Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

RODRIGO ROCHA JOSÉ

MINERAÇÃO DE DADOS: CONCEITOS FUNDAMENTAIS E APLICAÇÃO PRÁTICA NA AGRICULTURA

Assis/SP
2024

Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

RODRIGO ROCHA JOSÉ

MINERAÇÃO DE DADOS: CONCEITOS FUNDAMENTAIS E APLICAÇÃO PRÁTICA NA AGRICULTURA

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito à obtenção do Certificado de Conclusão.

Orientando(a): Rodrigo Rocha José
Orientador(a): Dr. Almir Rogério Camolesi

Assis/SP
2024

FICHA CATALOGRÁFICA

José, Rodrigo Rocha

J831m Mineração de dados: conceitos fundamentais e aplicação prática na agricultura / Rodrigo Rocha José. -- Assis, 2024.

51p.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) -- Fundação Educacional do Município de Assis (FEMA), Instituto Municipal de Ensino Superior de Assis (IMESA), 2024.

Orientador: Prof. Dr. Almir Rogério Camolesi.

Elaborada por Anna Carolina Antunes de Moraes – Bibliotecária – CRB-8/10982

MINERAÇÃO DE DADOS AGRÍCOLAS : CONCEITOS FUNDAMENTAIS E APLICAÇÃO PRÁTICA NA AGRICULTURA

RODRIGO ROCHA JOSÉ

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Dr. Almir Rogério Camolesi

Examinador: _____
Esp. Célio Desiró

Assis/SP
2024

DEDICATÓRIA

Dedico este trabalho a todos que sempre acreditaram em mim e me motivaram a buscar constantemente a realização dos meus objetivos.

AGRADECIMENTOS

Primeiramente, agradeço a todos os meus familiares, que sempre se dedicaram ao meu desenvolvimento pessoal e profissional.

Agradeço ao meu professor, orientador, amigo e colega de trabalho, Dr. Almir Rogério Camolesi, que, além de compartilhar valiosos conhecimentos em sala de aula, também me proporcionou diversas oportunidades profissionais.

Agradeço à minha namorada, Clara, por seu constante apoio e motivação nos momentos em que me senti mais cansado ou frustrado ao longo da graduação.

Sou grato também a todas as pessoas que, em algum momento, contribuíram para o meu crescimento, oferecendo-me oportunidades, ensinando ou auxiliando.

E, por fim, agradeço a mim mesmo por ter chegado até aqui, superando todas as barreiras que encontrei ao longo do caminho.

RESUMO

O trabalho apresenta a aplicação da mineração de dados como uma ferramenta estratégica para transformar grandes volumes de dados em informações úteis, com o objetivo de demonstrar sua relevância na otimização de processos e na tomada de decisões em diferentes setores. Para ilustrar os benefícios e a aplicabilidade da mineração de dados, foi realizado um exemplo prático utilizando dados do setor agrícola. A pesquisa mostrou que, embora essa área seja frequentemente considerada complexa, é possível alcançar resultados relevantes mesmo sem seguir rigorosamente todas as etapas tradicionais do processo. Conclui-se que a mineração de dados pode ser aplicada de maneira flexível e acessível, proporcionando insights valiosos e contribuindo para o avanço do conhecimento em diversas áreas.

Palavras-chave: 1. Mineração de Dados 2. Análise de dados

ABSTRACT

This work presents the application of data mining as a strategic tool to transform large volumes of data into useful information, with the aim of demonstrating its relevance in optimizing processes and decision-making across different sectors. To illustrate the benefits and applicability of data mining, a practical example was conducted using data from the agricultural sector. The research showed that, although this area is often considered complex, it is possible to achieve significant results even without strictly following all traditional stages of the process. It is concluded that data mining can be applied flexibly and accessibly, providing valuable insights and contributing to the advancement of knowledge in various fields.

Keywords: 1. Data Mining. 2. Data Analysis

LISTA DE ILUSTRAÇÕES

Figura 1 : Diferença entre dados, informação e conhecimento 15

Figura 2 : Processo de mineração de dados (FAYYAD, 1996) 16

Figura 3 : Processo de agrupamento de dados 22

Figura 4 : Principais Componentes de uma Colhedora 25

Figura 5 : Tabela após aplicação do PIVOT 30

Figura 6 : Função para tratar valores ausentes 32

Figura 7 : DataFrame antes do tratamento de ausência de valores 33

Figura 8 : DataFrame após o tratamento de ausência de valores 34

Figura 9 : Gráfico demonstrando ausência de valores 38

Figura 10 : Gráfico demonstrando dados com ruídos 39

Figura 11 : Gráfico demonstrando valores inconsistentes 40

Figura 12 : Lógica de Classificação do modo de operação 41

Figura 13 : Função responsável por classificar o modo de operação das colhedoras 42

Figura 14 : Funções de classificação 43

Figura 15 : Função para calcular o tempo restante de combustível 45

Figura 16 : Aplicação das funções para obter o tempo restante até o reabastecimento 46

Figura 17 : Resultado após a aplicação das funções 46

LISTA DE TABELAS

Tabela 1 : Dados brutos de pessoas e seus gêneros 18

Tabela 2 : Dados transformados de pessoas e seus gêneros 18

Tabela 3 : Exemplo de Transformação de dados..... 18

Tabela 4 : Dados brutos de temperatura nas cidades..... 19

Tabela 5 : Dados discretizados de temperatura nas cidades..... 19

Tabela 6 : Exemplo de Discretização de dados 19

Tabela 7 : Descrição dos principais componentes de uma colhedora de cana-de-açucar 26

Tabela 8 : Descrição das colunas da base de dados..... 27

LISTA DE ABREVIATURAS E SIGLAS

MD Mineração de Dados
KDD Knowledge discovery in databases - Descoberta de conhecimento em base de dados

SUMÁRIO

1. INTRODUÇÃO 12

1.1. OBJETIVOS 13

1.2. JUSTIFICATIVAS 13

1.3. MOTIVAÇÃO 13

1.4. METODOLOGIA 14

1.5. PERSPECTIVAS DE CONTRIBUIÇÃO 14

1.6. ESTRUTURA DO TRABALHO 14

2. MINERAÇÃO DE DADOS 15

2.1. CONCEITO DE MINERAÇÃO DE DADOS 15

2.2. TAREFAS DA MINERAÇÃO DE DADOS 16

2.2.1. Pré Processamento de dados 17

2.2.2. Análise descritiva dos dados 20

2.2.3. Análise de grupos 21

3. PROPOSTA 24

3.1. PROPOSTA E METODOLOGIA 24

3.2. VISÃO GERAL DAS COLHEDORAS E DA BASE DE DADOS 24

3.2.1. Conhecendo os componentes da colhedora 25

3.2.2. Descrição da base de dados 26

3.2.3. Descrição das variáveis 27

4. IMPLEMENTAÇÃO 30

4.1. LIMPEZA DOS DADOS 31

4.1.1. Valores Ausentes 31

4.1.2. Dados com rúidos 34

4.1.3. Dados inconsistentes 36

4.1.4. Transformação dos Dados 37

4.2. ANÁLISE 38

4.2.1. Exploração de um Problema e Solução Proposta 44

5. CONCLUSÃO 47

6. TRABALHOS FUTUROS 48

7. REFERÊNCIAS 50

1. INTRODUÇÃO

O volume de dados gerados pelas empresas atualmente é enorme e cresce exponencialmente a cada ano. De acordo com o estudo "Seagate Data Age 2025" (Reinsel, Gantz & Rydning, 2017), o volume global de dados atingirá a marca impressionante de 175 zettabytes até 2025, representando um aumento significativo em relação aos 64 zettabytes gerados em 2020.

Os dados brutos, gerados diariamente, por si só não possuem valor imediato ou significativo. No entanto, ao aplicar técnicas de análise e interpretação sobre esses dados, é possível extrair informações valiosas. Essas informações, por sua vez, podem ser transformadas em conhecimento útil, auxiliando a tomada de decisões. Nesse quesito surge a mineração de dados.

A mineração de dados é uma ferramenta amplamente aplicável em diversas áreas as quais geram volumes significativos de dados diariamente. Tendo em vista essa imensa quantidade de informações disponíveis, é de suma importância dedicar esforços para estudar, analisar esses dados e gerar conhecimentos valiosos.

Este estudo tem como objetivo apresentar a mineração de dados, seus processos e demonstrar como essa área pode contribuir para a tomada de decisões mais eficazes, a otimização de processos e a descoberta de insights valiosos a partir de grandes volumes de dados. Além disso, busca-se fomentar o interesse em mineração de dados entre estudantes e empresas, promovendo uma maior integração com as tecnologias emergentes. Com o intuito de facilitar a replicação e o aprofundamento da pesquisa, este estudo também apresenta a base de dados utilizada, simplificando seu entendimento e oferecendo um guia para sua exploração em futuros trabalhos.

1.1. OBJETIVOS

O objetivo desse trabalho foi realizar um estudo relacionado a mineração de dados e seus processos, aproximar tal conhecimento as empresas em geral e apresentar na prática, por meio de algoritmos, utilizando o setor agrícola como exemplo, os benefícios que a mineração de dados pode oferecer.

1.2. JUSTIFICATIVAS

A ciência de dados assume uma importância vital na solução de desafios complexos e no auxílio de tomada de decisões. Conforme progredimos e a quantidade de dados disponíveis cresce, é esperado que a demanda por profissionais qualificados aumente ainda mais. De acordo com a pesquisa "Panorama das Carreiras 2030: Previsões para o Futuro Profissional", a ocupação de analista e cientista de dados emergirá como uma das mais cruciais no cenário do emprego no Brasil até o ano de 2030.

Portanto, é esperado que este estudo contribua para destacar a relevância da especialização em ciência de dados entre os profissionais, incentivando o surgimento de novos estudos e iniciativas nessa área.

1.3. MOTIVAÇÃO

A motivação deste trabalho foi impulsionar o avanço da área de ciência de dados, especialmente em regiões onde sua aplicação ainda é limitada e pouco explorada. Considerando a crescente demanda por profissionais qualificados, espera se que esse trabalho contribua para o estudo e conhecimento de todos que desejam explorar e entender a área.

1.4. METODOLOGIA

A metodologia adotada neste trabalho segue uma abordagem estruturada que visa fornecer uma compreensão abrangente dos conceitos fundamentais da mineração de dados, seguida pela explicação da base de dados utilizada e conceitos auxiliares para o entendimento da mesma e, por fim, a aplicação prática desses conceitos utilizando Python e a biblioteca Pandas.

1.5. PERSPECTIVAS DE CONTRIBUIÇÃO

O presente trabalho buscou contribuir regionalmente, para as empresas e também aos estudantes, demonstrando os benefícios práticos do uso da mineração de dados e a relevância de investir nessa área. Além da contribuição com meu próprio desenvolvimento profissional.

1.6. ESTRUTURA DO TRABALHO

O trabalho está estruturado em seis seções principais. A seção de Introdução (1) contextualiza o tema abordado, apresentando seus objetivos, justificativas, motivação, perspectivas de contribuição e metodologias utilizadas. Na seção de Mineração de Dados (2), é explicado o conceito de Mineração de dados, descrito algumas de suas etapas e métodos que podem ser utilizados, assim como exemplos. Na seção (3) Proposta são apresentados a proposta, demonstrado alguns conceitos fundamentais para o entendimento da base de dados. Na seção (4) Implementação é demonstrado a implentação dos algoritmos e o processo da análise dos dados. A seção de Conclusões (5) apresenta os resultados obtidos e as conclusões. A seção Trabalhos Futuros (6) contém observações obtidas durante a análise e algumas sugestões de trabalhos futuros que podem ser realizados a partir deste. Por fim, a seção de Referências (7) lista todo o referencial teórico utilizado no trabalho.

2. MINERAÇÃO DE DADOS

Nesse tópico será introduzida a mineração de dados, seus conceitos básicos e alguns dos principais métodos utilizados.

2.1. CONCEITO DE MINERAÇÃO DE DADOS

De acordo com CASTRO (2016), o processo de mineração de dados é descrito como :

"*O processo de mineração corresponde à extração de minerais valiosos, como ouro e pedras preciosas, a partir de uma mina. Uma característica importante desses materiais é que, embora não possam ser cultivados ou produzidos artificialmente, existem de maneira implícita e muitas vezes desconhecida em alguma fonte, podendo ser extraídos. Esse processo requer acesso à mina, o uso de ferramentas adequadas de mineração, a extração dos minérios propriamente dita e o seu posterior preparo para comercialização. O termo mineração de dados (MD) foi cunhado como alusão ao processo de mineração descrito anteriormente, uma vez que se explora uma base de dados (mina) usando algoritmos (ferramentas) adequados para obter conhecimento (minerais preciosos). Os dados são símbolos ou signos não estruturados, sem significado, como valores em uma tabela, e a informação está contida nas descrições, agregando significado e utilidade aos dados, como o valor da temperatura do ar. Por fim, o conhecimento é algo que permite uma tomada de decisão para a agregação de valor, então, por exemplo, saber, que vai chover no fim de semana pode influenciar sua decisão de viajar ou não para a praia.*"

Figura 1 : Diferença entre dados, informação e conhecimento

A mineração de dados, embora frequentemente confundida com o KDD (Knowledge Discovery in Databases), é apenas uma etapa dentro de um processo mais amplo de descoberta de conhecimento em bancos de dados. O KDD, definido por Fayyad et al. (1996) como um processo que abrange "desde a seleção e preparação dos dados até a interpretação dos resultados", envolve a aplicação de técnicas de mineração de dados, mas também engloba outras etapas como a seleção de dados, a preparação dos dados e a interpretação dos resultados. Enquanto a mineração de dados se concentra na aplicação de algoritmos para extrair padrões e informações relevantes, o KDD busca transformar dados brutos em conhecimento útil e acionável, garantindo uma abordagem abrangente para a análise de dados.

2.2. TAREFAS DA MINERAÇÃO DE DADOS

Neste tópico serão apresentadas as principais tarefas que compõem a mineração de dados de maneira sucinta, sem aprofundamentos, visando proporcionar compreensão básica das tarefas que serão aplicadas na prática posteriormente.

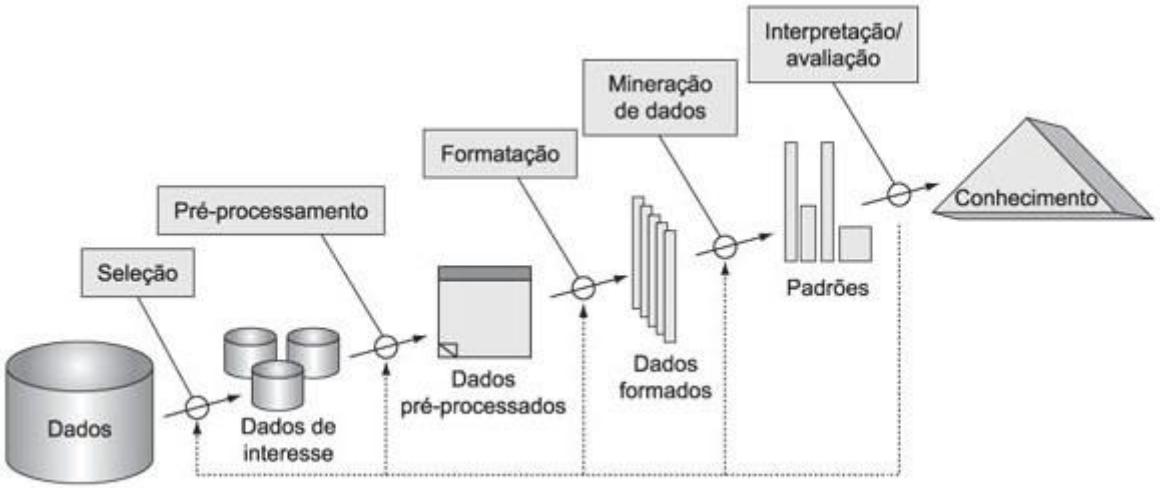

Figura 2 : Processo de mineração de dados (FAYYAD, 1996)

Essas tarefas são iterativas e interativas pois podemos fazer essas tarefas em qualquer momento e também podemos refazer essas tarefas caso haja necessidade.

2.2.1. Pré-Processamento de dados

Esta tarefa é responsável por preparar os dados brutos, integrando fontes de dados diferentes, imputando valores ausentes, reduzindo o tamanho do conjunto de dados para facilitar o processamento e uniformizando os dados em um formato consistente. Segundo Han et al. (2011, p. 83), o pré-processamento de dados é uma etapa essencial para garantir a qualidade e a precisão dos resultados obtidos em etapas posteriores da análise de dados.

As etapas do pré-processamento de dados que serão discutidas a seguir incluem técnicas e métodos específicos para alcançar seus objetivos, no entanto, não serão abordados todos os métodos e técnicas existentes.

2.2.1.1. Limpeza

A limpeza de dados é uma etapa que envolve a identificação e correção de erros, inconsistências e valores ausentes nos dados brutos (Witten, Frank & Hall, 2016), tornando-os mais consistentes e confiáveis para análise posterior. Os métodos de limpeza dos dados atuam no sentido de imputação ou preenchimento de valores ausentes, remoção de ruídos, identificação de valores discrepantes e correção de inconsistências.

2.2.1.2. Integração

Este processo é crucial quando os dados estão dispersos em diferentes fontes e formatos e precisam ser unificados para análise ou modelagem. A integração dos dados atua na concatenação desses dados de fontes diferentes em uma única base de dados.

2.2.1.3. Redução

O processo de redução é responsável por diminuir o tamanho da base de dados (linhas ou colunas) com o objetivo de simplificar o processamento, mantendo a integridade dos dados.

"É intuitivo pensar que, quanto maior a quantidade de objetos e atributos, mais informações estão disponíveis para o algoritmo de mineração de dados. Entretanto, o aumento do número de objetos e da dimensão do espaço (número de atributos na base) pode fazer com que os dados disponíveis se tornem esparsos e as medidas matemáticas usadas na análise tornem-se numericamente instáveis. Além disso, uma quantidade muito grande de objetos e atributos pode tornar o processamento dos algoritmos de mineração muito complexo, assim como os modelos gerados." (CASTRO, 2016)

2.2.1.4. Transformação

É comum nos depararmos com situações em que dados idênticos são inseridos de formas distintas, por exemplo, valores em unidades de medida diferentes. A transformação de dados desempenha o papel de padronizar e preparar os dados em um formato passível de aplicação das diferentes técnicas de mineração.

Nome	**Gênero**
João Victor	M
Pedro Paulo	masculino
Alicia Silva	Feminino
Erica Santos	fem
Lucas Junior	Homem
Bruna Souza	f

Tabela 1 : Dados brutos de pessoas e seus gêneros

>>>

Nome	**Gênero**
João Victor	M
Pedro Paulo	M
Alicia Silva	F
Erica Santos	F
Lucas Junior	M
Bruna Souza	F

Tabela 2 : Dados transformados de pessoas e seus gêneros

Tabela 3 : Exemplo de Transformação de dados

A Tabela 3 mostra a padronização dos dados, um dos métodos de transformação de dados que transforma os dados para garantir que eles sigam um padrão ou formato consistente. Inicialmente na Tabela 1 , a coluna "Gênero" continha valores variados e inconsistentes, como "M", "masculino", "Feminino", "fem", "homem", e "f". Após a transformação, todos esses valores foram padronizados de forma consistente, utilizando

apenas "M" para representar o gênero masculino e "F" para o feminino, como pode ser observado na Tabela 2.

2.2.1.5. Discretização

A discretização é responsável por transformar os valores dos atributos contínuos em dados categóricos, facilitando a aplicação de métodos que operam apenas com atributos nominais e permitindo que um conjunto maior de problemas seja abordado. Além disso, reduz a quantidade de valores para um dado atributo ao agrupar valores próximos, conforme discutido por Silva (2020).

Cidade	**Temperatura**
Assis	22.0
São Paulo	27.8
Salvador	36.0
Campinas	23.5
São Caetano	28.9
Palmital	33.5

Tabela 4 : Dados brutos de temperatura nas cidades

>>>

Cidade	**Temperatura**
Assis	Baixa
São Paulo	Média
Salvador	Alta
Campinas	Baixa
São Caetano	Média
Palmital	Alta

Tabela 5 : Dados discretizados de temperatura nas cidades

Tabela 6 : Exemplo de Discretização de dados

A Tabela 6 ilustra um processo de discretização aplicado à coluna "Temperatura" da Tabela 4. Originalmente, a coluna continha valores numéricos específicos representando as temperaturas em diferentes cidades. Após a discretização, esses valores foram categorizados em três faixas: "Baixa", "Média" e "Alta" (Tabela 5). Por exemplo, temperaturas como 22.0°C e 23.5°C foram classificadas como "Baixa", 27.8°C e 28.9°C foram classificadas como "Média", enquanto 33.5°C e 36.0°C foram classificadas como "Alta".

2.2.2. Análise descritiva dos dados

A análise descritiva de dados é utilizada para descrever e resumir as principais características de uma base de dados, além de auxiliar na detecção de quaisquer anomalias ou erros, como valores ausentes ou pontos discrepantes. Esta etapa é fundamental, pois, antes de adentrar na exploração dos dados em busca de conhecimento, é necessário compreender claramente o que está sendo analisado. Segundo Montgomery e Runger (2010), a análise descritiva é essencial para a compreensão inicial dos dados e para a identificação de padrões e irregularidades.

As análises descritivas podem ser univariadas ou bivariadas. As análises univariadas focam na descrição de um único atributo, organizando os dados em distribuições de frequência, visualizando através de gráficos, e identificando tendências centrais e variações. Field (2013) destaca que a análise univariada é crucial para entender a distribuição e a variabilidade de um único conjunto de dados, utilizando medidas como média, mediana, moda, variância e desvio padrão. Por outro lado, as análises bivariadas exploram relações entre pares de atributos, buscando compreender causas e correlações. Hair et al. (2010) explicam que a análise bivariada é fundamental para identificar e quantificar relações entre duas variáveis, utilizando técnicas como correlação e regressão.

O processo de análise descritiva dos dados pode ser desmembrado em três passos principais:

2.2.2.1. Distribuições de frequência

A distribuição de frequências é uma representação resumida dos dados, onde eles são agrupados em classes distintas e exclusivas, mostrando quantas vezes cada classe ocorre. Essa técnica é aplicável tanto a dados numéricos quanto categóricos, e é útil para compreender a distribuição dos dados e identificar padrões ou tendências.

2.2.2.2. Apresentação dos dados

A apresentação de dados em gráficos na análise descritiva visa facilitar a compreensão das distribuições dos dados, tornando a extração de conhecimento mais eficiente e permitindo o compartilhamento direto desse conhecimento entre diferentes públicos e entidades.

2.2.2.3. Medidas de resumo

Algumas medidas podem ser empregadas para resumir a informação presente nas distribuições de probabilidade de um atributo ou nos conjuntos de dados. Três tipos principais de medidas são fundamentais para essa finalidade: as medidas de tendência central, as medidas de dispersão e as medidas de forma da distribuição. Tais medidas oferecem uma compreensão abrangente das características dos dados e são essenciais para a análise e interpretação dos resultados

2.2.3. Análise de grupos

A Análise de grupos, também conhecida como agrupamento de dados ou clustering, é uma técnica de mineração de dados que envolve a divisão de um conjunto de dados em subconjuntos de modo que os dados em cada subconjunto compartilhem características comuns. Segundo Han, Kamber e Pei (2012), o clustering é uma técnica fundamental em mineração de dados, pois permite a organização de grandes volumes de dados em grupos que são mais fáceis de analisar e interpretar.

A organização de objetos em grupos distintos pode servir como uma maneira prática de estruturar conjuntos extensos de dados, tornando-os mais acessíveis e compreensíveis.

Além disso, essa prática pode ser empregada para a realização de tarefas complexas, como a tomada de decisões em processos cruciais. Berkhin (2006) explica que o clustering pode ser utilizado em diversas aplicações práticas, como segmentação de mercado, detecção de fraudes e análise de redes sociais, auxiliando na tomada de decisões estratégicas.

O agrupamento de dados é um processo que pode ser dividido em cinco etapas principais.

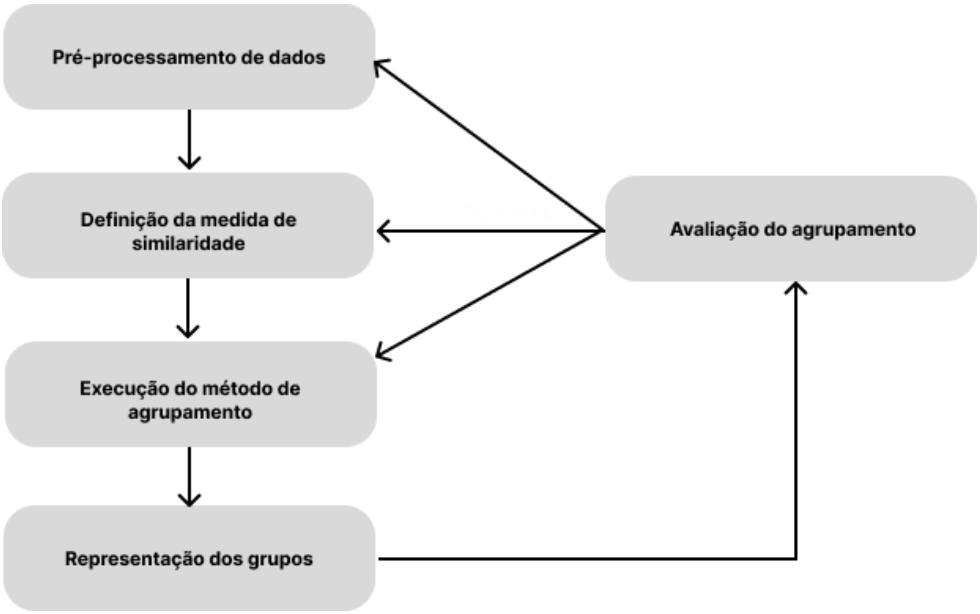

Figura 3 : Processo de agrupamento de dados

2.2.3.1. Definição da Medida de Similaridade

A escolha da métrica de similaridade é crucial, pois ela determina como a “distância” entre os pontos de dados é calculada. A escolha da métrica depende da natureza dos dados e do problema em questão.

2.2.3.2. Execução do Método de Agrupamento

Existem muitos algoritmos de agrupamento disponíveis, cada um com suas próprias características e suposições. A escolha do algoritmo depende da estrutura dos dados e do objetivo da análise. Podemos pensar em dois tipos principais de agrupamento: o particional, que define grupos fixos e organiza os dados dentro deles, e o hierárquico, que cria uma hierarquia de grupos mostrando as relações entre os grupos. A escolha do

algoritmo certo é crucial para a qualidade dos resultados e para entender o que os grupos encontrados significam. É importante analisar cuidadosamente os dados e o objetivo da análise para escolher o algoritmo que melhor se adapta à situação.

2.2.3.3. Representação dos Grupos

Após a execução do algoritmo de agrupamento, é útil visualizar os grupos formados. Isso pode ser feito através de técnicas de visualização de dados, como gráficos de dispersão ou mapas de calor. A visualização ajuda a interpretar os resultados e a entender a estrutura dos dados.

2.2.3.4. Avaliação do Agrupamento

A avaliação da qualidade dos grupos é uma etapa crucial para assegurar que os clusters sejam significativos e úteis. De acordo com Tan, Steinbach e Kumar (2019), existem várias medidas de validação, que podem ser classificadas como internas ou externas. As medidas internas avaliam a qualidade dos clusters com base apenas nos dados de entrada, considerando aspectos como a coesão e a separação dos clusters. Por outro lado, as medidas externas comparam os clusters obtidos com uma estrutura de grupo conhecida ou "verdadeira", permitindo uma avaliação mais objetiva da precisão dos clusters gerados.

3. PROPOSTA

Nessa seção, serão apresentadas a proposta do trabalho e também os conceitos essenciais para o entendimento da base de dados.

3.1. PROPOSTA E METODOLOGIA

A proposta deste estudo é desenvolver um material que sirva como guia introdutório para qualquer pessoa interessada no tema de mineração de dados e análise de dados. Além disso, busca-se simplificar o entendimento da base de dados utilizada, tendo em vista que esta será empregada em trabalhos futuros. Por fim, o estudo visa auxiliar a empresa fornecedora da base de dados na otimização de processos e, possivelmente, na realização de predições com os dados fornecidos.

A base de dados utilizada neste trabalho, foi disponibilizada pela empresa NovAmerica Agro - Tarumã, SP, Brasil, fornecedora agrícola de matéria-prima que atua principalmente no segmento de produção de cana-de-açúcar. A base de dados engloba um conjunto completo de informações sobre as colhedoras de cana-de-açúcar, incluindo dados operacionais, parâmetros de desempenho e informações relevantes para a análise da produção.

Este trabalho propõe o desenvolvimento de um algoritmo em Python. Para sua elaboração foram utilizadas as bibliotecas Pandas, Numpy e Matplotlib. O Pandas oferece ferramentas para manipulação e análise de dados de forma eficiente e intuitiva, enquanto o Matplotlib fornece uma interface poderosa para a visualização de dados. O NumPy, fornece estruturas de dados eficientes para arrays multidimensionais e uma vasta gama de funções matemáticas otimizadas para trabalhar com esses arrays.

3.2. VISÃO GERAL DAS COLHEDORAS E DA BASE DE DADOS

Nesta seção, são introduzidos os conceitos essenciais para a análise, abordando o processo de colheita da cana-de-açúcar, os componentes das máquinas colhedoras e suas funções, além da descrição da base de dados utilizada.

3.2.1. Conhecendo os componentes da colhedora

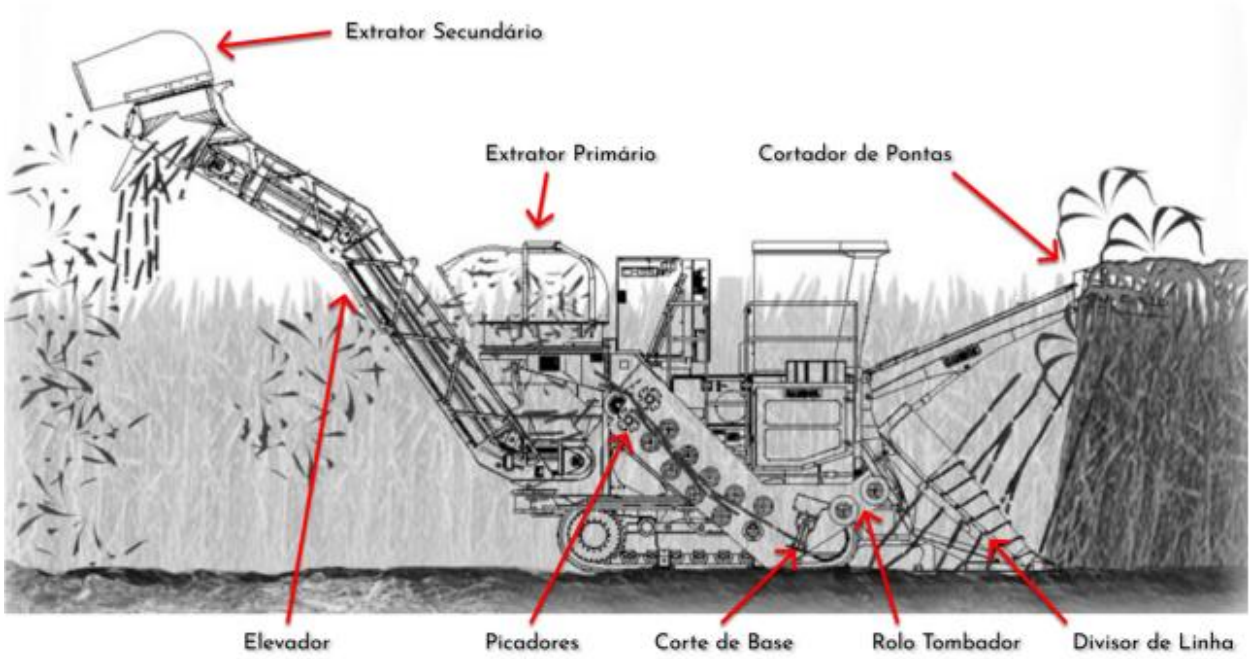

Figura 4 : Principais Componentes de uma Colhedora

A Figura 4, acima, apresenta o esquema de funcionamento de uma máquina colhedora de cana. Na figura pode ser observado todo o processo, desde o corte da cana-de-acúcar até o seu depósito em veículo de transporte e descarte de palhada. Na referida figura temos:

Componente	**Descrição**
Cortador de Pontas	Corta as pontas da cana, geralmente onde há menos açúcar, para otimizar o processamento da parte mais rica da planta.
Divisor de Linha	Separa as linhas de cana para que a colhedora possa operar de forma mais eficiente, evitando embaraçamentos e otimizando o corte.

Rolo Tombador	Auxilia no derrubamento da cana, facilitando o corte e a coleta.
Corte de Base	Realiza o corte rente ao solo, separando a cana da raiz de maneira eficiente.
Picadores	Pica a cana em pedaços menores, facilitando o transporte e processamento subsequente.
Elevador	Transporta a cana picada para um veículo ou armazenamento, mantendo o fluxo contínuo de material.
Extrator Primário	Atua como a primeira etapa de limpeza, removendo a maior parte das impurezas como folhas e terra, antes que a cana passe para a fase de processamento.
Extrator Secundário	Responsável por separar a palha do caldo da cana, removendo as impurezas após o corte.

Tabela 7 : Descrição dos principais componentes de uma colhedora de cana-de-açucar

3.2.2. Descrição da base de dados

Para este estudo foram disponibilizados registros de um maquinário durante o período de Julho à Setembro de 2023. A base de dados completa é composta por 12 arquivos e abrange informações operacionais sobre o maquinário como nível de combustível, temperatura do arrefecedor do motor, pressão do óleo do motor, fluxo de cana, peso da cana e diversas outras informações.

3.2.3. Descrição das variáveis

Neste estudo, a análise dos dados foi realizada com uma amostra dos dados de Setembro de 2023. A amostra utilizada contém 743.510 linhas. Abaixo serão apresentadas as colunas da base de dados.

Campo	Descrição
COMPANYID	ID da Companhia
CEQID	ID do Colhedora
CANID	ID do Tipo de Registro
DATA	Data do Registro
DT_INTEGRACAO	Data da Integração do Registro
LAT	Latitude
LON	Longitude
ALT	Altitude
DIRECTION	Direção da Colhedora
SPEED	Velocidade da Colhedora
FAMILYCODE	Descrição do Tipo de Registro
FUNCTION	Função aplicada para obter o valor do registro
ERRORTAGS	Tags de Erro
TYPE	Tipo do Registro
TEXT	Valor do Registro Caso tipo for Texto
UNIFIEDVALUE	Valor do Registro Arredondado caso tipo for numérico
UOM	Unidade de medida do Registro
VALUE	Valor do Registro caso tipo for numérico

Tabela 8 : Descrição das colunas da base de dados

A coluna "FAMILYCODE" representa diferentes componentes ou parâmetros associados a equipamentos e máquinas. Cada valor nesta coluna corresponde a um tipo específico de dado coletado dos equipamentos, que pode incluir status, medições ou outras informações relevantes. Como existem 93 variáveis para essa coluna, serão descritas abaixo somente as colunas/variáveis que serão utilizadas nesse estudo.

- **FUEL_LEVEL (Nível de Combustível):** O Nível de combustível é um parâmetro que mede em porcentagem a quantidade de combustível em cada colhedora. Analisando essa variável podemos identificar se a colhedora precisa de reabastecimento.

- **BC_HYD_PRESS (Pressão do Corte de Base):** A pressão do corte de base (BC_HYD_PRESS) é um parâmetro que mede a força hidráulica aplicada durante o processo de corte da colhedora. Monitorando essa pressão, é possível identificar se a colhedora está realizando o corte ou não.

- **GROUND_SPEED (Velocidade Efetiva da Máquina):** A velocidade efetiva da máquina (GROUND_SPEED) é um parâmetro que indica a velocidade com a qual a colhedora está se movimentando. Com este parâmetro, podemos identificar diferentes estados de operação, como:

  Parado: Quando a velocidade é zero, indicando que a máquina não está em movimento.

  Manobra: Quando a velocidade é baixa, o que pode indicar que a máquina em movimento para uma nova posição de trabalho.

  Deslocamento: Quando a velocidade é maior, sugerindo que a máquina está em movimento para a área de trabalho ou retornando dela.

- **ELEVATOR_STATUS (Status do Elevador):** O status do elevador (ELEVATOR_STATUS) refere-se à condição operacional do elevador da colhedora, que pode estar em uma posição de "FORWARD" (avançando) ou "NEUTRAL" (neutro). Esse parâmetro é importante para entender o estado de transporte do material cortado dentro da máquina. Por exemplo, se o elevador estiver em "NEUTRAL", pode indicar que a máquina está em pausa ou preparando-se para uma nova operação.

- **BC_HEIGHT (Altura do Corte de Base):** A altura do corte de base (BC_HEIGHT) mede a altura em que a colhedora está cortando o material. Como esse valor é configurável pelo operador da máquina, monitorando esse parâmetro podemos identificar se a máquina está colhendo (valor aproximado do padrão configurado) ou está parada (valor abaixo do padrão configurado).

- **ENG_SPEED (Rotação do Motor):** A rotação do motor (ENG_SPEED) mede a velocidade de rotação do motor da colhedora em rotações por minuto (rpm). Esse parâmetro é utilizado para monitorar a carga de trabalho do motor e garantir que ele opere dentro dos limites ideais para evitar superaquecimento ou desgaste prematuro. Dependendo da rotação, podemos inferir se a máquina está em operação normal, manobra ou em modo de espera.

- **CHOPPER_PRESS (Pressão do Picador):** A pressão do picador (CHOPPER_PRESS) mede a força aplicada pelo picador da colhedora durante o processo de trituração. Esse parâmetro pode nos indicar que a máquina está parada por exemplo, quando observarmos um valor baixo.

4. IMPLEMENTAÇÃO

Com base na proposta do capítulo anterior, essa seção apresenta a implementação da análise de dados.

À primeira vista, foi identificado, duas colunas referente a data, *'DATA'* e *'DT_INTEGRACAO'*. A coluna *'DATA'* se refere a data real da coleta dos dados pela colhedora, essa coleta dos dados é feita a cada minuto. Os maquinários atuam em campo, oque significa que nem sempre a conexão com a internet é presente, por isso, a coluna *'DT_INTEGRACAO'* se refere a data em que os dados foram enviados e recebidos para o servidor.

Para melhor visualização dos dados, foi feito um PIVOT da tabela. De acordo com Harrington (2003), a operação de PIVOT é utilizada para reorganizar e resumir dados, facilitando a análise ao transformar dados de uma forma tabular em uma forma de matriz. O PIVOT permite que os dados sejam movidos de linhas para colunas, criando um novo layout que pode ser mais intuitivo e fácil de interpretar. Ao aplicar o PIVOT, o seguinte resultado é obtido:

	DATA	HOUR	BATTERY_VOLTAGE	BC_HEIGHT	BC_HYD_PRESS	CHOPPER_PRESS
0	2023-09-01	00:00:42	12.5	55.0	12200.0	11085.0
1	2023-09-01	00:01:42	12.5	54.0	8625.0	10145.0
2	2023-09-01	00:02:43	12.5	46.0	4770.0	5395.0
3	2023-09-01	00:03:43	12.5	59.0	9495.0	11335.0
4	2023-09-01	00:04:43	12.6	67.0	9155.0	8325.0

Figura 5 : Tabela após aplicação do PIVOT

Dessa forma, a visualização dos valores das variáveis de minuto a minuto, é simplificada. Transformando aproximadamente 56 linhas em uma única linha.

Foi criada a coluna *'HOUR'* que possibilita o agrupamento dos valores por data e hora, permitindo a geração de gráficos de linha que mostram a variação das colunas/variáveis a cada minuto.

4.1. LIMPEZA DOS DADOS

É muito comum que bases de dados apresentem valores ausentes, dados com ruídos (dados que contêm erros aleatórios ou flutuações que distorcem o valor real) e dados inconsistentes (dados que não estão de acordo com as regras ou padrões estabelecidos). Para trabalhar com dados mais confiáveis, é necessário tratar esses tipos de dados.

Na base de dados utilizada neste estudo, foram encontrados todos os tipos de problemas citados acima, e cada um deles será detalhado a seguir.

4.1.1. Valores Ausentes

Como identificado, os dados são fornecidos de minuto a minuto, entretanto existem diversos intervalos maiores que um minuto entre os registros.

Para tratar registros ausentes com diferença de 1 minuto, como entre 1:10 e 1:12, foi utilizado o cálculo da média dos dois registros para as colunas numéricas, e para as colunas categóricas, o valor do registro anterior foi repetido.

Entretanto, é importante ressaltar que esse preenchimento de valores ausentes deve ser aplicado apenas em intervalos curtos, pois diferenças maiores, como 5 horas entre registros, podem indicar que a colhedora interrompeu suas operações e, após esse período, retomou seu funcionamento, o que é um comportamento comum.

Para isso foi desenvolvida uma nova função chamada "preencher_valores_ausentes", que permite que o intervalo ideal entre os registros seja determinado.

```
def preencher_valores_ausentes(df, menor_diferenca, maior_diferenca):
    novas_linhas = []

    colunas_categoricas = df.select_dtypes(include=['object', 'category']).columns.tolist()
    colunas_numericas = df.select_dtypes(include=['number']).columns.tolist()

    for i in range(len(df) - 1):
        linha_atual = df.iloc[i]
        linha_posterior = df.iloc[i + 1]

        datetime_atual = datetime.combine(linha_atual['DATA'], linha_atual['HOUR'])
        datetime_posterior = datetime.combine(linha_posterior['DATA'], linha_posterior['HOUR'])

        datetime_atual_calculo = datetime_atual.replace(second=0, microsecond=0)
        datetime_posterior_calculo = datetime_posterior.replace(second=0, microsecond=0)

        diferenca = datetime_posterior_calculo - datetime_atual_calculo

        if timedelta(minutes=menor_diferenca) < diferenca < timedelta(minutes=maior_diferenca):
            diferenca = int(diferenca.total_seconds())

            ultima_linha_inserida = linha_atual
            for j in range(1, diferenca):
                datetime_novo = datetime_atual + timedelta(minutes=j)
                nova_linha = ultima_linha_inserida.copy()

                for col in colunas_categoricas:
                    nova_linha[col] = linha_atual[col]

                for col in colunas_numericas:
                    nova_linha[col] = round((ultima_linha_inserida[col] + linha_posterior[col]) / 2, 1)

                nova_linha['HOUR'] = datetime_novo.time()
                novas_linhas.append(nova_linha)
                ultima_linha_inserida = nova_linha

    novas_linhas_df = pd.DataFrame(novas_linhas)

    df = pd.concat([df, novas_linhas_df], ignore_index=True)
    df = df.sort_values(by=['DATA', 'HOUR']).reset_index(drop=True)

    return df
```

Figura 6 : Função para tratar valores ausentes

A função “preencher_valores_ausentes” preenche lacunas de tempo em um DataFrame.

DataFrame é uma estrutura de dados bidimensional usada principalmente na biblioteca Pandas em Python. Ele organiza dados em linhas e colunas, similar a uma tabela de banco de dados ou uma planilha Excel.

A função percorre o DataFrame linha por linha e, para cada par de linhas consecutivas, calcula a diferença de tempo entre os registros. Caso essa diferença esteja entre o intervalo determinado pelas variáveis menor_diferenca e maior_diferenca, novos registros serão inseridos no DataFrame para preencher essa lacuna. Por exemplo, se a diferença for de 5 minutos, 4 novos registros serão inseridos. As colunas categóricas são preenchidas repetindo o valor do primeiro registro do intervalo, enquanto as colunas numéricas são preenchidas calculando a média entre o valor do último registro inserido e o valor do último registro do intervalo. A coluna que define o horário é preenchida somando 1 minuto ao horário do registro anterior. Todas essas novas linhas são armazenadas em uma lista, que posteriormente é convertida em um DataFrame e concatenada com o DataFrame original.

Essa abordagem garante que os intervalos de tempo no DataFrame sejam consistentes e que os valores numéricos sejam interpolados de maneira adequada, mantendo a integridade dos dados categóricos.

	DATA	HOUR	BATTERY_VOLTAGE	BC_HEIGHT	BC_HYD_PRESS	CHOPPER_PRESS
45	2023-09-01	00:45:48	12.6	400.0	425.0	2265.0
46	2023-09-01	00:46:48	12.5	118.0	440.0	2235.0
47	2023-09-01	00:47:21	13.1	111.0	315.0	1985.0
48	2023-09-01	00:54:13	12.6	50.0	4280.0	7380.0
49	2023-09-01	00:54:39	12.6	51.0	5545.0	8595.0
50	2023-09-01	00:55:40	12.6	60.0	4360.0	7375.0
51	2023-09-01	00:56:40	12.7	351.0	435.0	2005.0

Figura 7 : DataFrame antes do tratamento de ausência de valores

A Figura 7 apresenta dois registros com uma diferença de 7 minutos entre eles. Ao aplicar a função preencher_valores_ausentes utilizando um intervalo entre 2 e 11 minutos, 6 novos registros serão inseridos. Após a aplicação da função, o resultado obtido foi o seguinte:

	DATA	HOUR	FUEL_LEVEL	BC_HYD_PRESS	BC_DIRECTION	ELEVATOR_STATUS
45	2023-09-01	00:45:48	40.8	425.0	Neutral	Neutral
46	2023-09-01	00:46:48	41.6	440.0	Neutral	Neutral
47	2023-09-01	00:47:21	32.4	315.0	Neutral	Neutral
48	2023-09-01	00:48:21	36.4	2297.5	Neutral	Neutral
49	2023-09-01	00:49:21	38.4	3288.8	Neutral	Neutral
50	2023-09-01	00:50:21	39.4	3784.4	Neutral	Neutral
51	2023-09-01	00:51:21	39.9	4032.2	Neutral	Neutral
52	2023-09-01	00:52:21	40.2	4156.1	Neutral	Neutral
53	2023-09-01	00:53:21	40.3	4218.0	Neutral	Neutral
54	2023-09-01	00:54:13	40.4	4280.0	Neutral	Forward
55	2023-09-01	00:54:39	39.6	5545.0	Up	Forward
56	2023-09-01	00:55:40	38.0	4360.0	Neutral	Forward

Figura 8 : DataFrame após o tratamento de ausência de valores

A Figura 8 mostra o DataFrame após a aplicação da função. Agora o DataFrame possui 6 novos registros, preenchendo o intervalo de valores ausentes com dados consistentes.

4.1.2. Dados com ruídos

Geralmente, os locais onde são realizadas as colheitas não são planos. Isso significa que variáveis como "FUEL_LEVEL" e "BC_HEIGHT" possuem ruídos constantes, pois qualquer mudança de nível do solo pode distorcer esses valores.

Para limpar esses dados com ruídos, o intervalo entre os registros foi aumentado. Essa mudança, além de resolver os ruídos, também simplificou a análise dos dados, pois as

diferenças entre os registros de minuto a minuto eram muito pequenas e pouco relevantes para o estudo.

Essa mudança de intervalo foi realizada utilizando uma média truncada, que é uma medida estatística utilizada para calcular a média de um conjunto de dados, excluindo uma determinada porcentagem dos valores mais extremos, tanto do início quanto do fim da distribuição.

Essa técnica é particularmente útil para reduzir o impacto de outliers ou valores atípicos que podem distorcer a média aritmética simples. De acordo com Wilcox (2012), a média truncada é calculada removendo uma porcentagem específica dos menores e maiores valores do conjunto de dados e, em seguida, computando a média dos valores restantes. A justificativa para a escolha dessa medida estatística foi feita baseado na ausência do tratamento de outliers, utilizando a média truncada, mesmo que um conjunto possua um valor muito fora do comum, ele irá ser descartado.

```
def media_truncada(dados, proporcao=0.1):
    tamanho = len(dados)
    qnt_truncados = int(tamanho * proporcao)
    dados_ordenados = np.sort(dados)
    dados_truncados = dados_ordenados[qnt_truncados:tamanho-qnt_truncados]
    return np.mean(dados_truncados)
```


```
def media_truncada(dados, proporcao=0.1):
    tamanho = len(dados)
    qnt_truncados = int(tamanho * proporcao)
    dados_ordenados = np.sort(dados)
    dados_truncados = dados_ordenados[qnt_truncados:tamanho-qnt_truncados]
    return np.mean(dados_truncados)
```


Figura 9 : Função para obter a média truncada

A função “media_truncada” (Figura 9) é responsável pelo cálculo da média truncada. O processo começa determinando o tamanho do conjunto de dados. Em seguida, calcula-se a quantidade de elementos a serem removidos das extremidades. Após essa etapa, os dados são ordenados de forma crescente (ou decrescente), e os valores das extremidades são excluídos. Com isso, a média é calculada apenas sobre os elementos

restantes, proporcionando uma medida mais representativa ao minimizar a influência de valores extremos (outliers).

DATA	HOUR	FUEL_LEVEL
2023-09-03	20:58:43	69.6
2023-09-03	20:59:43	70.8
2023-09-03	21:00:43	73.2
2023-09-03	21:01:43	67.2
2023-09-03	21:02:44	67.2
2023-09-03	21:03:44	67.2
2023-09-03	21:04:44	67.6
2023-09-03	21:05:44	72.8
2023-09-03	21:06:44	68.8
2023-09-03	21:07:44	69.2
2023-09-03	21:08:44	68.8
2023-09-03	21:09:45	68.0
2023-09-03	21:10:45	69.2
2023-09-03	21:11:45	64.8
2023-09-03	21:12:45	66.0
2023-09-03	21:13:45	67.6
2023-09-03	21:14:46	68.0
2023-09-03	21:15:45	67.6
2023-09-03	21:16:46	70.4
2023-09-03	21:17:46	68.4
2023-09-03	21:18:46	68.0

DATA	HOUR	VALOR
2023-09-03	20:58:43	69.15
2023-09-03	21:08:44	67.95
2023-09-03	21:18:46	67.30
2023-09-03	21:52:34	64.40
2023-09-03	22:02:38	64.15
2023-09-03	22:12:39	61.05
2023-09-03	22:32:28	62.00

Figura 10 : Aplicação da média truncada

4.1.3. Dados inconsistentes

Por meio dos responsáveis pela base de dados, obtivemos uma tabela descritiva contendo os valores mínimos e máximos que cada variável poderia assumir. No entanto, alguns problemas foram identificados.

As variáveis "BC_HYD_PRESS" e "CHOPPER_PRESS" apresentaram metade de seus valores fora do padrão esperado. Além disso, a coluna "ENG_SPEED", que representa a velocidade de rotação do motor, foi descrita com um valor máximo de 8 milhões de rpm. A coluna "FUEL_LEVEL", que deveria representar a taxa de combustível em porcentagem com um valor máximo de 100, apresentou 25 registros com valores acima de 100, alguns chegando a 26.000.

Como metade dos valores de algumas variáveis estavam fora dos limites estabelecidos, surgiram dúvidas sobre a veracidade da tabela. No entanto, apenas próximo ao prazo de entrega deste trabalho foi esclarecido que a tabela descritiva com os limites dos dados utiliza uma unidade de medida diferente. Embora esses dados não tenham sido ajustados neste momento, eles serão tratados em trabalhos futuros.

Uma alternativa para tratar esses dados seria a simples exclusão dos registros identificados como outliers, especialmente quando esses valores forem claramente erros de entrada. Outra técnica é o filtro de limite, onde se definem limites superiores e inferiores e se removem dados fora desses limites (Barnett & Lewis, 1994). Outra abordagem seria a substituição de outliers, que é uma técnica onde os outliers são substituídos por valores como a média, mediana ou moda dos dados. A interpolação, que usa valores vizinhos para substituir outliers, também é uma técnica eficaz (Little & Rubin, 2002).

4.1.4. Transformação dos Dados

Com o objetivo de facilitar a implementação de funções, as seguintes variáveis passaram pelo processo de transformação de dados:

ELEVATOR_STATUS

Valores categóricos foram transformados em numéricos.

-1. Não identificado | 0. Neutral | 1. Forward | 2. Reverse

CHOPPER_PRESS e BC_HYD_PRESS

Unidade de medida em KPA (kilopascal), transformada para BAR.

4.2. ANÁLISE

O principal campo escolhido para a análise foi "FUEL_LEVEL". Então foi feita uma análise unidirecional. Inicialmente foi desenvolvida uma função para agrupar os valores por data e hora e em sequência, gráficos de cada dia foram gerados mostrando o consumo de combustível.

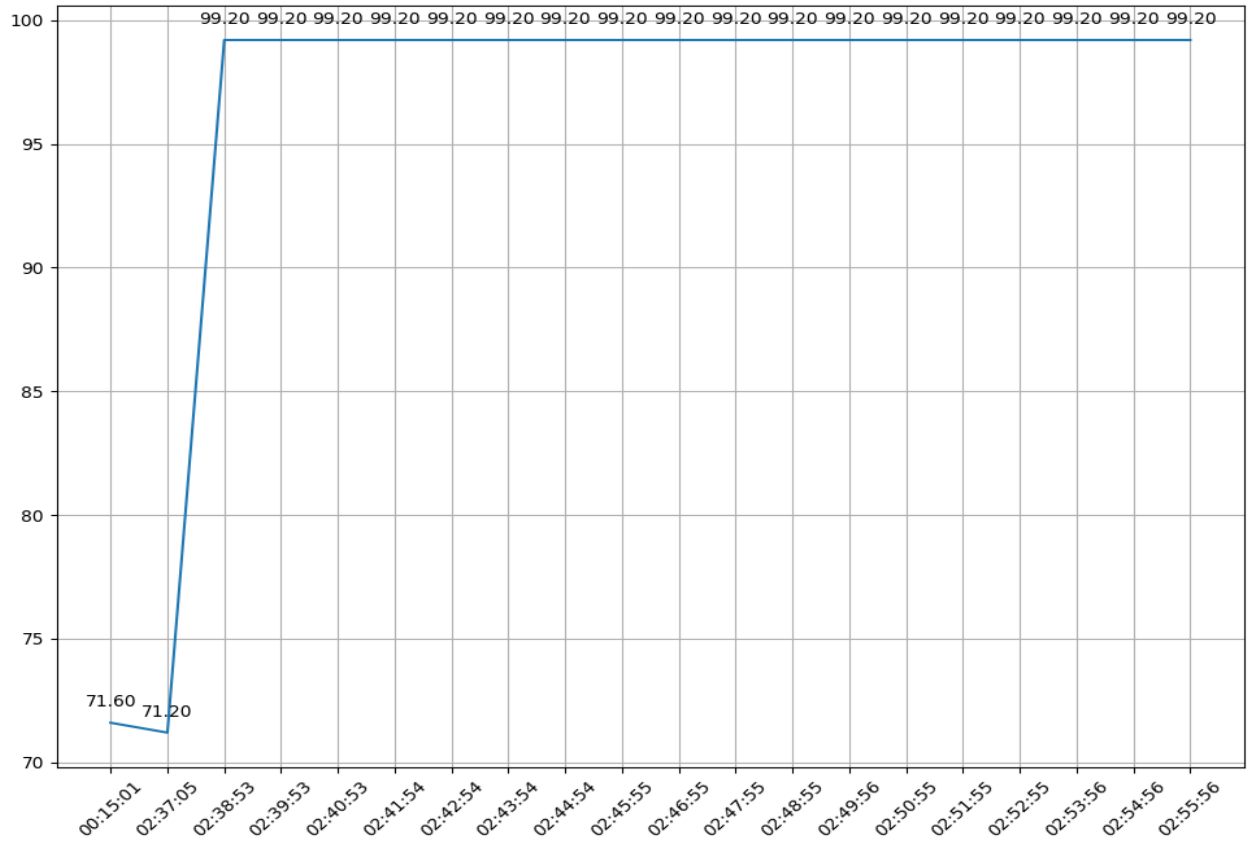

Figura 9 : Gráfico demonstrando ausência de valores

Como identificado na figura acima (Figura 9), o primeiro registro está com o horário das 00:15 e o segundo está as 02:37, ou seja, há uma ausência de valores de mais de duas horas entre os registros.

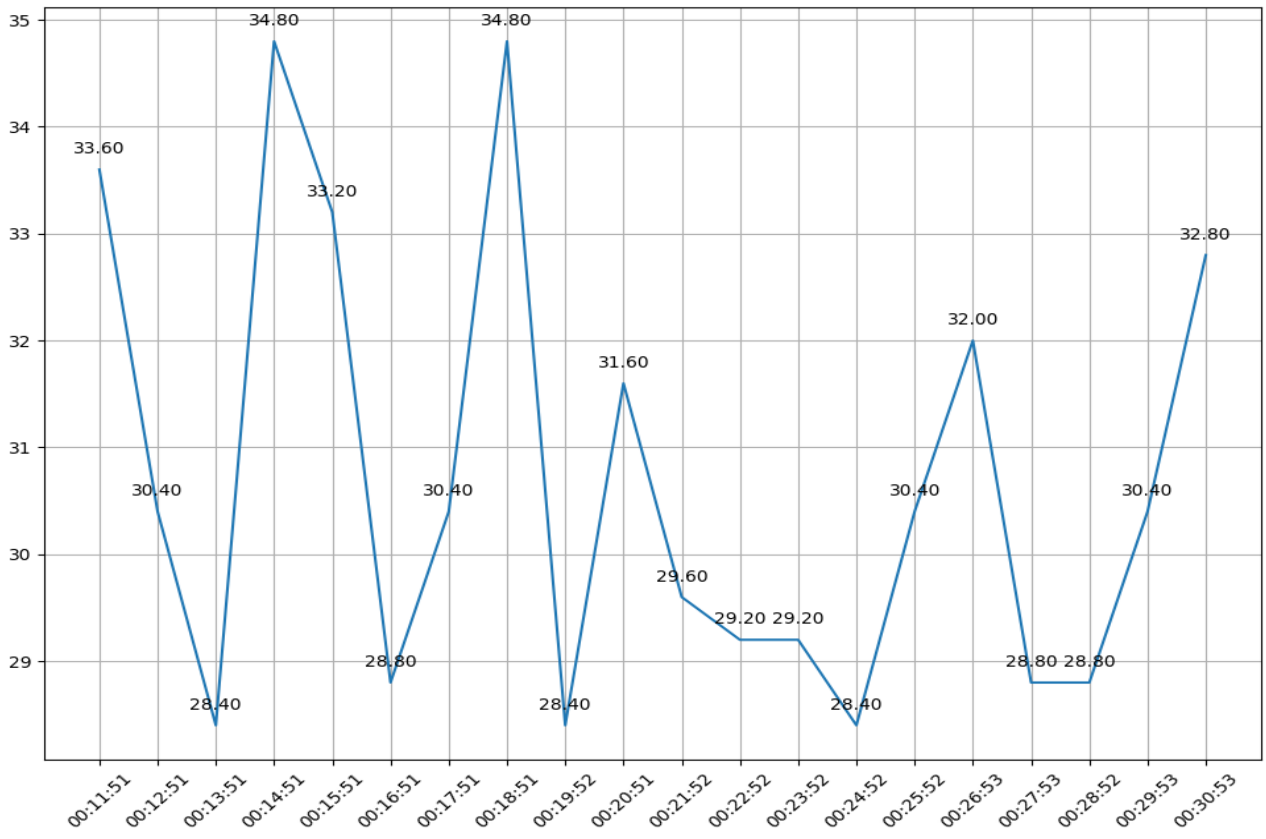

Figura 10 : Gráfico demonstrando dados com ruídos

Na figura acima (Figura 10), nota-se grandes variações entre os registros com poucos minutos de diferença, isso nos demonstra que os dados estão com ruídos, possivelmente pois qualquer mudança de nível do solo pode distorcer esses valores.

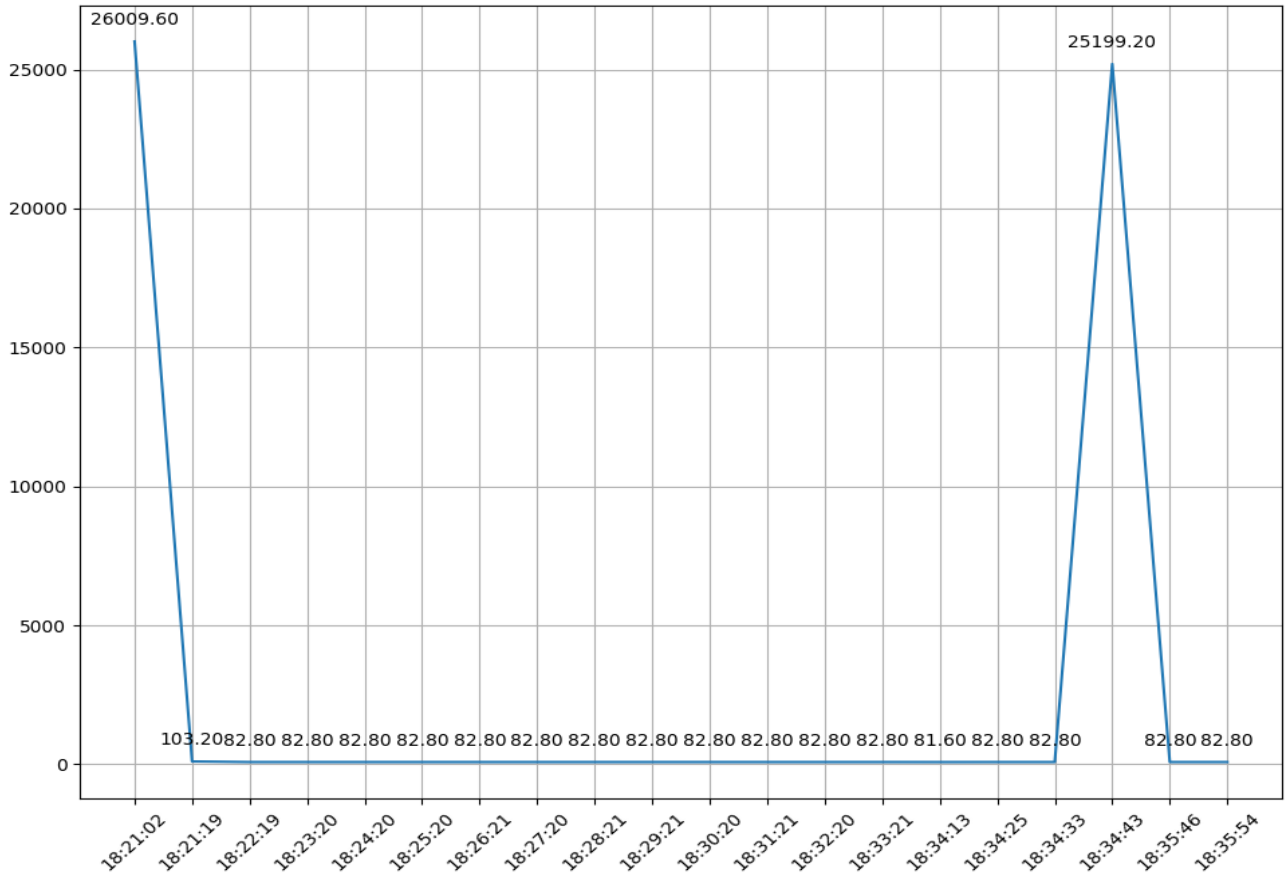

Figura 11 : Gráfico demonstrando valores inconsistentes

A figura acima (Figura 11), apresenta valores que atingiram mais que 20.000% do nível de combustível, esses são valores absurdos pois a unidade de medida é porcentagem e o valor máximo é 100%.

Então, ao analisar a variável "FUEL_LEVEL", nota-se três pontos importantes: a ausência de valores em alguns momentos, as variações significativas entre registros consecutivos e a persistência de um mesmo valor durante vários minutos.

Essas variações no nível de combustível sugerem que o maquinário foi abastecido em algum momento. Essa observação levanta uma questão crucial: como determinar o modo de operação do maquinário com base nesses dados?

A identificação de padrões no nível de combustível, em conjunto com outras variáveis relevantes, pode nos fornecer *insights* sobre o funcionamento do maquinário, como períodos de trabalho intenso, de inatividade ou de manutenção.

Ao apontar essa questão em reuniões com os responsáveis por fornecer a base de dados, foi fornecida uma tabela que descreve o modo de operação da colhedora de acordo com outros parâmetros

LÓGICAS COLHEDORA 9000				
Status de Operação	**Parâmetro**	**Lógica**	**Valor**	**Unidade**
Status "Colheita"	BC_HYD_PRESS	MAIOR QUE	30	bar
	GROUND_SPEED	MAIOR QUE	0,5	km/h
	ELEVATOR_STATUS	IGUAL	FOWARD (1) OU NEUTRAL (0)	-
	BC_HEIGHT	MENOR QUE	150	-
Status "Manobra"	BC_HYD_PRESS	MENOR QUE	30	bar
	BC_HEIGHT	MAIOR QUE	150	-
	GROUND_SPEED	MENOR QUE	3,0	km/h
	ENG_SPEED	MENOR QUE	1800	rpm
Status "Deslocamento"	GROUND_SPEED	MAIOR OU IGUAL A	3,0	km/h
	ENG_SPEED	MENOR QUE	1800	rpm
	BC_HEIGHT	MAIOR QUE	150	-
	CHOPPER_PRESS	MENOR QUE	80	bar
	ELEVATOR_STATUS	IGUAL	NEUTRAL (0)	-
Status "Embuchamento"	ENG_SPEED	MAIOR OU IGUAL A	1800	rpm
Status "Ociosa"	GROUND_SPEED	IGUAL	0	km/h
	ENG_SPEED	MAIOR QUE	0	rpm
		MENOR QUE	1800	rpm
Status "Chave Ligada"	GROUND_SPEED	IGUAL	0	km/h
	ENG_SPEED	IGUAL	0	rpm

Figura 12 : Lógica de Classificação do modo de operação

Com base nessa lógica, foi implementada uma função responsável por classificar esses registros:

```
def classificar_modo_operacao(df):
    df['OPERATION_MODE'] = 'Indefinido'

    for index, row in df.iterrows():
        if modo_chave_ligada(row.GROUND_SPEED, row.ENG_SPEED):
            df.loc[index, 'OPERATION_MODE'] = 'CHAVE LIGADA'

        elif modo_ociosa(row.GROUND_SPEED, row.ENG_SPEED):
            df.loc[index, 'OPERATION_MODE'] = 'OCIOSA'

        elif modo_embuchamento(row.ENG_SPEED):
            df.loc[index, 'OPERATION_MODE'] = 'EMBUCHAMENTO'

        elif modo_deslocamento(row.GROUND_SPEED, row.ENG_SPEED,
                               row.BC_HEIGHT, row.CHOPPER_PRESS, row.ELEVATOR_STATUS):
            df.loc[index, 'OPERATION_MODE'] = 'DESLOCAMENTO'

        elif modo_manobra(row.BC_HYD_PRESS, row.BC_HEIGHT, row.GROUND_SPEED, row.ENG_SPEED):
            df.loc[index, 'OPERATION_MODE'] = 'MANOBRA'

        elif modo_colheita(row.BC_HYD_PRESS, row.GROUND_SPEED, row.ELEVATOR_STATUS, row.BC_HEIGHT):
            df.loc[index, 'OPERATION_MODE'] = 'COLHEITA'

        else:
            df.loc[index, 'OPERATION_MODE'] = 'INDEFINIDO'

    return df
```

Figura 13 : Função responsável por classificar o modo de operação das colhedoras

A função *'classificar_modo_operacao'* (Figura 13) é projetada para analisar um DataFrame linha por linha, atribuindo a cada linha um modo de operação específico com base em critérios predefinidos. Inicialmente, todas as linhas do DataFrame são marcadas com o modo de operação 'INDEFINIDO', garantindo que cada linha tenha um valor padrão antes da avaliação. A função percorre cada linha do DataFrame e aplica uma série de verificações sequenciais através de funções auxiliares (Figura 14).

Para cada linha, a função 'classificar_modo_operação' começa verificando se a condição da função *'modo_chave_ligada'* é atendida. Se esta retornar falso, a função tenta a próxima classificação, como *'modo_ociosa'*, e assim por diante, até encontrar um modo de operação que corresponda.

Assim que uma condição é satisfeita, o modo de operação correspondente é atribuído à linha atual. Se nenhuma das condições é atendida após todas as verificações, o modo permanece como "INDEFINIDO".

```
def modo_colheita(BC_HYD_PRESS, GROUND_SPEED, ELEVATOR_STATUS, BC_HEIGHT):
  if (BC_HYD_PRESS > 30 and GROUND_SPEED > 0.5 and
      (ELEVATOR_STATUS == 0 or ELEVATOR_STATUS == 1) and BC_HEIGHT < 150):
    return True
  return False

def modo_manobra(BC_HYD_PRESS, BC_HEIGHT, GROUND_SPEED, ENG_SPEED):
  if (BC_HYD_PRESS < 30 and BC_HEIGHT > 150 and GROUND_SPEED < 3.0 and ENG_SPEED < 1800):
    return True
  return False

def modo_deslocamento(GROUND_SPEED, ENG_SPEED, BC_HEIGHT, CHOPPER_PRESS, EVATOR_STATUS):
  if (GROUND_SPEED >= 3.0 and ENG_SPEED < 1800 and
      BC_HEIGHT > 150 and CHOPPER_PRESS < 80 and EVATOR_STATUS == 0):
    return True
  return False

def modo_embuchamento(ENG_SPEED):
  if (ENG_SPEED >= 1800):
    return True
  return False

def modo_ociosa(GROUND_SPEED, ENG_SPEED):
  if (GROUND_SPEED == 0 and ENG_SPEED > 0 and ENG_SPEED < 1800):
    return True
  return False

def modo_chave_ligada(GROUND_SPEED, ENG_SPEED):
  if (GROUND_SPEED == 0 and ENG_SPEED == 0):
    return True
  return False
```

Figura 14 : Funções de classificação

Após aplicamos a função obtivemos o seguinte resultado:

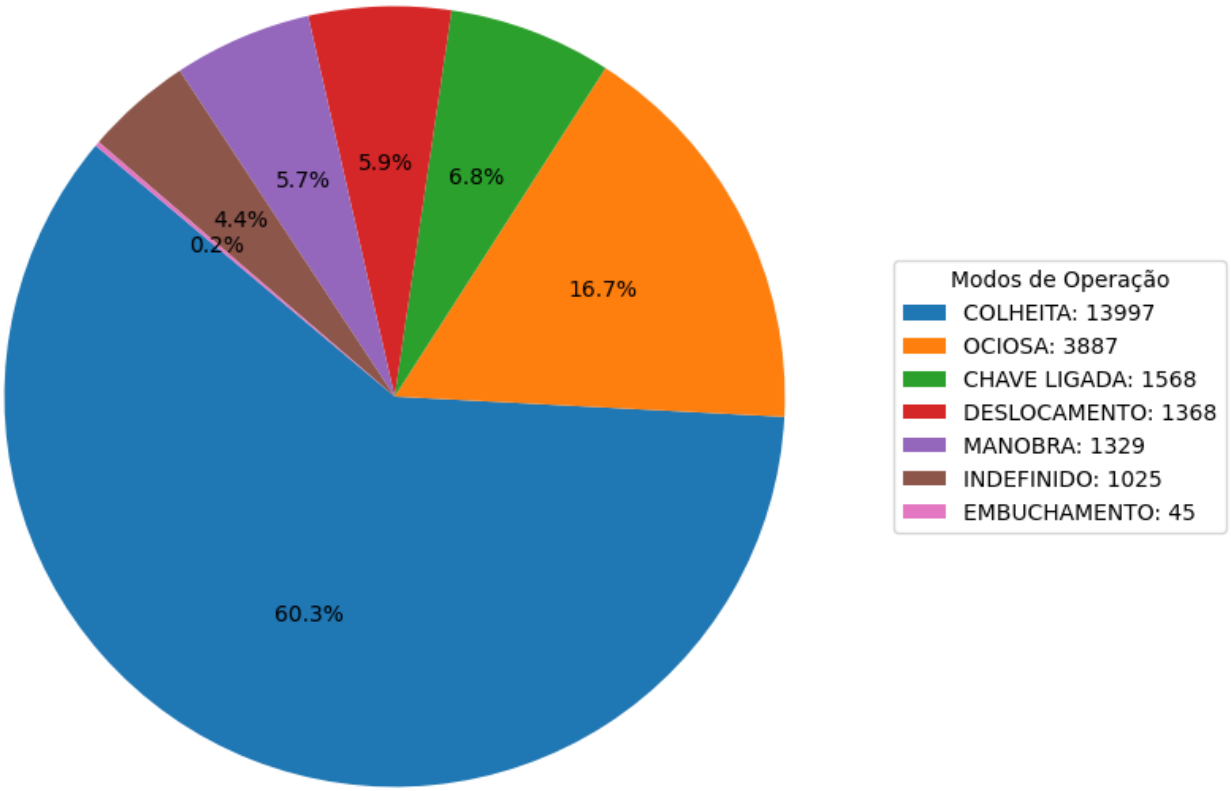

Figura 14: Distribuição dos Modos de Operação

Esse novo conhecimento obtido, pode ser utilizado para identificar problemas nos componentes da colhedora ou verificar integridade das informações. Por exemplo, ao analisar registros de nível de combustível em um intervalo de 30 minutos e observar uma queda atípica no consumo, é possível verificar o modo de operação correspondente a esses registros. Se estiverem classificados como "Parada", isso pode indicar um vazamento no tanque de combustível.

4.2.1. Exploração de um Problema e Solução Proposta

O abastecimento das colhedoras atualmente depende de comboios que chegam ao campo em horários fixos. Entretanto, como não há nenhuma lógica por trás desses horários, o comboio pode chegar no campo e precisar ficar esperando horas para poder

abastecer a colhedora. Então surgiu-se o interesse em otimizar esse processo, através de cálculos para alcançar informações como o tempo restante de combustível. Para isso, primeiramente é necessário informar a quantidade de litros consumido por hora que são 40 litros, fazendo um breve cálculo visto que a capacidade máxima do tanque da colhedora é 600 litros, chegamos no valor 6,7% de combustível por hora.

```
def modo_colheita(BC_HYD_PRESS, GROUND_SPEED, ELEVATOR_STATUS, BC_HEIGHT):
  if (BC_HYD_PRESS > 30 and GROUND_SPEED > 0.5 and
      (ELEVATOR_STATUS == 0 or ELEVATOR_STATUS == 1) and BC_HEIGHT < 150):
    return True
  return False

def modo_manobra(BC_HYD_PRESS, BC_HEIGHT, GROUND_SPEED, ENG_SPEED):
  if (BC_HYD_PRESS < 30 and BC_HEIGHT > 150 and GROUND_SPEED < 3.0 and ENG_SPEED < 1800):
    return True
  return False

def modo_deslocamento(GROUND_SPEED, ENG_SPEED, BC_HEIGHT, CHOPPER_PRESS, EVATOR_STATUS):
  if (GROUND_SPEED >= 3.0 and ENG_SPEED < 1800 and
      BC_HEIGHT > 150 and CHOPPER_PRESS < 80 and EVATOR_STATUS == 0):
    return True
  return False

def modo_embuchamento(ENG_SPEED):
  if (ENG_SPEED >= 1800):
    return True
  return False

def modo_ociosa(GROUND_SPEED, ENG_SPEED):
  if (GROUND_SPEED == 0 and ENG_SPEED > 0 and ENG_SPEED < 1800):
    return True
  return False

def modo_chave_ligada(GROUND_SPEED, ENG_SPEED):
  if (GROUND_SPEED == 0 and ENG_SPEED == 0):
    return True
  return False
```

Figura 15 : Função para calcular o tempo restante de combustível

A função *'calcularTempoRestanteCombustivel'* calcula quanto tempo a colhedora pode operar antes de precisar ser reabastecida. Inicialmente, é criada uma cópia do DataFrame original e adicionada uma nova coluna chamada 'REMAINING_HOURS_FUEL_LEVEL', que armazenará o horário estimado até o reabastecimento. A função percorre o DataFrame em intervalos definidos pela variável 'minutes', e para cada intervalo, é obtido

a média truncada dos registros. Se a média for maior que 20% (limite mínimo de combustível para o reabastecimento), o tempo restante é calculado com base na taxa de consumo de 6,7% por hora e convertido para um formato de horas, minutos e segundos. No entanto, se a média de combustível for menor ou igual a 20%, a função insere o valor 'REABASTECIMENTO URGENTE' na nova coluna, permitindo fácil identificação de registros críticos ou anormais. O resultado é então inserido na coluna "REMAINING_HOURS_FUEL_LEVEL" do DataFrame.

Após a definição das funções, aplicamos no DataFrame e determinamos o intervalo entre 10 minutos (ou registros).

```
dataFrame = calcularTempoRestanteCombustivel(df_classificado, 10)
dataFrame[['DATA', HOUR','REMAINING_HOURS_FUEL_LEVEL']][
~dataFrame['REMAINING_HOURS_FUEL_LEVEL'].isna()
]
```

Figura 16 : Aplicação das funções para obter o tempo restante até o reabastecimento

Após a aplicação da função obtivemos o seguinte resultado:

DATA	HOUR	REMAINING_HOURS_FUEL_LEVEL
2023-09-01	22:02:52	03:01:47
2023-09-01	22:12:53	02:51:02
2023-09-01	22:22:31	02:46:07
2023-09-01	22:32:56	02:32:41
2023-09-01	22:51:58	02:19:42
2023-09-01	23:01:58	02:04:01
2023-09-01	23:12:01	01:55:31
2023-09-01	23:22:01	01:36:16
2023-09-01	23:32:05	01:30:00
2023-09-01	23:41:52	01:31:20

Figura 17 : Resultado após a aplicação das funções

5. CONCLUSÃO

As áreas de dados têm crescido exponencialmente, impulsionadas pela abundância de dados e a necessidade de convertê-los em informações estratégicas. Com o aumento do volume de dados, a demanda por profissionais qualificados cresce, tornando essencial a produção de materiais que cubram desde conceitos básicos até técnicas avançadas.

Embora muitas pessoas associem as áreas de TI a uma complexidade intrínseca, a mineração de dados, apesar de ser vista como uma dessas áreas de difícil acesso, mostrou-se mais acessível do que se imagina.

Os objetivos propostos nesse trabalho foram atingidos, conseguimos obter dois novos conhecimentos da base de dados, sendo eles, o modo de operação da colhedora e o tempo restante de combustível, ao mesmo tempo, ao identificar problemas onde metade dos dados estavam fora do limite, localizamos um problema interno referente a base de dados, ou seja, a empresa fornecedora da base de dados também teve um retorno significativo.

Além disso, durante o desenvolvimento deste trabalho, acreditou-se que seria necessário seguir rigorosamente todos os processos de mineração de dados para alcançar resultados significativos, como a extração de novos conhecimentos ou a otimização de processos. No entanto, ao longo da pesquisa, ficou claro que cada etapa da mineração de dados pode oferecer informações valiosas por si só. É possível atingir objetivos concretos e gerar resultados palpáveis mesmo sem completar todas as etapas tradicionais, revelando que a mineração de dados pode ser aplicada de maneira mais flexível do que é esperado mas ao mesmo tempo isso não invalida as outras etapas pois, através delas é possível obter resultados mais precisos e concretos.

6. TRABALHOS FUTUROS

Ao analisarmos a base de dados, podemos identificar que existem inúmeras possibilidades para extração de conhecimento e a otimização dos processos. Embora o estudo atual tenha fornecido estimativas sobre o tempo restante até o reabastecimento de uma colhedora, existem várias abordagens que podem aprimorar a precisão e a aplicação desse conhecimento. A seguir, destacamos algumas direções para pesquisas futuras:

- I**ntegração com Dados Logísticos de Combustível:** Incorporar informações sobre a localização dos comboios (caminhões responsáveis pelo abastecimento) pode otimizar ainda mais o processo. Com dados sobre as rotas e tempos de deslocamento dos caminhões, é possível cruzar essas informações com o tempo estimado de reabastecimento, possibilitando a previsão mais precisa do momento ideal para o envio dos comboios ao local de colheita.

- **Aplicação para Frotas de Máquinas:** O conhecimento obtido, embora aplicável a uma máquina individual, pode ser expandido para todas as máquinas do mesmo modelo em operação simultânea. Considerando múltiplas colhedoras em um mesmo local, seria possível calcular o tempo de reabastecimento de todas as máquinas e definir um horário ideal para o comboio abastecer todas as unidades de forma eficiente.

- **Otimização de Rotas de Abastecimento:** Em cenários onde um comboio deve abastecer vários locais de colheita, a análise das rotas pode ser aprimorada. Mapeando o tempo restante de combustível em diferentes locais, é possível determinar a rota mais eficiente para o comboio, considerando tanto a proximidade quanto a urgência de abastecimento em cada local.

- **Análise do Modo de Operação:** A inclusão de dados sobre o modo de operação da máquina pode ser crucial para identificar anomalias, como o consumo excessivo de combustível em modos ociosos ou com a chave ligada. Esses insights podem ajudar a diagnosticar problemas.

- **Cálculo da Quantidade de Litros por Hora:** Desenvolver um cálculo da quantidade de litros consumido por hora, em vez de usar o valor conhecido, pode ser uma abordagem mais precisa. Dessa forma, seria possível identificar problemas relacionados ao consumo de combustível e aumentar a precisão do cálculo.

7. REFERÊNCIAS

BARNETT, V.; LEWIS, T. **Outliers in Statistical Data**. 3rd ed. New York: Wiley, 1994.

BERKHIN, P. **A Survey of Clustering Data Mining Techniques**. In: KOGAN, J.; NICHOLAS, C.; TEBER, M. (Eds.). Grouping Multidimensional Data. Berlin: Springer, 2006. p. 25-71.

CASTRO, Daniel Gomes Ferrari Leandro Nunes de. **Introdução à Mineração de Dados: Conceitos Básicos, Algoritmos e Aplicações**. São Paulo: Editora Saraiva, 2016. E-book. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/978-85-472-0100-5/. Acesso em: 26 mar. 2024.

FIELD, A. **Discovering Statistics Using IBM SPSS Statistics**. 4. ed. London: SAGE Publications, 2013.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P.; UTHURUSAMY, R. **From data mining to knowledge discovery in databases**. AI Magazine, v. 17, n. 3, p. 37-54, 1996.

HAIR, J. F. et al. **Multivariate Data Analysis**. 7. ed. Upper Saddle River: Prentice Hall, 2010.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. 3. ed. Waltham: Morgan Kaufmann, 2012.

LITTLE, R. J. A.; RUBIN, D. B. **Statistical Analysis with Missing Data**. 2nd ed. New York: Wiley, 2002.

MONTGOMERY, D. C.; RUNGER, G. C. **Applied Statistics and Probability for Engineers**. 5. ed. Hoboken: John Wiley & Sons, 2010.

REINSEL, D.; GANTZ, J.; RYDNING, J. **Seagate data age 2025**. [PDF]. [S.l.]: Seagate, 2017. Disponível em: https://www.seagate.com/files/www-content/our-story/trends/files/Seagate-WP-DataAge2025-March-2017.pdf. Acesso em: 1 abr. 2024.

SILVA, João. **Introdução à Análise de Dados**. 2. ed. São Paulo: Editora Exemplo, 2020.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining**. 2. ed. Boston: Pearson, 2019.

WILCOX, R. R. **Introduction to Robust Estimation and Hypothesis Testing**. 3. ed. San Diego: Academic Press, 2012.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data mining: practical machine learning tools and techniques**. Morgan Kaufmann, 2016.