Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

Aplicação de técnicas de mineração de dados para a predição de falhas corretivas em caminhões fora de estrada

Gabriel Henrique dos Santos Ferreira

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
Helen de Cassia Sousa da Costa Lima

Junho, 2022
João Monlevade–MG

Gabriel Henrique dos Santos Ferreira

Aplicação de técnicas de mineração de dados para a predição de falhas corretivas em caminhões fora de estrada

Orientador: Helen de Cassia Sousa da Costa Lima

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação.

Universidade Federal de Ouro Preto

João Monlevade

Junho de 2022

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

F383a Ferreira, Gabriel Henrique dos Santos.
Aplicação de técnicas de mineração de dados para a predição de falhas corretivas em caminhões fora de estrada. [manuscrito] / Gabriel Henrique dos Santos Ferreira. - 2022.
41 f.

Orientadora: Profa. Dra. Helen de Cassia Sousa da Costa Lima.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia de Computação .

1. Mineração de dados (Computação). 2. Minas e recursos minerais - Carregamento e transporte. 3. Caminhões. 4. Manutenção. I. Lima, Helen de Cassia Sousa da Costa. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.62:622.68

Bibliotecário(a) Responsável: Sione Galvão Rodrigues - CRB6 / 2526

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTACAO E SISTEMAS

FOLHA DE APROVAÇÃO

Gabriel Henrique dos Santos Ferreira

Aplicação de Técnicas de Mineração de Dados para a predição de falhas corretivas em caminhões fora de estrada

Monografia apresentada ao Curso de Engenharia da Computação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação

Aprovada em 15 de junho de 2022

Membros da banca

Dra. Helen de Cássia Sousa da Costa Lima - Orientadora (Universidade Federal de Ouro Preto)
Dra. Gilda Aparecida de Assis (Universidade Federal de Ouro Preto)
Dra. Janniele Aparecida Soares Araújo (Universidade Federal de Ouro Preto)

Helen de Cássia Sousa da Costa Lima, orientadora do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 12/07/2022

Documento assinado eletronicamente por **Helen de Cassia Sousa da Costa Lima**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 12/07/2022, às 20:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0352377** e o código CRC **1DDC5122**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.008462/2022-86 SEI nº 0352377

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: (31)3808-0819 - www.ufop.br

Agradecimentos

Gostaria de agradecer aos meus pais Geraldo Ponciano Ferreira e Sueli Maria dos Santos Ferreira pela paciência, conselhos, carinho, e força que sempre tiveram comigo ao longo desta jornada. Agradeço também à minha noiva Isabela Soares Avelino por toda a paciência para me suportar nos momentos díficeis e felizes nessa caminhada.

Agradeço aos meus irmãos Arthur Ponciano dos Santos Ferreira, Maria Eduarda dos Santos Ferreira, e Leticia Cristina dos Santos Ferreira pelo suporte e companhia em toda minha vida.

Agradeço a todos que fazem parte da minha vida profissional pelos concelhos, pelo incentivo a criar análises mais críticas, e pelo suporte em meu desenvolvimento profissional.

Agradeço a minha orientadora, Helen De Cassia Sousa da Costa Lima, por toda orientação, paciência, e desenvolvimento no tempo que estivemos juntos, que me ajudaram a elaborar este trabalho.

Agradeço a todas as pessoas que fizeram parte da minha caminhada durante o curso na UFOP, a todos os amigos, professores e funcionários que dedicam seu trabalho a manter esta importante instituição.

"Science is more than a body of knowledge; it is a way of thinking."

— Carl Sagan (1934 – 1996),
in: The Demon-Haunted World: Science as a Candle in the Dark.

Resumo

Este trabalho propõe a geração de conhecimento útil relacionado ao problema da predição de falhas em caminhões fora de estrada com a utilização de técnicas e tarefas de mineração de dados aplicadas a uma base de dados de uma grande mineradora do estado de Minas Gerais-Brasil. Este estudo faz uma análise sobre dados coletados por sensores de 19 caminhões e indicadores de performance dos mesmos. Para realizar a predição, foi utilizada a tarefa de classificação com os algoritmos *Random Forest*, *Naive Bayses* e *XGBoost*. Com essa abordagem foi possível predizer 62,22% das falhas nos caminhões com o algoritmo *Naive Bayses*.

Palavras-chaves: Mineração de dados. Predição. Manutenção. Caminhões fora de estrada. Mineradora.

Abstract

This work proposes the genaration of useful knowledge related to the problema of predicting failures in off-roal trucks with the use of data mining techniques and tasks Applied to a database of a large mining company in the state of Minas Gerais -Brazil. This study analyses data collected by sensors from 19 trucks and their performance indicators. To perform the prediction, we used the classification task with the algorithms Random Forest, Naive Bayses and XGBoost. With this approach was able to predict 62,22% of truck failures with the algorithms Naive Bayses.

Key-words: Data mining. Prediction. Maintenance. Off-road trucks. Mining company.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

IoT *Internet of Things*

KDD *Knowledge-Discovery in Databases* - em português, Descoberta de Conhecimento em Bases de Dados

PSF *Python Software Foundation*

RFID Identificação por Rádio Frequência

Sumário

1 Introdução

A sobrevivência nos mercados exige das empresas uma constante busca pelo aprimoramento do desempenho de suas operações (HIPKIN; De Cock, 2000). No cenário de empresas que possuem diversos veículos de transporte material ou humano, a otimização de custo com estes equipamentos é fundamental para a lucratividade da empresa.

Dessa forma, possuir um processo de manutenção de qualidade, demonstra um grande competitivo estratégico. Segundo Kardec e Nascif (2009), o processo de manutenção visa garantir a confiabilidade e a disponibilidade da função dos equipamentos e instalações de modo a atender a um processo de produção ou serviço com segurança, preservação do meio ambiente e custos adequados.

Para uma empresa de grande porte, com vários equipamentos que necessitam de manutenções de forma programada e emergencial, uma estratégia de manutenção adequada e otimizada implicaria na redução de custos e aumento da produção. Conforme a descrição de Kardec e Nascif (2009), as manutenções de forma corretiva sem planejamento implicam em alto custo, perda de qualidade de produção, devendo ser evitadas na medida do possível.

Com esse cenário em mente, na busca de reduzir os custos e otimizar os processos de manutenção, este trabalho irá apresentar uma solução através da mineração de dados que auxiliará a conhecer padrões de comportamento das falhas em caminhões de grande porte de uma mineradora. A partir do conhecimento gerado, será possível identificar o modo de apresentação da falha, o que permite que a empresa possa atuar no equipamento antes que torne uma falha mais grave e haja aumento dos custos com a manutenção.

1.1 Objetivos

O objetivo geral é realizar a mineração de dados para predizer quando um caminhão fora de estrada irá apresentar uma falha.

Este trabalho possui aos seguintes objetivos específicos:

- reduzir custos de manutenção;
- facilitar a programação de manutenções nos caminhões;
- elencar padrões de comportamentos de falha;
- desenvolver modelo preditivo com base nos dados dos sistemas dos caminhões.

1.2 Metodologia

A metodologia de pesquisa pode ser determinada como uma descrição detalhada do processo da pesquisa, ou seja, a determinação dos passos para coleta da informações, análise dos dados e desenvolvimento do trabalho. A seguir estão elencadas as etapas da metodologia utilizada neste trabalho:

- pesquisar trabalhos relacionados e elaborar a revisão bibliográfica;
- consolidar e tratar a base de dados que fornecerá informações necessárias para a predição das falhas;
- realizar a mineração de dados sobre os dados coletados;
- consolidar os resultados apresentados pela análise e aferir a acertividade da análise.

1.3 Organização do trabalho

Neste capítulo realizamos uma pequena contextualização sobre o problema abordado nesta pesquisa, o restante deste trabalho é organizado como se segue. O Capítulo 2 apresenta toda a fundamentação teórica utilizada para a elaboração deste estudo, tecnologias aplicadas no desenvolvimento da mineração de dados. O Capítulo 3 apresenta trabalhos relacionados ao tema desta pesquisa. O Capítulo 4 expõe qual é o problema abordado pela pesquisa, informando também o porque este problema deveria ser tratado. O Capítulo 5 demonstra todo o desenvolvimento da pesquisa, como foi elaborada a base de dados, o pré-processamento, aplicação dos algoritmos de mineração e resultados dos algoritmos. Por fim, no Capítulo 6 são apresentadas as conclusões frente aos resultados das análises e os trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo é apresentado todo o conhecimento necessário para desenvolvimento deste trabalho. Na Seção 2.1 são explicados como são os tipo de manutenções nos veículos. Na Seção 2.2 é apresentado o processo de descoberta de conhecimento por meio da base de dados. Na Seção 2.3 são explicados, brevemente, os algoritmos de mineração de dados que serão utilizados durante a pesquisa. Na Seção 2.4 são apresentadas as métricas utilizadas para aferir a acertividade dos algorimos analisados. Na Seção 2.5 são apresentadas as tecnologias utilizadas no desenvolvimentos das análises e implementação dos algoritmos.

2.1 Tipos de Manutenção de Equipamentos

A manutenção é uma série de intervenções e recursos aplicados aos equipamentos a fim de garantir o correto funcionamento, confiabilidade do ativo, aumento da produção, visando que o equipamento opere de forma segura.

Dessa forma, as empresas geram planos de manutenções para os equipamentos almejando o aumento da vida útil, redução dos custos e paradas inesperadas que possam prejudicar a produção da empresa. Entretanto, nem todas as manutenções ocorrem no mesmo formato, algumas ocorrem para prevenir falhas (preventiva e preditiva) e outras para corrigir as falhas (corretivas). A seguir é explicitado cada tipo de manutenção.

2.1.1 Manutenção Corretiva

Segundo Kardec, Nascif e Baroni (2002), a manutenção corretiva é a atuação para a correção da falha ou do desempenho menor do que o esperado. Sendo divididas em manutenções corretivas não planejadas e manutenções corretivas planejadas.

A manutenção corretiva não planejada, também conhecida como correção emergencial, é a correção da falha ou redução do desempenho do equipamento de forma aleatória. Geralmente este tipo de manutenção implica em altos custos, já que causa perda da produção, qualidade e aumento dos custos indiretos.

A manutenção corretiva planejada é a correção do desempenho menor do que é o esperado, realizada através de um acompanhamento preditivo, ou por uma decisão gerencial. Pela termologia deste tipo de manutenção, indica que tudo será planejado, e consequentemente tende a ficar mais barato, rápido e seguro.

2.1.2 Manutenção preventiva

Segundo Kardec, Nascif e Baroni (2002), a manutenção preventiva é a atuação realizada de forma a reduzir ou evitar a falha ou queda no desempenho, obedecendo a um plano previamente elaborado, baseado em intervalos definidos de tempo.

Ao contrário da manutenção corretiva, a manutenção preventiva busca evitar a incidência de falhas, ou seja, procura prevenir. A manutenção preventiva é mais conveniente à medida que a manutenção seja mais simples a reposição, quando os custos com as manutenções corretivas forem altos, quando as falhas impactarem no processo produtivo, segurança pessoal e operacional.

2.1.3 Manutenção detectiva

Com surgimento na década de 90, a manutenção detectiva está ligada a palavra detectar. Segundo Kardec, Nascif e Baroni (2002), a manutenção detectiva é a atuação efetuada em sistemas de proteção, comando e controle, buscando detectar falhas ocultas ou não perceptíveis ao pessoal de operação e manutenção.

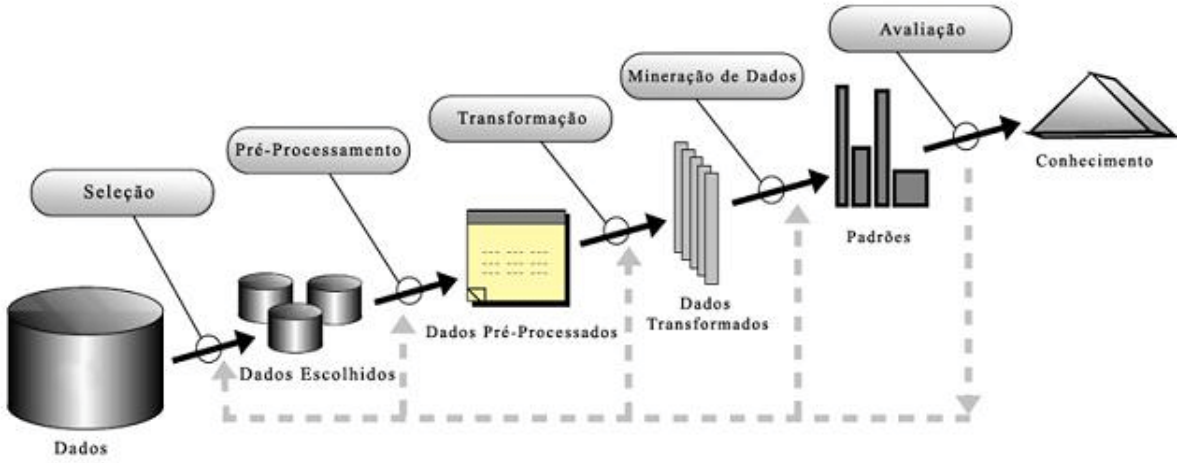
2.2 Descoberta de conhecimento em base de dados

A mineração de dados é um processo de descobrir vários modelos, resumos, e valores derivados de uma determinada coleção de dados (KANTARDZIC, 2011). A mineração de dados, que é uma das bases para o processo de *Knowledge-Discovery in Databases* - em português, Descoberta de Conhecimento em Bases de Dados (KDD), consiste no processo de extrair conhecimentos ocultos ou padrões não triviais e implícitos de uma grande quantidade de base de dados (ZHAO; BHOWMICK, 2003). Na Figura 1 podemos observar os passos para a execução de um KDD.

De acordo com a Figura 1, devemos selecionar os dados que são importantes para as análises; em seguida é feita a limpeza dos dados na etapa de pré-processamento; logo após é realizada a transformação dos dados para se adequarem na etapa de mineração para extração de informações; e por fim, é necessário realizar a interpretação e identificação dos padrões interessantes.

Dentro do processo de mineração de dados está imerso o uso de diversas técnicas e tarefas que auxiliam na resolução de problemas. As técnicas são os algoritmos utilizados, e as tarefas são as classes dos problemas utilizados. As tarefas da mineração de dados podem ser separadas em preditivas e descritivas, nas quais a preditiva gera uma predição sobre uma amostra de dados a partir do conhecimento adquirido do conjunto de dados, e a tarefa descritiva identifica padrões de comportamento em comum na base de dados.

Figura 1 – Etapas do KDD



Fonte: Fayyad e Piatetsky-Shapiro (1996)

As tarefas preditivas são compostas por duas categorias, a classificação e a regressão. A classificação busca informar qual classe que determinado elemento da base de dados pertence de acordo com suas características. Já na regressão, é realizada uma estimativa de um valor para determinado atributo a partir de suas características.

2.3 Algoritmos de mineração de dados

Nesta seção será apresentada brevemente os algoritmos de mineração de dados utilizados no desenvolvimento deste trabalho.

2.3.1 Algoritmo *Random Forest*

O algoritmo *Random Forest* é um algoritmo supervisionado de aprendizado de máquina que é bastante utilizado na resolução de problemas de regressão e classificação. Segundo Breiman (2001), esse algoritmo utiliza a combinação dos resultados de várias árvores de decisão para criar um resultado único.

De acordo com IBM (2020), este algoritmo fornece alguns benefícios, tais como reduzir o risco de *overfitting*, fornecer flexibilidade visto que funciona para problemas de classificação e regressão, e facilitar a determinação do atributo mais importante da base de dados. Como em qualquer recurso há os prós e contras, o *Random Forest* é árduo devido ao tempo de processamento necessário, requer mais recurso computacional e é mais complexo que uma árvore de decisão simples.

2.3.2 Algoritmo *Naive Bayses*

De acordo com Webb (2017), o *Naive Bayes* é um simples algoritmo de aprendizado de máquina que utiliza o teorema de Thomas Bayes (1701 - 1761) aliado a uma forte suposição que os atributos são condicionalmente independentes dada cada classe. Basicamente, este classificador assume que a presença de um atributo da classe não é correlata com a presença de outro atributo. Uma variação deste algoritmo é a utilização da curva de distribuição Gaussiana para assumir a probabilidade de cada atributo em relação à classe.

2.3.3 Algoritmo *XGBoost*

Segundo a documentação oficial do XGBoost (2014), o XGBoost é uma biblioteca para aplicação em aprendizado de máquina, na qual utiliza a técnica de *Gradient Boosting*, otimizado para ser mais eficiente, flexível e portátil. A técnica *Gradient Boosting* consiste na combinação do resultado de vários preditores que possuem baixa acurácia, como as árvores de decisão, para gerar um modelo preditivo mais apurado. A biblioteca do *XGBoost* permite o impulsionamento paralelo das árvores de decisão, que auxiliam na resolução dos problemas de forma mais rápida e com maior acurácia. Além disso, essa biblioteca é versátil, pois pode ser utilizada para problema de classificação e regressão.

2.4 Métricas de avaliação

Nesta seção, são apresentadas as métricas utilizadas para avaliar a execução dos algoritmos analisados neste trabalho. As métricas foram utilizadas de acordo com a metodologia de mineração de dados tais como acurácia, *F-score*, revocação, e matriz de confusão, de acordo com as definições do Harrison (2019). A seguir segue a definição de cada métrica utilizada:

- **Acurácia:** Esta métrica informa a divisão entre os resultados corretos sobre todas as tentativas de predição (corretas e erradas), de modo que quanto maior a acurácia, mais acertivo é o algoritmo. Basicamente, a acurácia é a porcentagem de classificações corretas;
- **Revocação (*recall*):** Também conhecido como sensibilidade, a revocação ou *recall* em inglês, determina a proporção de resultados que são positivos realmente foram classificados corretamente. Essa métrica segue a fórmula 2.1, onde o verdadeiro positivo representa a quantidade de vezes que classe foi predita como verdadeira e a verificação também foi verdadeira, enquanto o falso negativo demonstra que a predição era negativa e a verificação foi positiva;

$$Recall = \frac{verdadeiro\ positivo}{verdadeiro\ positivo + falso\ negativo} \tag{2.1}$$

- **Precisão (*precision*)** Segundo Harrison (2019), é a porcentagem de predições positivas que estavam corretas, de acordo com a fórmula abaixo;

$$Precision = \frac{verdadeiro\ positivo}{verdadeiro\ positivo + falso\ positivo} \tag{2.2}$$

- ***F-score:*** Essa métrica representa um número entre 0 e 1 que representa a média harmônica das métricas precisão e revocação, conforme a seguinte equação:

$$F\text{-}score = \frac{2 * recall * precision}{recall + precision} \tag{2.3}$$

- **Matriz de Confusão:** É uma forma de tabela que permite observar o desempenho de um algoritmo de classificação (DUDA PETER HART, 2001). Os resultados são analisados em verdadeiros positivos, falsos positivos, falso positivo, e falso negativo. Dessa forma, é indicado a quantidade dos acontecimentos que o algoritmo teve para cada um dos resultados;

Tabela 1 – Exemplo da matriz de confusão

Matriz de Confusão		**Classe Predita**	
		Positiva	**Negativa**
Classe original	**Positiva**	VP	FN
	Negativa	FP	VN

2.5 Tecnologias

Nesta seção são apresentadas as ferramentas tecnológicas usadas na aplicação das técnicas de mineração de dados para criação do modelo de predição. Entende-se por preditor os algoritmos de mineração de dados que geram determinadas predições.

2.5.1 *Python*

Python é uma linguagem de programação de alto nível criada por Guildo van Rossum em 1991, sendo ela multi-programada. O *Python* é muito versátil, o que faz ser utilizado para diversas aplicações, como mineração de dados, desenvolvimento web, análises científicas, desenvolvimento de jogos e muito mais (PYTHON, 2022).

Atualmente o *Python* está sob o modelo de código aberto, ou seja, permite seja utilizado sem custo e que seus usuários e colaboradores possam contribuir para o código fonte, mantendo a linguagem sempre atualizada. Essa linguagem é mantida pela empresa *Python Software Foundation* (PSF).

2.5.2 *Numpy*

Numpy[1] é uma biblioteca de código aberto do *Python* criada em 2005 para manipulações de matemáticas de alto nível e com o suporte a processamento de grandes matrizes e arranjos multidimensionais. O *Numpy* é bastante utilizados em tarefas de mineração de dados, processamento de imagens e computação gráfica, e diversos cálculos matemáticos (NUMPY, 2022).

2.5.3 *Pandas*

O *Pandas*[2] é uma biblioteca em código aberto e gratuita criada em 2008 por Wes McKinney que fornce uma gama de soluções para análises e manipulações de dados, para ser utilizada na linguagem *Python*.

Segundo o site oficial do *Pandas*, entre as diversas possibilidades, essa biblioteca destaca em permitir criar matrizes de dados de forma muito rápida e eficaz, possibilitar ler e escrever diversos arquivos de dados, e realizar as manipulações desejadas sobre os dados.

2.5.4 *Seaborn*

O *Seaborn*[3] é uma biblioteca em *Python* que fornece uma interface de alto nível para criar elegantes gráficos estatísticos, auxiliando a visualização sobre determinado conjunto de dados (WASKOM, 2021).

De acordo com Tukey (1977), a visualização de dados é uma parte imprescindível do processo científico. Uma visualização efetiva sobre os dados permite ao cientista entender os próprios dados e comunicar suas conclusões com outros.

2.5.5 *Scikit-learn*

O *Scikit-learn*[4], também conhecido como *sklearn*, é uma biblioteca em *Python* de código aberto com diversos algoritmos de aprendizado de máquina, para resolver problemas de supervisionado e não supervisionado (PEDREGOSA et al., 2011). Em sua gama de algoritmos de aprendizado de máquina de classificação e regressão, destaca-se o *Random Forest*, algoritmos de clusterização, entre outros.

[1] Numpy: <https://numpy.org/>
[2] Pandas: <https://pandas.pydata.org/>
[3] Seaborn: <https://seaborn.pydata.org/>
[4] Scikit-Learn: <https://scikit-learn.org/stable/>

2.6 *Big data* e análise de dados na indústria da mineração

De acordo com a definição da Oracle (2022), o *Big Data* é o conjunto de dados que possuem caracteristicas como alto volume de dados, velocidade de entrega das informações, e variedade da fonte de dados. Esses caracteristicas são os pilares do *Big Data*, também conhecido como os três V's.

De forma simplificada, *Big Data* é o conjunto avançado de dados gerados pela ação de computadores, celulares, circuitos embarcados e outros dispositivos eletrônicos. Essas informações tornaram-se mais predominantes à medida que inovações como Identificação por Rádio Frequência (RFID) e telemática progrediram (VAUGHAN, 2014).

O *big data* e análise de dados são uma realidade inevitável para vários tipos de indústrias, incluindo a indústria de mineração, visto que o preço dos minérios é bem variável e os equipamentos utilizados para o processo produtivo são caras para manter e adquirir. Dessa forma, a coleta e análise de dados de produção e manutenção dos equipamentos são de grande importância para o ramo da mineração a fim de produzir planos de manutenção e produção cada vez mais eficientes.

Atualmente, graças as tecnologias de *big data* e *Internet of Things* (IoT), as empresas podem coletar cada vez mais informações com um nível de detalhamento aprimorado, como alarmes de falhas nos equipamentos, localização geográfica, peso, velocidade, entre outros, em cada etapa do processo produtivo. Com isto, estes conjuntos de dados podem ser analisados para gerar um ambiente mais seguro e produtivo.

2.7 Industria 4.0

A indústria 4.0, termo bastante utilizado atualmente, demonstra a revolução tecnológica que ocorre nas empresas, aprimorando a maneira que produzem e distribuem seus produtos. Tecnologias como IoT, Inteligência Artificial, Aprendizado de Máquina e tecnologias de sensoriamento avançado, são utilizadas por diversas companhias que estão na transformação 4.0 (IBM, 2022).

As empresas nesta revolução 4.0 são caracterizadas também por serem "empresas inteligentes", ou seja, por meio dos diversos sensores avançados e tecnologias de software embarcados, elas coletam e analisam dados da cadeia de produção e permitem gerar conhecimento para uma tomada de decisão mais assertiva. Coletar e analisar essa grande quantidade de dados dos sensores do chão de fábrica asseguram uma visão em tempo real do processo, o que pode fornecer informações para predição de falhas nos equipamentos a fim de diminuir o tempo de máquina parada.

2.7.1 Sistema de telemetria na mineração

Um sistema de sensoriamento avançado da indústria 4.0 bastante utilizado na mineração, fórmula 1 e empresas de logística, é o sistema de telemetria. A telemetria é um processo de comunicação altamente automatizado que coleta dados de instrumentos localizados remotamente e transmite para um equipamento receptor para realizar monitoramento, visualização e gravação de informações (BRITANNICA, 2013).

Figura 2 – Exemplo do sistema de telemetria na mineração

Fonte: Mining (2022)

Na mineração, este sistema de telemetria é capaz de coletar diversas informações das frotas que transportam o minério, como frotas de caminhões, carregadeiras e trens. Na Figura 2 podemos ver um exemplo de como a telemetria funciona nas empresas de mineração e como é a interação entre os diversos equipamentos com este sistema. Com os caminhões fora de estrada, por exemplo, o sistema de telemetria auxilia as empresas com a transmissão de informações importantes para a cadeia produtiva, como velocidade, distância percorrida, toneladas carregadas, consumo de combustível, alarmes de manutenção e diversos outros indicadores de operação e manutenção. Na Figura 3 é possível visualizar os diversos sensores utilizados no caminhão fora de estrada e os quais o sistema de telemetria transmite as informações para demais sistemas. Através deste sistema, as mineradoras conseguem aprimorar sua cadeia produtiva com análises sobre dados obtidos e tomada de decisão assertiva, reduzindo custos e aumentando a produtividade.

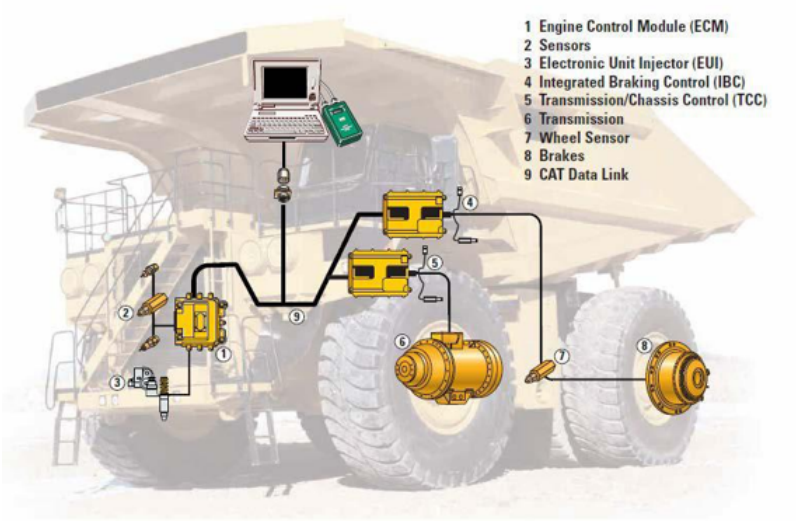


Figura 3 – Exemplo de sensores no caminhão fora de estrada coletados pela telemetria

3 Trabalhos Relacionados

Kahraman (2018) analisaram os alarmes gerados pela telemetria de caminhões fora de estrada que podem ocasionar uma falha corretiva nos mesmos. Os dados foram coletados do sistema de telemetria utilizado nos 11 caminhões de uma mineradora norte-americana durante o período de 9 meses. Foi utilizado os passos do KDD para preparar os dados e posteriormente aplicar os algoritmos. A base de dados contém três tabelas principais: "status", "produção", e "status da saúde dos caminhões". A tabela "status"apresenta o status que os caminhões se encontram, podendo ser falha, hibernado, produção, etc. A tabela "produção"informa os ciclos de movimentações realizadas pelos caminhões com informações de data, nome do operador, tipo do material, localização, entre outros. Por fim, a tabela "status da saúde dos caminhões"informa os dados gerados pelos sensores em diversas partes dos caminhões e transmitidos pela telemetria.

Nesta abordagem, Kahraman (2018) utilizou a técnica de mineração de padrão sequencial, que consiste em encontrar padrões que ocorrem consecutivamente na base de dados ou que podem estar relacionados ao tempo ou outros valores (HAO-EN, 2010). Na implementação da técnica do padrão sequencial, foram agrupados os três códigos da telemetria que estão mais relacionados as falhas nos equipamentos. Os resultados apresentados desta implementação indicam uma taxa de detecção de 90% em uma amostragem de 5 turnos de operação dos caminhões. No entanto, os resultados não mostram altas taxas de detecção para os alarmes e sinais dos últimos três turnos antes que ocorram falhas nos caminhões.

4 O problema da pesquisa

Neste trabalho, o problema estudado pertence a uma mineradora multinacional localizada em Minas Gerais, na qual possui diversos equipamentos de grande porte fundamentais ao processo produtivo, como os caminhões fora de estrada exemplificado na Figura 4. Para que não haja perca de produção, estes equipamentos necessitam estar disponíveis para a operação o maior tempo possível. Com isso, a empresa adota a estratégia de realizar manutenções preventivas ou programadas nestes equipamentos a fim de evitar que falhem de forma corretiva não planejada e evitarem custos excessivos, com mencionado na seção 2.1.1.

Figura 4 – Exemplar do caminhão fora de estrada Caterpillar 777F/G

A mineradora em estudo possui uma frota de 19 caminhões do modelo apresentado na Figura 4 localizado na mina de Água Limpa em Rio Piracicaba-MG. O custo para manter a operacionalização destes equipamentos é alto, devido ao custo com as manutenções necessárias, manutenções corretivas e custos com diesel e pneus. Dessa forma, otimizar o investimento financeiro na operacionalização dos equipamentos torna-se um diferencial competitivo para a mineradora.

Uma vez que em qualquer equipamento mecânico a manutenção corretiva não planejada é indesejada, pois é custosa e pode acarretar impactos no processo produtivo, a mineradora em estudo busca formas de prever quando determinada manutenção corretiva acontecerá, e assim, reduzir custos e aumentar a produtividade.

Para tentar prever algumas falhas nos equipamentos, a mineradora conta com

análises de colaboradores técnicos sobre dados fornecidos pelo sistema de telemetria instalados nos caminhões, os quais informam diversos parâmetros necessários para as análises. No entando, essa é uma tarefa despendiosa, visto que deve-se analisar caminhão por caminhão dos 19 existentes, e muitas das vezes não é possível identificar padrões que indicam uma possível falha. Reconhecer quando determinada falha pode acontecer é o principal objetivo deste trabalho, onde propõe-se um modelo preditivo através de algoritmos de classificação para identificar quais combinações dos parâmetros do caminhão podem ocasionar uma falha de manutenção.

5 Desenvolvimento

Neste Capítulo é apresentado o descritivo do banco de dados da mineradora em estudo e as etapas do desenvolvimento deste trabalho. O descritivo do banco de dados é apresentado na Seção 5.1, o pré-processamento na Seção 5.2, e a aplicação dos algoritmos na Seção 5.3.

5.1 Banco de dados

A base de dados deste trabalho é composta por informações fornecidas pelo sistema de telemetria instalado nos 19 caminhões fora de estrada do modelo Caterpillar 777 F/G da mineradora. A telemetria até a data deste estudo não fornece uma interface de comunicação direta para coleta dos dados automática. Dessa forma, foi necessário acessar o site da telemetria e baixar os relatórios mensalmente no formato de planilhas eletrônicas do tipo (.xlsx) entre o período de Março/2021 a Janeiro/2022 com diversas informações dos caminhões. Vale ressaltar que a telemetria existente coleta as informações dos caminhões de minuto em minuto e as armazena no banco de dados da empresa responsável pelo sistema.

Para consolidar a base de dados foram utilizados dois relatórios disponíveis no sistema de telemetria, o relatório de alarme e o relatório de produtividade. Na planilha do relatório de alarme, são informados os eventos de alarmes/falhas emitidos pelos caminhões e coletados pela telemetria, e no relatório de produtividade são apresentados os dados referentes à produtividade do caminhão durante o transporte do minério.

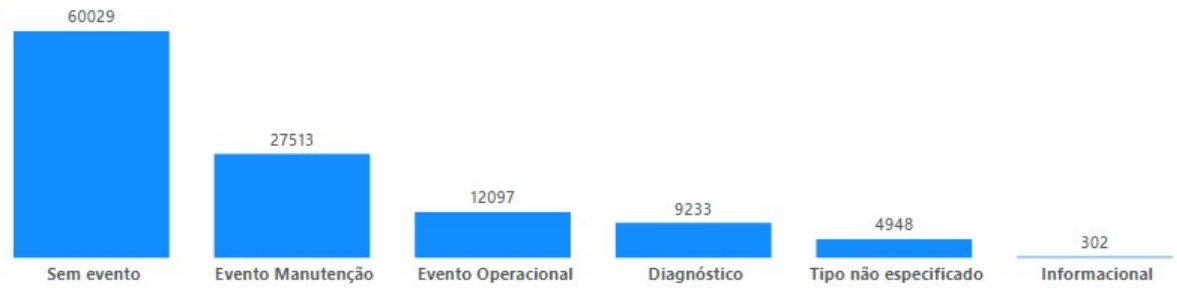
Para prever as falhas de manutenção nos caminhões é preciso correlacionar os dois relatórios, visto que um fornece os eventos dos caminhões e o outro fornece dados sobre produtividade dos caminhões por trajetos, de modo a haver correlação entre parâmetros de operação e alarmes de falha. Dessa forma, criou-se um arquivo consolidado no formato de planilha onde foram inseridos os dados do relatório de alarme em uma aba da planiha e os dados do relatório produtividade em outra aba. Em seguida, criou-se uma terceira aba onde consolidou-se os relatórios através de fórmulas em *Excel* correlacionando identificadores únicos entre os relatórios. Vale ressaltar que nem todos os trajetos percorrido pelo caminhão geram alarmes ou falhas, ou seja, o caminhão pode realizar um percurso sem evento algum. Desse modo, na base consolidada identificou-se registros sem preenchimento no atributo "Evento", significando que em determinado percurso não houve nenhum evento.

Assim, esses registros foram preenchidos automaticamente por fórmulas com a categoria "Sem evento".

Na planilha unificada, de aproxidamente 100 megabytes, está presente em cada linha a indicação da movimentação de minério realizado por determinado caminhão, com os parâmetros de produtividade e os eventos nas colunas. A Tabela 2 informa os atributos existentes na base de dados.

A fim de entender o comportamento dos dados presentes no arquivo, elaborou-se uma análise gráfica sobre a base de dados com a distribuição quantitativa por cada categoria do atributo "Evento", apresentada na Figura 5.

Figura 5 – Análise quantitativa das categorias do atributo "Evento"



Fonte: Elaborado pelo autor

O atributo "Evento" é o atributo classe que se quer prever na resolução do problema, visto que demonstra os eventos de manutenção emitidos pelos caminhões. A descrição de cada valor categórico disponível neste atributo é detalhado nos tópicos a seguir:

- **Evento de Manutenção:** alarme emitido quando determinado componente do caminhão falha;
- **Evento Operacional:** alarme emitido quando há alguma anormalidade na operacionalização do caminhão, por exemplo, carga excessiva no caminhão;
- **Diagnóstico:** alarme emitido quando há alguma anormalidade que dificulta o diagnóstico da telemetria, por exemplo, mal contato em sensores;
- **Informacional:** alarme emitido quando é utilizado alguma funcionalidade incorreta no sistema de telemetria instalado no caminhão, por exemplo, botão do *display* apertado por muito tempo;
- **Tipo não especificado:** alarme emitido quando não é encontrado um código na telemetria que especifique o evento;
- **Sem evento:** refere-se a quando não há nenhum alarme emitido pelo caminhão correlacionado a determinado trajeto no relatório de produtividade.

Tabela 2 – Descritivo dos atributos da base de dados

Atributo	**Tipo do dado**	**Descrição**
TAG	Categórico	Informa qual é a placa de identificação do caminhão fora de estrada.
Código do Cartão	Numérico	Informa qual é o código do cartão do operador.
Início	Data/Hora	Informa a data e hora do início da viagem do caminhão.
Fim	Data/Hora	Informa a data e hora do fim da viagem do caminhão.
Carga	Numérico	Informa a quantidade em toneladas transportadas pelo caminhão.
<80%	Binário	Informa se a carga do caminhão está mais de 20% abaixo da capacidade de carga do caminhão. 1 indica que sim, e 0 que não.
80% - 100%	Binário	Informa se a carga do caminhão está entre o limite de 100% e 20% abaixo da capacidade de carga do caminhão. 1 indica que sim, e 0 que não.
100% - 110%	Binário	Informa se a carga do caminhão está entre o limite de 100% e 10% acima da capacidade de carga do caminhão. 1 indica que sim, e 0 que não.
110% - 120%	Binário	Informa se a carga do caminhão está excede entre 10% e 20% da capacidade da carga do caminhão. 1 indica que sim, e 0 que não.
>120%	Binário	Informa se a carga do caminhão está mais do que 20% acima da capacidade de carga do caminhão. 1 indica que sim, e 0 que não.
Total	Numérico	Informa o consumo de combustível total em litros consumidos no percurso.
Carregado	Numérico	Informa o consumo de combustível em litros consumidos enquanto o caminhão estava carregado no percurso.
Vazio	Numérico	Informa o consumo de combustível em litros consumidos enquanto o caminhão estava vazio no percurso.
Média	Numérico	Informa a velocidade média executada pelo caminhão no percurso.
Média Vazio	Numérico	Informa a velocidade média executada pelo caminhão enquanto estava vazio durante o percurso.
Média Carregado	Numérico	Informa a velocidade média executada pelo caminhão enquanto estava carregado durante o percurso.
Média Sem Manobra	Numérico	Informa a velocidade média executada pelo caminhão no percurso sem considerar tempos de manobras de estacionamento e basculamento.
KM Total	Numérico	Informa a distância percorrida pelo caminhão no trajeto em quilômetros.
Evento	Categórico	Informa o evento registrado no caminhão. Este atributo pode assumir as seguintes categorias: Eventos manutenção, Evento operacional, Diagnóstico, Informacional e Tipo não especificado.
Descrição	Categórico	Informa a descrição do evento registrado. Caso a categoria do atributo Evento seja “Sem evento”, este atributo será nulo ou vazio.

Fonte: Produzida pelo autor

5.2 Pré-processamento

De acordo com Han, Kamber e Pei (2012), as bases de dados reais são altamente suscetíveis a ruídos, inconsistência dos dados, e ausência de informações devido ao tamanho

da base, que é geralmente grande (muitas vezes com vários gigabytes), e as múltiplas fontes heterogeneas da base. Com isso, existem diversas técnicas de pré-processamento que podem ser aplicadas na base de dados para solucionar os ruídos, entre elas: Limpeza, Integração, Redução e Transformação dos dados. Dessa forma, nesta seção serão apresentadas as etapas do pré-processamento utilizadas neste trabalho.

A primeira etapa foi realizar a limpeza dos dados, excluindo tuplas com registros incorretos e a atribuição de valores ausentes em atributos. Para a exclusão, foram identificados registros negativos em atributos numéricos que deveriam ser positivos, caracterizando erro no registro de dados da telemetria, como por exemplo, valores negativos para a distância percorrida pelo caminhão no atributo "KM Total". Já para a atribuição de valores, foram identificados valores ausentes no atributo "Evento", os quais foram substituidos pela categoria "Sem evento", visto que demonstra que para determinado trajeto não houve qualquer evento registrado no caminhão, conforme mencionado na seção 5.1.

A segunda etapa foi realizada a redução dos dados, onde foi removido tuplas que haviam as categorias (Diagnóstico, Evento operacional, Informacional, e Tipo não especificado) no atributo "Evento", visto que são categorias não representativas para o objetivo de prever eventos de manutenção. Dessa forma, resultou-se em duas classes únicas para o atributo "Evento", sendo elas, Evento manutenção e Sem evento. Sendo assim, este atributo foi selecionado para ser o atributo classe do problema, ou seja, os algoritmos identificarão se determinado registro é um evento de manutenção ou não com base nos demais atributos da base.

Ainda na etapa de redução dos dados, buscou-se reduzir a dimensionalidade da base, a fim de evitar o *overfitting*, diminuir o tempo para treinar o algoritmo e melhorar o desempenho. Para isso, foram realizadas análises sobre os dados onde identificou-se atributos desnecessários para a tarefa de classificação, pois são dados que não informam um possível causador de um evento de manutenção. Dessa forma, foram removidos os seguintes atributos "TAG", "Código do Cartão", "Início", "Fim" e "Descrição". A Tabela 3 informa os atributos restantes.

Após a conclusão das etapas de pré-processamento mencionadas, a base de dados está apta para ser utlizada na aplicação dos algoritmos de mineração de dados.

5.3 Aplicação dos algoritmos

Nesta Seção será apresentado todo o desenvolvimento sobre a aplicação dos algoritmos de mineração de dados citados na Seção 2.3. Os algoritmos foram implementados na ferramenta *Google Colab* utilizando a linguagem *Python* juntamente com a bibioteca *Scikit-learn*, *Seaborn*, e *Pandas*, explicados individualmente na Seção 2.5. Ressalta-se também que nesta Seção os algoritmos foram aplicados usando todos os 14 atributos

Tabela 3 – Atributos da base de dados.

ID	Atributo
1	<80%
2	80% - 100%
3	100% - 110%
4	110% - 120%
5	>120%
6	Total
7	Carregado
8	Vazio
9	Média
10	Média Vazio
11	Média Carregado
12	Média Sem Manobra
13	KM Total
14	Carga

Fonte: Produzida pelo autor

listados na Tabela 3 que estão disponíveis na base.

Para a validação da execução dos algoritmos foi utilizado o método de *Cross-Validation*, Validação Cruzada em português, de k partes. Confome cita Han, Kamber e Pei (2012), esse método particiona os dados da base em k partes aproximadamente do mesmo tamanho sendo mutualmente excludentes, onde em cada execução é escolhido um dos segmentos para teste, enquanto as demais partições são usadas para treinar o algoritmo. O algoritmo é executado k vezes, de modo que cada partição seja utilizada para testar o algoritmo exatamente uma vez. Han, Kamber e Pei (2012) cita que em geral, dez é o número recomendado de partições para estimar a acurácia devido ao seu viés e variancia relativamente baixos. Dessa forma, neste trabalho os algoritmos foram executados com 10 partições e também em 5 partições para assegurar a eficiência dos algoritmos.

Ao aplicar os algoritmos, adotou-se algumas métricas para avaliar o desempenho, demonstradas na Seção 2.4. Em todos os experimentos foram geradas tabelas com as informações das métricas para cada algoritmo e o determinado valor de k escolhido, verificando o desempenho de predição das duas categorias do atributo classe “Evento”.

5.3.1 Execução dos algoritmos com 5-fold

Neste experimento, executou-se os algoritmos com o parâmetro k definido em cinco para todos os algoritmos. Os resultados obtidos pelas métricas F1, revocação, precisão e acurácia estão demonstrados na Tabela 4. A Tabela 5 informa os resultados da métrica matriz de confusão. Os resultados destacados em negrito representam os melhores resultados para a execução.

Como o objetivo principal deste trabalho é prever os eventos de manuteção para que não haja perdas financeiras e de produção, o resultado de algumas das métricas utilizadas

Tabela 4 – Resultados execução - 5 fold

Algoritmo	Categoria predita	F1	Revocação	Precisão	Acurácia
Random Forest	Evento manutenção	**54,67%**	**47,38%**	**65,00%**	**62,40%**
	Sem evento	**67,88%**	76,18%	**61,00%**	**62,40%**
Naive Bayses	Evento manutenção	31,85%	23,11%	51,00%	52,69%
	Sem evento	63,77%	**79,84%**	53,00%	52,69%
XGBoost	Evento manutenção	48,07%	42,21%	56,00%	56,37%
	Sem evento	62,38%	69,36%	57,00%	56,37%

Tabela 5 – Matriz de confusão - 5-*fold*

		Classe predita	
Algoritmo	**Classe original**	**Evento manutenção**	**Sem evento**
Random Forest	**Evento Manutenção**	**47,38%**	**52,62%**
	Sem evento	23,82%	76,18%
Naive Bayses	**Evento Manutenção**	23,11%	76,89%
	Sem evento	**20,16%**	**79,84%**
XGBoost	**Evento Manutenção**	42,21%	57,79%
	Sem evento	30,64%	69,36%

são mais importantes, tal como a Revocação, que informa a capacidade do algoritmo de detectar com sucesso a classe desejada.

Dessa forma e de acordo com os resultados apresentados nas Tabelas 4 e 5, nota-se que o algoritmo *Random Forest* apresentou o melhor desempenho para o propósito deste trabalho e com o k=5. Vale ressaltar que apesar de haver resultados melhores para a classe "Sem evento", escolher o algoritmo com este resultado não necessariamente será o melhor para a resolução do problema alvo.

5.3.2 Execução dos algoritmos com 10-fold

A fim de melhorar o desempenho e assegurar a eficiência dos algoritmos, neste experimento o parâmentro k foi alterado para dez. De maneira semelhante ao experimento anterior, os resultados obtidos pelas métricas F1, revocação, precisão e acurácia estão demonstrados na Tabela 6. A Tabela 7 informa os resultados da métrica matriz de confusão.

Tabela 6 – Resultados execução - 10 fold

Algoritmo	Categoria predita	F1	Revocação	Precisão	Acurácia
Random Forest	Evento manutenção	**58,83%**	**52,26%**	**67,00%**	**64,99%**
	Sem evento	**69,55%**	76,68%	**64,00%**	**64,99%**
Naive Bayses	Evento manutenção	29,57%	20,72%	52,00%	52,75%
	Sem evento	64,46%	**82,15%**	53,00%	52,75%
XGBoost	Evento manutenção	51,01%	45,61%	58,00%	58,07%
	Sem evento	63,35%	69,50%	58,00%	58,07%

De acordo com os resultados apresentados, o algoritmo *Random Forest* ainda apresenta os melhores resultados para a classificação das categorias na maioria das métricas. Somente as métricas revocação e a matriz de confusão foram melhores para a classe "Sem

Tabela 7 – Matriz de confusão - 10-*fold*

		Classe predita	
Algoritmo	**Classe original**	**Evento manutenção**	**Sem evento**
Random Forest	**Evento Manutenção**	**52,26%**	**47,74%**
	Sem evento	23,32%	76,68%
Naive Bayses	**Evento Manutenção**	20,72%	79,28%
	Sem evento	**17,85%**	**82,15%**
XGBoost	**Evento Manutenção**	45,61%	54,39%
	Sem evento	30,50%	69,50%

evento". Ao analisar a execução anterior com k=5 e a execução com k=10, ambas indicaram o *Random Forest* como o melhor algoritmo, no entanto, a execução com k=10 apresentou melhores resultados, principalmente na métrica revocação, onde houve acréscimo de aproximadamente 5%. Sendo assim, o algoritmo recomendado para implementação com todos atributos da base selecionados, é o *Random Forest* com o parâmetro k=10.

5.4 Seleção de atributos

De acordo com os resultados da aplicação dos algoritmos apresentados na Seção 5.3, esta seção visa aplicar metricas de seleção de atributos a fim de aprimorar o desempenho dos algoritmos.

Para a seleção dos atributos foi utilizado o método Qui-quadrado, que de acordo com Liu e Setiono (1995), utiliza a estatística qui-quadrado de modo que, se o valor do teste for próximo de zero, o atributo investigado e o atributo classe são independentes, e caso contrário, os atributos são dependentes e devem ser selecionados pois são relevantes. Dessa forma, uma das vantagens da seleção de atributo é a redução da dimensionalidade da base de dados, diminuindo o espaço necessário para armazenar os dados e diminuir o tempo de execução dos algoritmos. Com isso, a seleção de atributos é recomendada até que não haja perda no desempenho do algoritmo. A depender do algoritmo adotado, a redução da dimensionalidade aumenta a desempenho do mesmo.

Sendo assim, foi executado a seleção de atributos sobre os 14 atributos apresentados na Tabela 3, onde obteve o fator de dependência do atributo analisado com o atributo classe apresentado na Tabela 8.

De acordo com os resultados do método qui-quadrado apresentado na Tabela 8, nota-se que os atributos "Carregado", "Total" e "Carga" apresentam um fator de correlação com o atributo classe bem maior que os demais atributos.

Para melhor embasamento sobre a quantidade ideal de atributos a serem selecionados, elaborou-se uma análise gráfica com o resultado da execução dos algoritmos com a métrica F-*Score* versus a variação da quantidade dos atributos selecionados. Nesta análise foi escolhida a métrica *F-Score* visto que informa a média harmônica entre as métricas mais

Tabela 8 – Resultado da seleção de atributos

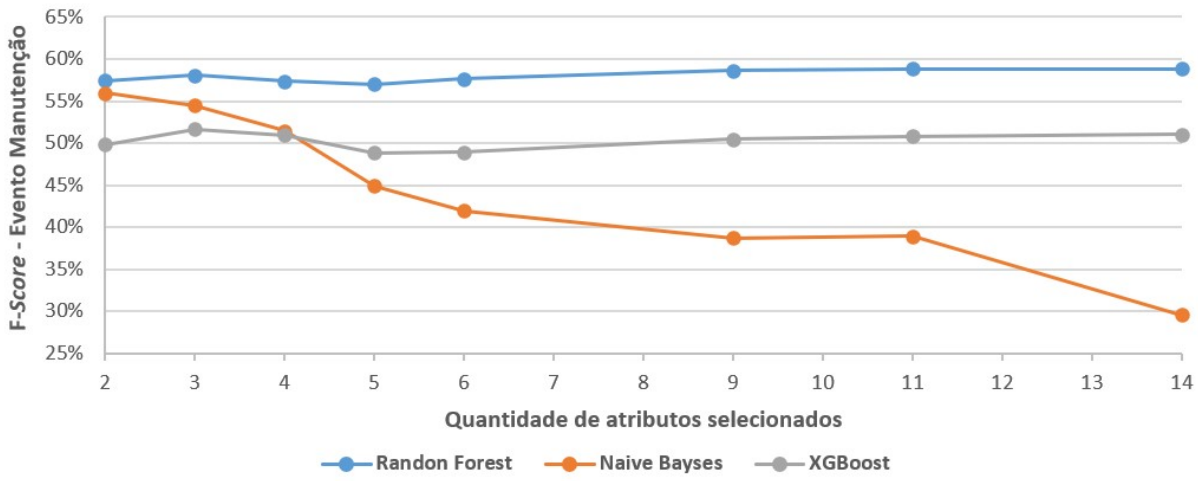
ID	Atributo	*Score* do Qui-quadrado
1	Carregado	6,72
2	Total	6,59
3	Carga	2,2
4	Vazio	0,87
5	KM Total	0,44
6	Média Vazio	0,22
7	100% - 110%	0,16
8	80% - 100%	0,1
9	Média	0,06
10	Média Sem Manobra	0,03
11	<80%	0,02
12	Média Carregado	0,02
13	110% - 120%	0,01
14	>120%	0

Fonte: Produzida pelo autor

importantes para este trabalho, a revocação e a precisão. Além disso, foram considerados somente os resultados para a classe "Evento de manutenção", visto que é a classe desejada para ter o melhor desempenho de predição. Conforme observado na Seção 5.3, a execução com k=10 apresenta melhores resultados na execução, sendo assim, a execução desta análise e as demais execuções dos algoritmos nesta seção, considera somente o parâmetro k=10.

Dessa forma, a Figura 6 apresenta a correlação entre a métrica F-*Score* versus quantidade de atributos selecionados para a classe "Evento de manutenção".

Figura 6 – Análise de correlação entre F-*Score* e quantidade de atributos



Fonte: Elaborado pelo autor

Ao analisar a Figura 6, observa-se que o algoritmo *Naive Bayses* apresentou

resultado inversamente proporcional à quantidade de atributos selecionados e os algoritmos *Random Forest* e *XGBoost* apresentaram pouca variabilidade nos resultados. Além disso, nota-se que cada algoritmo possui o melhor resultado com diferentes quantidade dos atributos selecionados, sendo o *Random Forest* com 14 atributos, o *Naive Bayses* com 2 atributos e *XGBoost* com 3 atributos. Na Tabela 9 são apresentados os atributos selecionados para cada algoritmo de acordo com a análise apresentada, onde os atributos marcados com um "x" representam os escolhidos pelo método qui-quadrado.

Tabela 9 – Atributos selecionados para cada algoritmo

ID	**Atributo**	**Qui-quadrado**	***Random Forest***	***Naive Bayses***	***XGBoost***
1	Carregado	6,72	x	x	x
2	Total	6,59	x	x	x
3	Carga	2,2	x		x
4	Vazio	0,87	x		
5	KM Total	0,44	x		
6	Média Vazio	0,22	x		
7	100% - 110%	0,16	x		
8	80% - 100%	0,1	x		
9	Média	0,06	x		
10	Média Sem Manobra	0,03	x		
11	<80%	0,02	x		
12	Média Carregado	0,02	x		
13	110% - 120%	0,01	x		
14	>120%	0	x		

Fonte: Produzida pelo autor

5.4.1 Execução dos algoritmos com atributos selecionados e 10-fold

Neste experimento, executou-se os algoritmos com o parâmetro k=10 para todos os algoritmos e utilizando os atributos selecionados apresentado na Tabela 9 para cada algoritmo. Os resultados obtidos pelas métricas F1, revocação, precisão e acurácia estão demonstrados na Tabela 10. A Tabela 11 informa os resultados da métrica matriz de confusão.

Tabela 10 – Resultados da execução com 10-*fold* e atributos relevantes

Algoritmo	**Atributos**	**Categoria predita**	**F1**	**Revocação**	**Precisão**	**Acurácia**
Random Forest	14	Evento manutenção	**58,83%**	**52,26%**	**67,00%**	**64,99%**
		Sem evento	**69,55%**	76,68%	**64,00%**	**64,99%**
Naive Bayses	2	Evento manutenção	55,92%	**62,22%**	51,00%	53,06%
		Sem evento	49,81%	44,66%	56,00%	53,06%
XGBoost	3	Evento manutenção	51,69%	48,20%	56,00%	56,89%
		Sem evento	61,08%	64,87%	58,00%	56,89%

De acordo com os resultados apresentados, o algoritmo *Random Forest* ainda apresenta o melhor desempenho para a maioria das métricas analisadas. No entando, de acordo com a Tabela 10, a métrica revocação obteve o melhor resultado para a categoria

Tabela 11 – Matriz de confusão com 10-*fold* e atributos relevantes

			Classe predita	
Algoritmo	**Atributos**	**Classe original**	**Evento manutenção**	**Sem evento**
Random Forest	14	**Evento Manutenção**	52,26%	47,74%
		Sem evento	**23,32%**	**76,68%**
Naive Bayses	2	**Evento Manutenção**	**62,22%**	**37,78%**
		Sem evento	55,34%	44,66%
XGBoost	3	**Evento Manutenção**	48,20%	51,80%
		Sem evento	35,13%	64,87%

"Evento de manutenção" com o algoritmo *Naive Bayses*, sendo 9,96% melhor que o *Random Forest*.

Como mencionado nas execuções anteriores, a métrica revocação possui grande importância visto que informa a eficiência do método em prever as classes corretamente, ou seja, quanto maior o resultado da revocação, maior é a probabilidade de prever a categoria alvo corretamente com baixo indice de falsos negativos. Dessa forma e de acordo com o objetivo do trabalho em prever os "Eventos de manutenção", a escolha do algoritmo deve-se estar pautada na melhor desempenho para a métrica revocação.

Portanto, de acordo com os resultados apresentados, o algoritmo *Naive Bayses* com os dois atributos mais relevantes selecionados, "Carregado" e "Total", e com o 10-*fold*, é o que obteve a melhor desempenho para o objetivo deste trabalho.

Além da análise por métricas estatísticas dos algoritmos, vale ressaltar também o tempo de execução destes algoritmos neste experimento. A Tabela 12 demonstra o tempo médio de execução para cada algoritmo. Observa-se que o *Naive Bayse* possui o menor tempo de execução, o que corrobora com a decisão por utilizar este algoritmo com os parâmetros mencionados, visto que em uma aplicação em tempo real, a tomada de decisão rápida e precisa é vital.

Tabela 12 – Tempo médio de execução dos algoritmos

ID	**Algoritmo**	**Tempo médio de execução**
1	*Random Forest*	27 minutos
2	***Naive Bayses***	**8 segundos**
3	*XGBoost*	30 segundos

Fonte: Produzida pelo autor

6 Considerações Finais

Este trabalho apresentou a aplicação de algoritmos de minerção de dados sobre uma base com eventos emitidos por uma frota de 19 caminhões fora de estrada de uma mineradora. A pesquisa foi iniciada com a revisão bibliográfica dos tipos de manutenções existentes a fim informar qual delas é a mais despendiosa de tempo e recurso e que deve ser evitada ao máximo. Em seguida buscou-se quais informações disponibilizadas pela mineradora para que fosse possível identificar quando determinado equipamento registrasse uma falha, onde foi apresentado o sistema de telemetria instalado nos caminhões.

Para o levantamento dos dados foram selecionados dois tipos de relatórios no sistema de telemetria, onde informam individualmente os dados de produtividade e falhas dos equipamentos. Assim, a próxima etapa da pesquisa foi realizar a organização e limpeza dos dados, consolidando-se em uma única base de dados para que pudesse ser normalizada e fornecida para os algoritmos de classificação. Em seguida, definiu-se as categorias alvo, sendo “Sem evento” e “Evento de manutenção”.

Para executar a tarefa de classificação dos eventos nos caminhões, foram utilizados três algoritmos de classificação (*Random Forest*, *Naive Bayses*, e *XGBoost*) e cinco métricas para a avaliação do desempenho deles.

No primeiro experimento realizado foram aplicados os três algoritmos utilizando todos os 14 atributos disponíveis na base de dados. Além disso, definiu-se a repartição da base de dados para teste e treino dos algoritmos com o método de validação cruzada setado em cinco. Dessa forma, o algoritmo que apresentou o melhor resultado foi o *Random Forest*, com o *F-Score* de 54,67% e revocação de 47,38% para a classe “Evento de manutenção”.

No segundo experimento realizado foi semelhante ao primeiro experimento, porém repartindo a base de dados no método de validação cruzada com dez partes. Dessa forma, o algoritmo que apresentou o melhor resultado foi novamente o *Random Forest*, com o *F-Score* em 58,83% e revocação de 52,26%, representando um aumento de aproximadamente 5% em relação a revocação do experimento anterior.

Para aprimorar os resultados, foi realizada uma seleção de atributos utilizando o método do qui-quadrado, onde identificou-se quais eram os atributos mais relevantes da base de dados e qual a melhor combinação entre a quantidade de atributos selecionados e os algoritmos. Dessa forma, identificou-se que o maior desempenho para o algoritmo *Random Forest* é com todos os 14 atributos da base de dados, para o algoritmo *Naive Bayses* é com os 2 atributos mais relevantes (“Carregado”, “Total”) e para o algoritmo *XGBoost* é com os 3 atributos mais relavantes (“Carregado”, “Total”, “Carga”).

Sendo assim, realizou-se um terceiro experimento aplicando os três algoritmos com os respectivos atributos selecionados para cada método citado anteriormente. Além disso, definiu-se a repartição da base de dados para teste e treino dos algoritmos com o método de validação cruzada setado em 10-*fold*. Dessa forma, o algoritmo com o melhor desempenho foi o *Naive Bayses*, pois apresenta o melhor resultado para a métrica revocação (62,22%) e o menor tempo de execução (Tempo médio de 8 segundos).

De acordo com os experimentos executados, o *Random Forest* apresentou a melhor perfomance na maioria dos experimentos e quase todas as métricas analisadas. No entanto, o *Naive Bayses* obteve a melhor perfomance entre os demais, quando alinhado com a quantidade ideal dos atributos selecionados e com 10-*fold*, com resultado de 62,22% para a revocação, além do tempo de execução ser consideravelmente menor.

É importante observar que mesmo com diversos experimentos e refinamentos, os resultados ainda estão baixos para a previsão das classes para a abordagem proposta neste trabalho. Nota-se também que a classe "Evento de manutenção" obteve taxa de previsão inferior à classe "Sem eventos".

6.1 Trabalhos Futuros

A base de dados contém um periodo curto de amostragem (Março/21 à Jan/22), devido ao tempo para conclusão deste trabalho, e a instalação do sistema de telemetria ter ocorrido em Março de 2021. Continuando este trabalho, pode-se inserir mais eventos coletados dos caminhões e experimentar outros algoritmos de classificação. Para contornar o problema de desbalanceamento entre as classes, algoritmos de sobreamostragem poderiam ser aplicados, como *Synthetic Minority Over-sampling TEchnique (SMOTE)* (CHAWLA et al., 2002) e *Conditional Tabular Generative Adversarial Network (CTGAN)* (XU et al., 2019). Além disso, pode-se avaliar com a mineradora outros relatórios da telemetria relevantes somando-se na base unificada e que podem gerar um melhor desempenho.

Referências

BREIMAN, L. *Random Forest*. Berkeley: *Statistics Department*, 2001. Universidade da Califórnia. Disponível em: <https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf>. Acesso em: 10 abr. 2022. Citado na página 17.

BRITANNICA. *"telemetry"*. 2013. Encyclopedia Britannica. Disponível em: <https://www.britannica.com/technology/telemetry>. Acesso em: 26 mar. 2022. Citado na página 22.

CHAWLA, N. V. et al. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, AI Access Foundation, El Segundo, CA, USA, v. 16, n. 1, p. 321–357, jun 2002. ISSN 1076-9757. Citado na página 38.

DUDA PETER HART, D. S. R. *Pattern classification*. [S.l.]: Wiley, 2001. ISBN 0471056693 9780471056690 0471429775 9780471429777. Citado na página 19.

FAYYAD, U.; PIATETSKY-SHAPIRO, P. S. G. From data mining to knowledge discovery in databases. *AI Magazine*, AAAI, v. 17, n. 3, p. 37–54, 1996. ISSN 0305-0483. Citado na página 17.

HAN, J.; KAMBER, M.; PEI, J. *Data mining concepts and techniques, third edition*. Waltham, Mass.: Morgan Kaufmann Publishers, 2012. ISBN 0123814790. Disponível em: <http://www.amazon.de/Data-Mining-Concepts-Techniques-Management/dp/0123814790/ref=tmm_hrd_title_0?ie=UTF8&qid=1366039033&sr=1-1>. Citado 2 vezes nas páginas 29 e 31.

HAO-EN, C. Mining target-oriented sequential patterns with time-intervals. *International Journal of Computer Science & Information Technology*, v. 2, 08 2010. Citado na página 24.

HARRISON, M. *Machine Learning – Guia de Referência Rápida: Trabalhando com dados estruturados em Python*. Novatec Editora, 2019. ISBN 9788575228180. Disponível em: <https://books.google.com.br/books?id=VvXADwAAQBAJ>. Citado 2 vezes nas páginas 18 e 19.

HIPKIN, I.; De Cock, C. Tqm and bpr: Lessons for maintenance management. *Omega: The International Journal of Management Science*, Elsevier, v. 28, n. 3, p. 277–292, 2000. ISSN 0305-0483. Citado na página 13.

IBM. *Random Forest*. 2020. IBM Cloud Education. Disponível em: <https://www.ibm.com/cloud/learn/random-forest#toc-what-is-ra-DaEaNVdG>. Acesso em: 10 abr. 2022. Citado na página 17.

IBM. *How Industry 4.0 technologies are changing manufacturing*. 2022. IBM site. Disponível em: <https://www.ibm.com/topics/industry-4-0>. Acesso em: 20 fev. 2022. Citado na página 21.

KAHRAMAN, A. Maintainability analysis of mining trucks with data analytics. *Electronic Theses and Dissertations*, University of Louisville, p. 2932, 2018. Citado na página 24.

KANTARDZIC, M. *Data Mining: Concepts, Models, Methods, and Algorithms*. [S.l.]: Wiley-IEEE Press, 2011. Citado na página 16.

KARDEC, A.; NASCIF, J. *Manutenção: função estratégica*. [S.l.]: Editora Quality Mark - Rio de Janeiro, 2009. Citado na página 13.

KARDEC, A.; NASCIF, J.; BARONI, T. *Gestão Estratégica e Técnicas Preditivas*. Rio de Janeiro: Editora Quality Mark, 2002. Coleção Manutenção. Citado 2 vezes nas páginas 15 e 16.

LIU, H.; SETIONO, R. Chi2: Feature selection and discretization of numeric attributes. In: . [S.l.: s.n.], 1995. p. 388 – 391. ISBN 0-8186-7312-5. Citado na página 33.

MINING, M. *Sistema de telemetria*. 2022. Modularmining.com. Disponível em: <https://www.modularmining.com/pt-br/>. Acesso em: 19 fev. 2022. Citado na página 22.

NUMPY. 2022. Disponível em: <https://numpy.org/>. Acesso em: 23 Mar. 2022. Citado na página 20.

ORACLE. 2022. Disponível em: <https://www.oracle.com/br/big-data/what-is-big-data/>. Acesso em: 24 Abr. 2022. Citado na página 21.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, JMLR.org, v. 12, n. null, p. 2825–2830, nov 2011. ISSN 1532-4435. Citado na página 20.

PYTHON. 2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Glossary/Python>. Acesso em: 23 Mar. 2022. Citado na página 19.

TUKEY, J. W. *Exploratory Data Analysis*. Addison-Wesley, 1977. Disponível em: <https://www.worldcat.org/title/exploratory-data-analysis/oclc/614720136>. Citado na página 20.

VAUGHAN, J. *What is machine data?* 2014. WhatIs.com. Disponível em: <https://www.techtarget.com/iotagenda/definition/machine-data>. Acesso em: 26 nov. 2021. Citado na página 21.

WASKOM, M. L. seaborn: statistical data visualization. *Journal of Open Source Software*, The Open Journal, v. 6, n. 60, p. 3021, 2021. Disponível em: <https://doi.org/10.21105/joss.03021>. Citado na página 20.

WEBB, G. I. Naïve bayes. In: _____. *Encyclopedia of Machine Learning and Data Mining*. Boston, MA: Springer US, 2017. p. 895–896. ISBN 978-1-4899-7687-1. Disponível em: <https://doi.org/10.1007/978-1-4899-7687-1_581>. Citado na página 18.

XGBOOST, D. *XGBoost*. 2014. Xgboost.ai. Disponível em: <https://xgboost.readthedocs.io/en/stable/>. Acesso em: 10 Mar. 2022. Citado na página 18.

XU, L. et al. *Modeling Tabular data using Conditional GAN*. arXiv, 2019. Disponível em: <https://arxiv.org/abs/1907.00503>. Citado na página 38.

ZHAO, Q.; BHOWMICK, S. *Sequential Pattern Mining: A Survey. Technical Report, Nanyang Technological University*, Singapore, n. 2003118, 2003. Citado na página 16.