de gás
VEDACAO_DEFICIENTE	Responsável por algum problema de vedação das portas do refrigerador

Tabela 1 – Descrição dos grupos de falhas.
Fonte: Elaborado pelo autor.

Além disso, os grupos de falha são divididos em 3 grandes categorias e cada um com uma forma de identificação da falha na linha, seja humana ou através de máquinas, conforme a tabela 2:

Categoria de falha	Grupo de falha	Forma de identificação da falha na linha
Vazamento de PU	VAZAMENTO_PU_CANTO_PLASTICO	Humana
	VAZAMENTO_PU_TRAVESSA	Humana
	VAZAMENTO_PU_GABINETE	Humana
	VAZAMENTO_PU_FUNDO_POLIONDAS	Humana
	VAZAMENTO_PU_CAIXA_INTERNA	Humana
	VAZAMENTO_PU	Humana
Estético	AMASSADO_PORTA	Humana
	RISCADO_SOVADO	Humana
	AMASSADO_GABINETE	Humana
	IDENTIFICACAO_AUSENTE	Humana
	AVARIA_LOGISTICA	Humana
	LIMPEZA_INADEQUADA	Humana
	FALHA_PINTURA	Humana
	COMPONENTE_MAL_FIXADO_SOLTO	Humana
	FALHA_ELETRICA_ELETRONICA	Máquina
	VAZAMENTO_GAS	Máquina

Funcional	AUSENCIA_COMPONENTES	Humana
	COMPONENTES_INTERNOS_DANIFICADOS	Humana
	FALHA_VACUO	Máquina
	VEDACAO_DEFICIENTE	Máquina

Tabela 2 – Descrição das categorias de falhas.
Fonte: Elaborado pelo autor.

Após a criação e atualização dos registros de falhas, foi criado uma tabela no banco de dados com objetivo de sumarizar quantitativamente o número de falhas. Essa nova tabela é responsável por armazenar a quantidade de falhas por grupo que ocorreram por dia, agrupado por data (diária) e SKU do produto.

Pode-se observar a modelagem da tabela na figura 29:

Resultados | Mensagens

	Name	Owner	Type	Created_datetime
1	TB_FALHA_MONTAGEM	dbo	user table	2023-09-12 15:38:34.317

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	SKU	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
2	DATA_CORRIGIDA	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
3	VAZAMENTO_PU	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
4	VAZAMENTO_PU_CANTO_PLASTICO	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
5	VAZAMENTO_PU_TRAVESSA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
6	VAZAMENTO_PU_GABINETE	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
7	VAZAMENTO_PU_FUNDO_POLIONDAS	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
8	VAZAMENTO_PU_CAIXA_INTERNA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
9	AMASSADO_PORTA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
10	RISCADO_SOVADO	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
11	AMASSADO_GABINETE	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
12	IDENTIFICACAO_AUSENTE	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
13	AVARIA_LOGISTICA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
14	LIMPEZA_INADEQUADA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
15	FALHA_PINTURA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
16	COMPONENTE_MAL_FIXADO_SOLTO	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
17	FALHA_ELETRICA_ELETRONICA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
18	VAZAMENTO_GAS	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
19	AUSENCIA_COMPONENTES	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
20	COMPONENTES_INTERNOS_DANIFIC...	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
21	FALHA_VACUO	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
22	VEDACAO_DEFICIENTE	int	no	4	10	0	yes	(n/a)	(n/a)	NULL

Figura 29 – Modelagem da tabela TB_FALHA_MONTAGEM.
Fonte: Elaborado pelo autor.

A tabela TB_FALHA_MONTAGEM foi criada e planejada para obter a melhor organização dos dados. Após a sua criação foi necessário realizar a ingestão dos dados da tabela TB_FPY_SKU_HORA para a nova tabela TB_FALHA_MONTAGEM. Essa ingestão foi realizada com os seguintes passos de programação SQL:

Criação de 20 tabelas temporárias representando cada um dos grupos de falha. O objetivo é recuperar, organizar e sumarizar as falhas sobre cada grupo, por data e produto.

```
SELECT Modelo
      ,DataCorrigida
      ,QTD = COUNT(*)
INTO #TMP_VAZAMENTO_PU
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'VAZAMENTO DE PU'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_VAZAMENTO_PU_CANTO_PLASTICO
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'VAZAMENTO DE PU CANTO PLASTICO'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_VAZAMENTO_PU_TRAVESSA
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'VAZAMENTO DE PU TRAVESSA'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_VAZAMENTO_GABINETE
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'VAZAMENTO DE PU GABINETE'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
```

```
INTO #TMP_VAZAMENTO_PU_FUNDO_POLIONDAS
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'VAZAMENTO DE PU FUNDO POLIONDAS'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_VAZAMENTO_PU_CAIXA_INTERNA
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'VAZAMENTO DE PU CAIXA INTERNA'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_AMASSADO_PORTA
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'AMASSADO PORTA'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_RISCADO_SOVADO
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'RISCADO/SOVADO'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_AMASSADO_GABINETE
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'AMASSADO GABINETE'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
```

```
        ,DataCorrigida
            ,QTD = COUNT(*)
INTO #TMP_IDENTIFICACAO_AUSENTE
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'IDENTIFICACAO AUSENTE'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_AVARIA_LOGISTICA
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'AVARIA LOGISTICA'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_LIMPEZA_INADEQUADA
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'LIMPEZA INADEQUADA'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_FALHA_PINTURA
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'FALHA DE PINTURA'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_COMPONENTE_MAL_FIXADO_SOLTO
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'COMPONENTE MAL FIXADO / SOLTO'
GROUP BY Modelo
        ,DataCorrigida
```

```
SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_FALHA_ELETRICA_ELETRONICA
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'FALHA ELETRICA/ELETRONICA'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_VAZAMENTO_GAS
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'VAZAMENTO DE GAS'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_AUSENCIA_COMPONENTES
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'AUSENCIA DE COMPONENTES'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_COMPONENTES_INTERNOS_DANIFICADOS
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'COMPONENTES INTERNOS DANIFICADOS'
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_FALHA_VACUO
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'FALHA DE VACUO'
```

```
GROUP BY Modelo
        ,DataCorrigida

SELECT Modelo
      ,DataCorrigida
          ,QTD = COUNT(*)
INTO #TMP_VEDACAO_DEFICIENTE
FROM TB_FPY_SKU_HORA
WHERE MOTIVO = 'VEDACAO DEFICIENTE'
GROUP BY Modelo
        ,DataCorrigida
```

Após a criação das 20 tabelas temporárias, foi criado uma única tabela, para reunir todos os dados normalizados e sumarizados em uma única tabela para posterior atualização na modelagem planejada.

```
SELECT Modelo
      ,DataCorrigida
INTO #TMP_ALL_MOTIVOS
FROM #TMP_VAZAMENTO_PU
WHERE 1 = 2

SELECT *
FROM #TMP_ALL_MOTIVOS

INSERT #TMP_ALL_MOTIVOS(
  Modelo
 ,DataCorrigida
)
SELECT Modelo
      ,DataCorrigida
FROM #TMP_VAZAMENTO_PU
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_VAZAMENTO_PU_CANTO_PLASTICO
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
```

```
        ,DataCorrigida
FROM #TMP_VAZAMENTO_PU_TRAVESSA
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_VAZAMENTO_GABINETE
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_VAZAMENTO_PU_FUNDO_POLIONDAS
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_VAZAMENTO_PU_CAIXA_INTERNA
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_AMASSADO_PORTA
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_RISCADO_SOVADO
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_AMASSADO_GABINETE
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
```

```
FROM #TMP_IDENTIFICACAO_AUSENTE
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_AVARIA_LOGISTICA
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_LIMPEZA_INADEQUADA
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_FALHA_PINTURA
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_COMPONENTE_MAL_FIXADO_SOLTO
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_FALHA_ELETRICA_ELETRONICA
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_VAZAMENTO_GAS
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_AUSENCIA_COMPONENTES
```

```
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_COMPONENTES_INTERNOS_DANIFICADOS
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_FALHA_VACUO
GROUP BY Modelo
        ,DataCorrigida
UNION
SELECT Modelo
      ,DataCorrigida
FROM #TMP_VEDACAO_DEFICIENTE
GROUP BY Modelo
        ,DataCorrigida
```

Por último, para cada grupo de falha, foi atualizado a quantidade geral de falhas para cada dia e cada produto.

```
INSERT TB_FALHA_MONTAGEM(
  SKU
 ,DATA_CORRIGIDA
 ,VAZAMENTO_PU
 ,VAZAMENTO_PU_CANTO_PLASTICO
 ,VAZAMENTO_PU_TRAVESSA
 ,VAZAMENTO_PU_GABINETE
 ,VAZAMENTO_PU_FUNDO_POLIONDAS
 ,VAZAMENTO_PU_CAIXA_INTERNA
 ,AMASSADO_PORTA
 ,RISCADO_SOVADO
 ,AMASSADO_GABINETE
 ,IDENTIFICACAO_AUSENTE
 ,AVARIA_LOGISTICA
 ,LIMPEZA_INADEQUADA
 ,FALHA_PINTURA
 ,COMPONENTE_MAL_FIXADO_SOLTO
 ,FALHA_ELETRICA_ELETRONICA
 ,VAZAMENTO_GAS
 ,AUSENCIA_COMPONENTES
 ,COMPONENTES_INTERNOS_DANIFICADOS
```

```
       ,FALHA_VACUO
       ,VEDACAO_DEFICIENTE
      )
      SELECT Modelo
            ,DataCorrigida
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
                ,0
      FROM #TMP_ALL_MOTIVOS
      GROUP BY Modelo
              ,DataCorrigida

      UPDATE A
      SET A.VAZAMENTO_PU = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_VAZAMENTO_PU B ON (A.SKU
= B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.VAZAMENTO_PU_CANTO_PLASTICO = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN
#TMP_VAZAMENTO_PU_CANTO_PLASTICO B ON (A.SKU = B.Modelo AND
A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.VAZAMENTO_PU_TRAVESSA = B.QTD
```

```
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_VAZAMENTO_PU_TRAVESSA B
ON (A.SKU = B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.VAZAMENTO_PU_GABINETE = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_VAZAMENTO_GABINETE B ON
(A.SKU = B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.VAZAMENTO_PU_FUNDO_POLIONDAS = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN
#TMP_VAZAMENTO_PU_FUNDO_POLIONDAS B ON (A.SKU = B.Modelo AND
A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.VAZAMENTO_PU_CAIXA_INTERNA = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN
#TMP_VAZAMENTO_PU_CAIXA_INTERNA B ON (A.SKU = B.Modelo AND
A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.AMASSADO_PORTA = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_AMASSADO_PORTA B ON
(A.SKU = B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.RISCADO_SOVADO = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_RISCADO_SOVADO B ON
(A.SKU = B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.AMASSADO_GABINETE = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_AMASSADO_GABINETE B ON
(A.SKU = B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.IDENTIFICACAO_AUSENTE = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_IDENTIFICACAO_AUSENTE B
ON (A.SKU = B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.AVARIA_LOGISTICA = B.QTD
```

```
      FROM  TB_FALHA_MONTAGEM  A  INNER  JOIN  #TMP_AVARIA_LOGISTICA  B  ON
(A.SKU = B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.LIMPEZA_INADEQUADA = B.QTD
      FROM  TB_FALHA_MONTAGEM  A  INNER  JOIN  #TMP_LIMPEZA_INADEQUADA  B  ON
(A.SKU = B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.FALHA_PINTURA = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_FALHA_PINTURA B ON (A.SKU
= B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.COMPONENTE_MAL_FIXADO_SOLTO = B.QTD
      FROM         TB_FALHA_MONTAGEM         A         INNER         JOIN
#TMP_COMPONENTE_MAL_FIXADO_SOLTO    B    ON    (A.SKU    =    B.Modelo    AND
A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.FALHA_ELETRICA_ELETRONICA = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_FALHA_ELETRICA_ELETRONICA
B ON (A.SKU = B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.VAZAMENTO_GAS = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_VAZAMENTO_GAS B ON (A.SKU
= B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.AUSENCIA_COMPONENTES = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_AUSENCIA_COMPONENTES B ON
(A.SKU = B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.COMPONENTES_INTERNOS_DANIFICADOS = B.QTD
      FROM         TB_FALHA_MONTAGEM         A         INNER         JOIN
#TMP_COMPONENTES_INTERNOS_DANIFICADOS   B   ON   (A.SKU   =   B.Modelo   AND
A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.FALHA_VACUO = B.QTD
```

```
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_FALHA_VACUO B ON (A.SKU =
B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)

      UPDATE A
      SET A.VEDACAO_DEFICIENTE = B.QTD
      FROM TB_FALHA_MONTAGEM A INNER JOIN #TMP_VEDACAO_DEFICIENTE B ON
(A.SKU = B.Modelo AND A.DATA_CORRIGIDA = B.DataCorrigida)
```

Obtendo assim, a tabela TB_FALHA_MONTAGEM normalizada para cada produto contendo o quantitativo de falhas por dia, separado por grupo de falhas, conforme figura 30:

Resultados | Mensagens

SKU	DATA_CORRIGIDA	VAZAMENTO_PU	VAZAMENTO_PU_CANTO_PLASTICO	VAZAMENTO_PU_TRAVESSA	VAZAMENTO_PU_GABINETE	VAZAMENTO_PU_FUNDO_POLIONDAS	VAZAMENTO_
	2021-11-13	0	0	0	0	0	0
	2021-11-16	2	1	0	0	0	0
	2021-11-19	0	0	0	0	0	0
	2021-11-20	1	0	0	0	0	0
	2021-11-22	0	0	1	0	0	0
	2021-11-23	0	0	0	0	0	0
	2021-11-25	1	1	1	0	0	0
	2021-12-07	2	1	0	0	0	0
	2021-12-08	0	0	0	0	0	0
	2021-12-14	1	2	0	0	0	0
	2021-12-15	2	0	2	1	0	0
	2021-12-16	2	0	0	0	0	0
	2021-12-17	0	0	0	0	0	0
	2022-01-13	0	4	0	0	0	0
	2022-01-14	0	0	0	0	0	0
	2022-01-15	0	1	0	0	0	0
	2022-01-17	0	3	1	0	0	0
	2022-01-21	0	5	0	0	0	5
	2022-01-24	0	1	0	0	0	1
	2022-01-25	1	0	0	0	0	0
	2022-01-26	6	1	0	0	0	0
	2022-01-27	2	7	1	1	0	0

Figura 30 – Estrutura da tabela TB_FALHA_MONTAGEM normalizada.
Fonte: Elaborado pelo autor.

2.2 ANÁLISE E ORGANIZAÇÃO DOS DADOS DE ORDENS DE SERVIÇO DE REFRIGERADORES

Para o desenvolvimento dessa etapa, os dados foram extraídos do sistema geral de OS da empresa, onde ele possui diversas informações sobre o produto, cliente etc. Foi extraído toda a base de dados de OS referentes a refrigeradores no período de fevereiro de 2021 até junho de 2023. Esse período compreende em um total de 64.076 registros de itens de ordem de serviço.

Essa etapa compreende as seguintes partes do processo (em vermelho), conforme ilustrado na figura 31:

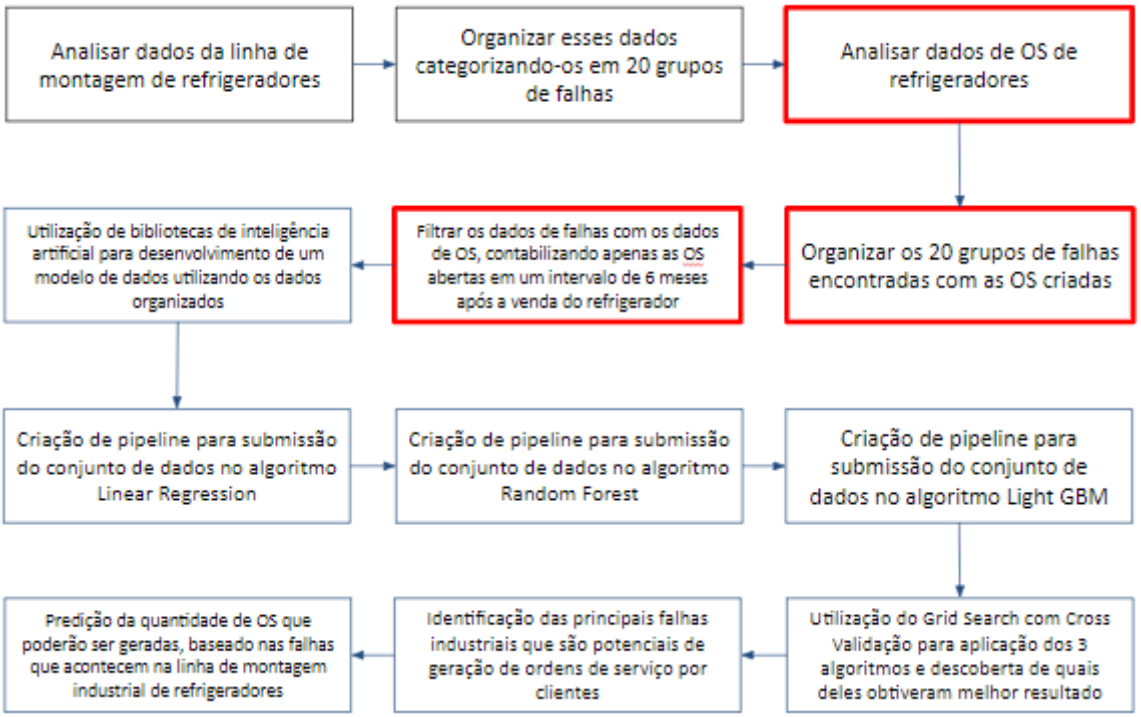

Figura 31 – Etapas desenvolvidas na segunda parte.
Fonte: Elaborado pelo autor.

Todo esse conteúdo de dados foi armazenado em um arquivo CSV, onde podemos observar a estrutura dos dados obtidos antes da organização, utilizando uma planilha Excel, conforme figura 32:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	NrOS	OS_Valida_GSIR	Item_OS	OS_Incidente	Desc_Tipo_OS	OS_Casada	Serie	Produto	Cod_Peca	Desc_Peca	Desc_Peca_Agrupac	Desc_Peca_Agrupac	Desc_FCTA	Desc_FCT
2	8002614414	0	1	Incidencia	NORMAL	Unica				VÁLVULA DO DREN	SISTEMA DEGELO	VALVE (5710)	CONSERTADA	REPAIRE
3	8002635002	0	2	Incidencia	NORMAL	Casada				CONTROLE ELETR	CONTROLE ELETR	CONTROL BOARD /	TROCADA	CHANGE
4	8002635002	0	3	Incidencia	NORMAL	Casada				PLACA ELETRONIC	CONTROLE ELETR	CONTROL BOARD /	TROCADA	CHANGE
5	8002635002	0	1	Incidencia	NORMAL	Casada				SENSOR TEMP DO	KIT SENSOR P CAM	SENSOR (6910)	TROCADA	CHANGE
6	8002707219	0	2	Incidencia	NORMAL	Casada				INTERRUPTOR DA	INTERRUPTOR BIPI	IGNITION - SWITCH	TROCADA	CHANGE
7	8002707219	0	1	Incidencia	NORMAL	Casada				LAMPADA LED KIT	LED	PILOT LIGHT / LED	TROCADA	CHANGE
8	8002873378	1	1	Incidencia	NORMAL	Unica				GABINETE EMBALA	GABINETE	CABINET (1800)	CONSERTADA	REPAIRE
9	8002931173	0	1	Incidencia	NORMAL	Casada				CARGA DE GAS R1	RUS	REFRIGERANT / GA	TROCADA	CHANGE
10	8002931173	0	2	Incidencia	NORMAL	Casada				FILTRO SECADOR	DESCONSIDERAR	FILTER (7630)	TROCADA	CHANGE
11	8002938980	0	1	Incidencia	NORMAL	Unica				SENSOR TEMP DO	KIT SENSOR P CAM	SENSOR (6910)	TROCADA	CHANGE
12	8002961410	0	1	Incidencia	NORMAL	Unica				MOTOR VENTILAD	MOTOR VENTILAD	MOTOR-FAN, CIRCU	TROCADA	CHANGE
13	8002962925	0	1	Incidencia	NORMAL	Unica				MOTOR VENTILAD	MOTOR VENTILAD	MOTOR-FAN, CIRCU	CONSERTADA	REPAIRE
14	8002963654	0	1	Incidencia	NORMAL	Unica				CALHA DEGELO EV	SISTEMA DEGELO	DRAIN-TUBE/-PIPE/	CONSERTADA	REPAIRE
15	8002965449	1	4	Incidencia	NORMAL	Casada				COMP EMB 513701	COMPRESSOR	COMPRESSOR (402	TROCADA	CHANGE
16	8002965449	1	3	Incidencia	NORMAL	Casada				MOTOR DO REFRIG	MOTOR VENTILAD	MOTOR-FAN, CIRCU	TROCADA	CHANGE
17	8002965449	1	1	Incidencia	NORMAL	Casada				PE NIVELADOR CIN	PE CJ	FOOT/SPACER (185	TROCADA	CHANGE
18	8002965449	1	2	Incidencia	NORMAL	Casada				RODAPE CINZA BAL	PE CJ	FOOT/SPACER (185	TROCADA	CHANGE
19	8002968007	0	1	Reincidencia	LGI	Unica				SERVICO	SERVICO	SERVICE RELATED	(9970)	
20	8002984690	1	1	Incidencia	NORMAL	Unica				PORTA EMB EXT F	PORTA FREEZER	DOOR FREEZER (10	TROCADA	CHANGE
21	8002986416	0	1	Incidencia	NORMAL	Casada				COMP EMB EM2P7	COMPRESSOR	COMPRESSOR (402	TROCADA	CHANGE
22	8002986416	0	5	Incidencia	NORMAL	Casada				VARETA SOLDA 15	DESCONSIDERAR	SUPPORT/BRACKE	TROCADA	CHANGE

Figura 32 – Estrutura dos dados de ordens de serviço antes da organização.
Fonte: Elaborado pelo autor.

Para dar início a etapa de organização, os dados já contidos em um arquivo CSV, foram importados na tabela TB_OS em uma base de dados SQL Server,

permitindo maior flexibilidade na análise e manipulação dos dados, obtendo a seguinte estrutura já em uma tabela, conforme figura 33:

Resultados | Mensagens

	Name	Owner	Type	Created_datetime
1	TB_OS	dbo	user table	2023-09-07 12:19:12.110

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	NrOS	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
2	OS_Valida_GSIR	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
3	Item_OS	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
4	OS_Incidente	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
5	Desc_Tipo_OS	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
6	OS_Casada	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
7	Serie	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
8	Produto	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
9	Cod_Peca	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
10	Desc_Peca	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
11	Desc_Peca_Ag...	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
12	Desc_Peca_Ag...	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
13	Desc_FCTA	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
14	Desc_FCTA_EN	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
15	Desc_Tipo_Oc...	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
16	Desc_Tipo_Oc...	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
17	Desc_Def_Rec...	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS
18	Desc_Def_Rec...	varchar	no	-1			yes	no	yes	Latin1_General_CI_AS

Figura 33 – Estrutura dos dados de ordens de serviço depois do início da organização.
Fonte: Elaborado pelo autor.

Foi extraído dessa tabela as informações referentes a dois modelos específicos de refrigeradores para que possamos submeter no modelo de dados posteriormente. Essas informações foram armazenadas em 2 tabelas temporárias para cada modelo, conforme código abaixo:

```
SELECT *
INTO TB_OS_MODELO 01
FROM TB_OS
WHERE PRODUTO LIKE 'MODELO 01%'

SELECT *
INTO TB_OS_MODELO 02
FROM TB_OS
WHERE PRODUTO LIKE 'MODELO 02%'
```

Para cada uma das tabelas foi criado uma nova coluna DATA_ANALISE. Essa coluna possui uma data com acréscimo de 6 meses sobre a data da fabricação do produto, localizada no campo DATA_FABRICACAO. Essa informação foi necessária para os próximos passos da organização dos dados.

```
ALTER TABLE TB_OS_MODELO 01
ADD DATA_ANALISE VARCHAR(50) NULL

UPDATE TB_OS_MODELO 01
SET DATA_ANALISE = FORMAT(DATEADD(MONTH, 6,
CAST(LTRIM(RTRIM(DATA_FABRICACAO)) AS date)), 'yyyy-MM-dd')
WHERE LEN(LTRIM(RTRIM(DATA_FABRICACAO))) > 9
```

```
        AND DATA_FABRICACAO BETWEEN '2021-02-11' AND '2023-06-13'

      ALTER TABLE TB_OS_MODELO 02
      ADD DATA_ANALISE VARCHAR(50) NULL

      UPDATE TB_OS_MODELO 02
      SET       DATA_ANALISE       =       FORMAT(DATEADD(MONTH,       6,
CAST(LTRIM(RTRIM(DATA_FABRICACAO)) AS date)), 'yyyy-MM-dd')
      WHERE LEN(LTRIM(RTRIM(DATA_FABRICACAO))) > 9
        AND DATA_FABRICACAO BETWEEN '2021-02-11' AND '2023-06-13'
```

Em um próximo passo, foram considerados apenas os dados de OS que foram criadas no intervalo de 6 meses após a fabricação do refrigerador. Isso se faz necessário para realmente identificar as potenciais correlações com as falhas encontradas na linha de montagem.

```
      SELECT NrOS
            ,Produto = SUBSTRING(Produto, 1, 5)
            ,DATA_FABRICACAO
      INTO TB_OS_MODELO 01_QTD
      FROM TB_OS_MODELO 01
      WHERE Dt_Chamado <= DATA_ANALISE
      GROUP BY NrOS
              ,SUBSTRING(Produto, 1, 5)
              ,DATA_FABRICACAO
      ORDER BY DATA_FABRICACAO

      SELECT NrOS
            ,Produto = SUBSTRING(Produto, 1, 5)
            ,DATA_FABRICACAO
      INTO TB_OS_MODELO 02_QTD
      FROM TB_OS_MODELO 02
      WHERE Dt_Chamado <= DATA_ANALISE
      GROUP BY NrOS
              ,SUBSTRING(Produto, 1, 5)
              ,DATA_FABRICACAO
      ORDER BY DATA_FABRICACAO
```

Após isso, os dados foram sumarizados por data de fabricação e modelos de produto.

```
      SELECT DATA_FABRICACAO
            ,Produto = SUBSTRING(Produto, 1, 5)
```

```
        ,QTD = COUNT(*)
INTO TB_OS_MODELO 01_QTD2
FROM TB_OS_MODELO 01_QTD
GROUP BY DATA_FABRICACAO
        ,SUBSTRING(Produto, 1, 5)
ORDER BY DATA_FABRICACAO

SELECT DATA_FABRICACAO
        ,Produto = SUBSTRING(Produto, 1, 5)
        ,QTD = COUNT(*)
INTO TB_OS_MODELO 02_QTD2
FROM TB_OS_MODELO 02_QTD
GROUP BY DATA_FABRICACAO
        ,SUBSTRING(Produto, 1, 5)
ORDER BY DATA_FABRICACAO
```

Seguindo no processo de organização, após os dados de OS estarem organizados em dois modelos de refrigerador, chegou o momento de contabilizar a quantidade de OS geradas por dia e por modelo de produto versus a quantidade de falhas geradas na linha de montagem.

Para isso, foi criado a coluna “QTD_OS” na tabela TB_FALHA_MONTAGEM.

```
ALTER TABLE TB_FALHA_MONTAGEM
ADD QTD_OS INT NULL
```

Gerando a seguinte alteração na tabela, conforme figura 34:

Resultados | Mensagens

	Name	Owner	Type	Created_datetime
1	TB_FALHA_MONTAGEM	dbo	user table	2023-09-12 15:38:34.317

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	SKU	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
2	DATA_CORRIGIDA	varchar	no	50			yes	no	yes	Latin1_General_CI_AS
3	VAZAMENTO_PU	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
4	VAZAMENTO_PU_CANTO_PLASTICO	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
5	VAZAMENTO_PU_TRAVESSA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
6	VAZAMENTO_PU_GABINETE	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
7	VAZAMENTO_PU_FUNDO_POLIONDAS	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
8	VAZAMENTO_PU_CAIXA_INTERNA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
9	AMASSADO_PORTA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
10	RISCADO_SOVADO	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
11	AMASSADO_GABINETE	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
12	IDENTIFICACAO_AUSENTE	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
13	AVARIA_LOGISTICA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
14	LIMPEZA_INADEQUADA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
15	FALHA_PINTURA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
16	COMPONENTE_MAL_FIXADO_SOLTO	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
17	FALHA_ELETRICA_ELETRONICA	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
18	VAZAMENTO_GAS	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
19	AUSENCIA_COMPONENTES	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
20	COMPONENTES_INTERNOS_DANIFIC...	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
21	FALHA_VACUO	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
22	VEDACAO_DEFICIENTE	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
23	QTD_OS	int	no	4	10	0	yes	(n/a)	(n/a)	NULL

Figura 34 – Estrutura dos dados de ordens de serviço durante a organização.
Fonte: Elaborado pelo autor.

Após isso, foi atualizada a quantidade de OS gerada no intervalo de 6 meses após sua fabricação, para todos os produtos que tiveram falhas na linha de montagem industrial.

```
SELECT *
INTO TB_FALHA_MONTAGEM2
FROM TB_FALHA_MONTAGEM
WHERE 1 = 2

INSERT TB_FALHA_MONTAGEM2
SELECT SUBSTRING(SKU, 1, 5)
      ,DATA_CORRIGIDA
,SUM(VAZAMENTO_PU)
,SUM(VAZAMENTO_PU_CANTO_PLASTICO)
,SUM(VAZAMENTO_PU_TRAVESSA)
,SUM(VAZAMENTO_PU_GABINETE)
,SUM(VAZAMENTO_PU_FUNDO_POLIONDAS)
,SUM(VAZAMENTO_PU_CAIXA_INTERNA)
,SUM(AMASSADO_PORTA)
,SUM(RISCADO_SOVADO)
,SUM(AMASSADO_GABINETE)
,SUM(IDENTIFICACAO_AUSENTE)
,SUM(AVARIA_LOGISTICA)
,SUM(LIMPEZA_INADEQUADA)
,SUM(FALHA_PINTURA)
,SUM(COMPONENTE_MAL_FIXADO_SOLTO)
,SUM(FALHA_ELETRICA_ELETRONICA)
,SUM(VAZAMENTO_GAS)
,SUM(AUSENCIA_COMPONENTES)
,SUM(COMPONENTES_INTERNOS_DANIFICADOS)
,SUM(FALHA_VACUO)
,SUM(VEDACAO_DEFICIENTE)
,NULL
FROM TB_FALHA_MONTAGEM
GROUP BY SUBSTRING(SKU, 1, 5), DATA_CORRIGIDA

UPDATE TB_FALHA_MONTAGEM2
SET QTD_OS = 0

UPDATE A
SET A.QTD_OS = B.QTD
```

```
      FROM  TB_FALHA_MONTAGEM2  A  INNER  JOIN  TB_OS_MODELO  01_QTD2  B  ON
(A.DATA_CORRIGIDA = B.DATA_FABRICACAO) AND (A.SKU = B.Produto)

      UPDATE A
      SET A.QTD_OS = B.QTD
      FROM  TB_FALHA_MONTAGEM2  A  INNER  JOIN  TB_OS_MODELO  02_QTD2  B  ON
(A.DATA_CORRIGIDA = B.DATA_FABRICACAO) AND (A.SKU = B.Produto)
```

Tendo como resultado a seguinte modelagem, finalizando o processo de organização de dados, para os produtos MODELO 01, conforme figuras 35 e 36:

Resultados | Mensagens

	SKU	DATA_CORRIGIDA	VAZAMENTO_PU	VAZAMENTO_PU_CANTO_PLASTICO	VAZAMENTO_PU_TRAVESSA	VAZAMENTO_PU_GABINETE	VAZAMENTO_PU_FUNDO_POLIONDAS	VAZAMENTO_PU_CA
178		2022-04-08	1	4	9	1	0	0
179		2022-04-09	2	1	2	0	0	0
180		2022-04-11	2	0	6	0	2	0
181		2022-04-12	0	0	0	0	0	1
182		2022-04-13	5	0	0	0	0	0
183		2022-04-14	1	2	2	0	0	0
184		2022-04-15	2	2	2	0	0	0
185		2022-04-19	2	1	2	0	0	1
186		2022-04-20	0	0	0	0	0	1
187		2022-04-21	2	3	1	0	0	1
188		2022-04-22	4	2	1	0	1	1
189		2022-04-23	1	4	2	0	0	0
190		2022-04-25	1	0	0	0	0	2
191		2022-04-26	1	1	0	0	0	0
192		2022-04-27	1	1	1	0	0	0
193		2022-04-28	1	4	1	0	0	0
194		2022-04-29	0	3	0	1	1	0
195		2022-04-30	4	3	0	0	0	0
196		2022-05-02	0	1	0	0	1	1
197		2022-05-04	3	1	0	0	0	2

Figura 35 – Estrutura dos dados de ordens de serviço depois da organização completa para os produtos MODELO 01.
Fonte: Elaborado pelo autor.

Resultados | Mensagens

	TE_MAL_FIXADO_SOLTO	FALHA_ELETRICA_ELETRONICA	VAZAMENTO_GAS	AUSENCIA_COMPONENTES	COMPONENTES_INTERNOS_DANIFICADOS	FALHA_VACUO	VEDACAO_DEFICIENTE	QTD_OS
178		1	1	0	1	2	0	19
179		0	1	2	0	0	0	14
180		0	3	1	2	1	0	15
181		2	2	0	2	0	0	18
182		1	0	0	0	0	0	12
183		1	0	0	0	0	1	13
184		3	0	0	2	0	4	2
185		1	0	1	1	2	1	18
186		2	0	0	0	1	3	14
187		1	5	0	2	1	0	15
188		3	0	2	0	1	0	5
189		0	0	0	2	0	1	7
190		0	4	0	5	1	0	19
191		2	2	1	1	2	0	12
192		1	0	2	0	0	0	8
193		1	1	1	2	0	0	26
194		1	0	0	1	2	0	16
195		0	4	1	0	2	0	5
196		1	0	0	0	5	1	7
197		1	5	0	0	1	0	11

Figura 36 – Estrutura dos dados de ordens de serviço depois da organização completa para os produtos MODELO 01.
Fonte: Elaborado pelo autor.

E também para os produtos MODELO 02, conforme figuras 37 e 38:

Resultados | Mensagens

	SKU	DATA_CORRIGIDA	VAZAMENTO_PU	VAZAMENTO_PU_CANTO_PLASTICO	VAZAMENTO_PU_TRAVESSA	VAZAMENTO_PU_GABINETE	VAZAMENTO_PU_FUNDO_POLIONDAS	VAZAMENTO_PU_CAI
378		2023-02-15	0	0	0	0	0	0
379		2023-02-16	0	0	0	0	0	0
380		2023-02-17	0	0	0	0	0	0
381		2023-02-18	0	0	0	0	0	0
382		2023-02-20	0	0	0	0	0	0
383		2023-02-21	0	0	0	0	0	0
384		2023-02-22	0	0	0	0	0	0
385		2023-02-23	0	0	0	0	0	0
386		2023-02-24	0	0	0	0	0	0
387		2023-02-27	0	0	0	0	0	0
388		2023-02-28	0	0	0	0	0	0
389		2023-03-01	0	0	0	0	0	0
390		2023-03-02	0	0	0	1	0	0
391		2023-03-03	0	0	0	0	0	0
392		2023-03-06	0	0	0	0	0	0
393		2023-03-07	0	0	0	0	0	1
394		2023-03-08	0	0	0	0	0	0
395		2023-03-09	0	0	0	0	0	0
396		2023-03-10	0	0	0	0	0	0
397		2023-03-11	0	0	0	0	0	0

Figura 37 – Estrutura dos dados de ordens de serviço depois da organização completa para os produtos MODELO 02.
Fonte: Elaborado pelo autor.

Resultados | Mensagens

	TE_MAL_FIXADO_SOLTO	FALHA_ELETRICA_ELETRONICA	VAZAMENTO_GAS	AUSENCIA_COMPONENTES	COMPONENTES_INTERNOS_DANIFICADOS	FALHA_VACUO	VEDACAO_DEFICIENTE	QTD_OS
378		0	2	1	1	0	0	NULL
379		0	0	1	1	0	0	2
380		2	1	0	0	0	0	2
381		0	0	1	0	0	0	6
382		0	0	0	1	0	0	2
383		2	0	1	0	0	0	4
384		1	0	0	1	0	0	10
385		1	2	0	1	0	0	NULL
386		0	2	1	0	0	0	7
387		0	0	0	0	0	0	2
388		0	0	0	0	0	0	4
389		2	0	0	0	0	0	5
390		0	0	1	0	0	0	7
391		0	1	0	0	0	0	4
392		0	0	1	0	0	0	1
393		2	0	2	0	0	0	5
394		1	0	0	0	0	0	1
395		0	0	1	0	0	0	1
396		0	2	0	0	0	0	1
397		0	0	0	0	0	0	NULL

Figura 38 – Estrutura dos dados de ordens de serviço depois da organização completa para os produtos MODELO 02.
Fonte: Elaborado pelo autor.

2.3 DESENVOLVIMENTO DO MODELO DE DADOS UTILIZANDO IA E APRENDIZAGEM DE MÁQUINA

Essa etapa compreende as seguintes partes do processo (em vermelho), conforme ilustrado na figura 39:

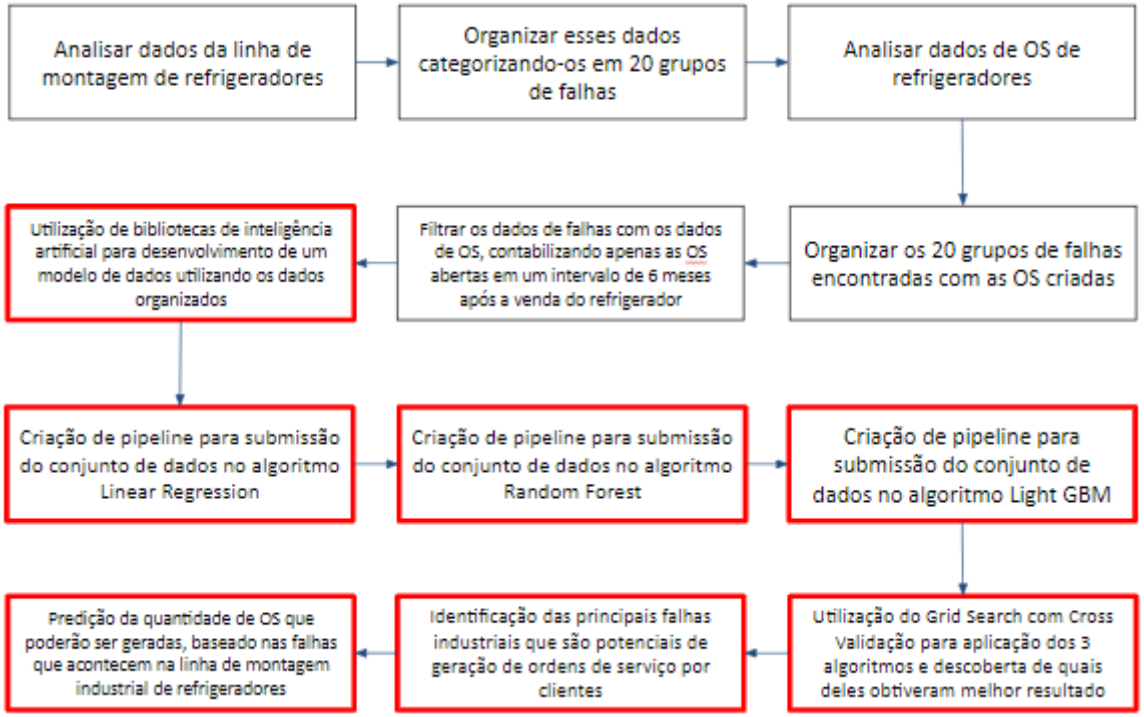

Figura 39 – Etapas desenvolvidas na terceira parte.
Fonte: Elaborado pelo autor.

Para o desenvolvimento desse modelo de dados, foi considerado os conjuntos de dados contendo produto, data, número de falhas para cada grupo e número de ordens de serviço geradas, conforme figuras 40 e 41:

Resultados | Mensagens

	SKU	DATA_CORRIGIDA	VAZAMENTO_PU	VAZAMENTO_PU_CANTO_PLASTICO	VAZAMENTO_PU_TRAVESSA	VAZAMENTO_PU_GABINETE	VAZAMENTO_PU_FUNDO_POLIONDAS	VAZAMENTO_PU_CA
252		2021-07-05	0	0	0	0	0	0
253		2021-07-06	0	0	0	0	0	0
254		2021-07-06	4	3	0	0	0	0
255		2021-07-07	0	0	0	0	0	0
256		2021-07-07	2	2	0	0	0	0
257		2021-07-08	1	0	0	0	0	0
258		2021-07-08	3	6	2	0	0	1
259		2021-07-09	0	0	1	1	0	0
260		2021-07-09	1	3	3	0	3	2
261		2021-07-10	0	0	0	0	0	0
262		2021-07-10	2	6	1	0	0	1
263		2021-07-11	0	0	0	0	0	0
264		2021-07-11	0	0	0	0	0	0
265		2021-07-12	0	0	0	0	0	0
266		2021-07-12	0	5	1	0	0	0
267		2021-07-13	1	0	1	0	0	0
268		2021-07-13	1	1	0	0	1	0
269		2021-07-14	1	0	0	0	0	0
270		2021-07-14	2	3	0	0	1	1
271		2021-07-15	0	0	0	0	0	0

Figura 40 – Conjunto de dados para o desenvolvimento do modelo de dados.
Fonte: Elaborado pelo autor.

Resultados | Mensagens

	TE_MAL_FIXADO_SOLTO	FALHA_ELETRICA_ELETRONICA	VAZAMENTO_GAS	AUSENCIA_COMPONENTES	COMPONENTES_INTERNOS_DANIFICADOS	FALHA_VACUO	VEDACAO_DEFICIENTE	QTD_OS
252		0	0	0	0	1	0	11
253		0	0	0	1	0	0	20
254		0	4	0	0	1	0	12
255		0	1	1	1	0	0	16
256		0	1	1	1	2	4	10
257		1	0	0	0	0	0	19
258		1	1	1	2	0	0	12
259		2	0	0	2	0	3	15
260		1	3	0	0	1	0	28
261		0	0	1	0	0	0	9
262		4	4	5	6	1	0	24
263		0	0	0	0	0	0	1
264		0	0	0	0	0	0	2
265		0	0	0	1	0	0	12
266		1	11	2	1	4	2	6
267		0	0	0	0	0	0	13
268		1	0	3	0	0	0	12
269		0	0	0	2	0	1	15
270		1	2	1	2	1	1	10
271		0	0	0	0	0	0	17

Figura 41 – Conjunto de dados para o desenvolvimento do modelo de dados.
Fonte: Elaborado pelo autor.

Após ter os dados normalizados e prontos para uso, foi gerado um arquivo CSV, contendo todo o conteúdo acima organizado em texto, conforme figura 42:

falhas_os_refri_2021_2023_NEW (2).csv - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```
SKU,DATA_CORRIGIDA,VAZAMENTO_PU,VAZAMENTO_PU_CANTO_PLASTICO,VAZAMENTO_PU_TRAVESSA,VAZAMENTO_PU_GABINETE,VAZAMENTO_PU_FUNDO_POLIONDAS,VAZAMENTO_P
U_CAIXA_INTERNA,AMASSADO_PORTA,RISCADO_SOVADO,AMASSADO_GABINETE,IDENTIFICACAO_AUSENTE,AVARIA_LOGISTICA,LIMPEZA_INADEQUADA,FALHA_PINTURA,COMPONEN
TE_MAL_FIXADO_SOLTO,FALHA_ELETRICA_ELETRONICA,VAZAMENTO_GAS,AUSENCIA_COMPONENTES,COMPONENTES_INTERNOS_DANIFICADOS,FALHA_VACUO,VEDACAO_DEFICIENTE
,
,2021-02-11,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,25,
,2021-02-11,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,18,
,2021-02-12,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,11,
,2021-02-12,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,23,
,2021-02-12,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,
,2021-02-13,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,
,2021-02-13,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,14,
,2021-02-14,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,
,2021-02-15,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,17,
,2021-02-15,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,11,
,2021-02-16,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,12,
,2021-02-16,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,20,
,2021-02-17,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,22,
,2021-02-17,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,9,
,2021-02-18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,
,2021-02-18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,14,
,2021-02-19,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,17,
```

Figura 42 – Arquivo CSV com conteúdo organizado em texto.
Fonte: Elaborado pelo autor.

Esse arquivo CSV foi utilizado como um Dataset para o modelo que foi desenvolvido.

Para o desenvolvimento do modelo de dados foi utilizado a linguagem Python em um notebook no Google Colab. O início do desenvolvimento do notebook contém todas as bibliotecas que utilizaremos no processo de desenvolvimento.

```
import numpy as np
import pandas as pd
import lightgbm as lgb
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LinearRegression
```

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error,
mean_absolute_error
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
```

Após a importação das bibliotecas, o arquivo CSV foi importado como um Dataset do modelo.

```
# Leitura da Base
df = pd.read_csv('/content/falhas_os_refri_2021_2023_NEW.csv')
```

Algumas correções foram feitas nos campos (header) do Dataset.

```
# Correção do header da base
tmp = df.dropna(axis=1).reset_index()
tmp.columns = df.columns
df = tmp
df.rename(columns={'Unnamed: 22':'NrOS'}, inplace=True)
df['DATA_CORRIGIDA'] = pd.todatetime(df['DATA_CORRIGIDA'])
```

O modelo desenvolvido é flexível e pode ser executado para cada modelo de refrigerador. Nesse caso, vamos iniciar com um dos modelos existentes no Dataset, o MODELO 01.

```
# Seleção de SKU para modelagem
sku = 'MODELO 01'
df_sku = df[df.SKU==sku].drop(columns='SKU')
```

Organiza os dados realizando a reamostragem temporal de acordo com o modelo do refrigerador selecionado (SKU).

```
# Reamostragem Temporal da série temporal do SKU selecionado
df_sku = df_sku.set_index('DATA_CORRIGIDA').resample('D').sum()
```

Retira os dias de Domingo da análise, pois não é um dia produtivo de trabalho nas linhas de montagem, para evitar interferências posteriores na análise.

```
df_sku['weekday'] = df_sku.index.weekday
df_sku = df_sku[df_sku.weekday!=6].drop(columns='weekday')
```

Exibe os dados organizados no Dataset, através da variável df_sku. Os dados utilizados para análise pertencem ao intervalo de fevereiro de 2021 até junho de 2023, conforme figura 43:

df_sku

DATA_CORRIGIDA	VAZAMENTO_PU	VAZAMENTO_PU_CANTO_PLASTICO	VAZAMENTO_PU_TRAVESSA	VAZAMENTO_PU_GABINETE	VAZAMENTO_PU_FUNDO_POLIONDAS	VAZAMENTO_PU_CAI
2021-02-11	0	0	0	0	0	
2021-02-12	0	0	0	0	0	
2021-02-13	0	0	0	0	0	
2021-02-15	0	0	0	0	0	
2021-02-16	0	0	0	0	0	
...	...	...	...	...	...	
2023-06-08	0	0	0	0	0	
2023-06-09	0	0	0	1	0	
2023-06-10	0	0	0	0	0	
2023-06-12	2	1	0	1	0	

Figura 43 – Dataset da variável df_sku no intervalo de fevereiro de 2021 até junho de 2023.
Fonte: Elaborado pelo autor.

Em uma próxima etapa, utilizando todo o conjunto de dados, foi separado uma parte dos dados para treino e outra para teste.

```
# Selecionando conjuntos de treino(+ validação) e teste
# O ultimo dado valido do conjunto é a data max - 6 meses (estamos contabilizando OS que tem probabilidade de ocorrer em um intervalo max de 6 meses para cada dia de fabricação)
df_valid = df_sku[df_sku.index<df_sku.index.max() - pd.Timedelta(days=180)]
df_valid_ = df_valid.sort_index().reset_index().drop(columns='DATA_CORRIGIDA')
```

O resultado é o apresentado conforme figura 44:

[ ] df_valid

DATA_CORRIGIDA	VAZAMENTO_PU	VAZAMENTO_PU_CANTO_PLASTICO	VAZAMENTO_PU_TRAVESSA	VAZAMENTO_PU_GABINETE	VAZAMENTO_PU_FUNDO_POLIONDAS	VAZAMENTO_PU_CAIXA_INTERNA	AMASSAI
2021-02-11	0	0	0	0	0	0	
2021-02-12	0	0	0	0	0	0	
2021-02-13	0	0	0	0	0	0	
2021-02-15	0	0	0	0	0	0	
2021-02-16	0	0	0	0	0	0	
...	...	...	...	...	...	...	
2022-12-09	1	0	0	1	0	0	
2022-12-10	0	0	5	0	0	0	
2022-12-12	1	1	0	0	0	0	
2022-12-13	1	0	0	0	0	1	
2022-12-14	1	0	0	0	0	0	

Figura 44 – Dataset do intervalo de fevereiro de 2021 até junho de 2023.
Fonte: Elaborado pelo autor.

Diante dos dados organizados, é iniciado o processo de entrada dos dados para treino. Os dados para treino serão submetidos da seguinte forma:

Entradas para treino:

sku, data(dia), qtd falhas por dia por cada um dos 20 grupos de falha, quantidade de os por dia pela data da fabricação do refrigerador.

Já os dados para teste, possuem quase a mesma estrutura, porém sem a informação de quantidade de os.

Entrada para teste:

sku, data(dia), qtd falhas por dia por cada um dos 20 grupos de falha.

Essa organização, tem como objetivo nos retornar a quantidade de os que poderá ser gerada para os próximos dias.

Saída esperada:

quantidade de OS para essa data de fabricação, que é a data de entrada.

Esse planejamento foi construído através do desenvolvimento abaixo:

```
# Inicio da Modelagem
X = df_valid_.drop(columns='NrOS')
y = df_valid_['NrOS']
# Conjunto treino e teste
# X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
X_train = X[:-60]
y_train = y[:-60]
X_test = X[-60:]
y_test = y[-60:]
```

Foi necessário o desenvolvimento de 2 funções auxiliares para utilização posterior no modelo de dados.

```
# Funções auxiliares
def create_pipeline(model):
    return Pipeline([
            ('scaler', StandardScaler()),  # Standardize features
(optional)
        ('model', model)
```

```
        ])
    # Funções auxiliares
    def train_and_evaluate_model(pipe, param_grid, X_train, y_train, X_test, y_test):
            grid_search = GridSearchCV(pipe, param_grid, cv=5, scoring='neg_mean_squared_error', n_jobs=-1)
        grid_search.fit(X_train, y_train)

        best_model = grid_search.best_estimator_
        best_score = grid_search.best_score_
        return best_model, best_score
```

Cria 3 pipelines selecionando 3 algoritmos de regressão para utilização na análise dos dados. Os algoritmos LinerRegression, RandomForest e LightGBM foram escolhidos, conforme desenvolvimento abaixo:

```
    # Definindo os pipelines para cada tipo de modelo para serem treinados via GridSearch
    linear_pipe = create_pipeline(LinearRegression())
    rf_pipe = create_pipeline(RandomForestRegressor(random_state=42))
    lgb_pipe = create_pipeline(lgb.LGBMRegressor(random_state=42))
```

Define os hiper parâmetros (hyper parameters) para cada algoritmo de regressão, para ser utilizado com Grid Search posteriormente.

```
    # Define hyperparameter grids for GridSearchCV
    linear_param_grid = {}
    rf_param_grid = {
            'model__n_estimators': [100, 200, 300],
            'model__max_depth': [None, 10, 20, 30]
        }
    lgb_param_grid = {
            'model__n_estimators': [100, 200, 300],
            'model__learning_rate': [0.05, 0.1, 0.2],
            'model__max_depth': [5, 10, 15]
        }

    models = {
            "Linear Regression": (linear_pipe, linear_param_grid), # modelo de regressao linear padrão
        "Random Forest Regression": (rf_pipe, rf_param_grid), # modelo de ensamble do tipo bagging
```

```
          "LightGBM Regression": (lgb_pipe, lgb_param_grid) # modelo de
ensamble do tipo boosting
      }
```

Aplica a técnica de Grid Search com Cross Validação para submeter o conjunto de dados nos 3 algoritmos de regressão utilizados no desenvolvimento.

```
      # Cross Validação com GridSearch (treinamento + validação)
      eval = dict()
      best_m = dict()
      for model_name, (pipe, param_grid) in models.items():
                best_model,  metrics  =  train_and_evaluate_model(pipe,
param_grid, X_train, y_train, X_test, y_test)
          eval.update({model_name:metrics})
          best_m.update({model_name:best_model})
          print(f"{model_name}:")
                                    print(f"Best        hyperparameters:
{best_model.named_steps['model'].get_params()}")
          print(f"Best metric {metrics}:")
          print()
```

O desenvolvimento abaixo foi utilizado para gerar o gráfico apresentando o resultado do treinamento.

```
      fig, ax = plt.subplots(1,1)
      ax.set_title('Backtest - Treinamento')
      ax.set_ylabel('NrOS')
      df_valid['NrOS'].iloc[:-60].plot(ax=ax, label='Real')
      pd.Series(y_bck_train,
index=df_valid['NrOS'].iloc[:-60].index).plot(ax=ax,label='Estim.')
      plt.legend()
```

Já o desenvolvimento abaixo foi utilizado para gerar o gráfico apresentando o resultado do teste, após o treinamento.

```
      fig, ax = plt.subplots(1,1)
      ax.set_title('Conj Teste')
      ax.set_ylabel('NrOS')
      df_valid['NrOS'].iloc[-60:].plot(ax=ax, label='Real')
      pd.Series(y_pred,
index=df_valid['NrOS'].iloc[-60:].index).plot(ax=ax,label='Estim.')
      plt.legend()
```

Encerrando assim, o processo de desenvolvimento do modelo de dados.

3. RESULTADOS E DISCUSSÕES

A inteligência artificial utilizando o Aprendizado de Máquina (Machine Learning, em inglês) é uma disciplina que capacita sistemas computacionais a aprenderem padrões a partir de dados e aperfeiçoarem seu desempenho com base nessa aprendizagem.

Diante disso, foi desenvolvido um modelo de analisar dados de falhas encontradas na linha de montagem de refrigeradores dos últimos 3 anos, categorizando-os em 20 grupos de falhas, sendo divididos em 3 grandes categorias. Todos esses dados foram correlacionados com dados de geração de OS para refrigeradores, geradas até 6 meses após sua fabricação.

O objetivo é normalizar os dados e descobrir padrões de comportamento através da aplicação de algoritmos de regressão e técnicas avançadas de machine learning para identificar as principais falhas ofensoras e a quantidade de OS que poderão ser geradas, baseado nas falhas que acontecem na linha de montagem industrial de refrigeradores.

A seguir são mostrados os resultados e discussões sobre análise de alguns modelos de refrigeradores.

3.1 REFRIGERADOR MODELO 01

Para o produto MODELO 01 foi utilizado o seguinte conjunto de dados, desde fevereiro de 2021 até dezembro de 2023, conforme figura 44:

DATA_CORRIGIDA	VAZAMENTO_PU	VAZAMENTO_PU_CANTO_PLASTICO	VAZAMENTO_PU_TRAVESSA	VAZAMENTO_PU_GABINETE	VAZAMENTO_PU_FUNDO_POLIONDAS	VAZAMENTO_PU_CAIXA_INTERNA	AMASSAD
2021-02-11	0	0	0	0	0	0	
2021-02-12	0	0	0	0	0	0	
2021-02-13	0	0	0	0	0	0	
2021-02-15	0	0	0	0	0	0	
2021-02-16	0	0	0	0	0	0	
...	...	...	...	...	...	...	
2022-12-09	1	0	0	1	0	0	
2022-12-10	0	0	5	0	0	0	
2022-12-12	1	1	0	0	0	0	
2022-12-13	1	0	0	0	0	1	
2022-12-14	1	0	0	0	0	0	

576 rows × 21 columns

Figura 44 – Dataset do intervalo de fevereiro de 2023 até dezembro de 2023 para o produto MODELO 01.
Fonte: Elaborado pelo autor.

É possível observar as correlações entre as falhas encontradas para esse produto. Quanto mais o valor é próximo de 0.8 a 1, mais correlacionado a falha é com outra. Nesse caso, é possível identificar que nenhuma falha impacta com outra, tornando a análise de falhas de forma independente entre as falhas, conforme figuras 45 e 46:

	VAZAMENTO_PU	VAZAMENTO_PU_CANTO_PLASTICO	VAZAMENTO_PU_TRAVESSA	VAZAMENTO_PU_GABINETE	VAZAMENTO_PU_FUNDO_POLIONDAS	VAZAMENTO_PU_CAIXA_INTERNA	AMASSADO_PORTA	RISCADO_SOVADO	AMASSADO_GABINETE	IDENTIFICACAO_AUSENTE	...	LIMPEZA_INADEQUADA	FALHA_PINTURA
VAZAMENTO_PU	1.000000	0.242523	0.280111	0.107704	0.116381	0.248894	0.259706	0.161607	0.290838	0.156614	...	0.119867	0.116653
VAZAMENTO_PU_CANTO_PLASTICO	0.242523	1.000000	0.343037	0.276916	0.231643	0.358602	0.222857	0.168996	0.304008	0.205994	...	0.024787	0.140873
VAZAMENTO_PU_TRAVESSA	0.280111	0.343037	1.000000	0.203997	0.198759	0.251290	0.288535	0.169077	0.299687	0.236470	...	0.096242	0.179607
VAZAMENTO_PU_GABINETE	0.107704	0.276916	0.203997	1.000000	0.173591	0.207663	0.174829	0.071341	0.251351	0.238233	...	0.082421	0.062068
VAZAMENTO_PU_FUNDO_POLIONDAS	0.116381	0.231643	0.198759	0.173591	1.000000	0.154120	0.167347	0.191248	0.267579	0.104079	...	-0.028532	0.019362
VAZAMENTO_PU_CAIXA_INTERNA	0.248894	0.358602	0.251290	0.207663	0.154120	1.000000	0.277546	0.174893	0.293079	0.208804	...	0.042530	0.201897
AMASSADO_PORTA	0.259706	0.222857	0.288535	0.174829	0.167347	0.277546	1.000000	0.386353	0.426951	0.220174	...	0.173519	0.146856
RISCADO_SOVADO	0.161607	0.168996	0.169077	0.071341	0.191248	0.174893	0.386353	1.000000	0.362633	0.126580	...	0.118316	0.054195
AMASSADO_GABINETE	0.290838	0.304008	0.299687	0.251351	0.267579	0.293079	0.426951	0.362633	1.000000	0.280494	...	0.043725	0.166465
IDENTIFICACAO_AUSENTE	0.156614	0.205994	0.236470	0.238233	0.104079	0.208804	0.220174	0.126580	0.280494	1.000000	...	0.087224	0.082344
AVARIA_LOGISTICA	-0.035354	-0.110785	-0.035005	-0.092967	-0.046301	-0.090014	0.037136	0.061345	-0.075839	-0.029978	...	-0.038804	-0.048419
LIMPEZA_INADEQUADA	0.119867	0.024787	0.096242	0.082421	-0.028532	0.042530	0.173519	0.118316	0.043725	0.087224	...	1.000000	-0.023580
FALHA_PINTURA	0.116653	0.140873	0.179607	0.062068	0.019362	0.201897	0.146856	0.054195	0.166465	0.082344	...	-0.023580	1.000000
COMPONENTE_MAL_FIXADO_SOLTO	0.278461	0.276318	0.285895	0.306324	0.218059	0.264280	0.378696	0.240085	0.459782	0.313976	...	0.095517	0.168930
FALHA_ELETRICA_ELETRONICA	0.211511	0.285714	0.217232	0.282415	0.234953	0.248651	0.312791	0.303261	0.425468	0.272540	...	0.061339	0.135773
VAZAMENTO_GAS	0.145773	0.124031	0.139749	0.169815	0.163827	0.093913	0.362216	0.279988	0.258691	0.199009	...	0.116152	0.049411
AUSENCIA_COMPONENTES	0.150634	0.262658	0.126784	0.252103	0.207597	0.184903	0.147183	0.114750	0.150298	0.134450	...	0.087642	0.006196
COMPONENTES_INTERNOS_DANIFICADOS	0.174298	0.298073	0.245080	0.327051	0.215289	0.227553	0.271576	0.240523	0.336474	0.233792	...	0.076947	0.116638
FALHA_VACUO	0.182262	0.133814	0.171012	0.081576	0.049897	0.052808	0.201908	0.252251	0.203953	0.098037	...	0.129959	0.037262
VEDACAO_DEFICIENTE	0.092333	0.164141	0.110480	0.116228	0.157918	0.125324	0.169051	0.144692	0.098736	0.106312	...	0.115680	0.062883

Figura 45 – Análise das falhas de forma independente para o produto MODELO 01.
Fonte: Elaborado pelo autor.

VAZAMENTO_PU_CAIXA_INTERNA	AMASSADO_PORTA	RISCADO_SOVADO	AMASSADO_GABINETE	IDENTIFICACAO_AUSENTE	...	LIMPEZA_INADEQUADA	FALHA_PINTURA	COMPONENTE_MAL_FIXADO_SOLTO	FALHA_ELETRICA_ELETRONICA	VAZAMENTO_GAS	AUSENCIA_COMPONENTES	COMPONENTES_INTERNOS_DANIFICADOS	FALHA_VACUO	VEDACAO_DEFICIENTE
0.248894	0.259706	0.161607	0.290838	0.156614	...	0.119867	0.116653	0.278461	0.211511	0.145773	0.150634	0.174298	0.182262	0.092333
0.358602	0.222857	0.168996	0.304008	0.205994	...	0.024787	0.140873	0.276318	0.285714	0.124031	0.262658	0.298073	0.133814	0.164141
0.251290	0.288535	0.169077	0.299687	0.236470	...	0.096242	0.179607	0.285895	0.217232	0.139749	0.126784	0.245080	0.171012	0.110480
0.207663	0.174829	0.071341	0.251351	0.238233	...	0.082421	0.062068	0.306324	0.282415	0.169815	0.252103	0.327051	0.081576	0.116228
0.154120	0.167347	0.191248	0.267579	0.104079	...	-0.028532	0.019362	0.218059	0.234953	0.163827	0.207597	0.215289	0.049897	0.157918
1.000000	0.277546	0.174893	0.293079	0.208804	...	0.042530	0.201897	0.264280	0.248651	0.093913	0.184903	0.227553	0.052808	0.125324
0.277546	1.000000	0.386353	0.426951	0.220174	...	0.173519	0.146856	0.378696	0.312791	0.362216	0.147183	0.271576	0.201908	0.169051
0.174893	0.386353	1.000000	0.362633	0.126580	...	0.118316	0.054195	0.240085	0.303261	0.279988	0.114750	0.240523	0.252251	0.144692
0.293079	0.426951	0.362633	1.000000	0.280494	...	0.043725	0.166465	0.459782	0.425468	0.258691	0.150298	0.336474	0.203953	0.098736
0.208804	0.220174	0.126580	0.280494	1.000000	...	0.087224	0.082344	0.313976	0.272540	0.199009	0.134450	0.233792	0.098037	0.106312
-0.090014	0.037136	0.061345	-0.075839	-0.029978	...	-0.038804	-0.048419	-0.074230	-0.052998	0.087862	0.025325	-0.029657	-0.080172	-0.082197
0.042530	0.173519	0.118316	0.043725	0.087224	...	1.000000	-0.023580	0.095517	0.061339	0.116152	0.087642	0.076947	0.129959	0.115680
0.201897	0.146856	0.054195	0.166465	0.082344	...	-0.023580	1.000000	0.168930	0.135773	0.049411	0.006196	0.116638	0.037262	0.062883
0.264280	0.378696	0.240085	0.459782	0.313976	...	0.095517	0.168930	1.000000	0.457719	0.289223	0.225617	0.412269	0.217641	0.230608
0.248651	0.312791	0.303261	0.425468	0.272540	...	0.061339	0.135773	0.457719	1.000000	0.322981	0.208855	0.436231	0.198515	0.256795
0.093913	0.362216	0.279988	0.258691	0.199009	...	0.116152	0.049411	0.289223	0.322981	1.000000	0.198390	0.276629	0.214239	0.210806
0.184903	0.147183	0.114750	0.150298	0.134450	...	0.087642	0.006196	0.225617	0.208855	0.198390	1.000000	0.299799	0.136056	0.163947
0.227553	0.271576	0.240523	0.336474	0.233792	...	0.076947	0.116638	0.412269	0.436231	0.276629	0.299799	1.000000	0.126485	0.205520
0.052808	0.201908	0.252251	0.203953	0.098037	...	0.129959	0.037262	0.217641	0.198515	0.214239	0.136056	0.126485	1.000000	0.125534
0.125324	0.169051	0.144692	0.098736	0.106312	...	0.115680	0.062883	0.230608	0.256795	0.210806	0.163947	0.205520	0.125534	1.000000

Figura 46 – Análise das falhas de forma independente para o produto MODELO 01.
Fonte: Elaborado pelo autor.

Foi possível identificar também as quantidades de falhas geradas para esse produto nesse mesmo intervalo, assim como a quantidade de OS criadas, conforme figura 47:

NrOS	12392
AMASSADO_PORTA	2938
COMPONENTE_MAL_FIXADO_SOLTO	2584
FALHA_ELETRICA_ELETRONICA	2292
RISCADO_SOVADO	2290
AMASSADO_GABINETE	2197
COMPONENTES_INTERNOS_DANIFICADOS	2011
VAZAMENTO_GAS	1984
AUSENCIA_COMPONENTES	1328
FALHA_VACUO	1203
VAZAMENTO_PU_CANTO_PLASTICO	1022
VAZAMENTO_PU_TRAVESSA	737
VAZAMENTO_PU	672
VEDACAO_DEFICIENTE	597
IDENTIFICACAO_AUSENTE	588
VAZAMENTO_PU_GABINETE	585
VAZAMENTO_PU_CAIXA_INTERNA	557
LIMPEZA_INADEQUADA	400
VAZAMENTO_PU_FUNDO_POLIONDAS	378
FALHA_PINTURA	74
AVARIA_LOGISTICA	66

Figura 47 – Quantidade de OS criadas e quantidade de falhas geradas para o produto MODELO 01.
Fonte: Elaborado pelo autor.

Para o conjunto de teste no produto (MODELO 01) foi identificado que as 3 principais falhas encontradas na linha industrial AMASSADO_PORTA, COMPONENTE_MAL_FIXADO_SOLTO e FALHA_ELETRICA_ELETRONICA. Além disso, pode-se observar o comportamento dessas 3 falhas nos últimos 2 meses do conjunto de dados em estudo, conforme gráfico 1:

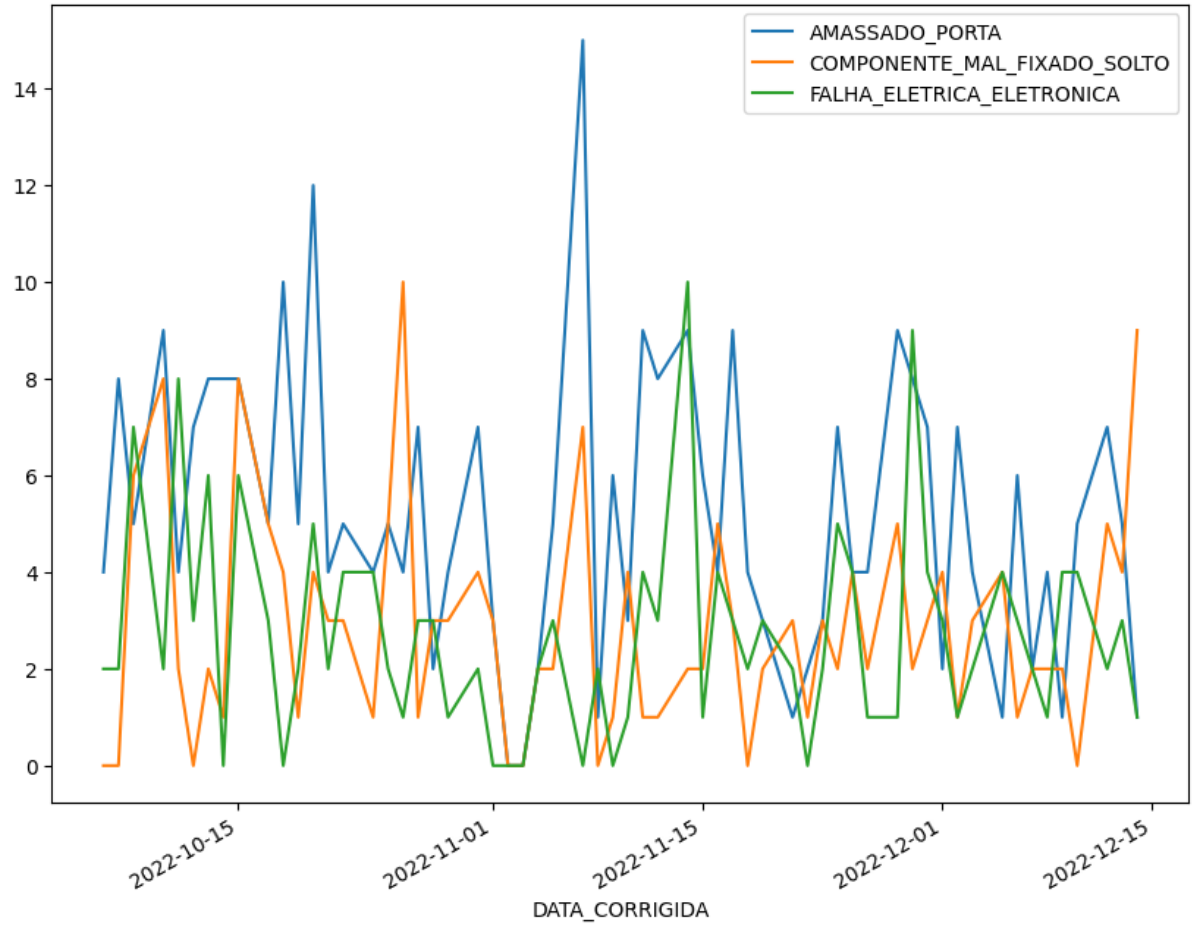

Gráfico 1 – Relação entre quantidade de OS e data de cada falha para o produto MODELO 01.
Fonte: Elaborado pelo autor.

Após aplicação das técnicas de *Cross* Validação com *Grid Search*, aplicando os algoritmos LinearRegression, RandomForest e LightGBM, foi obtido as seguintes métricas:

Linear Regression:

Best hyperparameters: {'copy_X': True, 'fit_intercept': True, 'n_jobs': None, 'positive': False}

Best metric -51.56139660266668:

Random Forest Regression:

Best hyperparameters: {'bootstrap': True, 'ccp_alpha': 0.0, 'criterion': 'squared_error', 'max_depth': 10, 'max_features': 1.0, 'max_leaf_nodes': None, 'max_samples': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1,

'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'n_estimators': 100, 'n_jobs': None, 'oob_score': False, 'random_state': 42, 'verbose': 0, 'warm_start': False}

Best metric -53.24355301714852:

LightGBM Regression:

Best hyperparameters: {'boosting_type': 'gbdt', 'class_weight': None, 'colsample_bytree': 1.0, 'importance_type': 'split', 'learning_rate': 0.05, 'max_depth': 5, 'min_child_samples': 20, 'min_child_weight': 0.001, 'min_split_gain': 0.0, 'n_estimators': 100, 'n_jobs': None, 'num_leaves': 31, 'objective': None, 'random_state': 42, 'reg_alpha': 0.0, 'reg_lambda': 0.0, 'subsample': 1.0, 'subsample_for_bin': 200000, 'subsample_freq': 0}

Best metric -51.84025513129977:

Nesse caso o modelo mais assertivo, entre os 3 aplicados, foi LinearRegression com uma taxa de erro de 51,56%.

Além disso, foi possível identificar também o MSE e o MAE:

Mean Squared Error (MSE): 42.43

Mean Absolute Error (MAE): 5.45

Em uma próxima etapa, foi gerado o gráfico do treinamento realizado entre o período de fevereiro de 2021 até setembro de 2022, para o produto MODELO 01, conforme gráfico 2:

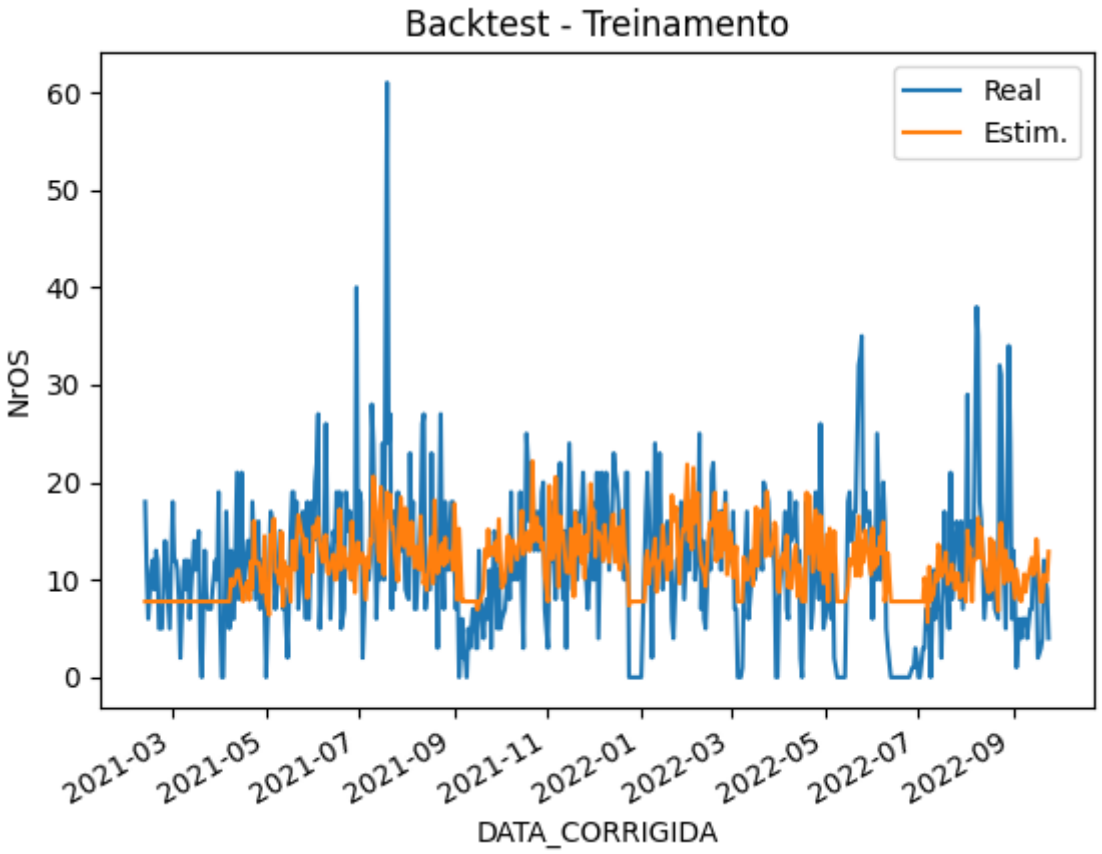

Gráfico 2 – Etapa de treinamento no período de fevereiro de 2021 e setembro de 2022 para o produto MODELO 01.
Fonte: Elaborado pelo autor.

Esse gráfico demonstra que o modelo em treinamento conseguiu capturar a informação, além de auxiliar também a entender se o modelo sofreu *overfitting*. Nesse caso o modelo não sofreu *overfitting*.

Após essa etapa, foi gerado o gráfico do teste realizado entre o período de outubro de 2022 a dezembro de 2022, para o produto MODELO 01, conforme gráfico 3:

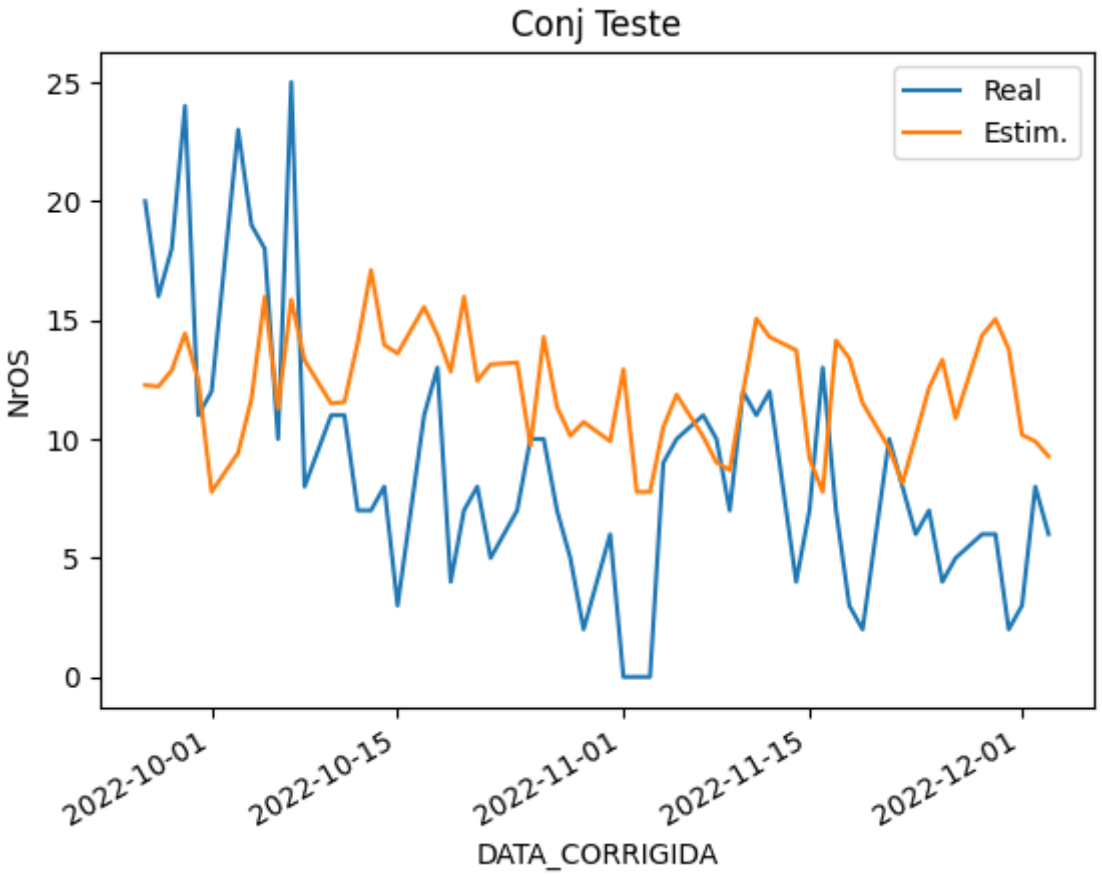

Gráfico 3 – Etapa de teste entre o período de outubro de 2022 e dezembro de 2022 para o produto MODELO 01.
Fonte: Elaborado pelo autor.

Diante disso, é possível observar que o modelo conseguiu responder bem a começou a identificar padrões para predição de número de ordens de serviço que podem ser geradas, com base nas falhas que acontecem nas linhas de produção.

Além dos resultados acima, foi possível também identificar, o quanto cada falha favoreceu para a análise realizada e submetida para o modelo de dados, através do gráfico 4:

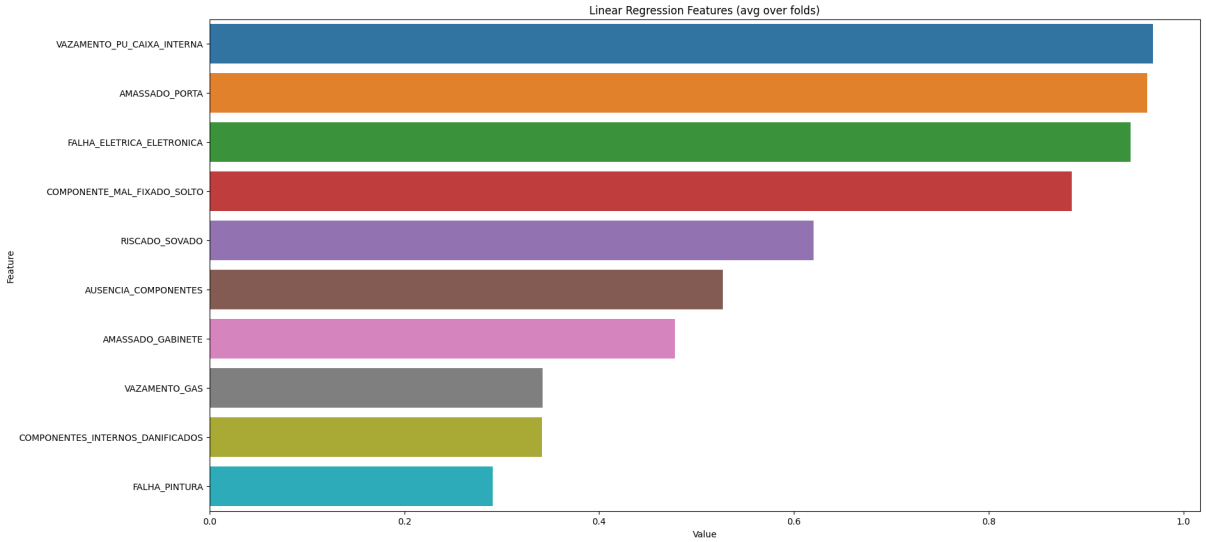

Gráfico 4 – Relação de favorecimento de cada falha para o resultado do produto MODELO 01.
Fonte: Elaborado pelo autor.

É possível perceber que VAZAMENTO_PU_CAIXA_INTERNA, AMASSADO_PORTA, FALHA_ELETRICA_ELETRONICA e COMPONENTE_MAL_FIXADO_SOLTO, foram os principais grupos de falha que influenciaram nessa análise.

Conforme figura 48, é possível identificar (em valores) as 10 falhas mais significativas para a análise deste produto:

	Value	Feature
19	0.968747	VAZAMENTO_PU_CAIXA_INTERNA
18	0.962581	AMASSADO_PORTA
17	0.945746	FALHA_ELETRICA_ELETRONICA
16	0.885643	COMPONENTE_MAL_FIXADO_SOLTO
15	0.620124	RISCADO_SOVADO
14	0.526926	AUSENCIA_COMPONENTES
13	0.477537	AMASSADO_GABINETE
12	0.342198	VAZAMENTO_GAS
11	0.341280	COMPONENTES_INTERNOS_DANIFICADOS
10	0.290474	FALHA_PINTURA

Figura 48 – As dez falhas mais significativas para a análise do produto MODELO 01.
Fonte: Elaborado pelo autor.

Em um cenário onde cada ordem de serviço tem custo médio de R$ 1.500, conclui-se que:

Para o Produto Modelo 01:

- Tivemos **12.392** ordens de serviço abertas;

- Gerando um custo médio de **R$ 18.588.000,00** ao longo de **28 meses** de análise desde **Fevereiro de 2021 até Junho de 2023**;

- Tendo como retorno do modelo de dados uma assertividade de **48,44%**;

- Com isso, podendo obter uma possível economia de **R$ 9.004.027,20** com relação ao valor total gasto com OS;

- E uma redução de **6.002** ordens de serviço nesse período.

3.2 REFRIGERADOR MODELO 02

Para o produto MODELO 02 foi utilizado o seguinte conjunto de dados, desde fevereiro de 2023 até dezembro de 2022, conforme figura 49:

DATA_CORRIGIDA	VAZAMENTO_PU	VAZAMENTO_PU_CANTO_PLASTICO	VAZAMENTO_PU_TRAVESSA	VAZAMENTO_PU_GABINETE	VAZAMENTO_PU_FUNDO_POLIONDAS	VAZAMENTO_PU_CAIXA_INTERNA	AMASSAI
2021-02-11	0	0	0	0	0	0	
2021-02-12	0	0	0	0	0	0	
2021-02-13	0	0	0	0	0	0	
2021-02-15	0	0	0	0	0	0	
2021-02-16	0	0	0	0	0	0	
...	...	...	...	...	...	...	
2022-12-09	0	0	0	0	0	0	
2022-12-10	0	0	0	0	0	0	
2022-12-12	0	0	0	0	0	0	
2022-12-13	0	0	0	0	0	0	
2022-12-14	0	0	0	0	0	0	

576 rows × 21 columns

Figura 49 – Dataset do intervalo de fevereiro de 2023 até dezembro de 2023 para o produto MODELO 02.
Fonte: Elaborado pelo autor.

É possível observar as correlações entre as falhas encontradas para esse produto. Quanto mais o valor é próximo de 0.8 a 1, mais correlacionado a falha é com outra. Nesse caso, é possível identificar que nenhuma falha impacta com outra, tornando a análise de falhas de forma independente entre as falhas, conforme figuras 50 e 51:

	VAZAMENTO_PU	VAZAMENTO_PU_CANTO_PLASTICO	VAZAMENTO_PU_TRAVESSA	VAZAMENTO_PU_GABINETE	VAZAMENTO_PU_FUNDO_POLIONDAS	VAZAMENTO_PU_CAIXA_INTERNA	AMASSADO_PORTA	RISCADO_SOVADO	AMASSADO_GABINETE	IDENTIFICACAO_AUSENTE	...	LIMPEZA_INADEQUADA	FALHA_PINTURA
VAZAMENTO_PU	1.000000	NaN	0.065781	0.072166	NaN	-0.018341	-0.008379	-0.019284	0.040468	-0.018694	...	0.010042	-0.024054
VAZAMENTO_PU_CANTO_PLASTICO	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
VAZAMENTO_PU_TRAVESSA	0.065781	NaN	1.000000	0.076659	NaN	-0.017339	0.062925	0.080219	0.106515	-0.017673	...	0.057745	-0.022740
VAZAMENTO_PU_GABINETE	0.072166	NaN	0.076659	1.000000	NaN	0.109881	0.083524	0.088005	0.106155	-0.019388	...	0.111873	-0.024947
VAZAMENTO_PU_FUNDO_POLIONDAS	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
VAZAMENTO_PU_CAIXA_INTERNA	-0.018341	NaN	-0.017339	0.109881	NaN	1.000000	0.034862	0.052949	0.106163	0.038066	...	0.117376	0.005643
AMASSADO_PORTA	-0.008379	NaN	0.062925	0.083524	NaN	0.034862	1.000000	0.114034	0.221987	0.171168	...	0.069477	0.066784
RISCADO_SOVADO	-0.019284	NaN	0.080219	0.088005	NaN	0.052949	0.114034	1.000000	0.154486	0.072431	...	0.230787	0.044380
AMASSADO_GABINETE	0.040468	NaN	0.106515	0.106155	NaN	0.106163	0.221987	0.154486	1.000000	-0.032201	...	0.120854	0.102422
IDENTIFICACAO_AUSENTE	-0.018694	NaN	-0.017673	-0.019388	NaN	0.038066	0.171168	0.072431	-0.032201	1.000000	...	0.140517	0.019555
AVARIA_LOGISTICA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
LIMPEZA_INADEQUADA	0.010042	NaN	0.057745	0.111873	NaN	0.117376	0.069477	0.230787	0.120854	0.140517	...	1.000000	0.001359
FALHA_PINTURA	-0.024054	NaN	-0.022740	-0.024947	NaN	0.005643	0.066784	0.044380	0.102422	0.019555	...	0.001359	1.000000
COMPONENTE_MAL_FIXADO_SOLTO	0.162992	NaN	0.144277	0.090597	NaN	0.062178	0.171168	0.118524	0.242599	0.087041	...	0.059838	0.032355
FALHA_ELETRICA_ELETRONICA	0.015607	NaN	0.152256	0.066903	NaN	0.123670	0.168081	0.172560	0.161935	0.076034	...	0.073035	-0.044033
VAZAMENTO_GAS	-0.036057	NaN	0.041431	-0.037397	NaN	0.064677	0.164995	0.068063	0.080132	-0.003157	...	-0.051573	-0.031915
AUSENCIA_COMPONENTES	-0.039707	NaN	-0.037539	0.039228	NaN	0.108439	0.137438	0.283360	0.114738	0.101876	...	0.151498	0.144602
COMPONENTES_INTERNOS_DANIFICADOS	0.029729	NaN	0.024857	0.105675	NaN	0.088120	0.165825	0.250305	0.133292	0.108796	...	0.164163	0.021371
FALHA_VACUO	-0.032220	NaN	-0.009619	-0.033417	NaN	0.007046	-0.025818	0.069518	0.022258	-0.015963	...	0.001050	0.032752
VEDACAO_DEFICIENTE	0.035021	NaN	0.112021	0.122893	NaN	-0.005061	0.173108	0.130388	0.029485	0.130023	...	0.120872	0.016119

Figura 50 – Análise das falhas de forma independente para o produto MODELO 02.
Fonte: Elaborado pelo autor.

VAZAMENTO_PU_CAIXA_INTERNA	AMASSADO_PORTA	RISCADO_SOVADO	AMASSADO_GABINETE	IDENTIFICACAO_AUSENTE	...	LIMPEZA_INADEQUADA	FALHA_PINTURA	COMPONENTE_MAL_FIXADO_SOLTO	FALHA_ELETRICA_ELETRONICA	VAZAMENTO_GAS	AUSENCIA_COMPONENTES	COMPONENTES_INTERNOS_DANIFICADOS	FALHA_VACUO	VEDACAO_DEFICIENTE
-0.018341	-0.008379	-0.019284	0.040468	-0.018694	...	0.010042	-0.024054	0.162992	0.015607	-0.036057	-0.039707	0.029729	-0.032220	0.035021
NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
-0.017339	0.062925	0.080219	0.106515	-0.017673	...	0.057745	-0.022740	0.144277	0.152256	0.041431	-0.037539	0.024857	-0.009619	0.112021
0.109881	0.083524	0.088005	0.106155	-0.019388	...	0.111873	-0.024947	0.090597	0.066903	-0.037397	0.039228	0.105675	-0.033417	0.122893
NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1.000000	0.034862	0.052949	0.106163	0.038066	...	0.117376	0.005643	0.062178	0.123670	0.064677	0.108439	0.088120	0.007046	-0.005061
0.034862	1.000000	0.114034	0.221987	0.171168	...	0.069477	0.066784	0.171168	0.168081	0.164995	0.137438	0.165825	-0.025818	0.173108
0.052949	0.114034	1.000000	0.154486	0.072431	...	0.230787	0.044380	0.118524	0.172560	0.068063	0.283360	0.250305	0.069518	0.130388
0.106163	0.221987	0.154486	1.000000	-0.032201	...	0.120854	0.102422	0.242599	0.161935	0.080132	0.114738	0.133292	0.022258	0.029485
0.038066	0.171168	0.072431	-0.032201	1.000000	...	0.140517	0.019555	0.087041	0.076034	-0.003157	0.101876	0.108796	-0.015963	0.130023
NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
0.117376	0.069477	0.230787	0.120854	0.140517	...	1.000000	0.001359	0.059838	0.073035	-0.051573	0.151498	0.164163	0.001050	0.120872
0.005643	0.066784	0.044380	0.102422	0.019555	...	0.001359	1.000000	0.032355	-0.044033	-0.031915	0.144602	0.021371	0.032752	0.016119
0.062178	0.171168	0.118524	0.242599	0.087041	...	0.059838	0.032355	1.000000	0.255008	0.076051	0.123602	0.277728	0.097934	0.195997
0.123670	0.168081	0.172560	0.161935	0.076034	...	0.073035	-0.044033	0.255008	1.000000	-0.008613	0.093216	0.172036	0.007178	0.131181
0.064677	0.164995	0.068063	0.080132	-0.003157	...	-0.051573	-0.031915	0.076051	-0.008613	1.000000	0.076486	0.050274	0.021444	-0.021984
0.108439	0.137438	0.283360	0.114738	0.101876	...	0.151498	0.144602	0.123602	0.093216	0.076486	1.000000	0.206568	-0.024214	0.135967
0.088120	0.165825	0.250305	0.133292	0.108796	...	0.164163	0.021371	0.277728	0.172036	0.050274	0.206568	1.000000	0.041400	0.161944
0.007046	-0.025818	0.069518	0.022258	-0.015963	...	0.001050	0.032752	0.097934	0.007178	0.021444	-0.024214	0.041400	1.000000	0.150035
-0.005061	0.173108	0.130388	0.029485	0.130023	...	0.120872	0.016119	0.195997	0.131181	-0.021984	0.135967	0.161944	0.150035	1.000000

Figura 51 – Análise das falhas de forma independente para o produto MODELO 02.
Fonte: Elaborado pelo autor.

Foi possível identificar também as quantidades de falhas geradas para esse produto nesse mesmo intervalo, assim como a quantidade de OS criadas, conforme figura 52:

NrOS	5866
AMASSADO_GABINETE	350
COMPONENTES_INTERNOS_DANIFICADOS	283
FALHA_ELETRICA_ELETRONICA	249
AMASSADO_PORTA	244
COMPONENTE_MAL_FIXADO_SOLTO	210
RISCADO_SOVADO	96
FALHA_VACUO	85
VEDACAO_DEFICIENTE	78
AUSENCIA_COMPONENTES	59
VAZAMENTO_GAS	52
LIMPEZA_INADEQUADA	40
FALHA_PINTURA	28
VAZAMENTO_PU_CAIXA_INTERNA	17
IDENTIFICACAO_AUSENTE	12
VAZAMENTO_PU	11
VAZAMENTO_PU_GABINETE	10
VAZAMENTO_PU_TRAVESSA	10
VAZAMENTO_PU_CANTO_PLASTICO	0
VAZAMENTO_PU_FUNDO_POLIONDAS	0
AVARIA_LOGISTICA	0

Figura 52 – Quantidade de OS criadas e quantidade de falhas geradas para o produto MODELO 02.
Fonte: Elaborado pelo autor.

Para esse produto (MODELO 02) foi identificado que as 3 principais falhas encontradas na linha industrial AMASSADO_GABINETE, COMPONENTES_INTERNOS_DANIFICADOS e FALHA_ELETRICA_ELETRONICA. Além disso, podemos observar o comportamento dessas 3 falhas nos últimos 2 meses do conjunto de dados em estudo, conforme gráfico 5:

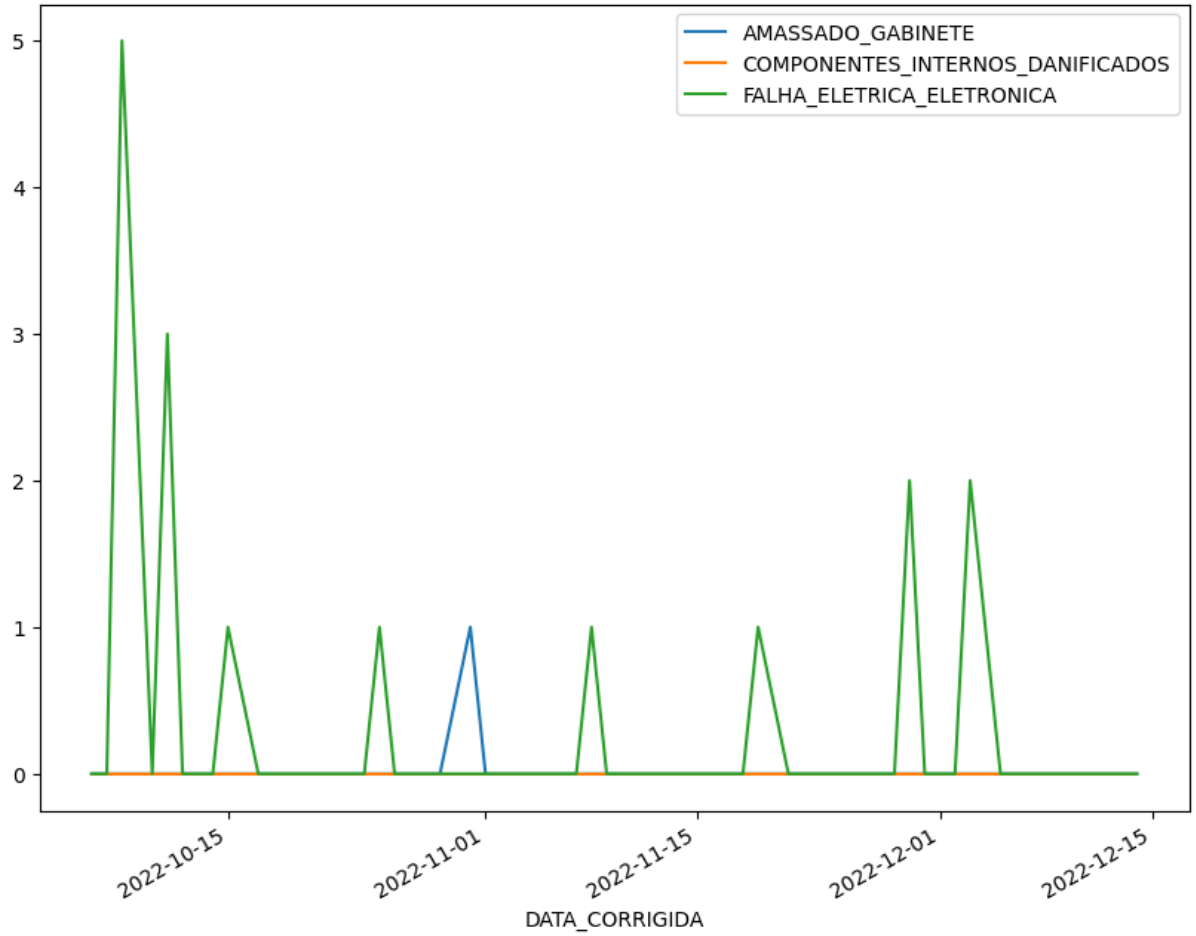

Gráfico 5 – Relação entre quantidade de OS e data de cada falha para o produto MODELO 02.
Fonte: Elaborado pelo autor.

Após aplicação das técnicas de Cross Validação com Grid Search, aplicando os algoritmos LinearRegression, RandomForest e LightGBM, foi obtido as seguintes métricas:

Linear Regression:

Best hyperparameters: {'copy_X': True, 'fit_intercept': True, 'n_jobs': None, 'positive': False}

Best metric -47.2636095380797:

Random Forest Regression:

Best hyperparameters: {'bootstrap': True, 'ccp_alpha': 0.0, 'criterion': 'squared_error', 'max_depth': 10, 'max_features': 1.0, 'max_leaf_nodes': None, 'max_samples': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1,

'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'n_estimators': 200, 'n_jobs': None, 'oob_score': False, 'random_state': 42, 'verbose': 0, 'warm_start': False}

Best metric -42.8934105274977:

LightGBM Regression:

Best hyperparameters: {'boosting_type': 'gbdt', 'class_weight': None, 'colsample_bytree': 1.0, 'importance_type': 'split', 'learning_rate': 0.05, 'max_depth': 5, 'min_child_samples': 20, 'min_child_weight': 0.001, 'min_split_gain': 0.0, 'n_estimators': 100, 'n_jobs': None, 'num_leaves': 31, 'objective': None, 'random_state': 42, 'reg_alpha': 0.0, 'reg_lambda': 0.0, 'subsample': 1.0, 'subsample_for_bin': 200000, 'subsample_freq': 0}

Best metric -40.58041005272089:

Nesse caso o modelo mais assertivo, entre os 3 aplicados, foi LightGBM com uma taxa de erro de 40,58%.

Além disso, foi possível identificar também o MSE e o MAE:

Mean Squared Error (MSE): 20.17

Mean Absolute Error (MAE): 3.26

Em uma próxima etapa, foi gerado o gráfico do treinamento realizado entre o período de fevereiro de 2021 até setembro de 2022, para o produto MODELO 01, conforme gráfico 6:

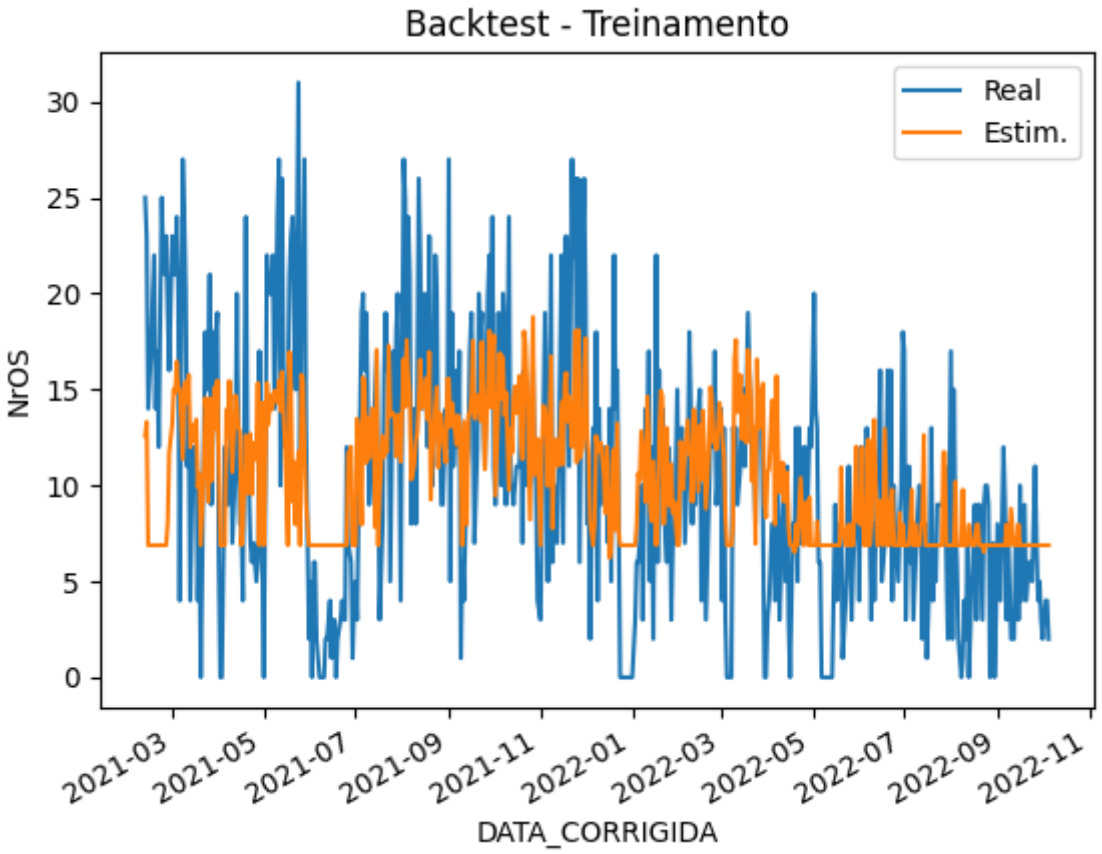

Gráfico 6 – Etapa de treinamento no período de fevereiro de 2021 e setembro de 2022 para o produto MODELO 02.
Fonte: Elaborado pelo autor.

Esse gráfico demonstra que o modelo em treinamento conseguiu capturar a informação, além de auxiliar também a entender se o modelo sofreu overfitting. Nesse caso o modelo não sofreu overfitting.

Após essa etapa, foi gerado o gráfico do teste realizado entre o período de outubro de 2022 a dezembro de 2022, para o produto MODELO 02, conforme gráfico 7:

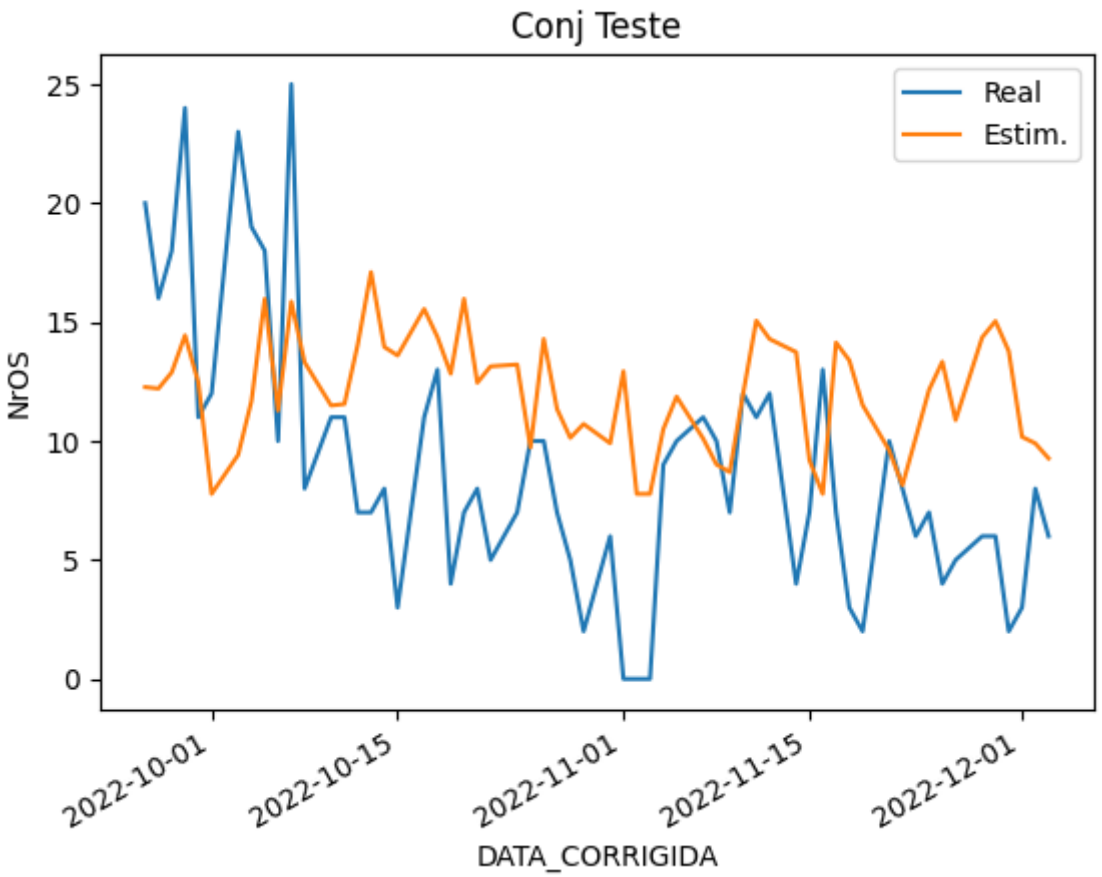

Gráfico 7 – Etapa de teste entre o período de outubro de 2022 e dezembro de 2022 para o produto MODELO 02.
Fonte: Elaborado pelo autor.

Diante disso, é possível observar que o modelo conseguiu responder bem a começou a identificar padrões para predição de número de ordens de serviço que podem ser geradas, com base nas falhas que acontecem nas linhas de produção.

Além dos resultados acima, foi possível também identificar, o quanto cada falha favoreceu para a análise realizada e submetida para o modelo de dados, através do gráfico 8:

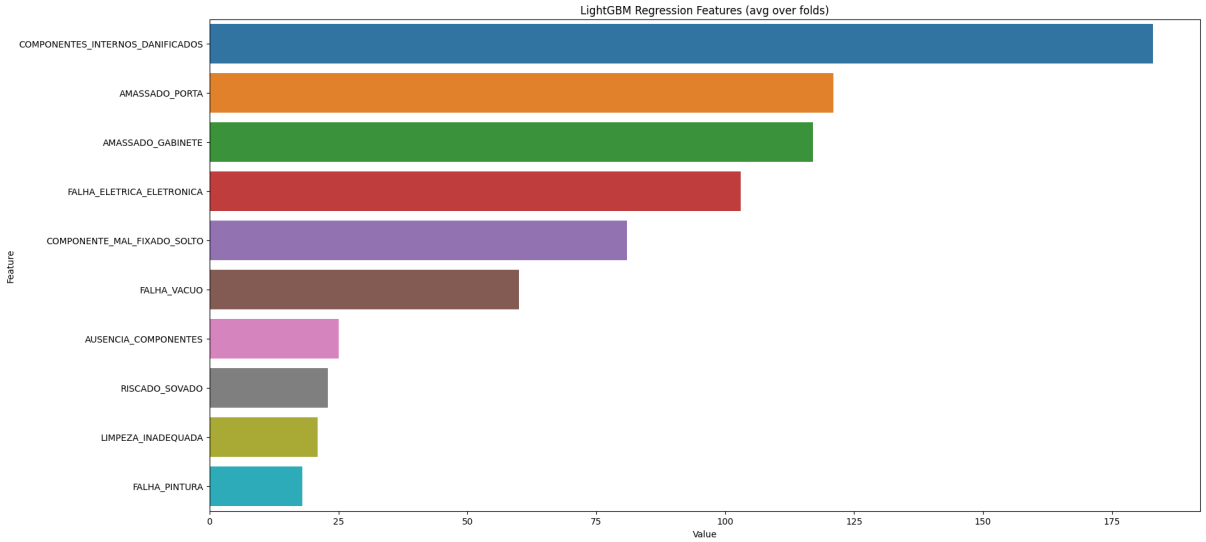

Gráfico 8 – Relação de favorecimento de cada falha para o resultado do produto MODELO 02.
Fonte: Elaborado pelo autor.

É possível perceber que COMPONENTES_INTERNOS_DANIFICADOS, AMASSADO_PORTA e AMASSADO_GABINETE, foram os principais grupos de falha que influenciaram nessa análise.

Conforme figura 53, é possível identificar (em valores) as 10 falhas mais significativas para a análise deste produto:

	Value	Feature
19	183	COMPONENTES_INTERNOS_DANIFICADOS
18	121	AMASSADO_PORTA
17	117	AMASSADO_GABINETE
16	103	FALHA_ELETRICA_ELETRONICA
15	81	COMPONENTE_MAL_FIXADO_SOLTO
14	60	FALHA_VACUO
13	25	AUSENCIA_COMPONENTES
12	23	RISCADO_SOVADO
11	21	LIMPEZA_INADEQUADA
10	18	FALHA_PINTURA

Figura 53 – As dez falhas mais significativas para a análise do produto MODELO 02.
Fonte: Elaborado pelo autor.

Em um cenário onde cada ordem de serviço tem custo médio de R$ 1.500, concluímos que:

Para o Produto Modelo 02:

- Tivemos **5.866** ordens de serviço abertas;

- Gerando um custo médio de **R$ 8.799.000,00** ao longo de **28 meses** de análise desde **Fevereiro de 2021 até Junho de 2023**;

- Tendo como retorno do modelo de dados uma assertividade de **59,42%**;

- Com isso, podendo obter uma possível economia de **R$ 5.228.365,80** com relação ao valor total gasto com OS;

- E uma redução de **3.485** OS nesse período.

4. CONCLUSÃO

Este estudo demonstrou a aplicação bem-sucedida da inteligência artificial na concepção de uma estrutura algorítmica composta de três principais algoritmos que funcionam em conjunto e técnicas de aprendizagem de máquina na otimização do controle de qualidade de uma indústria multinacional de bens de consumo, com foco na redução de ordens de serviço (OS). A análise aplicando a estrutura algorítmica construída abrangeu um período de três anos, categorizando 20 grupos de falhas em três principais categorias, correlacionando essas falhas com a geração de OS para produtos após sua fabricação.

Os resultados revelaram que as técnicas de IA representadas pela configuração da estrutura algorítmica mostrada desempenha um papel fundamental na identificação de falhas ofensivas e na previsão da quantidade de OS que podem ser geradas, com base nas falhas ocorridas na linha de montagem industrial. A aplicação de algoritmos de regressão, como Linear Regression, Random Forest e Light GBM, permitiu uma análise precisa dos dados, com taxas de erro aceitáveis, podendo gerar um impacto positivo financeiramente. Além disso, o estudo destacou as principais falhas que influenciaram a análise, oferecendo insights valiosos para o aprimoramento do processo de montagem.

Essa abordagem contribuiu para a melhoria do desempenho global da indústria, a redução de custos e o aumento da satisfação do cliente. Dessa forma, em um cenário em que a qualidade e eficiência são essenciais para o sucesso das indústrias de bens de consumo, a integração da inteligência artificial e da aprendizagem de máquina demonstrou ser uma estratégia promissora para enfrentar os desafios do controle de qualidade e da satisfação do cliente. Os resultados obtidos mostram que essa abordagem representa um passo importante em direção à excelência na produção de bens de consumo e a qualidade contínua no atendimento ao cliente.

REFERÊNCIAS

ALMEIDA, A.; CARVALHO, F.; MENINO, F. **Introdução ao Machine Learning**. Disponível em: https://dataat.github.io/introducao-ao-machine-learning/introdu%C3%A7%C3%A3o.html#machine-learning. Acesso em: 27 set. 2023.

BOSTROM, N. (2014). **Superintelligence: Paths, Dangers, Strategies**. Oxford University Press.

BREIMAN, L. (2001). **Random Forests**. Machine Learning, 45(1), 5-32.

BROOKEWENIG. **Linear Regression**. Disponível em: https://brookewenig.com/LinearRegression.html#/. Acesso em: 27 set. 2023.

CARDOSO, V., & CARDOSO, G. **Linguagem SQL Fundamentos e Práticas**. Editora Saraiva, 2013.

CHANDEKAR. **Understanding Linear Regression with Mathematical Insights!**. Disponível em: https://www.analyticsvidhya.com/blog/2021/08/understanding-linear-regression-with-mathematical-insights/. Acesso em: 21 set. 2023.

DA SILVA FILHO, J. I. da S.; ABE J. M.; TORRES, G. L. **Inteligência Artificial com as Redes de Análises Paraconsistentes: Teoria e Aplicações**. 1a. Edição. Editora LTC. Rio de Janeiro. 2008.

DOMINGOS, P. **A Máquina de Aprendizado**. Editora Paralela, 2018.

DOMINGOS, P. **“The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World”**. Editora Basic Books, 2018.

EDDY, D.; KRISHNAMURTY, S.; GROSSE, I.; WHITE, M.; BLANCHETTE D. **A Defect Prevention Concept Using Artificial Intelligence**. Volume 9: 40th Computers and Information in Engineering Conference (CIE)
ISBN: 978-0-7918-8398-3. DOI: 10.1115/DETC2020-22112

ELMASRI, R., & NAVATHE, S. B. **"Fundamentals of Database Systems"**. Editora Pearson, 2019.

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

GÉRON, A. (2019). **Hands-On Machine Learning with Scikit-Learn**, Keras, and TensorFlow. O'Reilly Media.

GOODFELLOW, I., BENGIO, Y., & COURVILLE, A. (2016). **Deep Learning**. MIT Press.

GRUS, J. (2015). **Data Science from Scratch**. O'Reilly Media.

HASTIE, T., TIBSHIRANI, R., & FRIEDMAN, J. (2009). **The Elements of Statistical Learning. Springer.**

INSIGHT. **Campo de conhecimento de inteligência artificial**. Disponível em:https://www.facebook.com/insightdatasciencelab/photos/a.2451544388425999/2782484875331947/?type=3. Acesso em: 21 set. 2023.

JAMES, G., WITTEN, D., Hastie, T., & TIBSHIRANI, R. (2013). **An Introduction to Statistical Learning**. Springer.

KE, G., MENG, Q., FINLEY, T., WANG, T., & CHEN, W. (2017). LightGBM: **A Highly Efficient Gradient Boosting Decision Tree**. Advances in Neural Information Processing Systems, 30.

LAROCK, T. **“Pro SQL Server 2019 Administration”**. Editora Microsoft Press, 2020.

KONONOW, P. **3 Basic Data Modeling Techniques - ERD, UML and Data Dictionary**. Disponível em: https://dataedo.com/blog/basic-data-modeling-techniques. Acesso em: 21 set. 2023.

LECUN, Y., BENGIO, Y., & HINTON, G. (2015). **Deep Learning**. Nature, 521(7553), 436-444.

MANOJ. **Artificial Intelligence in Synthetic Biology**. Disponível em: https://medium.com/@ananyaa.m/artificial-intelligence-in-synthetic-biology-fcfbf1c8eec. Acesso em: 21 set. 2023.

MCKINNEY, W. (2017). **Python for Data Analysis**. O'Reilly Media.

MICROSOFT. **Introducing SQL Server 2022**. Disponível em: https://www.microsoft.com/en-us/sql-server. Acesso em: 21 set. 2023.

MISTRY, R., & MISNER, S. **Introducing Microsoft SQL Server 2019**. Editora Microsoft Press, 2019.

MORALES, R.A.-G.; GARCIA MELO, J.-I.; MIYAGI, P.E. Diagnosis and treatment of faults in productive systems based on Bayesian networks and Petri net. In: Proceeding of IEEE International Conference on Automation Science and Engineering (CASE 2007), pp. 357 - 362, 2007.

MULLER, A. C., & GUIDO, S. (2016). **Introduction to Machine Learning with Python**. O'Reilly Media.

MURPHY, K. P. (2012). **Machine Learning**: A Probabilistic Perspective. MIT Press.

NIELSEN, P. **“SQL Server 2019 Fundamentals”**. Editora Microsoft Press, 2019.

NG, A. (2021). **Machine Learning**. Coursera.

PALADINI, E. P. **Controle de Qualidade: uma Abordagem Abrangente.** Editora Atlas, 1990.

PANDEY, B. **SQL in a Nutshell**. Disponível em: https://www.linkedin.com/posts/brijpandeyji_sql-doesnt-have-to-be-scary-or-overwhelming-activity-7105874918705160192-Oxdu/?trk=public_profile_like_view. Acesso em: 12 out. 2023.

PANDEY, B. **Python Roadmap**. Disponível em: https://www.linkedin.com/posts/brijpandeyji_if-youre-in-it-learning-python-can-open-activity-7125134849492815872-NDa-?utm_source=share&utm_medium=member_desktop. Acesso em: 12 out. 2023.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). **Scikit-learn: Machine learning in Python**. Journal of Machine Learning Research, 12, 2825-2830.

PESSANHA. **Random Forest: como funciona um dos algoritmos mais populares de ML**. Disponível em: https://medium.com/cinthiabpessanha/random-forest-como-funciona-um-dos-algoritmos-mais-populares-de-ml-cc1b8a58b3b4. Acesso em: 27 set. 2023.

Python.org. (s.d.). **Python Philosophy**. Retrieved from https://www.python.org/dev/peps/pep-0020/. Acesso em: 21/09/2023

RAINERGEWALT. **scikit-learn – Machine learning, Data Mining and Data Analysis in Python for free**. Disponível em: https://starship-knowledge.com/tag/scikit-learn-features. Acesso em: 27 set. 2023.

RAMAKRISHNAN, R., & Gehrke, J. **“Database Management Systems”**. Editora McGraw-Hill, 2003.

RENNER. **Projeto Analytics -Modelos pré-churn**. Disponível em: https://medium.com/@fintechrealize/projeto-analytics-modelos-pr%C3%A9-churn-612890d42f9. Acesso em: 27 set. 2023.

RIASCOS, L.A.M. **Metodologia para detecção e tratamento de falhas em sistemas de manufatura através de rede de Petri**, Tese de doutorado, Escola Politécnica da Universidade de São Paulo, 2002.

RIASCOS, L.A.M. MIYAGI, P.E. **Modeling and analysis of fault-tolerant systems for machining operations based on Petri nets**, Controle Engineering Practice, vol 14, pp. 397-408, 2006.

RUSSEL, S. J., & Norvig, P. **"Artificial Intelligence: A Modern Approach"**. Editora Pearson, 2021.

SANTANA. **Regressão Linear com Sklearn: Conceito e Aplicação**. Disponível em: https://medium.com/@lamartine_sl/regress%C3%A3o-linear-com-sklearn-modelo-de-previs%C3%A3o-de-custos-com-plano-de-sa%C3%BAde-5e963e590f4c#:~:text=Regress%C3%A3o%20Linear%20%C3%A9%20uma%20equa%C3%A7%C3%A3o,a%20rela%C3%A7%C3%A3o%20das%20suas%20vari%C3%A1veis. Acesso em: 27 set. 2023.

SEVERANCE. **Python for Everybody**. Disponível em: www.py4e.com. . Acesso em: Acesso em: 27 set. 2023.

SILBERSCHATZ, A., KORTH, H. F., & SUDARSHAN, S. **"Database System Concepts"**. Editora McGraw-Hill, 2020.

Testgorilla. **Top 11 advantages of Pandas for data analysis**. Disponível em: https://www.testgorilla.com/blog/pandas-for-data-analysis/. Acesso em: 27 set. 2023.

THOMPSON, R. **"Manufacturing Processes for Design Professionals"**. Editora Thames & Hudson, 2007.

VANDERPLAS, J. (2016). **Python Data Science Handbook**. O'Reilly Media.