

**GUIA PARA A UTILIZAÇÃO DE APRENDIZADO DE MÁQUINA NA  
CATEGORIZAÇÃO DE DADOS DE MANUTENÇÃO EM  
EQUIPAMENTOS ELETRÔNICOS – PROPOSTA DE FRAMEWORK**

**PRODUTO TECNOLÓGICO**

**Clayton da Silva Alves  
Eugênio de Oliveira Simonetto**

**Santa Maria, RS  
2024**

## SUMÁRIO

|   |   |    |
|---|---|----|
| 1 | INTRODUÇÃO.....   | 3  |
| 2 | ETAPAS PARA IMPLEMENTAÇÃO DO FRAMEWORK MANUTENÇÃO DE<br>EQUIPAMENTOS ELETRÔNICOS EM IFES..... | 5  |
| 3 | CONSIDERAÇÕES FINAIS .....  | 15 |
|   | REFERÊNCIAS .....   | 17 |

## 1 INTRODUÇÃO

As instituições públicas buscam adaptar-se as novas necessidades de seus usuários, onde questões como acesso, transparência, equidade, legalidade, confiabilidade e agilidade fazem com que os agentes públicos busquem desenvolver estratégias de gestão que facilitem suas atividades diárias frente a estes novos desafios (BENETTI, 2020). Além disso, a gestão pública tem sido afetada por mudanças gerenciais, estruturais, operacionais e legislativas que oportunizaram uma relação mais transparente e direta com a sociedade (MONTANA, 2020).

A gestão eficaz da manutenção consiste em um desafio significativo, requerendo habilidades humanas e tomada de decisão para identificar e remediar falhas que podem prejudicar o desempenho dos equipamentos e causar períodos de inatividade (SEGOVIA-RAMIREZ; GARCIA-MARQUEZ, 2020). Lidar com a manutenção corretiva (MC) envolve a identificação precisa da causa raiz da falha em meio a complexos sistemas e componentes (TSEREMOGLOU; SANTOS, 2023). No entanto, avanços tecnológicos estão simplificando esse processo, permitindo a aplicação de técnicas de aprendizado de máquina para a detecção de padrões de falhas e otimização de recursos (YEPEZ; ALSAYYED; AHMAD, 2019).

Além de otimizar a manutenção corretiva, a categorização eficaz das falhas também desempenha um papel crucial na gestão de recursos em instituições públicas, como nas universidades. A capacidade de identificar semelhanças nas falhas pode acelerar a solução de problemas e fornecer informações valiosas para a tomada de decisões relacionadas a compras, desempenho e eficiência (ÖZGÜR-ÜNLÜAKIN; TÜRKALI; AKSEZER, 2021). Isso se torna ainda mais relevante em um contexto de orçamentos limitados, onde o uso eficiente de recursos é fundamental, como é o caso das instituições de ensino superior (FARAGE, 2021).

A automação da categorização de dados de manutenção e o processamento de linguagem natural por meio do aprendizado de máquina não apenas economiza tempo e esforço, mas também permite que a equipe de manutenção se concentre em tarefas estratégicas e na resolução de problemas complexos, alinhando-se com os princípios da *New Public Management* (NPM) (PATRIARCA, et al., 2022). No setor educacional, a inteligência artificial (IA) conseguiu crescer de maneira expressiva devido à sua capacidade algorítmica de fazer recomendações, previsões, decisões e aprender em diferentes contextos (CHEN et al., 2022). A introdução da IA na educação teve como foco a possibilidade de tornar mais fácil a execução de tarefas de maneira mais eficaz e eficiente (MARTÍNEZ-COMESAÑA, et al., 2023).

De acordo com Alam (2021) até 40% do tempo de ensino ainda é comprometido em atividades que poderiam ser automatizadas, o que denota que o uso de IA tem muito espaço para crescer. Além disso, apresenta-se como uma tendência moderna, a de explorar o potencial das tecnologias de aprendizado de máquina, processamento de linguagem natural e *business intelligence* (BI) para melhorar a eficiência e a tomada de decisões em ambientes institucionais (GANDON, 2018).

Estes instrumentos são recursos que têm se mostrado como instrumentos poderosos na resolução de problemas que envolvem múltiplos objetivos, uma vez que podem apresentar dados de maneira a permitir que os usuários compreendam as compensações e escolhas entre objetivos conflitantes (HOFFENSON, et al., 2023). Isso os torna adequados para auxiliar em problemas complexos, nos quais diferentes objetivos, como custo, desempenho e características físicas, devem ser cuidadosamente equilibrados (PATRIARCA, et al., 2022).

Vários profissionais, bem como pesquisadores na literatura, desenvolveram ferramentas de análise que integram a exploração e a modelagem de dados por meio do AM e do PLN em diversas áreas. Essas aplicações abrangem uma ampla gama de campos, como finanças, saúde, gestão da cadeia de suprimentos, educação e segurança de sistemas (HAMOUD, et al., 2024; HILL, et al., 2023; HOFFENSON, et al., 2023; KALIISA; JIVET; PRINSLOO, 2023; KOBESSI et al., 2023; MAGHSOUDI; NEZAFATI, 2023; PATRIARCA, et al., 2022; SCHULZE, et al., 2023; TRIPATHI, et al., 2023).

Ademais, as instituições de ensino lidam diariamente com um extenso volume de dados educacionais, porém, sua utilização fica aquém do esperado e das potencialidades existentes (KABAKCHIEVA, 2015). Na educação, a categorização de dados, do BI e do uso do aprendizado de máquina podem ajudar gestores e professores a extrair e compreender informações de vários sistemas e processos para diagnosticar, monitorar ou avaliar o progresso institucional e acadêmico dos alunos (LEMES; DIAS; OLIVEIRA, 2023).

As instituições de ensino superior enfrentam constantes desafios na gestão de suas operações e recursos (COSTA, et al., 2023). Dentre estes, podem-se citar a eficácia na gestão da manutenção, a qual, muitas vezes, depende do acesso a informações precisas e atualizadas sobre as atividades de manutenção realizadas em todo o campus universitário, dos principais problemas, causas e efeitos (BONA, 2022).

Diante destes aspectos, este produto tecnológico busca servir como guia para replicação em outras organizações, em especial, em instituições federais de ensino superior ao propor um *Framework* que auxilia na aplicação de técnicas de aprendizado de máquina e processamento

de linguagem natural para automatizar e otimizar a categorização de dados de manutenção, visando a melhoria da gestão de manutenção e a eficiência na tomada de decisões.

## 2 ETAPAS PARA IMPLEMENTAÇÃO DO FRAMEWORK MANUTENÇÃO DE EQUIPAMENTOS ELETRÔNICOS EM IFES

Este produto tecnológico tem por base a dissertação desenvolvida ao Programa de Pós-Graduação em Gestão de Organizações Públicas – PPGOP/UFSM se propõe um *framework* de categorização de dados de manutenção, por meio do AM com PLN, que possa ser utilizado nas instituições federais de ensino superior. Neste sentido, este tópico busca delimitar uma proposta de *framework* que classifica componentes defeituosos, tendo por base o modelo treinado, delimitado nos tópicos anteriores, que busca classificar de maneira *multi-label* (vários rótulos), com mecanismos estruturados e automáticos, quais componentes em um *desktop* estão defeituosos. Para isto, utiliza como base, descrições textuais de manutenção e características reportadas sobre os problemas apresentados.

Em síntese, o *framework* proposto pode ser integrado com o processo de manutenção para identificar automaticamente componentes defeituosos. Assim, partindo do recebimento de um *ticket* o técnico do serviço descreve o problema encontrado no *desktop* ou equipamento; após, o *framework* processa a descrição textual de modo automático e prevê os componentes defeituosos, retornando uma lista de possíveis problemas. As previsões são registradas para atualizar o banco de dados de histórico e melhorar o *framework* ao longo do tempo.

Detalhadamente e trazendo a linguagem *python*, buscou-se descrever todo o processo de implementação do *framework*, subdividindo em 5 etapas, para na sequência sintetizar o *framework*. Ressalta-se que a codificação se adequa ao processo de manutenção de *desktops*, entretanto, mediante ajustes dos *hyper-parameters* (especificações do item a sofrer a manutenção e dos seus problemas a serem identificados) pode ser aplicado a diferentes equipamentos e máquinas e na detecção de novos problemas e componentes de manutenção.

Portanto, para implementar o *framework* de classificação de componentes defeituosos deve seguir as etapas a seguir:

### 1. **Coleta de dados: obtenção dos dados.**

Entrada (*Input*): o *framework* usará descrições textuais de problemas encontrados nos *desktops* como entrada de dados. Essas descrições podem ser coletadas a partir de tickets de manutenção, relatórios de problemas, ou registros internos;

**2. Pré-processamento de dados:** as ações desta etapa consistem, basicamente, em se limpar o texto, tokeinizá-lo e lematizá-lo. A partir desta etapa os processos e ações ocorrem em um ambiente de desenvolvimento, que visa implementar o aprendizado de máquina para processamento de linguagem natural. Destaca-se que o *framework* proposto sugere o uso da biblioteca *scikit-learn (Sklearn)* para a implementação do algoritmo em *Python*.

- Limpeza do texto: remover *stopwords*, pontuação, caracteres especiais e transformar as palavras para minúsculas, com fins de obter uma padronização nos dados, removendo aspectos desnecessários, normalizando o texto;
- Tokenização: dividir o texto em palavras individuais ou sequências de palavras. Este procedimento visa extrair informações principais em uma frase, do contrário o AM interpretaria a frase como uma única palavra;
- Lematização: busca reduzir as palavras a seu lema, reduzindo a variação morfológica, ou seja, um mesmo problema descrito com palavras em classe morfológicas diferentes (adjetivo, verbo, substantivo...) se tornam a representação de um mesmo aspecto ou problema;

A implementação desta etapa em Python, a ser utilizada na aplicação em outras instituições, prevê a adaptação do código conforme a necessidade de substituições específicas para adequação ao contexto local. Assim, o Quadro 1 implementa a importação e a combinação de listas de *stop words* da língua portuguesa utilizando duas bibliotecas de processamento de LN, Spacy e NLTK (*Natural Language Toolkit*), conforme expresso no Quadro 1.

Quadro 1 - Stop Words

```
# Importando as bibliotecas de stop words
import spacy
import nltk
from nltk.corpus import stopwords
# Carregando o modelo de idioma em português
nlp = spacy.load('pt_core_news_sm')
# Extraíndo as stop words do modelo
stop_words_spacy = list(spacy.lang.pt.stop_words.STOP_WORDS)
# Baixando o modelo de stop words
nltk.download('stopwords')
# Carregando as stop words em português
stop_words_nltk = stopwords.words('portuguese')
# Mesclando as duas listas de stop words
marge_stop_words = stop_words_spacy + stop_words_nltk
# Retirando as duplicatas e ordenando a lista
stop_words = sorted(list(set(marge_stop_words)), key=marge_stop_words.index)
stop_words = sorted(stop_words)
```

Fonte: Elaborado pelo autor.

Este código tem o objetivo de criar uma lista consolidada e única de *stop words*, combinando os vocabulários das duas bibliotecas tornando o banco de dados mais uniforme. O Quadro 5 implementa a tokenização, limpeza e remoção de *stop words* nas observações de um banco de dados, utilizando as bibliotecas de expressões regulares (re) e pandas para manipulação do banco de dados.

Quadro 2 - Tokenização, limpeza e remoção de stop words

Fonte: Elaborado pelo autor.

O processo inicia com a leitura de um arquivo contendo as observações, sendo convertido em um *DataFrame*. A função `clean_observacaos` é responsável por tokenizar o texto, transformando-o em minúsculas, e removendo as *stop words* definidas no Quadro 4. A tokenização é realizada por meio de uma expressão regular que extrai apenas palavras, preservando caracteres acentuados da língua portuguesa. Finalmente, as palavras limpas são unidas em uma *string* (conjunto de caracteres).

Por fim, o código aplica a função tokenização a cada observação no *DataFrame*, criando uma nova coluna chamada "Observações Limpas", que contém as observações processadas sem as *stop words* e com a limpeza aplicada. O Quadro 3 implementa a lematização das observações limpas de um *DataFrame*, utilizando uma função que aplica o processo de lematização.

### Quadro 3 - Lematização

```
# Função para aplicar lematização nas observações limpas
def lemmatize_text(text):
    doc = nlp(text)
    lemmas = [token.lemma_ for token in doc]
    return ''.join(lemmas)
# Aplicando lematização nas observações limpas
df_observacoes['Observações Lematizadas'] = df_observacoes['Observações Limpas'].apply(lemmatize_text)
```

Fonte: Elaborado pelo autor.

A função `lemmatize_text` utiliza um modelo de PLN, previamente carregado, para processar o texto de entrada. Esta converte o texto em um objeto de documento (*doc*), e, em seguida, extrai os lemas de cada palavra tokenizada, armazenando-os em uma lista. As palavras lematizadas são, então, recombinaadas em uma *string* (conjunto de caracteres).

Por fim, o código aplica a função de lematização a cada observação limpa no *DataFrame*, criando uma nova coluna chamada "Observações Lematizadas". Nessa coluna, as

observações são representadas com palavras em sua forma básica, o que facilita análises posteriores e melhora a uniformidade dos dados textuais.

**3. Dicionário de componentes:** O objetivo consiste em identificar diretamente os componentes dos equipamentos nas descrições textuais dos problemas, utilizando um dicionário predefinido. O dicionário foi desenvolvido criando um mapeamento de palavras ou frases que indicam os componentes específicos. Este mapeamento serve para facilitar a identificação automática dos componentes que podem estar defeituosos com base nas descrições fornecidas. Para isso, utiliza-se um código que percorre as observações textuais e identifica componentes potenciais conforme as palavras-chave ou expressões presentes.

Tal procedimento acelera o processo de classificação e auxilia o treinamento do modelo, fornecendo uma estrutura consistente para relacionar as descrições com os componentes afetados. O Quadro 4 implementa a identificação de componentes relacionados nas observações de um banco de dados de manutenção, utilizando a biblioteca pandas para manipulação dos dados.

Quadro 4 - Dicionário de componentes

(continua)

```

import pandas as pd
# Lendo do arquivo e salvando o banco de dados como um DataFrame
df = pd.read_excel(banco_registros.xlsx)
# Função para identificar os componentes nas observações
def identificar_componentes(observacao):
    observacao = observacao.lower()
    componentes = []
    if 'memória' in observacao or 'modulo' in observacao or 'ram' in observacao or 'pente' in observacao:
        componentes.append('Memória')
    if 'fonte' in observacao:
        componentes.append('Fonte')
    if 'hd' in observacao or 'disco' in observacao:
        componentes.append('HD')
    if any(term in observacao for term in ['placa de rede', 'ethernet', 'conexão de rede', 'adaptador de rede', 'wireless']):
        componentes.append('Placa de Rede')
    if 'placa mãe' in observacao or 'motherboard' in observacao or 'mobo' in observacao or 'placa onboard' in observacao:
        componentes.append('Placa Mãe')
    if any(term in observacao for term in ['cd', 'cdrw', 'dvd', 'gravador']):
        componentes.append('Unidade Ótica')
    if any(term in observacao for term in ['pilha', 'bateria']):
        componentes.append('Bateria de BIOS')
    if any(term in observacao for term in ['disquete', 'drive de disquete', 'floppy', 'floppy disk', 'unidade de disco flexível', 'disco flexível', 'unidade de disquete 3.5', 'drive de 3.5', 'diskette', 'floppy drive']):
        componentes.append('Unidade de Disquete')
    if any(term in observacao for term in ['processador', 'cpu', 'intel', 'amd', 'ryzen', 'i3', 'i5', 'i7', 'i9']):
        componentes.append('Processador')
    if any(term in observacao for term in ['botão', 'botões', 'liga/desliga', 'power']):
        componentes.append('Botões')
    if any(term in observacao for term in ['cabos', 'cabos', 'conector', 'fio', 'fios']):
        componentes.append('Cabos')
    if any(term in observacao for term in ['cooler', 'ventilador', 'fan']):
        componentes.append('Cooler')
    if any(term in observacao for term in ['placa de vídeo', 'gpu', 'nvidia', 'amd', 'radeon', 'geforce', 'graphics card', 'placa de captação de vídeo', 'instalação da placa de vídeo']):
        componentes.append('Placa de Vídeo')
    if any(term in observacao for term in ['som', 'áudio', 'alto-falante', 'speaker', 'placa de som', 'audio card']):
        componentes.append('Som')

```

Quadro 4 – Dicionário de componentes

(conclusão)

```

if 'bios' in observacao:
    componentes.append('BIOS')
if 'teclado' in observacao or 'keyboard' in observacao:
    componentes.append('Teclado')
if 'mouse' in observacao:
    componentes.append('Mouse')
if any(term in observacao for term in ['instalação do windows', 'formatar', 'instalado os
drivers', 'restauração do sistema', 'instalação de programas', 'reinstalação do sistema
operacional', 'xp instalado']):
    componentes.append('Sistema Operacional')
if 'limpeza' in observacao:
    componentes.append('Limpeza')
if any(term in observacao for term in ['montagem', 'jumper']):
    componentes.append('Montagem/Configuração')
if any(term in observacao for term in ['gabinete', 'estrutura']):
    componentes.append('Gabinete')
if any(term in observacao for term in ['tela', 'monitor', 'riscos']):
    componentes.append('Tela/Monitor')
if 'capacitor' in observacao and any(term in observacao for term in ['queimado',
'defeito']):
    if 'fonte' in observacao:
        componentes.append('Fonte')
    else:
        componentes.append('Placa Mãe')
if 'lubrificado' in observacao:
    componentes.append('Lubrificação')
if any(term in observacao for term in ['trocar o sistema operacional', 'sugerido trocar
sistema operacional']):
    componentes.append('Trocando Sistema Operacional')
if any(term in observacao for term in ['usb', 'porta usb', 'placa usb']):
    componentes.append('Periféricos/USB')
if any(term in observacao for term in ['não foi feita alteração', 'não foi executado nenhum
serviço', 'não foi constatada alteração']):
    componentes.append('Sem Intervenção')
if len(componentes) == 0:
    return 'Outro Componente'
return ', '.join(componentes)
# Aplicando a função ao DataFrame
df['Componentes'] = df['Observações Limpas'].apply(identificar_componentes)
# Salvando o DataFrame resultante em um novo arquivo Excel
df.to_excel('observacoes_e_componentes.xlsx', index=False)

```

Fonte: Elaborado pelo autor.

Deste modo, o processo inicia com a leitura de um arquivo Excel que contém as observações de manutenção e, em seguida, cria um *DataFrame* para armazenar e processar

esses dados. A função `identificar_componentes` é responsável por analisar cada observação, verificando a presença de palavras-chave associadas a diferentes componentes de hardware e software. A função converte a observação em letras minúsculas e compara com termos específicos de componentes, como "memória", "fonte", "placa de rede", "hd", entre outros. Para cada termo encontrado, o componente correspondente é adicionado a uma lista.

Após a execução da função `identificar_componentes`, uma nova coluna é criada chamada "Componentes", onde os componentes detectados são listados. Caso nenhuma correspondência seja encontrada, a função atribui "Outro Componente" à observação. Finalmente, o *DataFrame* resultante é salvo em um novo arquivo Excel para posterior análise e consulta.

**4. *Arquitetura do framework*:** o modelo de classificação proposto se baseia na classificação *multi-label*, que usa um *pipeline* composto por vetorização do texto e de um classificador *multi-output* (múltiplas saídas).

- *Pipeline* de treinamento – transforma texto em vetores para, na sequência, aplicar a classificação usando: *CountVectorizer* e *MultiOutputClassifier* com *LogisticRegression*. Para esta etapa de *pipeline*, a implementação em Python consiste no Quadro 5.

Quadro 5 - *Pipeline* de treinamento

```
# Importando as bibliotecas
from sklearn.Pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.multioutput import MultiOutputClassifier
from sklearn.linear_model import LogisticRegression
# Definir o Pipeline de classificação
Pipeline = Pipeline([
    ('vectorizer', CountVectorizer()), # Transformação do texto em vetor
    ('classifier', MultiOutputClassifier(LogisticRegression(max_iter=1000, random_state=42)))
    # Classificador multi-label
])
```

Fonte: Elaborado pelo autor.

- Funcionalidade e treinamento do modelo: consiste em verificar o comportamento do aprendizado de máquina em relação ao defeito reportado textualmente como entrada (input) e

o modelo reportará diversos componentes defeituosos como saída (output). Assim, o modelo, aqui delimitado, foi treinado usando dados históricos de manutenções de *desktops* (de 2003 a 2023) e categorizou múltiplos componentes defeituosos para cada observação de problema.

A técnica utilizada foi classificação *multi-label*, onde cada componente foi tratado como um rótulo separado, e o modelo foi treinado para prever mais de um rótulo por observação. Para novas aplicações, pode-se utilizar o banco de dados específico do setor em análise e aplicar a mesma abordagem, utilizando a técnica descrita a seguir em Python, que consiste no Quadro 6.

Quadro 6 - Treinando o modelo com os dados de treinamento

```
# Treinando o modelo com os dados de treinamento
Pipeline.fit(X_train, y_train)
```

Fonte: Elaborado pelo autor.

Onde:

X\_train: Descrições de problemas limpas.

y\_train: Componentes defeituosos associados, em formato *multi-label* (colunas com 0 e 1 para cada componente).

- Fase de predição: após o treinamento, o *framework* pode ser usado para classificar novos dados de entrada (descrições de problemas). O *Pipeline* vai automatizar todo o processo, desde a vetorização do texto até a classificação dos componentes defeituosos. O sistema melhora com o tempo à medida que mais dados de manutenção são adicionados.

Esse *framework* de classificação de componentes defeituosos pode ser implementado para melhorar o processo de manutenção de *desktops*, automatizando o diagnóstico de problemas e otimizando a eficiência das equipes de suporte técnico.

**5. Avaliação do *framework*** - após treinar e implementar o *framework*, ele deve ser avaliado regularmente para garantir sua eficácia em identificar corretamente os componentes defeituosos. Algumas métricas de avaliação, estão delimitadas no Quadro 10.

- Precisão: A proporção de previsões corretas feitas pelo *framework*;
- Recall: A capacidade do *framework* de identificar todos os componentes defeituosos;

- F1-Score: A média ponderada de precisão e recall.

Quadro 7 - Avaliação do *framework*

```
# Importando a biblioteca
from sklearn.metrics import classification_report
# Avaliando o desempenho do framework em dados de teste
y_pred = Pipeline.predict(X_test)
classification_report(y_test, y_pred)
```

Fonte: Elaborado pelo autor.

Com isso, espera-se que o *framework* proposto possa ser aplicado em diferentes contextos de manutenção de *desktops*, bem como, espera-se que, mediante ajustes nos *hyper-parameters*, este possa ser replicado em manutenções de outros equipamentos, máquinas e problemas, o que permite a escalabilidade do *framework* em outras organizações, em especial, Intituições Federais de Ensino Superior. Destaca-se que o *framework* proposto tem a capacidade de ser integrado com os sistemas de gestão de manutenção para automatizar os processos de diagnósticos tanto de falhas existentes quanto possibilidade de ocorrências futuras.

Além disso, o *framework* proposto tem a capacidade de estabelecer quais itens são mais críticos, por meio da análise ABC, sugerindo estoques de segurança a serem mantidos com base nas falhas mais comuns e críticas, otimizando a aplicação de recursos, muitas vezes escassos, como no caso do setor público. Outro ponto a ser destacado, que buscou-se propor um *framework* que se utilizasse de uma linguagem *Open Source* (código aberto), ou seja, disponível a todos, de maneira gratuita, facilitando sua aplicação.

### 3 CONSIDERAÇÕES FINAIS

O estudo desenvolvido teve como objetivo propor um *framework* que utilize técnicas de aprendizado de máquina e processamento de linguagem natural para categorizar dados de manutenção de equipamentos eletrônicos em instituições federais de ensino superior para otimizar a na gestão de manutenção e na melhoria da eficiência operacional. A pesquisa foi realizada por meio de uma pesquisa qualitativa, descritiva, documental e com a utilização de um banco de dados do setor LAMI-UFSM contemplando 21.177 registros de manutenções, durante os anos de 2003 a 2023.

A proposição deste *framework* de categorização de dados de manutenção, por meio do AM com PLN, que possa ser utilizado nas instituições federais de ensino superior – IFES, em especial, para otimizar a na gestão de manutenção e na melhoria da eficiência operacional, visa atender o preceito de produto tecnológico do Programa de Pós-Graduação em Gestão de Organizações Públicas – PPGOP. Assim, a replicação deste, em diferentes IFES pode facilitar a detecção de problemas recorrentes, auxiliar na tomada de decisão e facilitar que pessoas menos familiarizadas em aprendizado de máquina se beneficiem do seu uso para otimizar a gestão de manutenções. Entretanto, ressaltamos que este *framework* pode ser aplicado em outras organizações, não se restringindo a organizações públicas.

O uso de PLN para extrair informações críticas das descrições de manutenção oferece uma nova perspectiva sobre como gerenciar dados não estruturados, que são abundantes em muitos setores. Ademais, demonstra-se a relevância do uso de técnicas de inteligência artificial, as quais, podem ser aplicadas em outras áreas de gestão de ativos em instituições públicas, o que pode impactar desde a manutenção até o planejamento estratégico e o gerenciamento de riscos. O trabalho contribui, ainda, ao explorar e aplicar o aprendizado de máquina na gestão pública no contexto das IFES.

A relevância prática deste estudo, consiste no fato de que o *framework* proposto, ao qual pode ser reaplicado, automatiza uma tarefa que tradicionalmente é manual, economizando tempo e recursos. O uso de aprendizado de máquina elimina a necessidade de categorizações manuais demorados, permitindo que a equipe de manutenção se concentre em tarefas mais estratégicas, como diagnósticos mais complexos e planejamento preventivo. Além disso, ao implementar uma análise de estoque de segurança com base nos resultados da categorização, o estudo oferece aspectos relevantes que podem ajudar a prever necessidades futuras de peças e

componentes, otimizando assim a gestão de estoques. Tal fato, torna-se ainda mais relevante em um contexto onde a escassez de recursos, como em instituições públicas.

Do ponto de vista social, este trabalho contribui para a melhoria da qualidade dos serviços públicos. Ao automatizar e otimizar a gestão de manutenção, instituições como universidades federais podem oferecer um serviço mais rápido e eficiente, impactando positivamente na experiência dos usuários. Além disso, o uso eficiente de recursos públicos traz benefícios à sociedade, garantindo que os valores destinados à manutenção sejam utilizados de maneira otimizada e eficiente. A transparência na evidenciação dos dados de manutenção, proporcionadas pelo *framework*, também contribuem para uma maior confiança nas instituições públicas, alinhando-se aos princípios da nova gestão pública, que preza pela eficiência, transparência e prestação de contas.

Em relação à linha de pesquisa de inovação e modernização de sistemas e processos administrativos, o trabalho oferece uma contribuição relevante ao desenvolver uma abordagem inovadora, utilizando um *framework* de aprendizado de máquina para categorizar automaticamente os dados de manutenção, o que tem potencial para transformar os processos administrativos e operacionais das IFES, tornando-os mais ágeis, eficazes e alinhados aos princípios de gestão pública moderna e fazendo uso da inteligência artificial no setor público.

## REFERÊNCIAS

- ALAM, A. Possibilities and apprehensions in the landscape of artificial intelligence in education. In: 2021 **International Conference on Computational Intelligence and Computing Applications (ICCICA)**. IEEE, 2021. p. 1-8. DOI: <https://doi.org/10.1109/ICCICA52458.2021.9697272>
- BENETTI, G. Estratégias de gestão da informação do processo ambulatorial do HUSM face à Lei de Acesso à Informação. Dissertação (**Mestrado Profissional em Gestão de Organizações Públicas**) - Universidade Federal de Santa Maria (UFSM), Santa Maria, RS, 2020, 97p.
- BONA, R. S. Evaluación de políticas de integridad pública y anticorrupción: los desafíos de una iniciativa nacional para gobiernos locales en Brasil. **Administração Pública e Gestão Social**, v. 14, n. 4, p. 1-19, 2022. DOI: <https://doi.org/10.21118/apgs.v14i4.13513>
- CHEN, X.; ZOU, D.; XIE, H.; CHENG, G.; LIU, C. Two decades of artificial intelligence in education. **Educational Technology & Society**, v. 25, n. 1, p. 28-47, 2022. DOI: <https://doi.org/10.1016/j.caear.2020.100002>
- COSTA, W. P. L. B.; SILVA, J. D.; BRITO, L. M. P.; SILVA, S. L. P. Modelo de gestão do conhecimento: Uma proposta para o contexto da universidade pública. **Revista de Administração, Sociedade e Inovação**, v. 9, n. 2, p. 46-68, 2023. DOI: <https://orcid.org/0000-0002-6494-1454>
- FARAGE, E. Educação superior em tempos de retrocessos e os impactos na formação profissional do Serviço Social. **Serviço Social & Sociedade**, p. 48-65, 2021. DOI: <https://doi.org/10.1590/0101-6628.237>
- GANDON, F. A survey of the first 20 years of research on semantic Web and linked data. **Revue des Sciences et Technologies de l'Information-Série ISI: Ingénierie des Systèmes d'information**, 2018. DOI: <https://dx.doi.org/10.3166/ISI.23.3-4.11-56>
- HAMOUD, A. K.; HADI, M. A.; ABOUD, A. M.; ABDULLAH, A. N.; ABDULLATIF, A. K. Decision Tree Analysis Approaches to Classify Sensors Data in a Water Pumping Station. **International Journal of Computing and Digital Systems**, v. 14, n. 1, p. 555-563, 2024. DOI: <http://dx.doi.org/10.12785/ijcds/160142>
- HILL, D.; DUNHAM, C.; LARSEN, D. A.; COLLINS, M. Operationalizing an open-source dashboard for communicating results of wastewater-based surveillance. **MethodsX**, v. 11, p. 102299, 2023. DOI: <https://doi.org/10.1016/j.mex.2023.102299>
- HOFFENSON, S.; PHILIPPE, C.; CHEN, Z.; BARRIENTOS, C.; YU, Z.; CHELL, B.; BLACKBURN, M. Graphical features of interactive dashboards have little influence on engineering students performing a design task. **International Journal of Human-Computer Studies**, v. 180, p. 103121, 2023. DOI: <https://doi.org/10.1016/j.ijhcs.2023.103121>

KABAKCHIEVA, D. Business intelligence systems for analyzing university students data. **Cybernetics and Information Technologies**, v. 15, n. 1, p. 104-115, 2015. DOI: <https://doi.org/10.1515/cait-2015-0009>

KALIISA, R.; JIVET, I.; PRINSLOO, P. A checklist to guide the planning, designing, implementation, and evaluation of learning analytics dashboards. **International Journal of Educational Technology in Higher Education**, v. 20, n. 1, p. 28, 2023. DOI: <https://doi.org/10.1186/s41239-023-00394-6>

LEMES, T. de C.; DIAS, M. O. de S.; OLIVEIRA, T. de. Análise do uso de dashboard como ferramenta de apoio a tomada de decisão em instituições de ensino: uma Revisão Sistemática da Literatura. **Revista Novas Tecnologias na Educação**, Porto Alegre, v. 21, n. 1, p. 281–290, 2023. DOI: <https://doi.org/10.22456/1679-1916.134356>.

MACHIRAJU, S.; MODI, R.; MACHIRAJU, S.; MODI, R. Natural language processing. **Developing Bots with Microsoft Bots Framework: Create Intelligent Bots using MS Bot Framework and Azure Cognitive Services**, p. 203-232, 2018. DOI: [https://doi.org/10.1007/978-1-4842-3312-2\\_9](https://doi.org/10.1007/978-1-4842-3312-2_9)

MARTÍNEZ-COMESANÁ, M.; RIGUEIRA-DÍAZ, X.; LARRAÑAGA-JANEIRO, A.; MARTÍNEZ-TORRES, J.; OCARRANZA-PRADO, I.; KREIBEL, D. Impact of artificial intelligence on assessment methods in primary and secondary education: Systematic literature review. **Revista de Psicodidáctica (English ed.)**, v. 28, n. 2, p. 93-103, 2023. DOI: <https://doi.org/10.1016/j.psicoe.2023.06.002>

MONTANA, M. M. Análise do processo de controle patrimonial de livros nas bibliotecas da Universidade Federal de Santa Maria. **Dissertação (Mestrado Profissional em Gestão de Organizações Públicas)** - Universidade Federal de Santa Maria (UFSM), Santa Maria, RS, 2020, 143p.

ÖZGÜR-ÜNLÜAKIN, D.; TÜRKALI, B.; AKSEZER, S. Ç. Cost-effective fault diagnosis of a multi-component dynamic system under corrective maintenance. **Applied Soft Computing**, v. 102, p. 107092, 2021. DOI: <https://doi.org/10.1016/j.asoc.2021.107092>

PATRIARCA, R.; DI GRAVIO, G.; CIOPONEA, R.; LICU, A. Democratizing business intelligence and machine learning for air traffic management safety. **Safety science**, v. 146, p. 105530, 2022. DOI: <https://doi.org/10.1016/j.ssci.2021.105530>

SEGOVIA RAMIREZ, I.; GARCIA MARQUEZ, F. P. Fault detection and identification for maintenance management. In: **Proceedings of the Fourteenth International Conference on Management Science and Engineering Management**: Volume 1. Springer International Publishing, 2020. p. 460-469. DOI: [10.1007/978-3-030-49829-0\\_34](https://doi.org/10.1007/978-3-030-49829-0_34)

SCHULZE, A.; BRAND, F.; GEPPERT, J.; BÖL, G. F. Digital dashboards visualizing public health data: a systematic review. **Frontiers in Public Health**, v. 11, p. 999958, 2023. DOI: <https://doi.org/10.3389/fpubh.2023.999958>

TSEREMOGLOU, I.; SANTOS, B. F. Condition-Based Maintenance scheduling of an aircraft fleet under partial observability: A Deep Reinforcement Learning approach.

**Reliability Engineering & System Safety**, p. 109582, 2023. DOI:  
<https://doi.org/10.1016/j.ress.2023.109582>

TRIPATHI, M. A.; MADHAVI, K., KANDI, V. P.; NASSA, V. K., MALLIK, B.; CHAKRAVARTHI, M. K. Machine learning models for evaluating the benefits of business intelligence systems. **The Journal of High Technology Management Research**, v. 34, n. 2, p. 100470, 2023. DOI: <https://doi.org/10.1016/j.hitech.2023.100470>

YEPEZ, P.; ALSAYYED, B.; AHMAD, R. Intelligent assisted maintenance plan generation for corrective maintenance. **Manufacturing Letters**, v. 21, p. 7-11, 2019.