

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA  
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

LEONARDO CABRAL ADONIS  
SULIVAN MATHEUS SANTOS ROCHA

**MÉTODOS MULTICRITÉRIO E MECANISMOS DE APRENDIZAGEM PARA  
SUPORTE À TOMADA DE DECISÃO EM MANUTENÇÃO INDUSTRIAL**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA  
2019

LEONARDO CABRAL ADONIS  
SULIVAN MATHEUS SANTOS ROCHA

**MÉTODOS MULTICRITÉRIO E MECANISMOS DE APRENDIZAGEM PARA  
SUPORTE À TOMADA DE DECISÃO EM MANUTENÇÃO INDUSTRIAL**

Trabalho de conclusão de curso de Graduação em Engenharia de Controle e Automação apresentado à disciplina de Trabalho de conclusão de curso 2, do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR), como requisito para obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Eduardo de Freitas  
Rocha Loures

CURITIBA

2019

Leonardo Cabral Adonis  
Sulivan Matheus Santos Rocha

# MÉTODOS MULTICRITÉRIO E MECANISMOS DE APRENDIZAGEM PARA SUPORTE À TOMADA DE DECISÃO EM MANUTENÇÃO INDUSTRIAL

Este Trabalho de Conclusão de Curso de Graduação foi julgado e aprovado como requisito parcial para a obtenção do Título de Engenheiro de Controle e Automação, do curso de Engenharia de Controle e Automação do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR).

Curitiba, 21 de novembro de 2019.

---

Prof. Paulo Sérgio Walenia, Esp.  
Coordenador de Curso  
Engenharia de Controle e Automação

---

Prof. Marcelo de Oliveira Rosa, Dr.  
Responsável pelos Trabalhos de Conclusão de Curso  
de Engenharia de Controle e Automação do DAELT

## ORIENTAÇÃO

---

Eduardo de Freitas Rocha Loures, Dr.  
Universidade Tecnológica Federal do Paraná  
Orientador

## BANCA EXAMINADORA

---

Eduardo de Freitas Rocha Loures, Dr.  
Universidade Tecnológica Federal do Paraná

---

Daniel Balieiro Silva, Me.  
Universidade Tecnológica Federal do Paraná

---

Jorge Assade Leludak, Dr.  
Universidade Tecnológica Federal do Paraná

A folha de aprovação assinada encontra-se na Coordenação do Curso de Engenharia de Controle e Automação.

## **AGRADECIMENTOS**

Por Leonardo

Primeiramente, agradeço a minha família – em especial a meus pais – por todo o apoio e incentivo que me deram durante toda a minha formação acadêmica. Agradeço também aos colegas que sempre me apoiaram, aconselharam e motivaram durante todas as etapas.

Por Sullivan

Agradeço aos meus pais, Sérgio e Eugênia, pela vida e por tudo que proporcionaram para que eu chegasse até aqui. Ao meu irmão, Sérgio Lucas, pelos constantes ensinamentos e por sempre se fazer presente; e à minha namorada, Valéria, por todo o companheirismo, carinho, dedicação, paciência e por me apoiar e incentivar incondicionalmente em todos os momentos.

Por ambos

Agradecemos ao nosso orientador Prof. Dr. Eduardo de Freitas Rocha Loures, por todos os ensinamentos e pela dedicação em repassar o seu conhecimento de uma maneira exemplar. Ao mestrando Ewerton Gusthavo Gorski, pelo interesse, disposição e suporte, sendo essencial para o desenvolvimento desse trabalho. Ao grupo de pesquisa IAAS (Integração Avaliação e Automação de Sistemas) do professor orientador, que forneceu diretrizes para estruturação e definição do escopo do trabalho. A todos os nossos amigos, que fizeram parte da nossa formação e nos apoiaram ao longo desses anos. A todos os professores que participaram de forma ativa em nossas vidas acadêmicas.

A todos que, direta ou indiretamente, colaboraram com o desenvolvimento deste trabalho, o nosso muito obrigado.

*“A sorte não existe. Aquilo a que chamamos sorte  
é o cuidado com os pormenores.”*

- Winston Churchill

## RESUMO

ADONIS, Leonardo C.; ROCHA, Sullivan M. S. **Métodos multicritério e mecanismos de aprendizagem para suporte à tomada de decisão em manutenção industrial.** 2019. 82 f. Trabalho de Conclusão de Curso (Graduação – Curso de Engenharia de Controle e Automação). Universidade Tecnológica Federal do Paraná, Curitiba, 2019.

O cenário industrial atual passa por transformações que impactam em todo o contexto econômico mundial, existindo uma crescente necessidade de melhor aplicação de recursos, redução de desperdícios e, principalmente, dinamismo no desenvolvimento de atividades devido ao crescimento na concorrência. A manutenção industrial é um setor que desempenha papel essencial para a operação dos parques industriais, sendo importante possuir um sistema que auxilie o processo de tomada de decisão em relação à esta área da indústria. Com o avanço tecnológico, existe a possibilidade de realizar a implementação de aplicações através de plataformas em *cloud computing*, desenvolvidas de acordo com as necessidades e especificidades de cada cenário. Neste trabalho é apresentado um sistema para priorização das ordens de manutenção, desenvolvido com a utilização de métodos multicritério de apoio à tomada de decisão e algoritmos de *machine learning*. Este sistema possui o objetivo de aprimorar a atividade de gestão de ativos, otimizando o processo de atendimento às ordens de manutenção, realizando a priorização e destinação das mesmas conforme critérios estabelecidos em base informacional. O sistema resultante possui abordagem em paralelo entre o método multicritério e o *machine learning*, possibilitando avaliar o desempenho de ambos e identificar características de sua aplicação. Além disso, a implementação em plataforma comercial buscou explorar a aplicabilidade das atuais plataformas de tecnologia na indústria.

**Palavras-chave:** Manutenção industrial. Tomada de decisão. Método multicritério. Machine Learning. Indústria 4.0.

## ABSTRACT

ADONIS, Leonardo C.; ROCHA, Sullivan M. S. **Multicriteria methods and learning mechanisms to support decision making in industrial maintenance**. 2019. 82 f. Trabalho de Conclusão de Curso (Graduação – Curso de Engenharia de Controle e Automação). Universidade Tecnológica Federal do Paraná, Curitiba, 2019.

The current industrial scenario is undergoing transformations that impact the entire world economic context, and there is a growing need for better use of resources, reduction of waste and, mainly, dynamism in the development of activities due to the growth in competition. Industrial maintenance is a sector that plays an essential role for the operation of industrial parks, and it is important to have a system that assists the decision making process in relation to this area of industry. With technological advancement, there is the possibility of implementing applications through cloud computing platforms, developed according to the needs and specificities of each scenario. This work presents a system for prioritizing maintenance orders, developed using multicriteria decision support methods and machine learning algorithms. This system aims to improve the asset management activity, optimizing the process of meeting maintenance orders, prioritizing and allocating them according to criteria established on an information basis. The resulting system has a parallel approach between the multicriteria method and machine learning, allowing to evaluate the performance of both and identify characteristics of their application. In addition, commercial platform implementation aimed to explore the applicability of current technology platforms in the industry.

**Keywords:** Industrial maintenance. Decision making. Method multicriteria. Machine learning. Industry 4.0.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Cronograma de desenvolvimento do trabalho.....	18
Figura 2 – Exemplo de hierarquia de critérios/objetivos.....	27
Figura 3 – Escala Saaty.....	29
Figura 4 – Funções de preferência do método PROMETHEE.....	32
Figura 5 – Estrutura em camadas.....	35
Figura 6 – Serviços oferecidos pelo Microsoft Azure.....	36
Figura 7 – Rede de múltiplas camadas.....	40
Figura 8 – Rede recorrente.....	40
Figura 9 – Microsoft Azure Machine Learning: <i>Algorithm Cheat Sheet</i> .....	42
Figura 10 – Etapas do trabalho.....	44
Figura 11 – Estrutura da abordagem em paralelo.....	49
Figura 12 – Distribuição em forma de sino.....	50
Figura 13 – Trecho do código de pré-processamento implementado.....	55
Figura 14 – Implementação do pré-processamento dos dados do MCDM.....	55
Figura 15 - Trecho do código do MCDM implementado.....	56
Figura 16 - Trecho do código para avaliação do erro do MCDM implementado.....	58
Figura 17 – Implementação do pré-processamento dos dados do ML.....	61
Figura 18 – Implementação do ML através de dois módulos diferentes.....	62
Figura 19 – Resultados do teste para diversos cenários.....	65
Figura 20 – Resultado do algoritmo de floresta de decisões.....	66
Figura 21 – Resultado do algoritmo de redes neurais.....	66

## LISTA DE TABELAS

Tabela 1 – Exemplo da matriz “Mantenedores” .....	46
Tabela 2 – Informações selecionadas para o ML sobre os mantenedores .....	59
Tabela 3 – Informações selecionadas para o ML sobre as OMs.....	60

## LISTA DE ABREVIATURAS E SIGLAS

ABRAMAN	Associação Brasileira de Manutenção
AHP	<i>Analytic Hierarchy Process</i>
API	<i>Application Programming Interface</i>
CNI	Confederação Nacional da Indústria
CMMS	<i>Computerized Maintenance Management System</i>
COPIN	Conselho Temático Permanente de Política Industrial e Desenvolvimento Tecnológico
EAM	<i>Enterprise Asset Management</i>
FIEP	Federação das Indústrias do Estado do Paraná
IAAS	Integração, Avaliação e Automação de Sistemas
IaaS	<i>Infrastructure as a Service</i>
IoT	<i>Internet of Things</i>
IIoT	<i>Industrial Internet of Things</i>
MCDM/A	<i>Multiple-Criteria Decision-Making/Analysis</i>
ML	<i>Machine Learning</i>
PaaS	<i>Platform as a Service</i>
PROMETHEE	<i>Preference Ranking Organization Method for Enrichment Evaluation</i>
RNA	Redes Neurais Artificiais
SaaS	<i>Software as a Service</i>
TOPSIS	<i>Technique for Order of Preference by Similarity to Ideal Solution</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>13</b>
1.1	TEMA.....	14
<b>1.1.1</b>	<b>Delimitação do Tema</b> .....	<b>14</b>
1.2	PROBLEMAS E PREMISSAS .....	14
1.3	OBJETIVOS .....	16
<b>1.3.1</b>	<b>Objetivo Geral</b> .....	<b>16</b>
<b>1.3.2</b>	<b>Objetivos Específicos</b> .....	<b>16</b>
1.4	JUSTIFICATIVA.....	16
1.5	PROCEDIMENTOS METODOLÓGICOS .....	17
1.6	ESTRUTURA DO TRABALHO .....	17
1.7	CRONOGRAMA .....	18
<b>2</b>	<b>REFERENCIAL TEÓRICO</b> .....	<b>19</b>
2.1	MANUTENÇÃO INDUSTRIAL .....	19
<b>2.1.1</b>	<b>Modelos de Manutenção</b> .....	<b>22</b>
2.1.1.1	Manutenção Corretiva .....	22
2.1.1.2	Manutenção Preventiva .....	22
2.1.1.3	Manutenção Preditiva .....	23
<b>2.1.2</b>	<b>Gerenciamento de Ativos</b> .....	<b>24</b>
2.2	TOMADA DE DECISÃO .....	25
<b>2.2.1</b>	<b>Apoio Multicritério à Tomada de Decisão</b> .....	<b>26</b>
2.2.1.1	AHP .....	27
2.2.1.2	TOPSIS .....	29
2.2.1.3	PROMETHEE .....	31
<b>2.2.2</b>	<b>Tomada de Decisão em Manutenção Industrial</b> .....	<b>33</b>
2.3	COMPUTAÇÃO EM NUVEM.....	34
<b>2.3.1</b>	<b>Modelos de Computação em Nuvem</b> .....	<b>35</b>
<b>2.3.2</b>	<b>Azure</b> .....	<b>36</b>
<b>2.3.3</b>	<b>Python</b> .....	<b>37</b>
2.4	MACHINE LEARNING .....	37
<b>2.4.1</b>	<b>Aprendizado</b> .....	<b>38</b>
<b>2.4.2</b>	<b>Métodos de <i>Machine Learning</i></b> .....	<b>39</b>
2.4.2.1	Redes Neurais Artificiais.....	39

2.4.2.2	Classificadores Naive Bayes .....	41
2.4.2.3	Árvores de Decisões.....	41
<b>2.4.3</b>	<b>Recursos da Plataforma Azure.....</b>	<b>41</b>
<b>3</b>	<b>ABORDAGEM METODOLÓGICA .....</b>	<b>43</b>
3.1	CONTEXTUALIZAÇÃO DO PROBLEMA.....	43
3.2	ESTRUTURAÇÃO .....	43
<b>3.2.1</b>	<b>Revisão Bibliográfica .....</b>	<b>44</b>
<b>3.2.2</b>	<b>Análise de Dados e Definição dos Critérios.....</b>	<b>44</b>
<b>3.2.3</b>	<b>Definição dos Modelos MCDM.....</b>	<b>46</b>
3.2.3.1	AHP .....	46
3.2.3.2	TOPSIS .....	47
3.2.3.3	PROMETHEE .....	47
<b>3.2.4</b>	<b>Implementação em Plataforma <i>Cloud</i>.....</b>	<b>47</b>
3.2.4.1	Abordagem Paralelo entre MCDM e ML.....	48
3.2.4.2	Condição para Aplicação do ML.....	49
<b>4</b>	<b>APLICAÇÃO .....</b>	<b>52</b>
4.1	MCDM .....	52
<b>4.1.1</b>	<b>Pré-processamento dos Dados para Aplicação do MCDM .....</b>	<b>53</b>
<b>4.1.2</b>	<b>Implementação do MCDM .....</b>	<b>56</b>
<b>4.1.3</b>	<b>Avaliação do Erro do MCDM.....</b>	<b>57</b>
4.2	ML.....	58
<b>4.2.1</b>	<b>Pré-processamento dos Dados para Aplicação no ML .....</b>	<b>59</b>
<b>4.2.2</b>	<b>Implementação do ML .....</b>	<b>61</b>
<b>4.2.3</b>	<b>Avaliação da Acurácia do ML .....</b>	<b>63</b>
<b>5</b>	<b>ANÁLISE DOS RESULTADOS.....</b>	<b>64</b>
5.1	RESULTADOS DO MCDM.....	64
5.2	RESULTADOS DO ML.....	65
<b>6</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>67</b>
6.1	CONCLUSÕES.....	67
6.2	DIFICULDADES E DESAFIOS FUTUROS.....	68
	<b>REFERÊNCIAS .....</b>	<b>69</b>
	<b>APÊNDICES.....</b>	<b>76</b>
A.	ALGORITMO EM PYTHON PARA PRÉ-PROCESSAMENTO DOS DADOS NA IMPLEMENTAÇÃO DO MCDM COM MÉTODO TOPSIS .....	76

B.	ALGORITMO EM PYTHON PARA IMPLEMENTAÇÃO DO MCDM COM MÉTODO TOPSIS.....	77
C.	ALGORITMO EM PYTHON PARA AVALIAÇÃO DO ERRO DO MCDM COM MÉTODO TOPSIS.....	78
D.	ALGORITMO EM PYTHON PARA IMPLEMENTAÇÃO DO MODELO MCDM COM MÉTODO TOPSIS PARA CENÁRIOS DIVERSOS COM SELEÇÃO ALEATÓRIA DA OM.....	79

## 1 INTRODUÇÃO

A indústria 4.0, ou quarta revolução industrial, marca um período de evolução tecnológica onde a realidade de todas as empresas é alterada profundamente. A evolução da tecnologia permitiu a viabilização do uso de novas técnicas e a criação de novos modelos de negócios e operações, com a aplicação de sistemas de produção inteligentes. No Brasil, a transformação digital das indústrias de manufatura ainda é definida principalmente pela integração de equipamentos nos setores de produção e controle, mas ainda não há um número expressivo de empresas que estão totalmente preparadas para implementar esse conceito. Incentivos governamentais aliados à experiência das organizações pioneiras desse novo cenário resultarão em estímulo para as outras empresas, com o risco iminente de um novo ambiente competitivo (LIAO et al. 2017, p. 14).

Com o atual cenário industrial tornando-se cada vez mais competitivo, é notável a crescente necessidade de otimização e eficácia em todos os processos realizados pelas empresas, reduzindo os riscos e mantendo a competitividade no mercado. Por isso, destaca-se a relevância da busca contínua pela qualidade total no processo produtivo e a inibição de falhas, que acarretam perdas com desperdício de capacidade produtiva e de material. Os processos produtivos e a manutenção industrial apresentam grande variedade de indicadores e uma enorme quantidade de dados, além de inúmeros critérios essenciais à análise para as tomadas de decisão. Logo, torna-se imprescindível a capacitação do gestor responsável para realização de avaliação fundamentada com precisão de dados quantitativos e qualitativos para todos os procedimentos na esfera de tomada de decisão, desde a operação até a manutenção (LIAO et al. 2017, p. 15).

Segundo Mazzaferro (2018), a utilização adequada do conhecimento pode fazer a diferença entre uma revolução tecnológica bem-sucedida e a enorme capacidade de multiplicação de um erro com auxílio de tecnologia avançada. Assim, visando aprimorar a condução dos processos em manutenção industrial, auxiliando as dimensões de tomada de decisão com base em análise de grande quantidade de dados provenientes dos indicadores de produção e manutenção, vislumbra-se a utilização integrada de métodos multicritério e *machine learning*. Tal utilização será orientada à luz de requisitos de aplicação de plataforma industrial de gerenciamento de manutenção e ativos.

## 1.1 TEMA

O tema abordado neste trabalho é a análise e tomada de decisão na manutenção industrial e gerenciamento de ativos, visto que a criação de modelos para auxiliar, padronizar e sofisticar o processo de tomada de decisão desempenha papel fundamental para a evolução da gestão da manutenção na indústria através da perspectiva de sua aplicação em plataformas de gerenciamento de ativos.

### 1.1.1 Delimitação do Tema

Como previamente introduzido, o estudo de métodos de análises para apoio à tomada de decisão em ambiente industrial é de extrema importância para o planejamento, operação e gestão da produção e manutenção. Considerando a extensão da área de estudo da aplicação de tais métodos, referenciados em inglês *Multicriteria Decision Making* - MCDM, este trabalho teve como foco o estudo da associação destes à mecanismos de *Machine Learning* – ML para aprimoramento dos processos decisórios em manutenção industrial e gerenciamento de ativos.

Um case industrial caracterizou o universo de aplicação através de parceria com mestrando inserido em empresa de gerenciamento de ativos com grande participação no parque industrial nacional e internacional. O estabelecimento desta cooperação forneceu diretrizes para caracterização do escopo à tratativa de requisitos de análise e tomada de decisão em plataforma de gerenciamento de ativos<sup>1</sup>. Uma premissa inicial foi a investigação e utilização da plataforma Microsoft Azure<sup>2</sup> integrando seus resultados à métodos e plataformas MCDM e ML.

## 1.2 PROBLEMAS E PREMISAS

Com a implementação das tecnologias ligadas a Quarta Revolução Industrial, a flexibilização da produção, o aumento da eficiência e a segurança se encontram entre os principais atributos buscados nos sistemas industriais. Os equipamentos modernos, além de permitirem a customização dos bens produzidos, não atendem apenas a produção em massa, mas também geram um grande volume de informações que podem agora ser coletados do chão de fábrica (Confederação Nacional da Indústria, 2016).

---

<sup>1</sup> Manusis 4.0 (manusis4.com)

<sup>2</sup> Microsoft Azure (azure.microsoft.com)

O aumento da quantidade de informações disponíveis, combinado a ferramentas e métodos computacionais adequados, permitem a implementação atual de algoritmos de análise e aprendizagem com mais facilidade e aderência aos problemas industriais. A análise de dados é considerada um dos pilares dos avanços recentes indústria. Projetos de melhorias voltadas a análise avançada de dados, quando implementados de forma adequada, podem levar a aumento de 20% a 25% na capacidade de produção de uma planta industrial (MCKINSEY, 2015).

Entretanto, os algoritmos e plataformas computacionais *analytics* e ML não atendem de forma adequada a etapa final de decisão pertencente ao fluxo informacional preconizado pela I4.0 (aquisição, transmissão, armazenagem, análise e decisão). Os esforços atuais concentram-se nas etapas de aquisição (dispositivos *smart box*), transmissão (IoT, IIoT), armazenagem (*cloud*) e análise (*analytics/ML*), com pouco enfoque na decisão, oportunizando iniciativas de pesquisa e desenvolvimento. A integração de mecanismos de aprendizagem a métodos MCDM atende esta lacuna na tratativa destas últimas etapas – análise e decisão.

Na área de manutenção, as expectativas a respeito do desempenho das equipes que atuam neste setor têm aumentado substancialmente. Enquanto no passado, a função central da manutenção era a prevenção de falhas, agindo em geral de forma reativa, a manutenção na Indústria 4.0 tem por objetivo a eliminação completa das falhas. Isto passa a exigir uma postura proativa por parte das equipes de manutenção (DUNN, 2016) e uma capacidade de tomada de decisões de forma rápida e assertiva.

Os métodos MCDM oferecem diferentes técnicas para análise de situações em que duas ou mais variáveis influenciam no resultado e os objetivos do tomador da decisão apresentam certo grau de conflito entre si (SANTOS, 2015).

Neste trabalho, dentre os diferentes MCDM existentes, serão estudados os métodos AHP (*Analytic Hierarchy Process*), TOPSIS (*Technique for Order of Preference by Similarity to Ideal Solution*) e PROMETHEE (*Preference Ranking Organization Method for Enrichment Evaluation*) associados a mecanismos de *Machine Learning* através da plataforma Azure Machine Learning. Tal associação orienta a caracterização de critérios quali-quantitativos usados pelos métodos, assim como seus pesos.

## 1.3 OBJETIVOS

### 1.3.1 Objetivo Geral

Este trabalho tem por objetivo a investigação e utilização de métodos MCDM associados a mecanismos *Machine Learning* para análise informacional e suporte à tomada de decisão em empresa de gerenciamento de ativos.

### 1.3.2 Objetivos Específicos

- Investigar métodos de avaliação multicritério em manutenção industrial;
- Investigar mecanismos de análise de dados e aprendizagem da plataforma Azure Machine Learning;
- Avaliar as atividades de manutenção e gerenciamento de ativos juntamente com a base informacional da empresa escolhida;
- Selecionar os métodos multicritério que mais se adequam às mais críticas dimensões de decisão;
- Modelar os métodos MCDM escolhidos considerando a associação de mecanismos de *Machine Learning*.
- Implementação e análise dos resultados obtidos com a empresa.

## 1.4 JUSTIFICATIVA

Para a competitividade no cenário global, a indústria brasileira terá de adotar as tecnologias criadas pela Quarta Revolução Industrial. Iniciativas como a agenda de propostas elaborada pela CNI, para o COPIN, buscam acelerar a implementação destas tecnologias (Confederação Nacional da Indústria, 2016).

Este trabalho será aplicado em uma empresa de gerenciamento de ativos, visando empregar os conceitos apresentados neste trabalho em um cenário real, demonstrando, assim, os benefícios que podem ser obtidos a partir de uma melhor tomada de decisão em manutenção com reflexos na operação produtiva.

Há uma grande demanda do parque industrial brasileiro para que haja a evolução de sua planta instalada no atendimento de requisitos da indústria 4.0, uma vez que o entendimento do comportamento dos ativos e suporte às diferentes dimensões de decisão em manutenção representam uma lacuna industrial. A maior quantidade de informação de chão de fábrica e sistemas de gestão propiciada por

facilitadores tecnológicos, implica na necessidade de um consumo mais eficiente de toda a informação gerada. Como reflexo, as plantas e sistemas de operação tornam-se mais responsivos elevando a capacidade de produção com aumento na disponibilidade de seus ativos.

## 1.5 PROCEDIMENTOS METODOLÓGICOS

Para atingir os objetivos propostos, foi realizada revisão literária sobre o tema em artigos, livros, teses, dissertações e demais publicações com foco em tomada de decisão em sistemas de produção utilizando multicritério, expondo os principais métodos de análise existentes e também pesquisando sobre os mecanismos de aprendizagem da plataforma utilizada. Os trabalhos realizados pelo grupo de pesquisa IAAS (Integração Avaliação e Automação de Sistemas) do professor orientador forneceram diretrizes para definição do escopo (métodos e abordagem).

Na sequência, será realizado estudo sobre as atividades de manutenção e gerenciamento de ativos, coleta de dados da base informacional de uma empresa de gerenciamento de ativos, obtida a partir do trabalho do mestrando Ewerton Gusthavo Gorki, com enfoque em OMs e critérios preferenciais para definição da gestão dos ativos. Posteriormente serão indicados os métodos multicritério adequados para cada cenário industrial e modelados através de inteligência de máquina com o auxílio de plataforma.

Por fim, serão implementados os modelos, com a aplicação dos dados disponíveis, bem como será realizada análise e comparação dos resultados obtidos.

## 1.6 ESTRUTURA DO TRABALHO

O presente trabalho foi dividido em seis capítulos, objetivando maior clareza nas explicações e análises propostas sobre o tema. O primeiro capítulo apresenta a introdução ao conceito de “tomada de decisão”, os objetivos e justificativas para a realização do trabalho, além dos procedimentos metodológicos utilizados na pesquisa e a forma como o trabalho foi estruturado. O segundo capítulo indica o referencial teórico e a revisão de literatura sobre manutenção industrial e gerenciamento de ativos, os métodos multicritério no apoio à tomada de decisão, a computação em nuvem e os diferentes mecanismos de *machine learning*. No terceiro capítulo encontra-se a abordagem metodológica utilizada para o desenvolvimento do trabalho,

a contextualização do problema, a análise dos dados coletados da base informacional que permitiram o desenvolvimento do presente trabalho, além da explicação sobre a importância da utilização dos indicadores no processo de tomada de decisão e a descrição da implementação dos métodos escolhidos. O quarto capítulo apresenta os detalhes sobre a aplicação dos métodos de tomada de decisão através dos modelos MCDM e dos algoritmos de ML. O quinto capítulo apresenta a comparação dos resultados obtidos através dos diferentes métodos utilizados e do sistema desenvolvido através das duas abordagens em uma única aplicação. Por fim, o sexto capítulo apresenta as considerações finais sobre os resultados obtidos e sobre o trabalho desenvolvido.

## 1.7 CRONOGRAMA

O desenvolvimento deste trabalho foi realizado conforme o cronograma apresentado na Figura 1.

Figura 1 – Cronograma de desenvolvimento do trabalho

Atividades	Metodologia Aplicada ao TCC 2018.2					TCC 1 2019.1						TCC 2 2019.2					
	08	09	10	11	12	01	02	03	04	05	06	07	08	09	10	11	12
Reuniões	■	■	■	■	■	■	■	■	■	■	■	■	■	■			
Elaboração da Proposta		■	■														
Revisão Bibliográfica		■	■	■	■	■	■	■	■	■	■	■	■	■			
Estudo dos métodos de análise e mecanismos de aprendizagem				■	■	■	■										
Coleta de Dados							■	■									
Definição e modelagem dos métodos de análise									■	■	■	■					
Comparação dos resultados													■	■			
Conclusão e apresentação dos resultados															■	■	

Fonte: Própria (2018).

## 2 REFERENCIAL TEÓRICO

Para a produção do presente trabalho, foi necessário realizar uma pesquisa aprofundada sobre áreas e tópicos de relevância para o tema. Foram abordados os assuntos que justificam a necessidade da realização do trabalho e a sua importância para o cenário industrial; e na sequência, explicou-se de que forma esses assuntos podem ser levados para o campo de atuação industrial e aplicados de forma otimizada através de métodos desenvolvidos na área da tecnologia.

Primeiramente foram realizadas pesquisas com foco nos conceitos e teorias sobre manutenção industrial e gerenciamento de ativos, fundamentando o desenvolvimento do trabalho através de conhecimento científico e justificando os pontos focais para uma aplicação prática do trabalho desenvolvido. Na sequência foi realizada pesquisa sobre as bases teóricas da tomada de decisão e a utilização dos MCDM como ferramentas de apoio à tomada de decisão, além da aplicação do conceito de tomada de decisão em manutenção industrial.

Feita a análise sobre a fundamentação teórica que justifica a aplicação do trabalho, deu-se início a pesquisa sobre como esses conceitos podem evoluir para a aplicação prática, utilizando tecnologias e ferramentas atuais, onde foram estudados métodos de computação em nuvem e a aplicabilidade de *machine learning* como meio inovador para a resolução de problemas e melhoria de processos.

### 2.1 MANUTENÇÃO INDUSTRIAL

A crescente competitividade econômica em escala global gera um ambiente industrial em que a inovação, tecnologia, qualidade, atualização nos modelos de negócios e os diferenciais competitivos são essenciais para as indústrias continuarem presentes no mercado. Segundo a Confederação Nacional da Indústria (2018), desde 2013 o Brasil vem sucessivamente perdendo posições de competitividade no ranking do Fórum Econômico Mundial e chegou à pior posição dos últimos dez anos em 2017, comprometendo o crescimento econômico e o desenvolvimento do país. A produtividade nacional brasileira, que possui influência direta na competitividade, teve a pior evolução entre seus dez principais parceiros comerciais, com isso o país perdeu competitividade no mercado internacional e também no mercado nacional, onde é ameaçado pelas importações.

A manutenção industrial tem impacto direto na produtividade e em vários fatores importantes para a indústria, podendo ser o elemento decisivo para o sucesso ou fracasso de um modelo de negócio industrial. Segundo Slack (2006, p. 491), “manutenção é o termo usado para abordar a forma pela qual as organizações tentam evitar as falhas cuidando de suas instalações físicas. ”

Para Xenos (1998), as atividades de manutenção devem possuir uma finalidade muito mais amplas do que simplesmente manter as condições de disponibilidade dos equipamentos, pois também deve fazer parte dos trabalhos dos departamentos de manutenção a introdução de melhorias visando o aumento da produtividade. Com isso, as atividades de manutenção em qualquer empresa podem ser divididas entre atividades de manutenção e atividades de melhoria.

De acordo com Motter (1992), a manutenção esteve presente no cotidiano social desde as épocas mais remotas, se desenvolveu de forma consistente durante a Revolução Industrial e se consolidou como uma necessidade absoluta durante a Segunda Guerra Mundial. Na reconstrução pós-guerra, a manutenção auxiliou os investimentos realizados nas bases da engenharia para o desenvolvimento industrial, econômico e social.

Segundo Moubray (1997), entre a década de 1930 e a década de 1990, a manutenção pode ser dividida em três gerações:

- A primeira geração abrange o período que antecede a Segunda Guerra Mundial, onde as indústrias não eram muito avançadas e os maquinários eram de fácil reparo. Além de que o *downtime*<sup>3</sup> não representava grande ameaça pois a concorrência não era grande para a maioria dos setores.
- A segunda geração abrange todo o período da Segunda Guerra Mundial e as décadas de 1950 e de 1960, onde existiu intenso crescimento na demanda por vários tipos de produtos e bens de consumo. Com isso também houve um crescimento na complexidade das máquinas e a dependência dos produtos, levando a conclusão de que a ocorrência de *downtimes* deveria ser prevenida, ocasionando a criação do conceito de manutenção preventiva. Os custos de manutenção se tornaram pontos

---

<sup>3</sup> Segundo ABRAMAN: O tempo de paralisação “*Downtime*”, significa o tempo total que o ativo permaneceu fora de serviço, em função de uma determinada falha. <http://www.abraman.org.br/Arquivos/192/192.pdf>

relevantes na administração das indústrias, dando início aos primeiros sistemas de controle e planejamento de manutenção.

- A terceira geração teve início na década de 1970 com o aumento da competitividade em escala global com a popularização dos sistemas de produção *just-in-time*. O aumento da mecanização e da automação resultou na elevada percepção sobre a importância dos conceitos de confiabilidade e disponibilidade nos mais variados setores da indústria. Além disso a crescente presença de sistemas automatizados eleva os padrões de qualidade, mas também resulta no aumento de possíveis falhas por representar um aumento no número de maquinários empregados nas linhas de produção.

Após o ano 2000, a evolução das economias modernas fez com que os objetivos estratégicos corporativos priorizassem o desempenho produtivo, dando início à quarta geração da manutenção industrial que tende a alinhar a gestão de riscos com a gestão de ativos. (PINTO, 2002).

De mal necessário à uma parte integrante dos esforços estratégicos de produtividade das empresas. Da preocupação única com a disponibilidade do equipamento à priorização da efetividade do negócio por meio do gerenciamento dos custos e à priorização da integridade das pessoas e do meio ambiente. Do simples atendimento a Produção à peça fundamental na garantia do atendimento ao cliente por meio da melhoria da confiabilidade dos equipamentos e processos. Essas são algumas formas de como se via a Manutenção e em alguns casos ainda se vê, e como se deve vê-la nos dias de hoje. (MORAES, p. 21, 2004).

A principal função da manutenção industrial é possibilitar o funcionamento ótimo dos equipamentos produtivos, permitindo que a indústria possa operar em sua capacidade produtiva total e com o menor índice possível de perdas.

Segundo Borlido (2017), destacam-se como objetivos da manutenção a redução de custos da empresa, a inibição da ocorrência de qualquer forma de indisponibilidade da linha de produção, a melhoria contínua dos padrões de qualidade da produção e o aumento da segurança.

### 2.1.1 Modelos de Manutenção

Os modelos de manutenção podem ser classificados em dois grandes grupos: (i) as *planejadas*, que consistem em manutenções previamente especificadas pela indústria e possuem o objetivo de inibir a ocorrência das (ii) *não-planejadas*, que ocorrem após uma falha da linha de produção, geralmente resultando em um alto custo para a indústria.

Segundo Pinto (2002), a gestão estratégica da manutenção se desenvolve com a evolução da manutenção, partindo da não-planejada até a consolidação da engenharia de manutenção como cultura empresarial, ao custo da alteração de processos e costumes. Para atender a operação de forma ótima, é necessário que a manutenção atue de forma que integre as áreas de engenharia, suprimentos, inspeção de equipamentos, dentre outras.

Conforme publicado pela Associação Brasileira de Manutenção (2017), no Brasil os custos de manutenção representam em média 4% do faturamento das empresas e 7% do valor imobilizado. Enquanto 38% das empresas aplicam manutenção corretiva, 41% utilizam manutenção preventiva, 14% utilizam manutenção preditiva e 7% utilizam outros modelos de manutenção.

#### 2.1.1.1 Manutenção Corretiva

É a manutenção realizada após a apresentação de uma falha com o objetivo de corrigir a situação para que o maquinário volte a estar disponível para sua função.

A manutenção corretiva sempre é feita depois que a falha ocorreu. Em princípio, a opção por este método de manutenção deve levar em conta fatores econômicos: é mais barato consertar uma falha do que tomar ações preventivas? Se for, a manutenção corretiva é uma boa opção. Logicamente, não podemos nos esquecer de levar em conta também as perdas por paradas na produção, pois a manutenção corretiva pode acabar saindo muito mais cara do que imaginávamos em princípio. (XENOS, p. 23, 1998).

#### 2.1.1.2 Manutenção Preventiva

É a manutenção realizada com determinada periodicidade para reduzir a incidência de falhas e otimizar o desempenho do equipamento.

(...) envolve algumas tarefas sistemáticas, tais como as inspeções, reformas e trocas de peças, principalmente. Uma vez estabelecida, a manutenção

preventiva deve ter caráter obrigatório. Se comparada com a manutenção corretiva – somente do ponto de vista de custo de manutenção – a manutenção preventiva é mais cara pois as peças têm que ser trocadas e os componentes têm que ser reformados antes de atingirem seus limites de vida. (XENOS, p. 24, 1998).

### 2.1.1.3 Manutenção Preditiva

Ao contrário das clássicas manutenções corretiva e preventiva, ainda amplamente utilizadas por grande parte das empresas no Brasil, a manutenção preditiva melhora a performance dos equipamentos e fortalece os modelos de negócios das indústrias. Em consequência da inclusão de um conjunto de tecnologias de sensoriamento, monitoramento, análise preditiva e sistemas de distribuição, atualmente é possível executar e fornecer uma assistência técnica remota baseada em monitoramento contínuo e suporte da manutenção à distância. (FERNÁNDEZ, 2016).

A manutenção preditiva pode ser definida como um conjunto de atividades de acompanhamento e análise de variáveis que apresentam o desempenho atual dos equipamentos, com o objetivo de definir a necessidade ou não de intervenção (MACHADO; OTANI, 2008).

O primeiro indício da manutenção preditiva pode ter ocorrido quando um mecânico utilizou a própria audição para alertar que o ruído proveniente de uma máquina indicava que ela tinha grandes chances de apresentar uma falha, caso não recebesse manutenção adequada. Desde então foram criadas variadas tecnologias para analisar o que está acontecendo dentro das máquinas, mas a necessidade de uma pessoa experiente e bem informada para usar a tecnologia não mudou. (DUNN, 2002).

Segundo Pinto (2002, p. 43), a manutenção preditiva pode ser definida como “qualquer atividade de monitoramento que seja capaz de fornecer dados suficientes para uma análise de tendências, emissão de diagnóstico e tomada de decisão, no entanto, é comumente interpretada de uma forma generalista e descrita como um grupo de medições e análises. ”

O setor de manutenção deve possuir a capacidade de avaliar a “saúde” do maquinário produtivo e ter uma atitude proativa em relação à linha de produção, antecipando a resolução de problemas antes da ocorrência de falhas, desgastes ou de qualquer evento que afete a disponibilidade do maquinário. Além disso, o

conhecimento preciso sobre a “saúde” das máquinas possibilita melhorias no planejamento do departamento de manutenção e também a identificação das características que geram tais condições, melhorando o desenvolvimento de novos projetos e o ajuste adequado conforme o uso do maquinário. (FERNÁNDEZ, 2016).

Para (MACHADO; OTANI, p. 8, 2008), “(...) a estruturação da manutenção preditiva é basicamente definida em 8 etapas:

- Pesquisa das necessidades (custos);
- Seleção de componentes/equipamentos prioritários em classes ABC;
- Coleta de informações externas à empresa para obtenção de tecnologias;
- Estrutura para execução e fluxo do sistema de manutenção preditiva;
- Treinamento dos responsáveis;
- Testes;
- Seleção de *software* e *hardware*;
- Desenvolvimento gradual do sistema na empresa: seleção e ampliação dos equipamentos e partes que são objetos da manutenção preditivas, e; desenvolvimento de equipamentos e tecnologias de diagnóstico”.

### **2.1.2 Gerenciamento de Ativos**

Os ativos são bens materiais que geram valor para as indústrias, possuem desempenho dinâmico e necessitam de gerenciamento adequado para garantir a sua disponibilidade e funcionamento apropriado. Segundo Fuentes (2006), o crescimento da concorrência no mercado industrial implica na evolução da manutenção tradicional para um modelo de gestão de ativos industriais, onde questões como ciclo de vida do maquinário, relações de custo/benefício, melhorias em produtividade e retorno sobre o investimento têm prioridade durante o desenvolvimento da estratégia da organização.

Para Alves e Falsarella (2005), o gerenciamento de ativos pode ser representado como a interseção entre as áreas de inteligência organizacional e gestão da manutenção, pois ambas possuem foco na sustentação e preservação de uma posição estratégica com o objetivo de minimizar incertezas e devem atuar de maneira conjunta otimizando processos e antecipando ameaças internas e externas à continuidade operacional dos ativos.

Segundo Nascif (2012), o sistema de gestão deve ser baseado em um plano estratégico e gerenciar as diretrizes para a gestão de ativos, levando em consideração aspectos de durabilidade, custos, escalabilidade e previsibilidade.

Os processos operacionais também devem ser geridos como ativos e interpretados como bens que geram valor para a companhia. Por exemplo, a otimização dos processos de emissão, distribuição e alocação de ordens de manutenção podem representar resultados extremamente benéficos para o setor de manutenção e para toda a área de produção.

## 2.2 TOMADA DE DECISÃO

Para Saaty (2008), todos os humanos são fundamentalmente tomadores de decisão, sendo que tudo o que fazem consciente ou inconscientemente é o resultado de alguma decisão. Coletam informações com o objetivo de entender acontecimentos, desenvolvendo julgamentos para tomar decisões sobre esses próprios acontecimentos.

Supondo a existência de um conjunto de ideias e soluções para uma determinada situação, comparáveis ou não e sujeitos a restrições e limitações, em um momento onde é necessário realizar uma priorização ou escolha, é conveniente utilizar o planejamento estratégico padrão, ou seja, aplicar um conjunto de procedimentos sequenciais para encontrar o resultado mais conveniente. Porém, o fato de normalmente não se saber qual será o melhor resultado que pode ser alcançado, qual a confiabilidade das informações disponíveis em relação às questões subjetivas e qual a precisão na identificação e estimativa dos riscos potenciais da tomada de decisão eleva a dificuldade da situação. (MUNIER, 2011).

Bonissone, Subbu e Lizzi (2009) apresentam a tomada de decisão multicritério representada como a interseção de três áreas fundamentais que geralmente são tratadas e desenvolvidas independentemente.

- Solução via pesquisa: Apresenta a necessidade da pesquisa sobre as soluções viáveis, geralmente com desafios significativos conforme a complexidade da situação e a incidência de soluções não-lineares e probabilísticas.

- Solução via agregação de preferências e conflitos: Apresenta a necessidade de elencar e agregar as preferências do tomador de decisão, via combinação linear ou outro modelo.
- Visualização interativa: Apresenta a necessidade de melhoria da representação do problema, incorporando o tomador de decisão no aperfeiçoamento da solução através da compreensão dos impactos e os conflitos presentes na decisão.

Para Araya, Carignano e Gomes (2004), os problemas complexos de tomada de decisões possuem, pelo menos, uma das características a seguir:

- Existem no mínimo dois critérios resolução do problema são que conflitam entre si;
- Inexiste determinação clara sobre os critérios, alternativas e as consequências da escolha de determinada alternativa com relação a pelo menos um critério;
- Os critérios e as alternativas estão de alguma forma interligados, com implicações diferentes decorrentes de cada opção de escolha;
- Existem opiniões conflitantes entre os tomadores de decisão;
- Inexiste determinação clara sobre o que deve ser considerado como critério e o que deve ser considerado como restrição;
- Existem critérios subjetivos;
- Podem existir diferentes escalas para classificação dos critérios.

### **2.2.1 Apoio Multicritério à Tomada de Decisão**

O processo de tomada de decisão pode alcançar elevado nível de complexidade conforme as variáveis existentes e os diversos fatores influenciados pelo resultado de uma escolha. Segundo Araya, Carignano e Gomes (2004), os primeiros métodos de apoio ou auxílio multicritério à decisão surgiram na década de 1970 agregando todas as características importantes do problema, com a finalidade de possibilitar a sistematização do processo de decisão em busca de objetivos múltiplos e simultâneos.

Como ferramentas auxiliares ao processo de tomada de decisão, existem diversos métodos com capacidade de avaliar uma grande quantidade de critérios em busca da opção ótima para o contexto em que a decisão é necessária.

Para Baran (2015), a escolha do método multicritério deve ser realizada conforme as características do problema, aspectos dos critérios e da modelagem mais adequada, inclinações do tomador de decisão e domínio do analista. Após identificar teoricamente os aspectos e atributos necessários para realizar a avaliação da importância de um sistema industrial, é possível classificar quais métodos são mais adequados para aplicação no modelo sugerido.

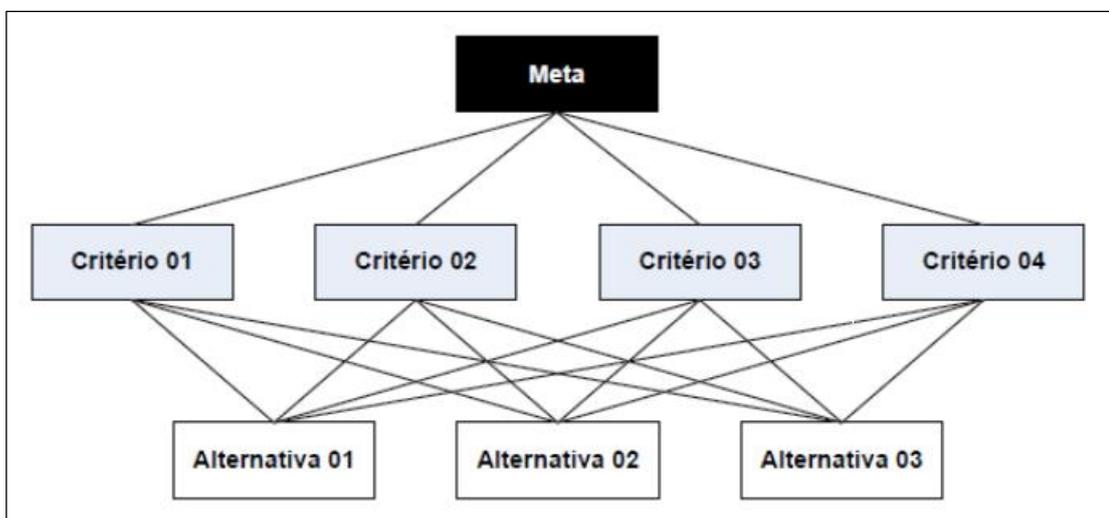
Segundo Bouyssou (1993 apud ARAYA; CARIGNANO; GOMES, 2004, p. 03), a abordagem multicritério em tomada de decisão representa as seguintes vantagens:

- A criação de um cenário com o ponto de vista de cada um dos tomadores de decisão;
- Facilidade de incorporar as incertezas nos dados conforme cada ponto de vista;
- Analisar as alternativas como soluções diretas para cada um dos objetivos.

#### 2.2.1.1 AHP

Criado pelo professor Thomas L. Saaty em 1980, o método AHP (*Analytic Hierarchy Process*) foi um dos primeiros desenvolvidos no campo das decisões multicritério discretas. Este método divide o problema de decisão em níveis hierárquicos para facilitar a compreensão e avaliação. (ARAYA; CARIGNANO; GOMES, 2004).

Figura 2 – Exemplo de hierarquia de critérios/objetivos



Fonte: Vargas (2010).

Segundo Araya, Carignano e Gomes (2004), após o problema ser dividido em níveis hierárquicos, é possível determinar através dos tomadores de decisão uma medida global para cada uma das alternativas. Depois da construção da hierarquia, cada tomador de decisão deve fazer uma comparação, par a par, de cada elemento em cada nível hierárquico, criando-se uma matriz de decisão quadrada. Nessa matriz, o tomador de decisão representará sua preferência entre as opções comparadas, a partir de uma escala predefinida, usualmente a escala Saaty.

Para Vargas (2010), o AHP transforma as comparações empíricas em valores numéricos que podem ser analisados de forma otimizada. O peso individual dos fatores considerados permite a análise independente dentro da hierarquia definida e essa capacidade de conversão de dados empíricos em modelos matemáticos é o principal diferencial do modelo AHP.

Para Vargas (2010), após a realização de todas as comparações e o estabelecimento dos pesos relativos entre os critérios, é calculada a probabilidade de cada uma das alternativas. Esse cálculo determina a probabilidade de a alternativa atender a meta estabelecida, quanto maior, mais aquela alternativa contribui para a meta final do modelo. Os cálculos envolvendo o AHP parecem simples em primeiro momento, porém em casos complexos, as análises e cálculos exigem o uso de *softwares* específicos para cálculo.

A comparação par a par do modelo AHP é comumente realizada conforme a escala proposta por Saaty (2005), onde a importância relativa entre dois critérios é avaliada atribuindo valores entre 1 e 9, conforme apresentada na Figura 3. O resultado da comparação par a par em um grupo possibilita a construção de uma matriz de comparação.

Figura 3 – Escala Saaty

Scales	Degree of preference	Explanation
1	Equally	Two activities contribute equally to the objective
3	Moderately	Experience and judgment slightly to moderately favor one activity over another
5	Strongly	Experience and judgment strongly or essentially favor one activity over another
7	Very strongly	An activity is strongly favored over another and its dominance is showed in practice
9	Extremely	The evidence of favoring one activity over another is of the highest degree possible of an affirmation
2, 4, 6, 8	Intermediate values	Used to represent compromises between the preferences in weights 1, 3, 5, 7, and 9
Reciprocals	Opposites	Used for inverse comparison

Fonte: Saaty (2005).

### 2.2.1.2 TOPSIS

O método TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) foi desenvolvido como um método multicritério para se estabelecer qual a melhor alternativa com base na maior proximidade com a alternativa ideal e a maior distância da pior alternativa. Para Duarte Junior e Costa (2013), o TOPSIS é um método que vem sendo muito utilizado para ordenar preferências e se destaca como vantagem a capacidade de aplicação para um grande número de alternativas e critérios. A primeira etapa corresponde a montagem de uma matriz de decisão composta por alternativas “m” e critérios “n”.

Equação 1 – Matriz de decisão para o método TOPSIS

$$M = \begin{bmatrix} m_{11} & \cdots & m_{1j} & \cdots & m_{1m} \\ \vdots & & \vdots & & \vdots \\ m_{i1} & \cdots & m_{ij} & \cdots & m_{im} \\ \vdots & & \vdots & & \vdots \\ m_{n1} & \cdots & m_{nj} & \cdots & m_{nm} \end{bmatrix}$$

Fonte: Adaptado de Duarte Junior e Costa (2013).

Na sequência é realizada a normalização da matriz e posteriormente a multiplicação pelos pesos dos critérios estabelecidos pelo tomador de decisão.

Equação 2 – Normalização linear

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum x_{ij}^2}}$$

Fonte: Adaptado de Duarte Junior e Costa (2013).

Equação 3 – Atribuição linear dos pesos dos critérios

$$v_{ij} = w_{ij}r_{ij}$$

Fonte: Adaptado de Duarte Junior e Costa (2013).

Após a atribuição dos pesos, é possível calcular os melhores e piores níveis para cada um dos critérios analisados e também as medidas de separação para cada alternativa em relação às soluções ideal e anti-ideal.

Equação 4 – Cálculo da solução ideal e da solução anti-ideal, respectivamente

$$S^+ = \{(\max v_{ij} | j \in J), (\min v_{ij} | j \in J')\}$$

$$S^- = \{(\min v_{ij} | j \in J), (\max v_{ij} | j \in J')\}$$

Fonte: Adaptado de Duarte Junior e Costa (2013).

Equação 5 – Cálculo da distância ideal positiva e anti-ideal, respectivamente

$$D_i^+ = \sqrt{\sum_{j=1}^n [v_{ij}(x) - v_j^+(x)]^2}$$

$$D_i^- = \sqrt{\sum_{j=1}^n [v_{ij}(x) - v_j^-(x)]^2}$$

Fonte: Adaptado de Duarte Junior e Costa (2013).

Por fim, encontra-se ao resultado da aproximação à situação ideal e a definição da hierarquização das alternativas. A solução ideal é definida pela opção mais próxima ou igual a 1, portanto a mais próxima ou igual a 0 será a solução anti-ideal. Com este método é possível ranquear as alternativas com base na priorização por critérios.

Equação 6 – Resultado da aproximação da situação ideal

$$C_i = \frac{D_i^-}{D_i^+ + D_i^-}$$

Fonte: Adaptado de Duarte Junior e Costa (2013).

### 2.2.1.3 PROMETHEE

O método PROMETHEE (*Preference Ranking Organization Method for Enrichment Evaluation*) foi criado por Jean-Pierre Brans e apresentado pela primeira vez em 1982. Em virtude das propriedades matemáticas utilizadas e da facilidade em sua utilização, a metodologia obteve sucesso ao ser aplicada nos mais variados campos de estudos, como investimentos e finanças, planejamento de recursos humanos, medicina, turismo, gestão e engenharia. (BRANS; MARESCHAL, 2005).

Segundo Brans e Vincke (1985), a metodologia utilizada pelo PROMETHEE tem como base a sobre classificação baseada em um grau de dominância de uma alternativa sobre a outra e sobre os pesos individuais de cada alternativa. A aplicação é iniciada através da criação de uma matriz de decisão igual ao do método TOPSIS, exemplificada na Equação 1Equação 7.

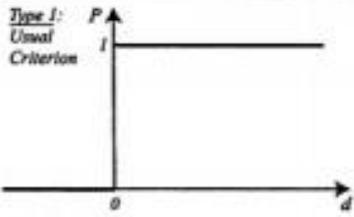
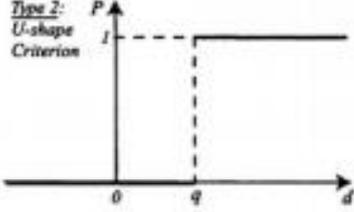
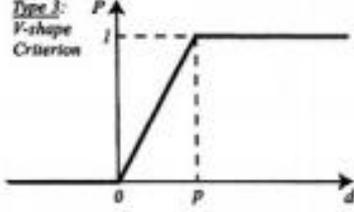
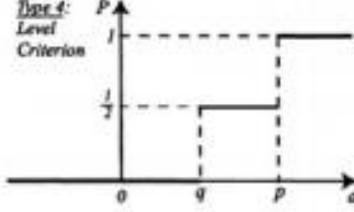
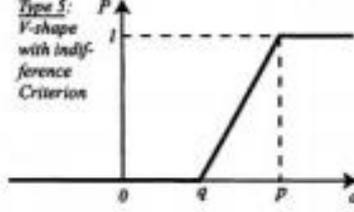
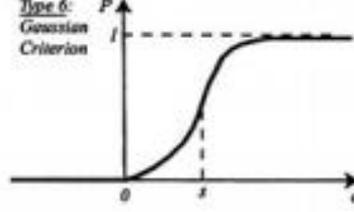
Equação 7 – Matriz de decisão para o método PROMETHEE

$$M = \begin{bmatrix} m_{11} & \dots & m_{1j} & \dots & m_{1m} \\ \vdots & & \vdots & & \vdots \\ m_{i1} & \dots & m_{ij} & \dots & m_{im} \\ \vdots & & \vdots & & \vdots \\ m_{n1} & \dots & m_{nj} & \dots & m_{nm} \end{bmatrix}$$

Fonte: Adaptado de Brans e Mareschal (2005).

Na sequência, são atribuídos pesos individuais para cada critério e definida qual das funções de preferência, apresentadas na Figura 4, do método é adequada para a aplicação. Após, são calculadas as diferenças das alternativas quando analisadas sobre cada um dos critérios e as mesmas são inseridas na função de preferência.

Figura 4 – Funções de preferência do método PROMETHEE

Generalised criterion	Definition	Parameters to fix
<p>Type 1: Usual Criterion</p> 	$P(d) = \begin{cases} 0 & d \leq 0 \\ 1 & d > 0 \end{cases}$	-
<p>Type 2: U-shape Criterion</p> 	$P(d) = \begin{cases} 0 & d \leq q \\ 1 & d > q \end{cases}$	q
<p>Type 3: V-shape Criterion</p> 	$P(d) = \begin{cases} 0 & d \leq 0 \\ \frac{d}{p} & 0 \leq d \leq p \\ 1 & d > p \end{cases}$	p
<p>Type 4: Level Criterion</p> 	$P(d) = \begin{cases} 0 & d \leq q \\ \frac{1}{2} & q < d \leq p \\ 1 & d > p \end{cases}$	p, q
<p>Type 5: V-shape with indif- ference Criterion</p> 	$P(d) = \begin{cases} 0 & d \leq q \\ \frac{d-q}{p-q} & q < d \leq p \\ 1 & d > p \end{cases}$	p, q
<p>Type 6: Gaussian Criterion</p> 	$P(d) = \begin{cases} 0 & d \leq 0 \\ 1 - e^{-\frac{d^2}{2s^2}} & d > 0 \end{cases}$	s

Fonte: Brans e Mareschal (2005).

Depois, deve ser calculado o índice de preferência de cada alternativa, assim como os fluxos positivo e negativo de preferência. Por fim, é possível calcular o fluxo de rede e obter a relação parcial entre as alternativas.

### 2.2.2 Tomada de Decisão em Manutenção Industrial

Segundo Almeida e Bohoris (1995), a tomada de decisão em manutenção industrial pode ocorrer em vários contextos com diferentes modelos de sistemas em termos de tecnologia, requisitos, confiabilidade e disponibilidade.

De acordo com Băban, Florin Băban e Şuteu (2019), a abordagem de manutenção preditiva baseada nas condições do maquinário pode ser utilizada para a programação das atividades de manutenção, pois as falhas das máquinas usualmente são precedidas por sintomas de deterioração ou algum mau funcionamento específico. A aplicação de um *framework* voltado para a manutenção preditiva inclui o desenvolvimento de três atividades principais:

- Aquisição de dados: O desenvolvimento tecnológico acelerado dos sensores industriais e a concorrência do ramo industrial tornou o processo de coleta de dados mais acessível para as indústrias, porém devem ser empregados métodos específicos de aquisição e escolha de parâmetros de acordo com as características das máquinas e dos processos envolvidos. Dentre os parâmetros mais comuns escolhidos para monitoramento, destacam-se vibração, ruído e temperatura.
- Análise de dados: Os dados coletados representam o comportamento da máquina e do processo executado, portanto apresentam características específicas do sistema industrial e necessitam do uso de algoritmos e técnicas de análise adequadas ao objetivo do monitoramento.
- Tomada de decisão: Existem diferentes métodos de abordagem disponíveis para apoio à tomada de decisão, geralmente podem ser divididos com base nas características físicas do processo ou nas características dos dados obtidos. Os métodos com base nas características dos dados são mais utilizados pois podem ser modelados com mais facilidade e tratar aspectos complexos utilizando métodos estatísticos, modelos de regressão, modelos de espaço de estados, análise probabilística, entre outros. A utilização desses métodos fornece informações importantes e auxiliam o processo de tomada de decisão.

Para Rocco (2012), a política de manutenção e os critérios de tomada de decisão em manutenção industrial devem ser adotados levando em consideração

aspectos como a relevância do maquinário para o processo produtivo, o custo do equipamento e de sua reposição, os efeitos das falhas do maquinário no processo, a cadência da produção e quaisquer outros fatores que influenciem na disponibilidade e comportamento ótimo dos ativos.

### 2.3 COMPUTAÇÃO EM NUVEM

A computação em nuvem tem por objetivo, disponibilizar o uso de recursos computacionais através da internet, diminuindo assim a necessidade de utilização de recursos computacionais por parte do usuário final. O termo é usado para se referir tanto as aplicações disponibilizadas através da internet, quanto a estrutura necessária para o provedor fornecer estas aplicações (ARMBRUST et al., 2009).

Segundo Buyya, Calheiros e Ranjan (2009), os sistemas em nuvem costumam integrar as funcionalidades de *hardware*, banco de dados, *software* de *back-end* e interface com o usuário.

A disponibilização das funcionalidades citadas anteriormente, de forma integrada de acordo com a demanda por parte do usuário, com uma cobrança proporcional as utilizações representam um grande ganho de eficiência e flexibilidade dos sistemas de computação em nuvem, em relação aos sistemas tradicionais. Desta forma, a nuvem torna o fornecimento de *software* mais próximo de um serviço, eliminando muitas das preocupações e custos com *hardware* por parte dos usuários (ARMBRUST et al., 2009).

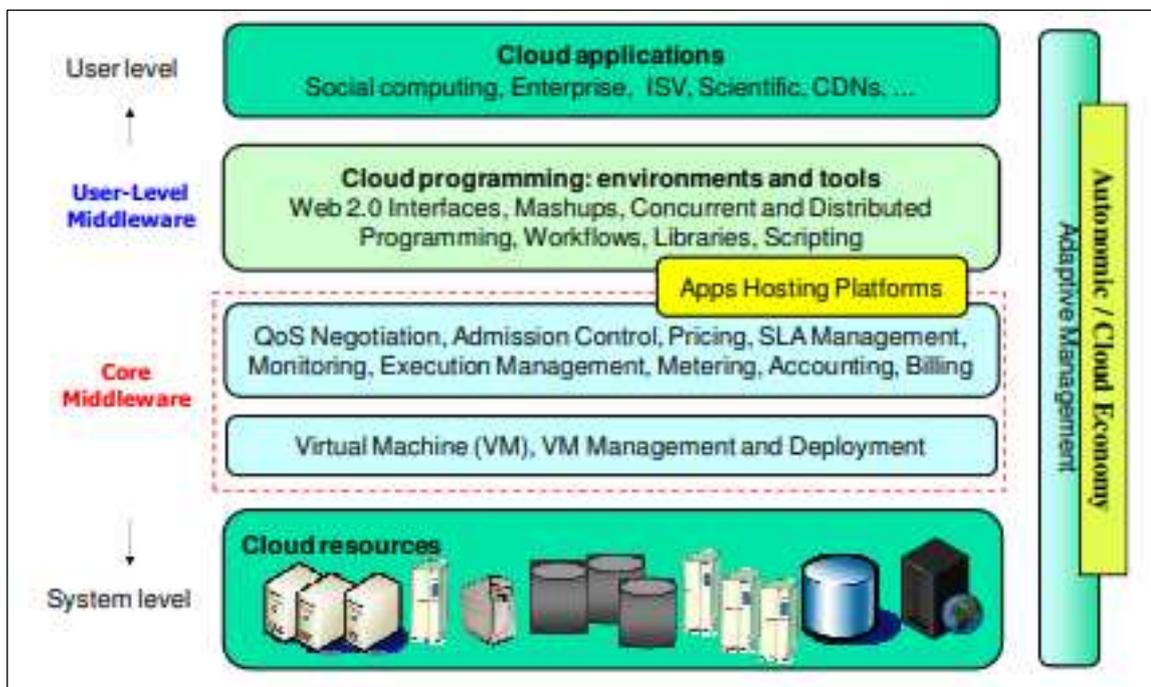
Apesar de todas as vantagens apresentadas pela computação em nuvem, suficientes para atrair o interesse das maiores empresas de tecnologia do mundo, alguns aspectos ainda trazem preocupações quanto à nuvem. Para o Programa de Computação em Nuvem da NIST (*National Institute of Standards and Technology*) – entidade americana responsável por assistir o governo dos Estados Unidos em assuntos relacionados à computação em nuvem – segurança, interoperabilidade e portabilidade são apontados como as principais barreiras enfrentadas por esta tecnologia.

### 2.3.1 Modelos de Computação em Nuvem

O ganho de flexibilidade e a desoneração dos custos associados a uma grande estrutura de TI própria, levou a computação em nuvem a convergir para três modelos principais de implementação e comercialização. São estes: SaaS, PaaS e IaaS.

Segundo Buyya, Calheiros e Ranjan (2009) estas categorias podem ser vistas como camadas, com diferentes níveis de complexidade e que podem se sobrepor, a descrição de cada camada é apresentada na Figura 5.

Figura 5 – Estrutura em camadas



Fonte: Buyya, Calheiros e Ranjan (2009).

- IaaS: este modelo compõe a camada mais básica da computação em nuvem, em que o objetivo é fornecer serviços de armazenamento de dados e capacidade de processamento computacional através da internet.
- PaaS: o modelo de Plataforma como Serviço, além incluir as funcionalidades da camada de IaaS, atende também as necessidades de middleware, hospedando o *software* da aplicação, ficando responsável por sua execução, permitindo ao cliente focar no desenvolvimento e uso do sistema.
- SaaS: o modelo de *Software* como Serviço se encontra no topo da arquitetura em camadas proposta. Utilizando as funcionalidades das

duas camadas anteriores, esta camada possui foco na aplicação para o usuário final.

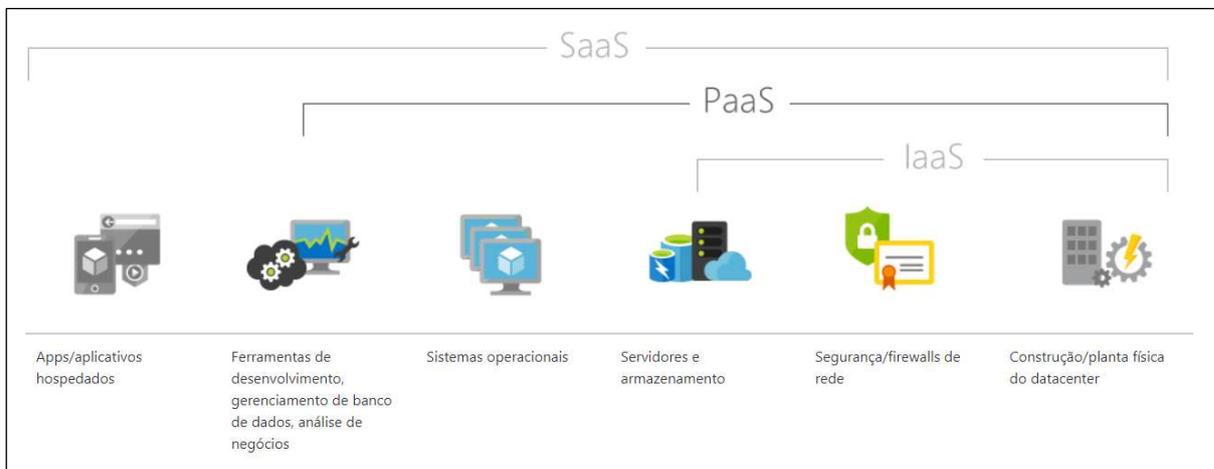
### 2.3.2 Azure

A plataforma, nomeada Windows Azure, foi lançada em 2010, sendo renomeada Microsoft Azure em 2014, é um sistema de computação em nuvem desenvolvido pela Microsoft (JENNINGS, 2009).

Segundo a Microsoft (2019), o Azure é um conjunto de serviços de *cloud computing* para auxílio às organizações. A empresa tem a liberdade de criar, gerenciar e implementar programas em rede global usando as ferramentas e estruturas que preferir.

O Azure segue o modelo PaaS, como um ambiente de desenvolvimento com diversas funcionalidades para o desenvolvimento, teste e execução de aplicações criadas pelo usuário. As funcionalidades de IaaS também estão integradas à plataforma, conforme mostrado na Figura 6.

Figura 6 – Serviços oferecidos pelo Microsoft Azure



Fonte: Microsoft (2019).

A plataforma para serviços de computação em nuvem da Microsoft obteve grande aceitação comercial, sendo utilizada em 95% das empresas que compõem a Fortune 500 (lista das 500 maiores empresas dos Estados Unidos).

O Microsoft Azure possui uma enorme quantidade de ferramentas e serviços, dentre os quais, o Azure SQL Services aplicação permite a implementação do Microsoft SQL Server, uns serviços de banco de dados amplamente difundido, em aplicações do Azure.

A plataforma conta com o Machine Learning Studio, um ambiente de desenvolvimento de ML, ferramenta de grande relevância para o desenvolvimento deste trabalho. A análise do seu potencial e recursos serão parte do escopo de desenvolvimento orientado à luz dos pilares conceituais apresentados na seção seguinte e considerando requisitos de integração aos métodos MCDM/A.

### 2.3.3 Python

Python é uma linguagem de programação aberta gerenciada pela organização sem fins lucrativos Python Software Foundation<sup>4</sup>. Segundo Bird, Klein e Loper (2009), a linguagem Python é muito utilizada por possuir características como uma menor curva de aprendizado em relação à outras linguagens de programação, estrutura dinâmica que permite a adição ativa de atributos à objetos e também que as variáveis sejam digitadas dinamicamente, facilitando o desenvolvimento rápido dos códigos. Entre as linguagens de programação, uma das características onde o Python se destaca é por sua grande e ativa comunidade de computação científica.

As bibliotecas de Python são coleções de módulos para facilitar o processo de programação através de funções pré-programadas que realizam ações sem a necessidade de programar individualmente todos os comandos que a função realiza. Dentre as bibliotecas mais utilizadas, destaca-se para a disposição e manipulação de dados a biblioteca “pandas”, que segundo McKinney (2017), proporciona funções e estruturas para desenvolver projetos com dados estruturados de maneira ágil e fácil. A biblioteca “pandas” combina a alta performance da computação utilizando vetores com a flexibilidade da manipulação de dados através de tabelas e bancos de dados, destacando-se para aplicações a estrutura *DataFrame*, uma tabela de duas dimensões com dados estruturados em colunas.

## 2.4 MACHINE LEARNING

Sistemas capazes de aperfeiçoar seu desempenho a partir de suas experiências são o ponto central do *Machine Learning* (ML). Esta área tem atraído crescente atenção e apresentado grandes avanços, tanto no campo teórico quanto em suas aplicações comerciais (MITCHELL, 1997).

---

<sup>4</sup> Python Software Foundation ([www.python.org/psf/](http://www.python.org/psf/))

Para a realização de inferências, envolvendo uma grande quantidade de dimensões, os algoritmos de ML tendem a apresentar desempenho consideravelmente superior ao de métodos de estatísticos tradicionais (BEGHI, 2015).

Na indústria, o aumento da quantidade de dados disponíveis e os aperfeiçoamentos de *hardware* viabilizaram a aplicação de modelos de ML para o gerenciamento de manutenção e qualidade. A utilização de ML é especialmente atrativa neste tipo de aplicação, devido à grande quantidade de dados (BEGHI, 2015).

#### **2.4.1 Aprendizado**

O aprimoramento do desempenho dos modelos de ML se dá através do aprendizado destes com os dados aos quais é exposto. Mitchell (1997, p. 11) define o processo de aprendizado de algoritmos como “um programa de computador é capaz de aprender com a experiência E, uma classe de tarefas T e uma medida de performance P se a sua performance na tarefa T, medida por P, é aperfeiçoada através da experiência E.”

Os algoritmos de ML podem ser categorizados quanto a sua forma de aprendizado, sendo estas aprendizado supervisionado e não-supervisionado (BENGIO, 2016).

No processo de aprendizado supervisionado, o modelo recebe um conjunto de dados a respeito de um determinado problema e recebe também um conjunto de valores objetivo para aquele determinado conjunto de dados. Desta forma o algoritmo é capaz de observar os resultados desejados para o conjunto de dados a que foi exposto (BENGIO, 2016).

No caso do aprendizado não-supervisionado, são apresentados dados sem que valores objetivo sejam informados. Esta classe de algoritmos se baseia em analisar distribuições de probabilidade e outras características comuns aos dados apresentados (BENGIO, 2016).

Além do tipo de supervisão, o processo de treinamento do modelo envolve a seleção de quais casos serão apresentados - dentre os cenários possíveis - ao algoritmo. A seleção destes casos pode ser realizada de forma aleatória, ou respeitando uma distribuição ou ordem predeterminada (MITCHELL, 1997).

## 2.4.2 Métodos de *Machine Learning*

As subseções 2.4.2.1 a 2.4.2.3 apresentam alguns dos métodos utilizados em *machine learning*.

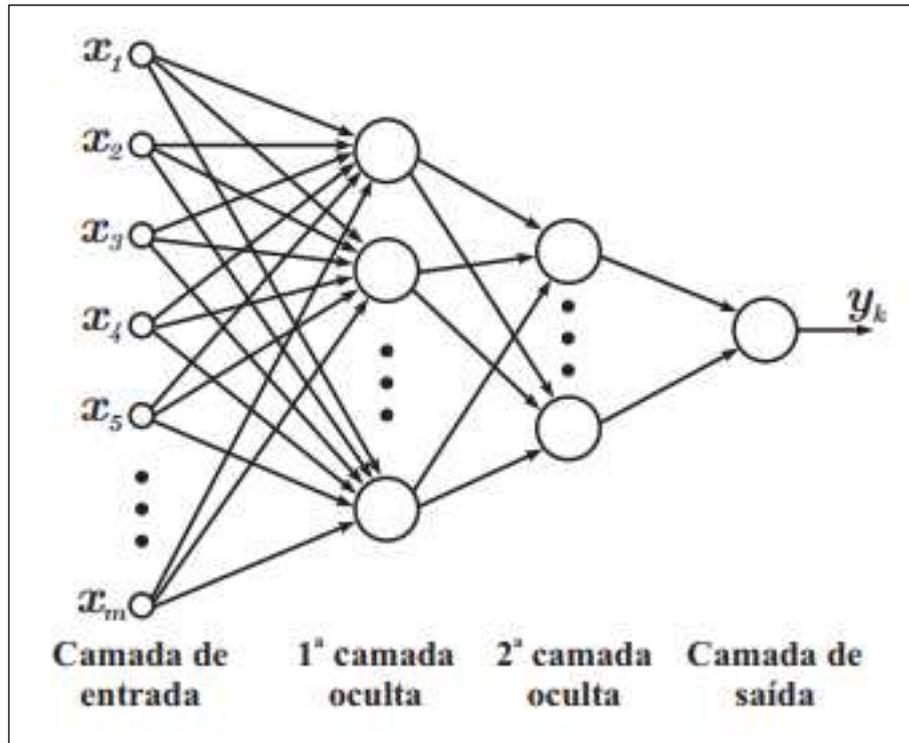
### 2.4.2.1 Redes Neurais Artificiais

A interconexão de unidades simples, capazes de receber um ou mais valores de entrada e fornecer um valor de saída, de forma que a informação de saída de uma das unidades atue como informação de entrada para outra permite às redes neurais biológicas produzam respostas rápidas e sofisticadas. Baseando-se parcialmente nas redes biológicas, as Redes Neurais Artificiais (RNAs) utilizam um modelo similar de rede de unidades simples, permitindo assim a representação distribuída de informação, com elevado nível de paralelismo (MITCHELL, 1997).

As conexões entre neurônios das RNAs se distribuem em forma de camadas. As RNAs podem ser divididos em três grupos, classificados a partir da quantidade de camadas que compõe suas redes e do tipo de conexões de que estas redes são compostas. São estes grupos: redes de camada única, redes de camadas múltiplas e redes recorrentes (MORETO, 2005).

No caso das redes de camada única, cada neurônio da rede é responsável por receber uma ou mais entradas da rede e produzir um valor, que será utilizado diretamente como saída da rede. A composição de múltiplas camadas permite que uma camada receba os valores de entrada da rede, e uma outra camada forneça os valores de saída da rede, podendo haver camadas intermediárias. A Figura 7 evidencia a arquitetura das redes de múltiplas camadas.

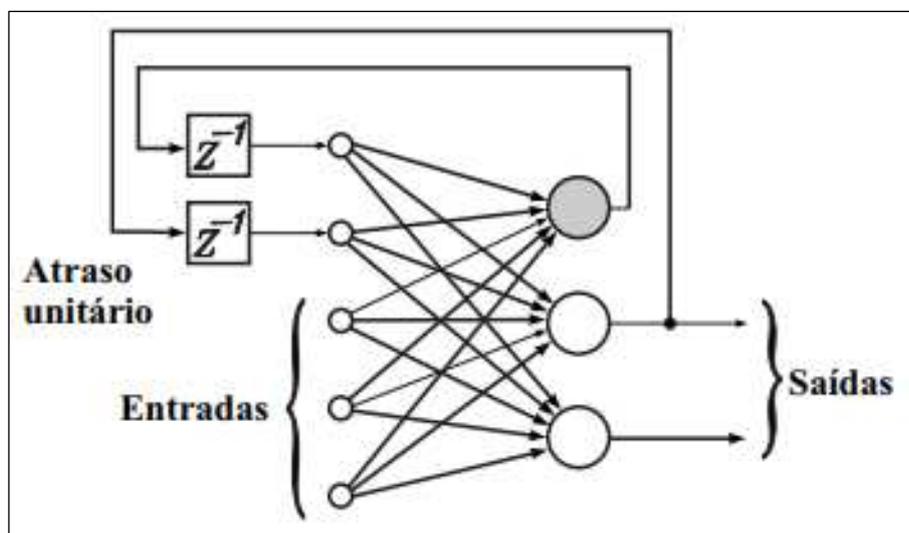
Figura 7 – Rede de múltiplas camadas



Fonte: Moreto (2005).

Pelo tipo de conexão existentes, as redes de camada única e múltipla são caracterizadas como sendo do tipo *feedforward*, visto que a saída de um neurônio não é ligada a entrada de nenhum neurônio de uma camada anterior. Já as redes recorrentes, possuem realimentação de informações entre as camadas, conforme mostrado na Figura 8.

Figura 8 – Rede recorrente



Fonte: Moreto (2005).

### 2.4.2.2 Classificadores Naive Bayes

Este método, pertence à família de métodos Bayesianos e se baseia na suposição de que, para um determinado conjunto de instâncias, contendo certos conjuntos de atributos, os atributos são condicionalmente independentes entre si, para um determinado valor objetivo da instância. Esta consideração permite que o problema, a partir de um conjunto de instâncias de treinamento, seja abordado de forma estatística e o Teorema de Bayes seja aplicado. (MITCHELL, 1997).

Equação 8 – Teorema de Bayes

$$P(h|D) = \frac{P(D|h) * P(h)}{P(D)}$$

Fonte: Mitchell (1997).

### 2.4.2.3 Árvores de Decisões

O método de representação através de Árvore de Decisões, dispendo as instâncias do problema na posição de “raiz” (nó superior do diagrama) e cada nó abaixo deste representam variações nos atributos das instâncias (MITCHELL, 1997).

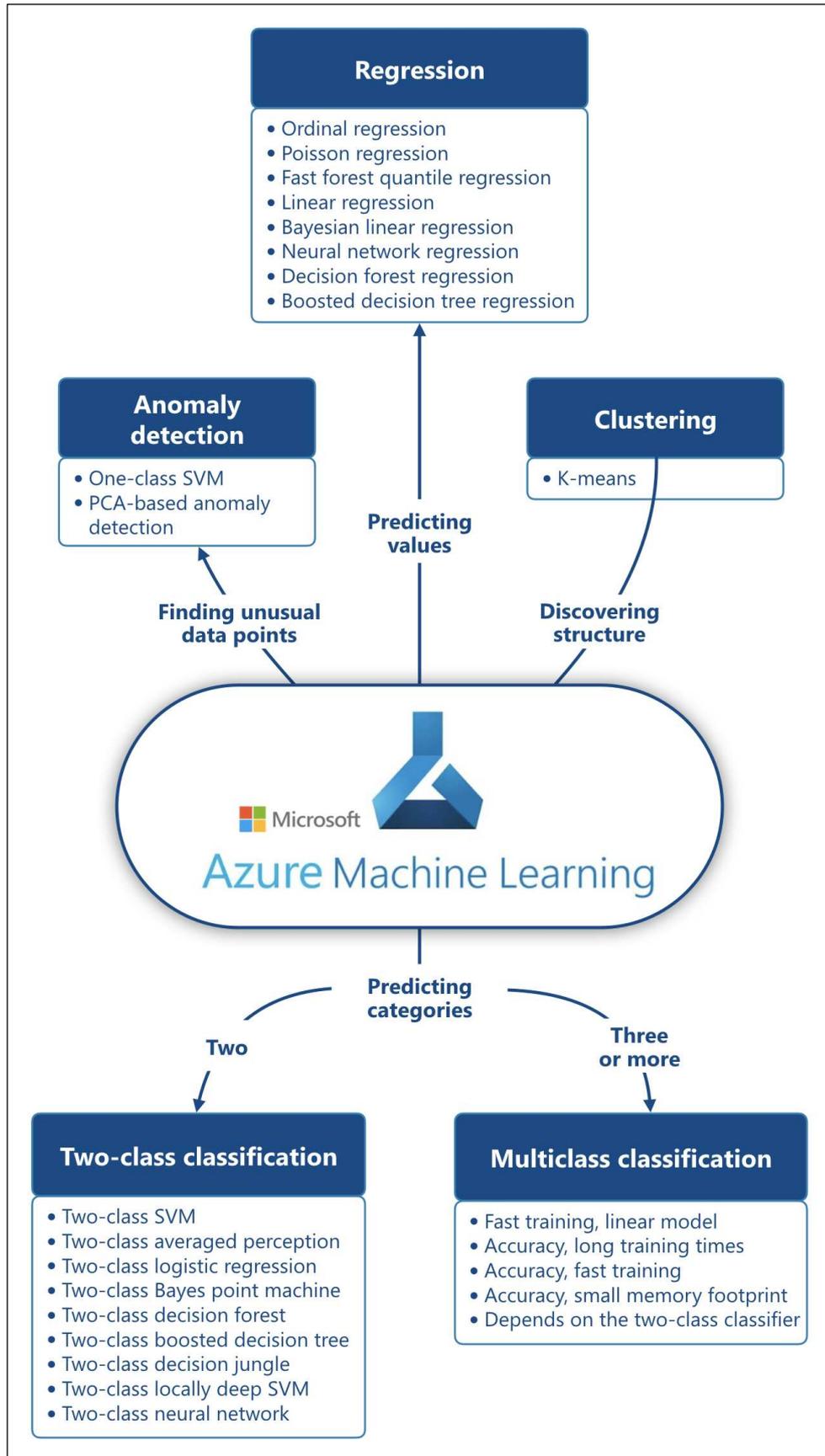
Esta forma de representação está entre as mais utilizadas em aplicações de *machine learning*, por fornecer uma forma de representação de problemas que pode ser facilmente compreendida e possuir considerável robustez em relação a irregularidades (“ruídos”) nos dados do problema (MITCHELL, 1997).

### 2.4.3 Recursos da Plataforma Azure

O Microsoft Azure conta com um ambiente próprio para o desenvolvimento e implementação de aplicações que utilizam *machine learning*, a sua estrutura é apresentada na Figura 9, o Machine Learning Studio. Este ambiente permite a utilização de diversos métodos e algoritmos, através de diagramas de blocos que podem ser construídos facilmente. O Machine Learning Studio possui algoritmos, previamente implementados, para: detecção de anomalias; agrupamento de dados; classificação binária; classificação multiclasse, e; regressão.

Além dos modelos e algoritmos já contidos no Machine Learning Studio, novos pacotes, desenvolvidos pelos usuários, podem ser implementados, nas linguagens Python e R.

Figura 9 – Microsoft Azure Machine Learning: *Algorithm Cheat Sheet*



Fonte: Adaptado de Microsoft (2019).

### 3 ABORDAGEM METODOLÓGICA

#### 3.1 CONTEXTUALIZAÇÃO DO PROBLEMA

O presente trabalho teve como propósito principal apresentar um complemento à dissertação desenvolvida pelo mestrando Ewerton Gusthavo Gorski, intitulada “Smart workflow no gerenciamento de manutenção com base em *machine learning* e métodos de apoio à tomada de decisão”. A dissertação teve como matéria a automação do processo de geração de OMs, com o objetivo de realizar o preenchimento automatizado de formulários, considerando o contexto de I4.0 e a integração e eficácia das ferramentas e processos industriais. O tema desenvolvido no trabalho do mestrando foi o ponto focal que norteou o desenvolvimento deste trabalho.

Segundo Gorski (2019), em um fluxo típico de um sistema CMMS/EAM, o início do processo é a criação de OMs, que irão descrever todas as atividades que deverão ser executadas e armazenar todas as informações sobre o processo. Na manutenção preditiva, quando é detectada uma possível falha em um processo, uma OM preditiva é aberta automaticamente pelo sistema e é utilizado um classificador para preencher os campos necessários no formulário. Por fim, a OM é alocada manualmente aos mantenedores.

Com a implementação do sistema automatizado responsável por gerar as OMs em uma planta industrial, vislumbrou-se a oportunidade de complementá-lo através do desenvolvimento de uma aplicação para realizar automaticamente a alocação das OMs com a priorização da escolha dos mantenedores de as suas características, como habilidade, especialidade e disponibilidade.

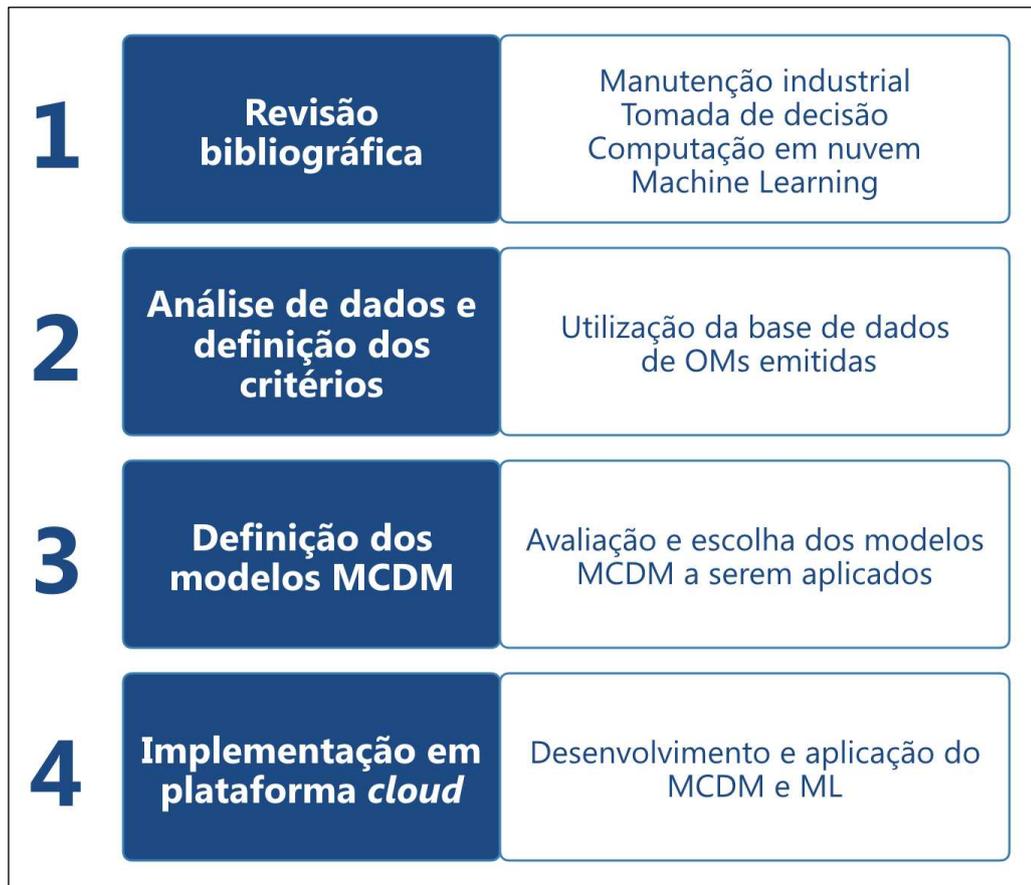
#### 3.2 ESTRUTURAÇÃO

O trabalho foi desenvolvido em duas frentes principais:

- Utilização de métodos multicritério e *machine learning* como auxílio à tomada de decisão em manutenção industrial e gerenciamento de ativos para desenvolvimento de ferramenta para priorização de OMs, e;
- Aplicabilidade de plataforma comercial de *cloud computing* para desenvolvimento e execução da ferramenta.

As duas frentes de desenvolvimento foram complementares para o êxito na aplicação da solução desenvolvida e para análise dos resultados obtidos após a realização do estudo. A abordagem para atingir os objetivos propostos foi estruturada em quatro etapas, apresentadas na Figura 10 e detalhadas na sequência:

Figura 10 – Etapas do trabalho



Fonte: Própria (2019).

### 3.2.1 Revisão Bibliográfica

A primeira etapa teve como objetivo o estabelecimento da fundamentação teórica do trabalho, através da revisão da bibliografia disponível sobre as quatro grandes áreas de estudo propostas: Manutenção industrial; Tomada de decisão; Computação em nuvem, e; *Machine Learning* - além dos conceitos abordados e as ferramentas disponíveis para alcançar os objetivos propostos.

### 3.2.2 Análise de Dados e Definição dos Critérios

Os dados sobre as OMs e sobre os mantenedores foram obtidos através da dissertação do mestrando Ewerton Gusthavo Gorski - os quais foram criados a partir

da base de dados de empresa responsável por desenvolver sistemas de gerenciamento de ativos - foram utilizados como fundamento para o desenvolvimento do presente trabalho.

A partir da análise dos dados sobre os mantenedores responsáveis pela realização das manutenções, utilizados na dissertação do Mestrando, foram estabelecidos neste trabalho os critérios a serem utilizados no modelo MCDM para a criação do ranking de alternativas de mantenedores responsáveis por atender às OMs, apresentados a seguir. Os seis critérios são aplicados no modelo com o formato de matriz, um exemplo é apresentado na Tabela 1.

- Especialidade (*specialty\_id*): cada número representa uma especialidade, por exemplo, 1 (mecânico), 2 (eletricista) e assim por diante, conforme a indústria e as características dos mantenedores. Critério pertinente para determinar o quão relevante o mantenedor é para cada situação em que o sistema gere uma OM.
- Equipe (*employee\_team\_id*): cada número descreve a equipe a que o mantenedor pertence. Critério que representa quão próxima fisicamente a equipe de mantenedores está alocada em relação ao local onde deverá ser realizada a manutenção, otimizando os tempos de manutenção da planta através da redução do deslocamento de equipes.
- Data de admissão (*admission\_date*): descreve a data de admissão do mantenedor e é utilizado pelo sistema após ser convertido para “tempo de casa”, em dias. Critério utilizado por levar em conta o nível de conhecimento tácito do funcionário devido à experiência prática e tempo desempenhando a função.
- Habilidade (*hability*): descreve o grau de habilidade dentro da especialidade (júnior, pleno, sênior, etc). Critério relevante por apresentar o nível de conhecimento teórico e prático do funcionário, conforme sua habilidade na função desempenhada.
- Mão de obra prevista (*est\_worktime*): descreve a quantidade de mão-de-obra prevista, em horas, para executar a manutenção.
- Ausência (*abstance\_worktime*): descreve a quantidade de ausências previstas em horas do mantenedor para os próximos 30 dias.

Tabela 1 – Exemplo da matriz “Mantenedores”

employee_id	specialtyid	employee team_id	admission date	hability	est_worktime	abstance worktime
1	23	3	671	3	10	2
2	15	1	611	2	8	0
3	8	5	586	4	5	1
4	2	1	340	3	6	0
5	5	4	286	2	12	1

Fonte: Própria (2019).

### 3.2.3 Definição dos Modelos MCDM

A partir da combinação dos critérios escolhidos com os dados disponíveis e critérios escolhidos, foi possível definir qual o método MCDM mais adequado para alcançar os objetivos de uma priorização adequada das OMs.

Inicialmente foram propostos três métodos multicritério que podem ser aplicados na situação estudada: AHP, TOPSIS e PROMETHEE. A seguir estão detalhadas as qualificações destes com base em sua metodologia para decidir se a aplicação do método é interessante do ponto de vista teórico e se a mesma se justifica com base no resultado esperado.

#### 3.2.3.1 AHP

Conforme apresentado no referencial teórico, o método AHP possui em sua essência um objetivo comparativo entre duas alternativas, conhecido como *pairwise comparison*, portanto não foi aplicado para o modelo de priorização. Além de possuir natureza diferente do objetivo proposto pelo trabalho, a elevada quantidade de critérios e indicadores implica em maior complexidade de implementação do método AHP, pois necessita da comparação de critérios em pares, e também no aumento do nível de inconsistência da avaliação.

### 3.2.3.2 TOPSIS

O método TOPSIS apresenta vantagens para a aplicação proposta em relação aos outros métodos propostos. O modelo possui em sua essência o objetivo de avaliação e priorização, a implementação computacional pode ser realizada com baixa complexidade e a literatura acusa o uso do TOPSIS em problemas análogos.

Segundo Shih, Shyur e Lee (2006), o método TOPSIS é a técnica mais direta para a utilização de MCDM e devido ao seu raciocínio lógico o método TOPSIS têm resolvido muitos problemas do mundo real. Quando comparado ao AHP, o TOPSIS alivia o requisito de comparações pareadas e a limitação da capacidade pode não dominar significativamente o processo.

Para Gorski (2019), os critérios que devem ser analisados para determinar a alocação de mantenedores variam de acordo com cada organização e por esse motivo não há como determinar quais critérios deverão ser utilizados de forma genérica, sendo necessário levantar os pontos avaliados com cada organização.

### 3.2.3.3 PROMETHEE

O método PROMETHEE pode ser considerado uma evolução do método TOPSIS, porém não foi aplicado para o modelo de priorização pois apresenta expressivo aumento na complexidade de implementação sem um expressivo diferencial em relação à metodologia aplicada no TOPSIS, que já atende o objetivo suficientemente bem.

## 3.2.4 Implementação em Plataforma *Cloud*

Essa etapa consistiu na aplicação do método multicritério TOPSIS, considerado o mais adequado para o objetivo proposto, utilizando os critérios definidos anteriormente. A implementação na plataforma Azure permitiu a aplicação do MCDM em paralelo com o algoritmo de ML.

Os métodos MCDM e ML consomem os dados do banco de dados e auxiliam na alocação das OMs. Para o ML, existe o pré-requisito de que exista uma base de treinamento (ML supervisionado), baseado em histórico de manutenções alocadas manualmente, onde o *software* possa abstrair as informações e treiná-las.

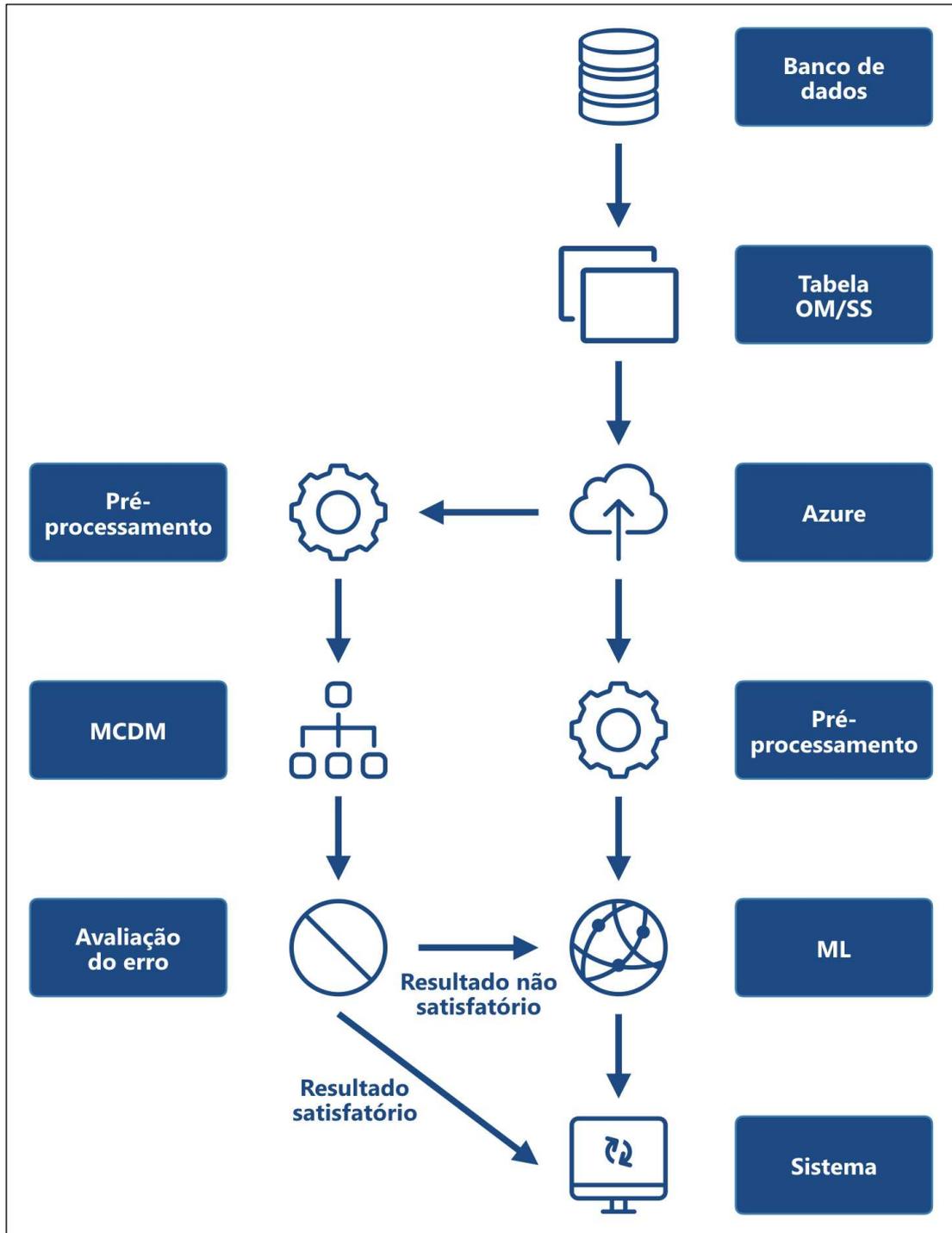
#### 3.2.4.1 Abordagem Paralelo entre MCDM e ML

Foi utilizada a abordagem em paralelo entre MCDM e ML para auxiliar na alocação das OMs, onde serão utilizadas ferramentas para indicar qual o melhor mantenedor para alocar cada OM gerada. Para cada OM gerada automaticamente pelo sistema, o usuário deverá analisar e quantificar a mão de obra prevista para a execução da manutenção, e posteriormente a OM será processada pelos métodos MCDM e, se necessário, ML e irá alocar a manutenção para o mantenedor que mais se adeque nas especificações da manutenção a ser realizada.

A abordagem em paralelo é apresentada na Figura 11 e consiste na classificação por ambos os métodos após a disponibilização das informações em relação às OMs através do banco de dados. O MCDM, através do método TOPSIS, deverá classificar primeiramente e se a priorização não atingir o resultado esperado entre os itens ranqueados, o algoritmo de ML deverá ser usado para classificar, por se demonstrar mais preciso uma vez que utiliza o banco de dados para realizar o treinamento e a classificação.

O MCDM tem baixo custo computacional comparado ao ML, dessa forma sempre será executado e o ML somente quando o resultado do MCDM não atingir a condição proposta. Segundo Gorski (2019), essa estratégia foi adotada pois normalmente utiliza-se métodos de ML para esse modelo de classificação, porém estamos explorando métodos com menor custo computacional e facilidade de desenvolvimento que podem trazer os mesmos resultados, como é o caso do MCDM.

Figura 11 – Estrutura da abordagem em paralelo



Fonte: Adaptado de Gorski (2019).

### 3.2.4.2 Condição para Aplicação do ML

Segundo Martins (2013), o desvio padrão de uma amostra de dados quantitativos é uma medida de dispersão dos dados em relação à média, que se obtém através da raiz quadrada da variância amostral. Uma vez que a variância amostral se exprime nas unidades dos dados elevados ao quadrado, considera-se

como medida de dispersão a sua raiz quadrada. Se representarmos os dados por  $x_1, x_2, \dots, x_n$  e  $\bar{x}$  a sua média, o desvio padrão amostral  $S$  é obtido através da expressão apresentada na **Equação 1**Equação 9, onde  $n$  é a quantidade de alternativas.

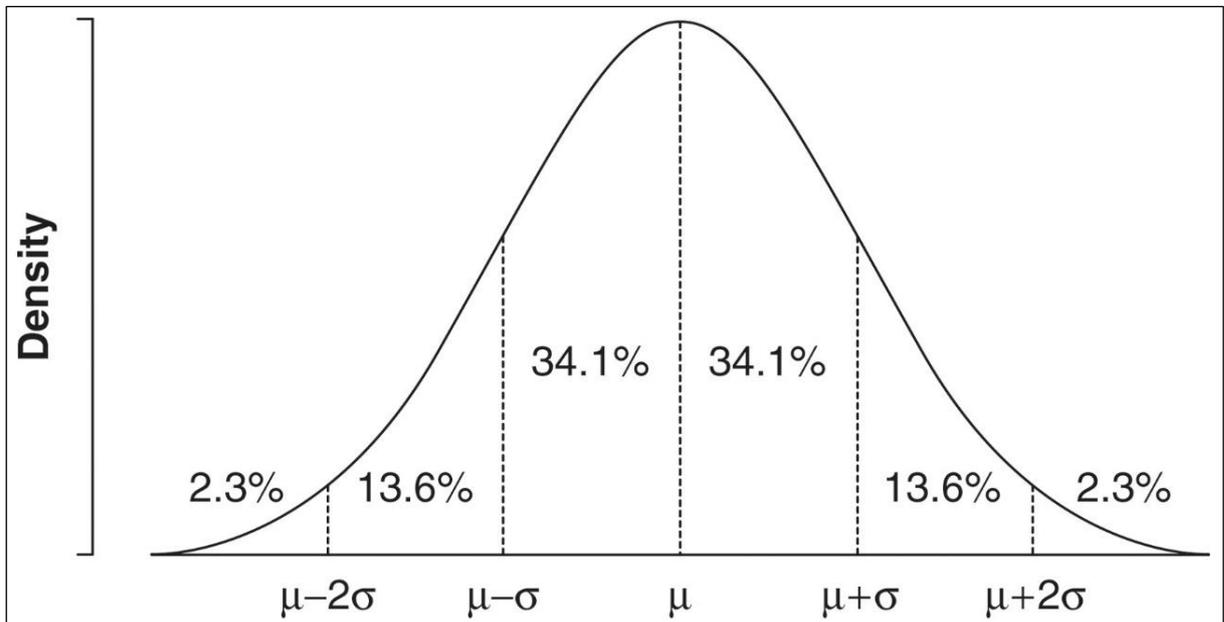
Equação 9 – Desvio padrão amostral

$$S = \sqrt{S^2} = \sqrt{\frac{\sum_i^n (x_n - \bar{x})^2}{n - 1}}$$

Fonte: Martins (2013).

Obtendo-se o desvio padrão amostral é possível mapear a dispersão dos dados e saber a quantidade de dados dentro de cada intervalo, esse mapeamento é conhecido como distribuição em forma de sino. A Figura 12 apresenta a distribuição nos dois primeiros intervalos, onde 68,2% dos dados estão contidos entre  $\bar{x} \pm S$  e 95,4% dos dados estão contidos entre  $\bar{x} \pm 2S$ .

Figura 12 – Distribuição em forma de sino



Fonte: Allen (2017).

A condição proposta para a necessidade de aplicação do ML foi estabelecida através do uso do primeiro intervalo de distribuição normal dos resultados de priorização obtidos pelo MCDM. Se o ranking de mantenedores para alocação das OMs apresentar alternativas com valores próximos, dentro do primeiro intervalo de distribuição normal (68%), deverá ser aplicado o ML e utilizar como resultado final a resposta encontrada por ele. Caso o ranking apresente a melhor alternativa sendo a

única posicionada fora do primeiro intervalo de distribuição no intervalo positivo, a resposta do MCDM será considerada como resultado final e não haverá necessidade de aplicação do ML, pois pode-se assumir que a discriminação entre as alternativas é alta e obtivemos uma alternativa elitista.

## 4 APLICAÇÃO

A aplicação segue o modelo do sistema automatizado responsável por gerar as OMs, portanto o evento de término de um estágio do sistema é o gatilho para o início do estágio seguinte. O final do estágio de emissão das OMs dá início ao estágio de aplicação do MCDM utilizando o método TOPSIS, iniciando pelo pré-processamento dos dados da OM gerada.

### 4.1 MCDM

A base de dados original contém um total de 72 tipos de registros divididos em colunas, consistindo nas ordens de manutenção geradas entre agosto de 2018 e agosto de 2019 e os mantenedores alocados para realizá-las.

Os registros consistem em 761 ocorrências – cada ocorrência é descrita em uma linha da base de dados – com diversas informações e características a respeito da execução da manutenção descritas nas colunas, como: tempo em que o serviço foi executado, prioridade que foi atribuída a este, custo, etc.

Para a implementação do TOPSIS a base de dados foi adaptada, selecionando seis características dos mantenedores e desconsiderando as informações referentes às OMs antigas. O modelo TOPSIS é acionado quando existe uma nova ordem de manutenção a ser alocada.

Uma segunda base de dados deve ser fornecida ao modelo – esta não provém do trabalho do mestrando – e relaciona um valor para cada uma das alternativas das seis características sendo avaliadas pelo modelo, podendo estes valores serem atribuídos por especialistas na área de manutenção, familiarizados com as características da planta e do processo em questão.

O aprimoramento do modelo MCDM pode ser desenvolvido através de três métodos: Ajuste dos valores de cada uma das alternativas disponíveis para cada critério; Ajuste das relações entre alternativas disponíveis e OM, e; Ajuste dos pesos de cada critério.

A execução destes três métodos de aprimoramento até o atingimento de um valor ótimo está além do escopo do presente trabalho – tendo sido utilizados valores considerados suficientes para um trabalho cujo objetivo é a análise e desenvolvimento de métodos de avaliação, não o estudo de um caso específico.

As características retiradas da base de dados para aplicação do modelo MCDM foram:

- *Employee\_id*: número de identificação do funcionário.
- *Specialty\_Id*: número da área de especialidade do funcionário de manutenção.
- *Employee\_Team\_id*: número de identificação da equipe do funcionário.
- *Admission\_date*: data que o funcionário começou a atuar na empresa, este dado está relacionado a experiência do mantenedor e foi passado para o modelo como a quantidade de dias desde a admissão do funcionário.
- *Hability*: descreve o grau de habilidade do mantenedor. Este dado é expressado através de um valor numérico. Os valores mais elevados representam funcionários mais habilitados.
- *Est\_work*: descreve a quantidade de horas prevista para a execução do serviço de manutenção.
- *Abstence\_worktime*: descreve a quantidade de horas de ausência prevista para o funcionário nos 30 dias seguintes, em horas.

#### 4.1.1 Pré-processamento dos Dados para Aplicação do MCDM

A base de dados obtida a partir da OM consiste em dois arquivos em formato “csv”. O primeiro contém uma matriz em que cada linha representa um dos funcionários cadastrados e cada coluna um dos critérios que serão avaliados pelo MCDM. Os valores desta matriz representam o atributo que o funcionário possui em termos absolutos – independente das especificações de uma determinada OM. Esta matriz será tratada como “Matriz Absoluta”.

O segundo arquivo possui uma matriz que contém os valores que cada uma das pontuações disponíveis para um determinado critério representa para a OM específica que está sendo avaliada. Esta matriz será tratada como “Matriz de Correlação”.

O pré-processamento dos dados da OM é necessário pois cada OM estabelecerá critérios diferentes para o ranqueamento realizado pelo MCDM, isso ocorre através da correlação dos dados da “Matriz Absoluta” e da “Matriz de

Correlação”. A partir deste processo, é gerada uma terceira matriz, chamada de “Matriz Relativa”.

A necessidade da obtenção da “Matriz Relativa” se deve ao fato de ser necessário quantificar o quanto cada uma das alternativas (mantenedores) contribui para um determinado critério de avaliação, em relação aos critérios específicos de avaliação para a OM que está sendo processada. Por exemplo, a habilidade ideal que o mantenedor deve possuir para a alocação da OM irá variar conforme a necessidade específica da OM gerada. Portanto, os critérios utilizados no MCDM devem ser dinâmicos em função da OM que está sendo processada.

O estágio de pré-processamento garante também que os dados respeitem os limites de intervalo de pontuação do TOPSIS – valores entre 0 e 10 – e a escala de valor correta – quanto maior o valor, mais a alternativa se aproxima do ideal para o critério em questão. Com esse estágio, o modelo criado garante que os dados são adequados para a implementação no modelo TOPSIS.

O estágio de pré-processamento foi implementado no Azure Machine Learning Studio através de um módulo na linguagem de programação Python, o código desenvolvido foi aplicado no módulo “*Execute Python Script*”, conforme apresentado na Figura 13, e pode ser consultado na íntegra no Apêndice A.

A rotina para este processo foi elaborada a partir do conhecimento das informações que são aplicadas no modelo, consistindo em garantir que para cada uma das colunas, os valores nela contidos seguissem o formato de escala definido e, quando necessário, fosse aplicada na base de dados uma escala de equivalência linear entre o maior e o menor valor disponível para o critério em questão.

Figura 13 – Trecho do código de pré-processamento implementado

```

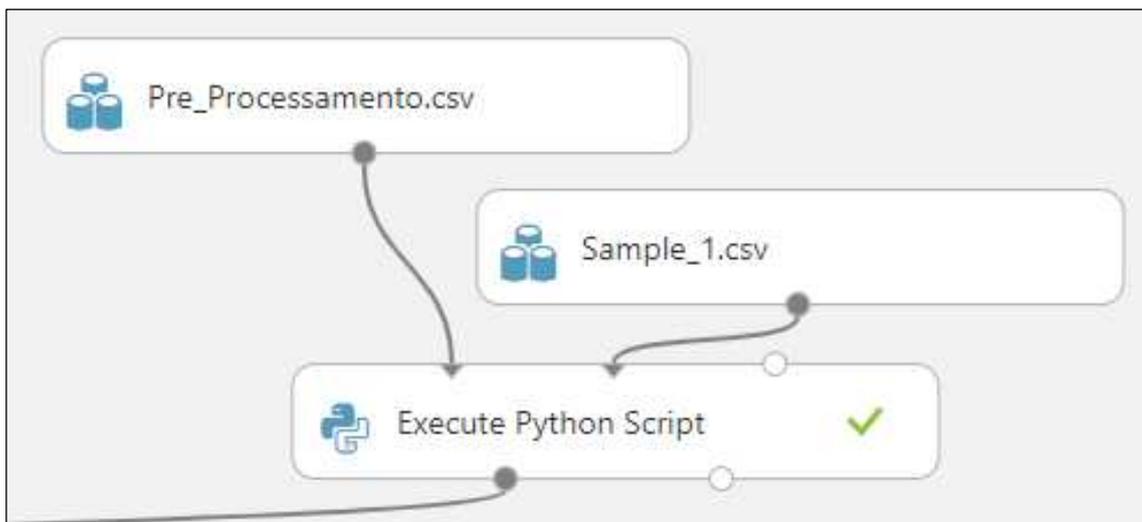
Execute Python Script
Python script
11 def azureml_main(dataframe1 = None, dataframe2 = None):
12
13     # Execution logic goes here
14     cab_b = list(dataframe1.columns)
15     cab_s = list(dataframe2.columns)
16     dictionary2 = {}
17
18     #for each in cab:
19     for i in range(0, 5):
20         j = i*2
21         limp1 = [x for x in dataframe1[cab_b[j]].values if str(x) != 'nan']
22         limp2 = [x for x in dataframe1[cab_b[j+1]].values if str(x) != 'nan']
23         dictionary = dict(zip(limp1, limp2))
24         dictionary2[cab_b[j]] = dictionary
25
26     resp = dataframe2.copy()
27     resp.drop('employee_id', axis=1)

```

Fonte: Própria (2019).

O módulo de processamento de dados recebe como entradas as duas bases de dados em “csv”, conforme apresentado na Figura 14. Após correlacionar e normalizar as duas matrizes, o módulo produz um arquivo “*Dataframe*” como saída.

Figura 14 – Implementação do pré-processamento dos dados do MCDM

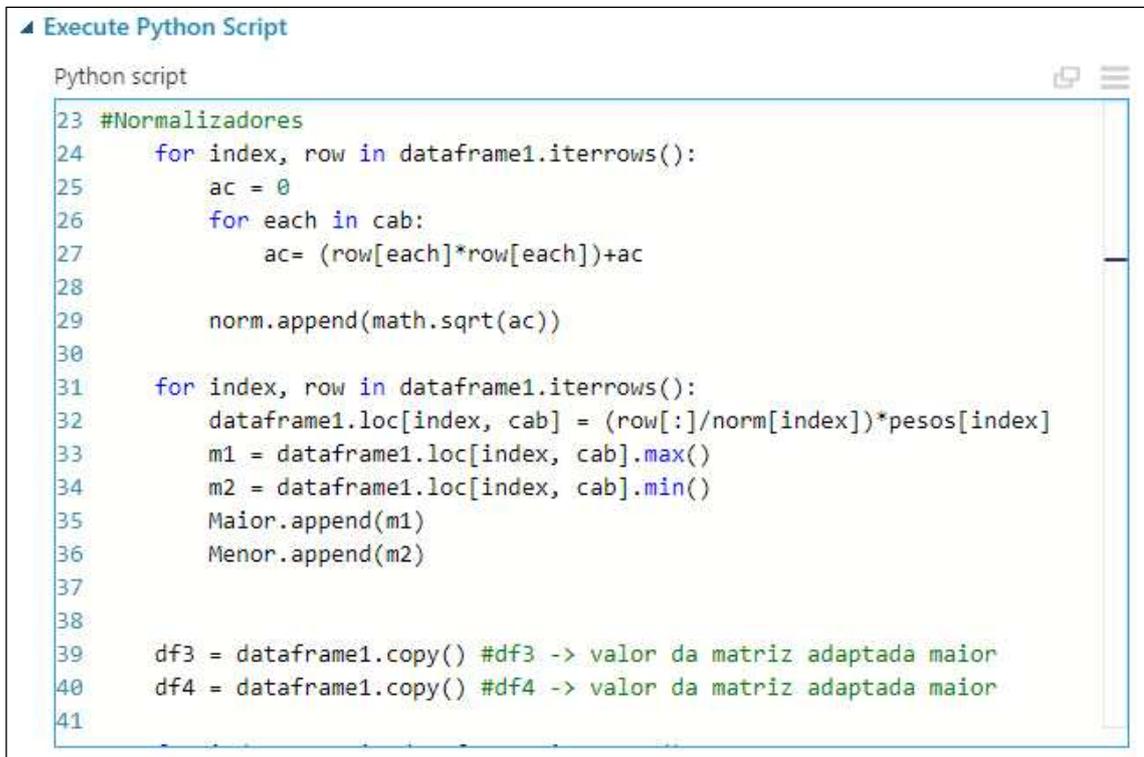


Fonte: Própria (2019).

### 4.1.2 Implementação do MCDM

A criação do modelo no ambiente de desenvolvimento Azure Machine Learning Studio demandou a implementação do MCDM em um formato compatível com o ambiente. Mesmo não contendo nenhum método MCDM em sua biblioteca de algoritmos, foi possível realizar a implementação utilizando um módulo na linguagem Python. O código desenvolvido foi aplicado no módulo “Execute Python Script”, conforme trecho apresentado na Figura 15, e pode ser consultado na íntegra no Apêndice B.

Figura 15 - Trecho do código do MCDM implementado



```

Python script
23 #Normalizadores
24     for index, row in dataframe1.iterrows():
25         ac = 0
26         for each in cab:
27             ac= (row[each]*row[each])+ac
28
29         norm.append(math.sqrt(ac))
30
31     for index, row in dataframe1.iterrows():
32         dataframe1.loc[index, cab] = (row[:]/norm[index])*pesos[index]
33         m1 = dataframe1.loc[index, cab].max()
34         m2 = dataframe1.loc[index, cab].min()
35         Maior.append(m1)
36         Menor.append(m2)
37
38
39     df3 = dataframe1.copy() #df3 -> valor da matriz adaptada maior
40     df4 = dataframe1.copy() #df4 -> valor da matriz adaptada maior
41

```

Fonte: Própria (2019).

A implementação do método como um módulo em Python exigiu alguns cuidados para que o módulo fosse compatível com o restante do modelo desenvolvido:

- O módulo Python foi desenvolvido de forma a tratar os dados recebidos como entrada como “*Dataframes*” – estrutura de dados implementada pela biblioteca Pandas, disponível para a linguagem Python – visto que esta estrutura é compatível com a estrutura de dados de “*Datasets*”, utilizada como padrão no Azure Machine Learning Studio.

- O método MCDM foi implementado de forma dinâmica, podendo aceitar diferentes quantidades de critérios e opções disponíveis, sendo estas definidas pelas dimensões dos “*Datasets*” de entrada.
- Os resultados obtidos pelo método devem ser dispostos como um “*Dataframe*”, para serem utilizados como saída do módulo e poderem ser reconhecidos como um “*Dataset*” pelo modelo.

O método TOPSIS recebeu como *input* dois “*Datasets*”: um contendo as alternativas a serem avaliadas e a sua respectiva pontuação para cada um dos critérios, em formato matricial; e o outro incluindo a matriz de pesos relativa aos critérios considerados pelo método.

A matriz de pesos utilizada possui valor unitário, garantindo resultados considerados suficientes para o objetivo proposto. Como explicado anteriormente, alguns métodos de aprimoramento possíveis estão além do escopo do presente trabalho.

Por sua vez, o TOPSIS produz como *output* um “*Dataset*” contendo duas colunas: a primeira delas listando as alternativas avaliadas pelo método e a segunda contendo a pontuação obtida por cada uma na classificação realizada pelo método. Os valores de pontuação variam de 0 a 1, sendo 1 a alternativa ideal e 0 a alternativa mais distante possível da ideal.

#### **4.1.3 Avaliação do Erro do MCDM**

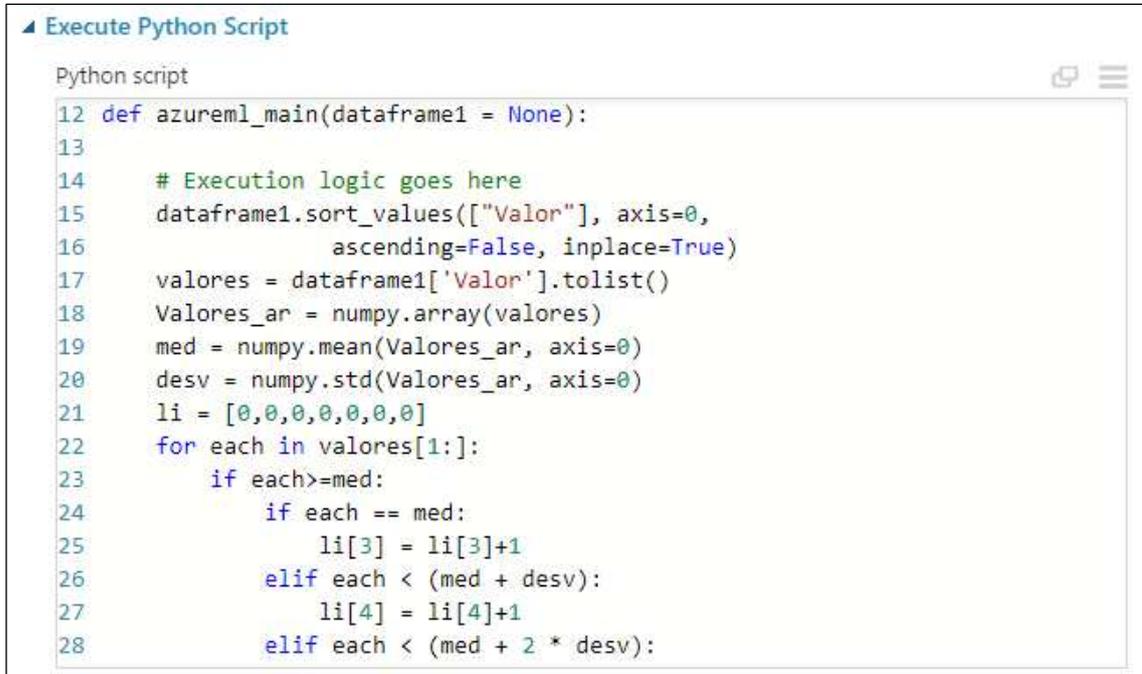
Conforme explicado na página 49, a condição proposta para a necessidade de aplicação do ML foi o primeiro intervalo de distribuição normal dos resultados de priorização obtidos pelo MCDM. A avaliação do erro do método TOPSIS visou identificar se a alternativa apontada como mais adequada pelo modelo é uma alternativa elitista, considerando o primeiro intervalo de distribuição normal.

O módulo implementado recebe como alimentação o “*Dataset*” contendo o ranking de alternativas fornecido pelo método TOPSIS. A classificação de cada alternativa no ranking consiste de um valor entre 0 e 1, sendo um a alternativa ideal. O algoritmo implementado calcula então o valor médio dos valores de cada uma das alternativas no ranking e o desvio padrão destes valores.

A avaliação do erro do modelo MCDM foi implementada através de um módulo em Python. O código utilizado na avaliação do erro do MCDM foi aplicado no módulo

“Execute Python Script”, conforme trecho apresentado na Figura 16, e pode ser consultado na íntegra no Apêndice C.

Figura 16 - Trecho do código para avaliação do erro do MCDM implementado



```

12 def azureml_main(dataframe1 = None):
13
14     # Execution logic goes here
15     dataframe1.sort_values(["Valor"], axis=0,
16                           ascending=False, inplace=True)
17     valores = dataframe1['Valor'].tolist()
18     Valores_ar = numpy.array(valores)
19     med = numpy.mean(Valores_ar, axis=0)
20     desv = numpy.std(Valores_ar, axis=0)
21     li = [0,0,0,0,0,0,0]
22     for each in valores[1:]:
23         if each>=med:
24             if each == med:
25                 li[3] = li[3]+1
26             elif each < (med + desv):
27                 li[4] = li[4]+1
28             elif each < (med + 2 * desv):

```

Fonte: Própria (2019).

A melhor alternativa apontada deve ser a única a possuir valor superior ao do primeiro intervalo de distribuição normal, representado pelo intervalo de 68% com centro no valor da média dos resultados. A alternativa será considerada elitista apenas se for a única em seu intervalo e este for superior ao primeiro intervalo – obedecendo assim o pré-requisito de 68% da distribuição.

## 4.2 ML

Para o modelo de ML também foi utilizada a base de dados proveniente da dissertação do mestrando, tendo sido realizado um pré-processamento em três etapas. Diferente do modelo MCDM, no modelo ML é utilizado o histórico de pareamento de ordens de manutenção com os mantenedores para o treinamento dos algoritmos. Nesta aplicação não foi necessária a implementação de nenhum módulo utilizando a linguagem Python, pois o ambiente Azure Machine Learning Studio já apresenta todas as funções para processamento de dados utilizando ML.

#### 4.2.1 Pré-processamento dos Dados para Aplicação no ML

A primeira etapa consistiu na seleção das colunas que continham valores relevantes para a aplicação do modelo e na criação de uma nova coluna, descrevendo o tempo em que a ordem de manutenção ficou em aberto – a base de dados traz a data de abertura e fechamento da OM, onde o cálculo do intervalo de tempo pode ser realizado facilmente no próprio Azure. A seleção das colunas a serem utilizadas foi realizada através do módulo “*Select Columns in Dataset*”, disponível no Azure Machine Learning Studio. As informações selecionadas referentes aos mantenedores podem ser visualizadas na Tabela 2 e as referentes às OMs podem ser visualizadas na Tabela 3.

Tabela 2 – Informações selecionadas para o ML sobre os mantenedores

<b>Informação</b>	<b>Descrição</b>
<i>employee_id</i>	Número de identificação do funcionário. Valor que deseja-se classificar.
<i>specialty_id</i>	Número da área de especialidade do funcionário de manutenção.
<i>employee_team_id</i>	Número de identificação da equipe do funcionário.
<i>admission_date</i>	Data de admissão do funcionário.
<i>admission_date</i>	Data que o funcionário começou a atuar na empresa.
<i>hability</i>	Grau de habilidade do mantenedor, expressado através de um valor numérico.
<i>est_worktime</i>	Quantidade de horas previstas para a execução do serviço de manutenção.
<i>abstace_worktime</i>	Quantidade de horas de ausência previstas para o funcionário nos próximos 30 dias.

Fonte: Própria (2019).

Tabela 3 – Informações selecionadas para o ML sobre as OMs

<b>Informação</b>	<b>Descrição</b>
<i>company_id</i>	Número de identificação da empresa em que a manutenção será realizada.
<i>area_id</i>	Número de identificação da área física da empresa onde a manutenção será realizada.
<i>asset_group_id</i>	Número de identificação do ativo que necessita de manutenção.
<i>priority</i>	Prioridade constatada para a execução do serviço
<i>performed_worktime</i>	Tempo de duração do serviço de manutenção
<i>estimated_worktime</i>	Tempo previsto para a duração do serviço de manutenção
<i>maint_order_status_id</i>	Status atual da ordem de manutenção
<i>maint_order_activities_type</i>	Tipos de atividade descritas na OM
<i>is_scheduled</i>	Agendamento prévio da OM
<i>priority_calculated</i>	Prioridade calculada
<i>need_asset_stop</i>	Parada necessária do ativo
<i>total_cost</i>	Custo total

Fonte: Própria (2019).

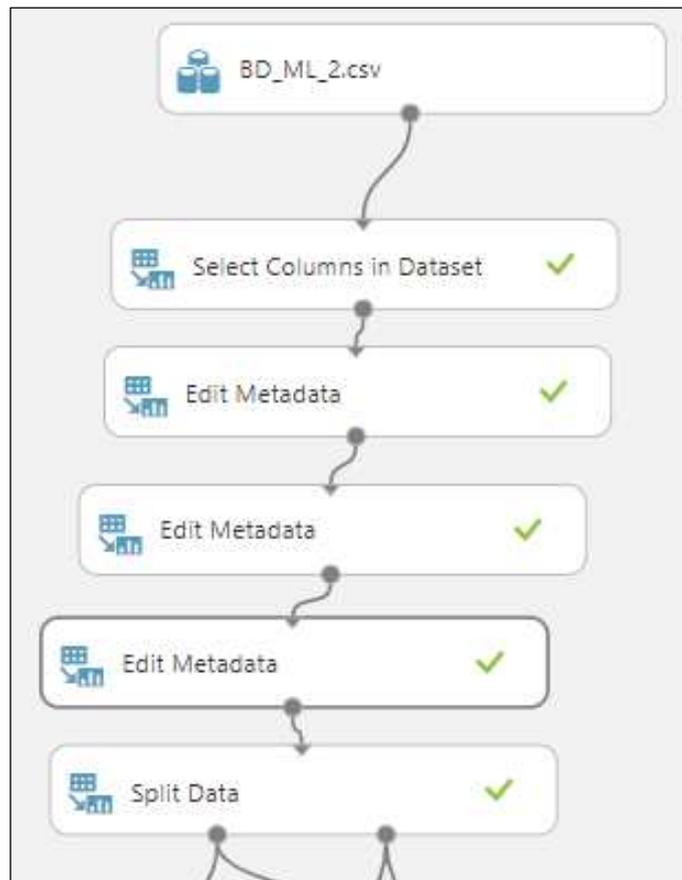
A segunda etapa consistiu na classificação dos dados, de forma que os modelos de ML pudessem interpretá-los de maneira correta. A classificação dos dados foi realizada com o módulo “*Edit Metadata*”, disponível no Azure Machine Learning Studio. As informações foram classificadas quanto a sua categoria e tipo de informação – a classificação foi feita com base nas colunas da base de dados.

- Categoria (*categorical*): os dados foram classificados como categóricos ou não categóricos. Dados que sejam constituídos por um conjunto limitado de possíveis valores, como por exemplo valores booleanos devem ser classificados como categóricos. Já informações que possam adotar um número infinito de valores devem ser classificadas como não-categóricos – por exemplo valores numéricos que representem datas, número de ocorrências de um determinado fenômeno, etc.
- Tipo de informação (*fields*): os dados cuja classificação não é o objetivo do modelo devem ser identificados como características (*features*). Já os dados que deverão ser classificados, devem ser identificados como rótulos (*labels*).

Os valores de “*Employee\_id*” foram categorizados como “*labels*”, por serem o valor que cuja classificação é o objetivo do método.

Como última etapa do pré-processamento, foi realizada a divisão entre os dados que seriam utilizados no processo de treinamento dos modelos e os utilizados para a avaliação do desempenho destes, característica do ML supervisionado. A divisão ocorreu com base no número total de tipos de registros contidos na base de dados e para auxiliar neste processo, foi utilizado o método “*Split*”, disponível no Azure Machine Learning Studio – 70% dos registros foram utilizados para treinamento e 30% para a validação do modelo. Todas as etapas do pré-processamento são apresentadas na Figura 17.

Figura 17 – Implementação do pré-processamento dos dados do ML



Fonte: Própria (2019).

#### 4.2.2 Implementação do ML

A implementação dos métodos de ML se deu através de duas etapas, uma de treinamento dos modelos e ajuste dos parâmetros e outra de avaliação do desempenho destes e reajuste dos parâmetros utilizados. O processo de

implementação se deu após a preparação dos dados, realizada durante a fase de pré-processamento.

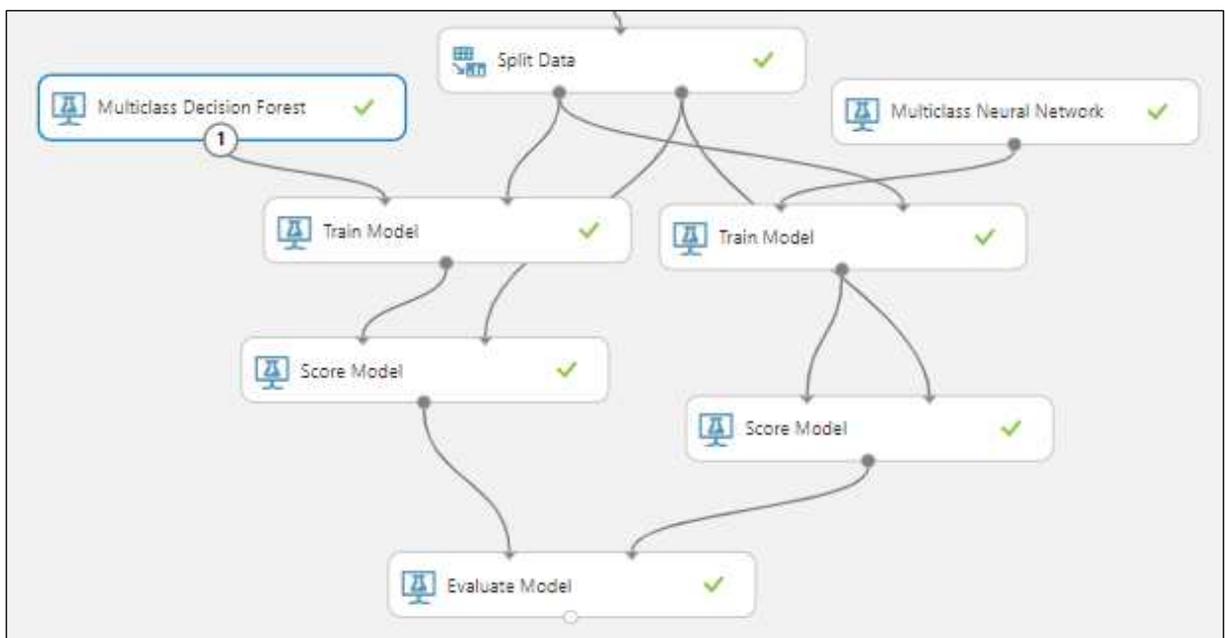
A implementação dos modelos de ML ocorreu inteiramente dentro do ambiente de desenvolvimento do Azure Machine Learning Studio com a utilização dos módulos disponíveis, apresentada na Figura 18. Dois algoritmos de ML foram implementados simultaneamente para que os resultados pudessem ser comparados ao final do processo.

Os dados processados são utilizados como entrada para os blocos de treinamento (*“Train Model”*): dois módulos de treinamento foram implementados, sendo um modelo para cada algoritmo.

Os módulos de cada um dos algoritmos de ML utilizados são ligados ao seu respectivo módulo de treinamento. Os algoritmos foram selecionados com base dentre as opções de algoritmos de classificação multiclasse disponíveis no Azure Machine Learning Studio. Foram selecionados:

- Rede Neural Multiclasse: consiste na aplicação de uma rede neural ao problema de classificação multicritério.
- Floresta de Decisão Multiclasse: método baseado na utilização de diversas árvores de decisão interligadas, aplicado ao problema de classificação multicritério.

Figura 18 – Implementação do ML através de dois módulos diferentes



Fonte: Própria (2019).

As saídas dos modelos de treinamento foram ligadas aos módulos de pontuação (“*Score Module*”). Estes módulos fornecem como saída um “*Dataset*” contendo as informações com as quais o modelo foi alimentado, com as respostas de classificação produzidas pelo modelo adicionadas como colunas ao final do “*Dataset*”.

Neste estágio de implementação já é possível verificar as respostas produzidas pelos algoritmos implementados e compará-las ao resultado obtido no pareamento entre mantenedor e OM que foi realmente realizado – obtido através da base de dados do mestrando Ewerton Gorski.

Para viabilizar a análise de acurácia do modelo, um módulo de avaliação (“*Evaluation Model*”) também foi implementado. Este módulo recebe como entrada o resultado do modelo de pontuação de ambos os algoritmos. A saída do módulo de avaliação é deixada em aberto.

#### **4.2.3 Avaliação da Acurácia do ML**

A avaliação da acurácia do ML foi realizada através de um módulo onde foi possível verificar os indicadores de desempenho e acurácia dos algoritmos de ML implementados.

O módulo de avaliação oferece quatro métricas de acurácia:

- Acurácia geral (“*Overall Accuracy*”): avalia a quantidade de classificações corretas em relação a quantidade total de classificações realizadas – considerando apenas os acertos exatos.
- Acurácia média (“*Average Accuracy*”): avalia a quantidade média de classificações corretas em relação a quantidade total de classificações realizadas.
- Precisão micro-média (“*Micro-Averaged Precision*”): a micro-média leva em conta a quantidade total de resultados verdadeiro-positivos em relação ao total de predições, independente das classes sendo consideradas pelo modelo.
- Precisão macro-média (“*Macro-Averaged Precision*”): a macro-média leva em consideração a média não ponderada de taxa de acerto dos algoritmos para cada classe.

## 5 ANÁLISE DOS RESULTADOS

### 5.1 RESULTADOS DO MCDM

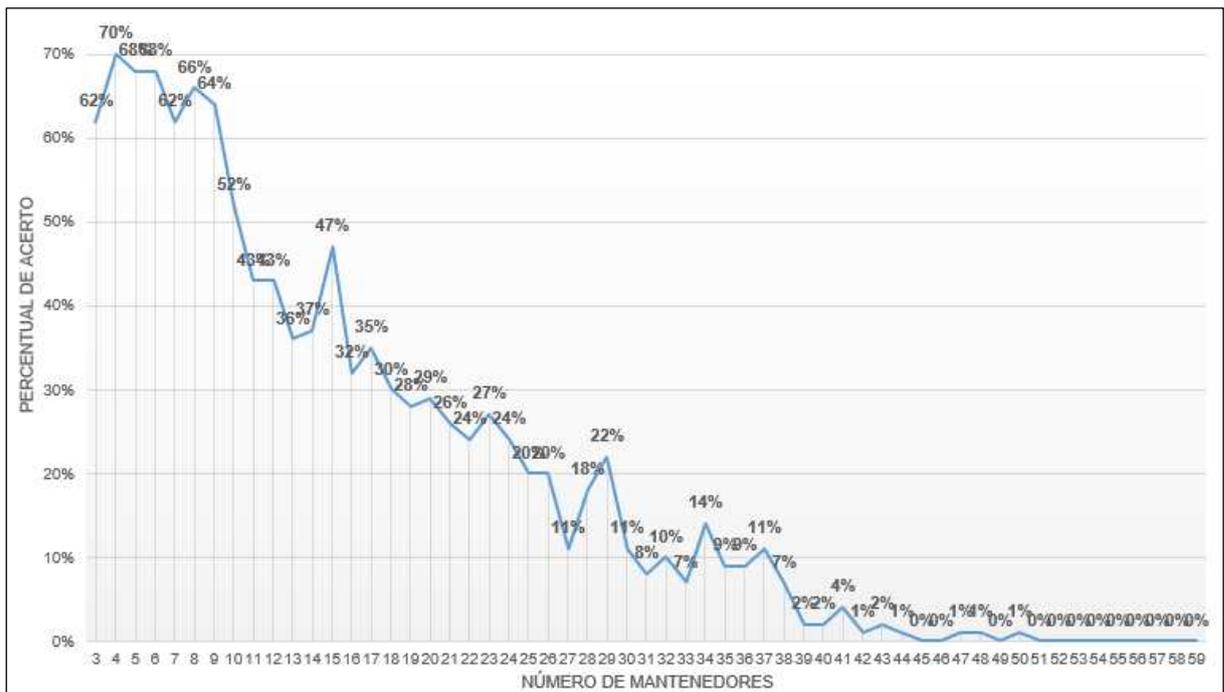
O modelo criado no ambiente de desenvolvimento Azure Machine Learning Studio realizou testes para diversos cenários com a base de dados para realizar a priorização e o ranqueamento das alternativas de mantenedores responsáveis por atender às OMs.

Como a estruturação do modelo foi desenvolvida para buscar uma alternativa elitista, vislumbrou-se a oportunidade de testá-lo repetidas vezes utilizando um número crescente de mantenedores. Para isso, foi criado um módulo em Python para realizar uma avaliação dinâmica com número variável de mantenedores para diferentes OMs, selecionadas de forma aleatória. O módulo avaliou cenários com três mantenedores até sessenta mantenedores, realizando cem iterações com diferentes OMs para cada um dos cenários.

Foi possível observar uma correlação do crescimento do número de mantenedores com a diminuição da probabilidade de o modelo encontrar uma alternativa elitista e de atender ao critério desejado do primeiro intervalo de distribuição normal, explicado anteriormente.

A implementação desse módulo exigiu desenvolvimento aprimorado do modelo para realizar a seleção em caráter randômico do número de mantenedores e de qual OM seria analisada em cada cenário. O módulo para implementação foi desenvolvido na linguagem Python e pode ser consultado na íntegra no Apêndice D. Os resultados foram tabulados e foi possível gerar o gráfico apresentado na Figura 19.

Figura 19 – Resultados do teste para diversos cenários



Fonte: Própria (2019).

É possível observar que o modelo criado utilizando MCDM é efetivo em mais da metade dos casos, para o grau de refinamento proposto, com até dez opções de mantenedores. Ainda, o algoritmo implementado realiza a seleção aleatória da OM e do número de mantenedores, com isso possui um caráter não determinístico e pode apresentar variações em seu resultado para diferentes implementações. Com tudo, é possível perceber que existe um padrão de redução de percentual de acerto conforme o aumento do número de mantenedores, mas a resposta não é linear e existem pequenos desvios para alguns cenários devido às características da avaliação realizada e ao número limitado de OMs analisadas dentre as cem iterações.

## 5.2 RESULTADOS DO ML

O modelo de ML criado foi implementado utilizando dois métodos de classificação multiclasse através do ambiente de desenvolvimento Machine Learning Studio. Os resultados para cada uma das implementações são apresentados na Figura 20 e na Figura 21, utilizando o método de floresta de decisões e o método de redes neurais, respectivamente.

Figura 20 – Resultado do algoritmo de floresta de decisões

Metrics	
Overall accuracy	0.925439
Average accuracy	0.997515
Micro-averaged precision	0.925439
Macro-averaged precision	NaN
Micro-averaged recall	0.925439
Macro-averaged recall	NaN

Fonte: Própria (2019).

Figura 21 – Resultado do algoritmo de redes neurais

Metrics	
Overall accuracy	0.912281
Average accuracy	0.997076
Micro-averaged precision	0.912281
Macro-averaged precision	NaN
Micro-averaged recall	0.912281
Macro-averaged recall	NaN

Fonte: Própria (2019).

Com a base de dados implementada, os dois métodos de classificação apresentaram resultados satisfatórios, e também muito próximos, porém com melhor desempenho no algoritmo de “Floresta de Decisões”. A implementação mostra que a problemática específica dessa situação e dessa base de dados é melhor atendida por um algoritmo que utiliza a abordagem de diversas árvores de decisão, com uma acurácia geral de 92.54%.

## 6 CONSIDERAÇÕES FINAIS

### 6.1 CONCLUSÕES

O trabalho desenvolvido possuiu dois objetivos principais: avaliar o desempenho de métodos multicritério de apoio à tomada de decisão em relação ao de algoritmos de *machine learning* sob à ótica de manutenção industrial e gerenciamento de ativos; e implementar esses métodos em plataforma comercial de *cloud computing*, buscando analisar a aplicabilidade dessas ferramentas auxiliares no processo de tomada de decisão.

Foi possível estudar diversos métodos multicritério e realizar a implementação do método TOPSIS – o mais adequado dentre os estudados – para diversos cenários de combinações de OMs e mantenedores, assim como avaliar o desempenho em uma aplicação com um algoritmo de *machine learning* em paralelo para uma determinada condição de erro proposta.

O método TOPSIS obteve resultados satisfatórios para determinadas circunstâncias específicas, sendo possível observar que, para a condição de erro proposta, existe uma diminuição expressiva da sua eficácia conforme o aumento do número de opções disponíveis para classificação e priorização. Em relação aos algoritmos de *machine learning*, os dois métodos de classificação testados apresentaram resultados satisfatórios para o cenário avaliado pelo presente trabalho, com mais de 99,7% de acurácia geral nos dois métodos.

O uso da plataforma comercial de *cloud computing* Microsoft Azure facilitou a aplicação do método multicritério, bem como do algoritmo de *machine learning*. Para adaptar o formato das informações recebidas da base de dados para os padrões de aplicação do método multicritério TOPSIS e para os algoritmos de *machine learning*, foi necessário desenvolver e implementar algoritmos na linguagem Python, que também tiveram a implementação facilitada com o uso da referida plataforma.

Ainda, vislumbrou-se a oportunidade de realizar a aplicação futura desta plataforma *cloud* através de API, otimizando o processo de funcionamento da ferramenta de auxílio à tomada de decisão e realizando comunicação do sistema desenvolvido com a base de dados e o processamento automático para a avaliação de critérios e priorização de alternativas para seleção.

Os resultados da ferramenta desenvolvida mostraram que esta não possui eficácia em uma aplicação generalista, podendo ser aprimorada e ajustada para cenários específicos através de estudo aprofundado dos processos da planta industrial, análise do critério de precisão necessário para obter resultados satisfatórios e validação das importâncias relativas de cada um dos critérios.

Por fim, a plataforma comercial utilizada no desenvolvimento do trabalho apresentou diversos artifícios e módulos que auxiliaram e facilitaram a implementação do sistema, se mostrando um excelente componente para a produção de aplicações da área de tecnologia.

## 6.2 DIFICULDADES E DESAFIOS FUTUROS

Durante a elaboração do estudo, foram encontradas algumas dificuldades e pontos que mereciam análise aprofundada para garantir o desenvolvimento de uma solução completa para auxílio à tomada de decisão, no entanto, estavam além do escopo do presente trabalho.

O processo de avaliação do erro do MCDM com método TOPSIS através de uma metodologia elitista, apresentado na página 49, mostrou que a sua eficácia é afetada conforme o aumento do número de opções. Com isso, existe campo para estudo de outros métodos para avaliação do erro e determinação da qualidade do resultado do sistema desenvolvido, possibilitando a sua aplicação independentemente do número de opções avaliadas.

Ademais, o estágio de pré-processamento para a aplicação do MCDM com método TOPSIS, apresentado na página 53, demandou a criação de uma “Matriz de Correlação” para relacionar a representatividade e importância de cada critério para a OM específica que é processada. Para o desempenho ótimo do modelo, é necessário que essa matriz possua elevada precisão ao realizar a correlação quali-quantitativa entre os critérios avaliados e OMs processadas, devendo ser desenvolvida após estudos aprofundados sobre o cenário de atuação do sistema.

Por fim, a implementação do método TOPSIS, apresentada na página 56, foi realizada utilizando-se peso unitário para os critérios avaliados. Porém, há a possibilidade de o modelo ser aprimorado com o refinamento dos valores utilizados como peso para os critérios, podendo até ser considerada a aplicação de pesos dinâmicos, conforme a circunstância.

## REFERÊNCIAS

ALLEN, Mike. **The SAGE encyclopedia of communication research methods**. London: SAGE publications, 2017. Disponível em: <<http://methods.sagepub.com/reference/the-sageencyclopedia-of-communication-research-methods>>. Acesso em: 16 out. 2019.

ALMEIDA, A. T.; BOHORIS, G. Decision theory in maintenance decision making. **Journal of Quality in Maintenance Engineering**, Bingley, v. 1, n. 1, 1995. Disponível em: <[https://www.researchgate.net/publication/235320296\\_Decision\\_theory\\_in\\_maintenance\\_decision\\_making](https://www.researchgate.net/publication/235320296_Decision_theory_in_maintenance_decision_making)>. Acesso em: 27 abr. 2019.

ALVES, Robson de P.; FALSARELLA, Orandi M. Modelo conceitual de inteligência organizacional aplicada à manutenção. **Gestão da Produção**, São Carlos, v. 16, n. 2, 2009. Disponível em: <<http://ref.scielo.org/86rdtq>>. Acesso em: 27 abr. 2019.

ARAYA, Marcela C. G.; CARIGNANO, Claudia; GOMES, Luiz F. A. M. **Tomada de decisões em cenários complexos**. São Paulo: Pioneira Thomson Learning, 2004.

ARMBRUST, Michael. et al. Above the clouds: a berkeley view of cloud computing. **Berkeley**, Berkeley, 2009. Disponível em: <<https://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>>. Acesso em: 10 abr. 2019.

Associação Brasileira de Manutenção – ABRAMAN. Documento Nacional. Rio de Janeiro: ABRAMAN, 2017. Disponível em: <<https://docplayer.com.br/105100568-Resultado-do-documento-nacional-2017.html>>. Acesso em 25 fev. 2019.

BĂBAN, Marius; FLORIN BĂBAN, Călin; ŞUTEU, Marius Darius. Maintenance decision-making support for textile machines: a knowledge-based approach using fuzzy logic and vibration monitoring. **IEEE Access**, Adelaide, 2019. Disponível em: <<https://ieeexplore.ieee.org/document/8740857>>. Acesso em 29 abr. 2019.

BARAN, Leandro R. **Proposta de um modelo multicritério para determinação da criticidade na gestão da manutenção industrial**. 2015. Dissertação (Mestrado em Engenharia de Produção) – Universidade Tecnológica Federal do Paraná, 2015. Disponível em: <<http://repositorio.utfpr.edu.br/jspui/handle/1/1450>>. Acesso em: 18 mar. 2019.

BEGHI, Alessandro. et al. Machine learning for predictive maintenance: a multiple classifiers approach. **IEEE Transactions on industrial informatics**, Taipei, v. 11, 2015. Disponível em: <<https://ieeexplore.ieee.org/document/6879441>>. Acesso em: 08 mar. 2019.

BENGIO, Yoshua.; COURVILLE, Aaron.; GOODFELLOW, Ian. **Deep learning**. Cambridge: The MIT Press, 2016.

BIRD, Steven; KLEIN, Ewan; LOPER, Edward. **Natural language processing with Python**: analyzing text with the natural language toolkit. Boston: O'Reilly Media, 2009.

BONISSONE, Piero P.; LIZZI, John; SUBBU, Raj. Multicriteria Decision Making (MCDM): a framework for research and applications. **IEEE Computational Intelligence Magazine**, New Jersey, v. 4, 2009. Disponível em: <<https://ieeexplore.ieee.org/document/5190938>>. Acesso em: 17 mar. 2019.

BORLIDO, David José Araújo. **Indústria 4.0**: aplicação a sistemas de manutenção. 2017. Dissertação (Mestrado em Engenharia Mecânica) – Universidade do Porto. Porto, 2017. Disponível em: <<https://repositorio-aberto.up.pt/bitstream/10216/102740/2/181981.pdf>>. Acesso em: 26 fev. 2019.

BRANS, Jean-Pierre; MARESCHAL, Bertrand. **Multiple criteria decision analysis**: state of the art surveys. New York: Springer, 2005. Disponível em: <[https://www.researchgate.net/publication/287688423\\_Chapter\\_5\\_PROMETHEE\\_methods](https://www.researchgate.net/publication/287688423_Chapter_5_PROMETHEE_methods)>. Acesso em: 27 abr. 2019.

BRANS, Jean-Pierre; VINCKE, Philippe. Management Science. **A preference ranking organisation method**: the PROMETHEE method for multiple criteria, Catonsville, v. 31, n. 6, jun. 1985. Informs. Disponível em: <<https://pdfs.semanticscholar.org/edd6/f5ae9c1bfb2fdd5c9a5d66e56bdb22770460.pdf>>. Acesso em 29 abr. 2019.

BUYA, Rajkumar; CALHEIROS, Rodrigo. N; RANJAN, Rajiv. Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: challenges and opportunities. In: INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE COMPUTING & SIMULATION, 2009, Leipzig. **Anais...** Leipzig: IEEEExplore, 2009. Disponível em: <<https://ieeexplore.ieee.org/document/5192685>>. Acesso em: 04 abr. 2019.

Confederação Nacional da Indústria – CNI. Desafios para Indústria 4.0 no Brasil. Brasília: CNI, 2016. Disponível em: <<http://www.portaldaindustria.com.br/publicacoes/2016/8/desafios-para-industria-40-no-brasil/#>>. Acesso em: 03 out. 2018.

Confederação Nacional da Indústria – CNI. Mapa Estratégico da Indústria 2018-2022. Brasília: CNI, 2018. Disponível em: <<http://www.portaldaindustria.com.br/cni/canais/mapa-estrategico-da-industria/downloads/>>. Acesso em: 02 mar. 2019.

COSTA, Leandro S. da; DUARTE JUNIOR, Antônio M. Uma metodologia para a pré-seleção de ações utilizando o método multicritério TOPSIS. In: XLV SIMPÓSIO BRASILEIRO DE PESQUISA, 2013, Natal. **Anais...** Natal: SBPO, 2013. Disponível em: <<http://www.din.uem.br/sbpo/sbpo2013/pdf/arq0123.pdf>>. Acesso em: 26 abr. 2019.

DUNN, Richard L. **Predictive Maintenance Technologies**. Plant Engineering, 2002. Disponível em: <<https://www.plantengineering.com/articles/predictive-maintenance-technologies/>>. Acesso em: 15 mar. 2019.

DUNN, Sandy. **The Fourth Generation of Maintenance**. Perth, 2016. Disponível em: <[http://www.plant-maintenance.com/maintenance\\_articles.shtml](http://www.plant-maintenance.com/maintenance_articles.shtml)>. Acesso em: 05 out. 2018.

Federação das Indústrias do Estado do Paraná - FIEP. **Observatórios SESI/SENAI/IEL: Setor Metal Mecânico**. Curitiba, 2018. Disponível em: <<http://www.fiepr.org.br/observatorios/metal-mecanico/>>. Acesso em: 20 out. 2018.

FERNÁNDEZ, Santiago. et al. Industry 4.0: predictive intelligent maintenance for production equipment. In: EUROPEAN CONFERENCE OF THE PROGNOSTICS AND HEALTH MANAGEMENT SOCIETY, 3., 2016, Bilbao. **Anais...** Bilbao: PHM Society, 2016. Disponível em: <<http://www.phmsociety.org/node/2058/>>. Acesso em 15 mar. 2019.

FUENTES, Fernando F. E. **Metodologia para inovação da gestão da manutenção industrial**. 2006. Tese (Doutorado em Engenharia Mecânica) – Universidade Federal de Santa Catarina, 2006. Disponível em: <<https://core.ac.uk/download/pdf/30369953.pdf>>. Acesso em: 25 abr. 2019.

GORSKI, Ewerton G. **Smart Workflow no gerenciamento de manutenção com base em machine learning e métodos de apoio a tomada de decisão**. 2019. Dissertação (Mestrado em Engenharia de Produção e Sistemas) – Pontifícia Universidade Católica do Paraná, 2019.

JENNINGS, Roger. **Cloud computing: with the windows Azure Platform**. New Jersey: John Wiley & Sons, 2009.

LIAO, Yongxin. et al. The Impact of the Fourth Industrial Revolution: a cross-country/region comparison. **Production**, São Paulo, v. 28, 2018. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0103-65132018000100401&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-65132018000100401&lng=en&nrm=iso)>. Acesso em: 10 out. 2018.

MACHADO, Waltair V.; OTANI, Mario. A proposta de desenvolvimento de gestão da manutenção industrial na busca da excelência ou classe mundial. **Gestão Industrial**, Ponta Grossa, v. 04, 2008. Disponível em: <<http://www.mantenimentomundial.com/notas/proposta.pdf>>. Acesso em: 10 abr. 2019.

MARTINS, E. G. Desvio Padrão Amostral. **Revista de Ciência Elementar**, n. 1, v. 1, 2013. Universidade do Porto, Portugal, 2013.

MAZZAFERRO, José A. E. Indústria 4.0 e a Qualidade da Informação. **Soldagem & Inspeção**, São Paulo, v. 23, n. 1, jan./mar. 2018. Editorial. Disponível em: <<https://doaj.org/article/ac6392ac5f7445459919057d5cad7357?frbrVersion=2>>. Acesso em: 05 out. 2018.

MCKINNEY, Wes. **Phyton for data analysis: data wrangling with Pandas, Numpy and IPython**. Boston: O'Reilly Media, 2017.

MCKINSEY. **Industry 4.0: how to navigate digitization of the manufacturing sector**. New York: McKinsey & Company Digital, 2015. Disponível em: <<https://www.santossey.com/business-functions/operations/our-insights/industry-four-point-o-how-to-navigae-the-digitization-of-the-manufacturing-sector>>. Acesso em: 12 out. 2018.

MICROSOFT. **Azure**. Disponível em: <<https://azure.microsoft.com/pt-br/overview/>>. Acesso em: 18 abr. 2019.

\_\_\_\_\_. **Machine learning algorithm cheat sheet for Azure Machine Learning Studio**. Disponível em: <<https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-cheat-sheet>>. Acesso em: 23 abr. 2019.

\_\_\_\_\_. **Machine Learning Studio**. Disponível em: <<https://azure.microsoft.com/pt-br/services/machine-learning-studio/>>. Acesso em: 23 abr. 2019.

\_\_\_\_\_. **O que é PaaS?**. Disponível em: <<https://azure.microsoft.com/pt-r/overview/>>. Acesso em: 18 abr. 2019.

MITCHELL, Thomas. **Machine learning**. New York: McGraw-Hill, 1997.

MORAES, Paulo H. de A. **Manutenção produtiva total: estudo de caso em uma empresa automobilística**. 2004. Dissertação (Mestrado em Gestão e Desenvolvimento Regional) – Universidade de Taubaté. Taubaté, 2004. Disponível em: <[http://www.ppga.com.br/mestrado/2003/moraes-paulo\\_henrique\\_de\\_almeida.pdf](http://www.ppga.com.br/mestrado/2003/moraes-paulo_henrique_de_almeida.pdf)>. Acesso em: 01 mar. 2019.

MORETO, Miguel. **Localização de faltas de alta impedância em sistemas de distribuição de energia: uma metodologia baseada em redes neurais artificiais**. 2005. 126 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Rio Grande do Sul. Porto Alegre, 2005. Disponível em: <<https://lume.ufrgs.br/handle/10183/7602>>. Acesso em 10 mar. 2019.

MOTTER, Osir. **Manutenção industrial: o poder oculto na empresa**. São Paulo: Hemus, 1992.

MOUBRAY, John. **Reliability-centered maintenance**. New York: Industrial Press, 1997.

MUNIER, Nolberto. **A strategy for using multicriteria analysis in decision-making**. New York: Springer, 2011.

NASCIF, Júlio. **Manutenção classe mundial**. Tecem Tecnologia Empresarial. 2016. Disponível em: <<ftp://ftp.ufv.br/dta/disciplinas/tal420/2002/MANUTEN%C7%C3O/14-Manuten%E7%E3o%20Classe%20Mundial.doc>>. Acesso em: 29 abr. 2019.

National Institute of Standards and Technology – NIST. Programa de Cloud Computing. Gaithersburg: NIST, 2019. Disponível: <<https://www.nist.gov/programs-projects/nist-cloud-computing-program-nccp>>. Acesso em: 29 mar. 2019.

PINTO, Alan Kardec. **Gestão estratégica e técnicas preditivas**. Rio de Janeiro: ABRAMAN, 2002.

ROCCO, Mauricio de. **Planejamento industrial**: um estudo sobre tomada de decisão em manutenção. 2012. Monografia (Especialização em Gestão Industrial) – Universidade Tecnológica Federal do Paraná, 2012. Disponível em: <<http://repositorio.roca.utfpr.edu.br/jspui/handle/1/8260>>. Acesso em: 27 mar. 2019.

SAATY, Thomas L. Decision making with the analytic hierarchy process. **International Journal of Services Sciences**, Pittsburgh, v. 01, 2008. Disponível em: <[https://www.researchgate.net/publication/228628807\\_Decision\\_making\\_with\\_the\\_Analytic\\_Hierarchy\\_Process](https://www.researchgate.net/publication/228628807_Decision_making_with_the_Analytic_Hierarchy_Process)>. Acesso em: 25 mar. 2019.

SAATY, Thomas L. **Theory and applications of the analytic network process**: decision making with benefits, opportunities, costs and risks. Pittsburgh: RWS Publications, 2005. Disponível em: <[https://www.researchgate.net/publication/226556079\\_The\\_Analytic\\_Network\\_Process](https://www.researchgate.net/publication/226556079_The_Analytic_Network_Process)>. Acesso em: 10 mar. 2019.

SANTOS, Antônio C. de Q. **Abordagem multicritério para classificação de equipamentos críticos e determinação de tempos de inspeções para manutenção**. 2015. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Pernambuco. Recife, 2015. Disponível em: <<https://repositorio.ufpe.br/handle/123456789/13838>>. Acesso em: 04 out. 2018.

SHIH, Hsu-Shih; SHYUR, Huan-Jyh; LEE, E. Stanley. An Extension of TOPSIS for Group Decision Making. **Mathematical and Computer Modelling**, v. 45, p. 801-813, 2007. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0895717706003025>>. Acesso em: 27 set. 2019.

SLACK, Nigel. et al. **Administração da produção**. São Paulo: Atlas, 2006.

VARGAS, Ricardo. Using the analytic hierarchy process (AHP) to select and prioritize projects in a portfolio. In: PMI GLOBAL CONGRESS, 2010, Washington D.C. **Anais...**

Washington D.C.: Soft Expert, 2010. Disponível em: <<http://www.urisan.tche.br/~pbetencourt/engsoftIII/SPPGP.pdf>>. Acesso em: 18 mar. 2019.

XENOS, Harilaus G. P. **Gerenciando a manutenção produtiva**. Belo Horizonte: EDG – Editora de Desenvolvimento Gerencial, 1998.

## APÊNDICES

### A. ALGORITMO EM PYTHON PARA PRÉ-PROCESSAMENTO DOS DADOS NA IMPLEMENTAÇÃO DO MCDM COM MÉTODO TOPSIS

```

import pandas as pd
import math

# The entry point function can contain up to two input arguments:
# Param<dataframe1>: a pandas.DataFrame
# Param<dataframe2>: a pandas.DataFrame
def azureml_main(dataframe1 = None, dataframe2 = None):

    # Execution logic goes here
    cab_b = list(dataframe1.columns)
    cab_s = list(dataframe2.columns)
    dictionary2 = {}

    #for each in cab:
    for i in range(0, 5):
        j = i*2
        limp1 = [x for x in dataframe1[cab_b[j]].values if str(x) != 'nan']
        limp2 = [x for x in dataframe1[cab_b[j+1]].values if str(x) != 'nan']
        dictionary = dict(zip(limp1, limp2))
        dictionary2[cab_b[j]] = dictionary

    resp = dataframe2.copy()
    resp.drop('employee_id', axis=1)
    print(dictionary2)
    maxi = dataframe2['admission_date'].max()
    for index, row in dataframe2.iterrows():
        for each in cab_s:
            print(each)
            print(row[each])
            if each in cab_b:
                resp[each][index]= dictionary2[each][row[each]]
            if each == 'admission_date':
                resp[each][index]= (dataframe2[each][index]/maxi)*10
    #teste = dictionary2[cab[0]][datas[cab[0]]]
    # If a zip file is connected to the third input port is connected,
    # it is unzipped under ".\Script Bundle". This directory is added
    # to sys.path. Therefore, if your zip file contains a Python file
    # mymodule.py you can import it using:
    # import mymodule
    resp.index =dataframe2['employee_id']
    resp = resp.drop('employee_id', axis=1)
    # Return value must be of a sequence of pandas.DataFrame
    respT = resp.T
    return respT,

```

## B. ALGORITMO EM PYTHON PARA IMPLEMENTAÇÃO DO MCDM COM MÉTODO TOPSIS

```

import pandas as pd
import math

# The entry point function can contain up to two input arguments:
# Param<dataframe1>: a pandas.DataFrame
# Param<dataframe2>: a pandas.DataFrame
def azureml_main(dataframe1 = None, dataframe2 = None):

    # Execution logic goes here
    cab = list(dataframe1.columns)
    norm = []
    Maior = []
    Menor = []
    SMAior = []
    SMenor = []
    pesos = dataframe2['pesos'].tolist()

#Normalizadores
for index, row in dataframe1.iterrows():
    ac = 0
    for each in cab:
        ac= (row[each]*row[each])+ac

    norm.append(math.sqrt(ac))

for index, row in dataframe1.iterrows():
    dataframe1.loc[index, cab] = (row[:]/norm[index])*pesos[index]
    m1 = dataframe1.loc[index, cab].max()
    m2 = dataframe1.loc[index, cab].min()
    Maior.append(m1)
    Menor.append(m2)

df3 = dataframe1.copy() #df3 -> valor da matriz adaptada maior
df4 = dataframe1.copy() #df4 -> valor da matriz adaptada maior

for index, row in dataframe1.iterrows():
    df3.loc[index, cab] = (row[:]-Maior[index])**2
    df4.loc[index, cab] = (row[:]-Menor[index])**2

SMAior = df3.sum(axis=0).apply(lambda x:math.sqrt(x))
SMenor = df4.sum(axis=0).apply(lambda x:math.sqrt(x))

val = []
cabs = pd.Series(cab)

for each in cab:
    val.append(SMAior[each]/(SMenor[each]+SMAior[each]))

vals = pd.Series(val)
dic = {'Alternativa':cabs, 'Valor':vals}
SResult = pd.DataFrame(dic)

print(val)
print("-----")
print(vals)
print("-----/n")

```

```

print(SResult)
# If a zip file is connected to the third input port is connected,
# it is unzipped under ".\Script Bundle". This directory is added
# to sys.path. Therefore, if your zip file contains a Python file
# mymodule.py you can import it using:
# import mymodule

# Return value must be of a sequence of pandas.DataFrame
return SResult,

```

### C. ALGORITMO EM PYTHON PARA AVALIAÇÃO DO ERRO DO MCDM COM MÉTODO TOPSIS

```

import pandas as pd
import math
import numpy

# The entry point function can contain up to two input arguments:
# Param<dataframe1>: a pandas.DataFrame
# Param<dataframe2>: a pandas.DataFrame
def azureml_main(dataframe1 = None):

    # Execution logic goes here
    dataframe1.sort_values(["Valor"], axis=0,
                           ascending=False, inplace=True)
    valores = dataframe1["Valor"].tolist()
    Valores_ar = numpy.array(valores)
    med = numpy.mean(Valores_ar, axis=0)
    desv = numpy.std(Valores_ar, axis=0)
    li = [0,0,0,0,0,0,0]
    for each in valores[1:]:
        if each >= med:
            if each == med:
                li[3] = li[3]+1
            elif each < (med + desv):
                li[4] = li[4]+1
            elif each < (med + 2 * desv):
                li[5] = li[5]+1
            elif each < (med + 3 * desv):
                li[6] = li[6]+1

    erro = 0
    if valores[0] >= med:
        if valores[0] == med:
            if li[3] > 0:
                erro = 1
        elif valores[0] < (med + desv):
            if li[4] > 0:
                erro = 1
        elif valores[0] < (med + 2 * desv):
            if li[5] > 0:
                erro = 1
        elif valores[0] < (med + 3 * desv):
            if li[6] > 0:
                erro = 1

    #numpy.array([(med-2*desv), (med-desv), med, (med+desv), (med+2*desv)])
    dic = {'Media':[med, desv, erro]}
    SResult = pd.DataFrame(dic)

```

```

# If a zip file is connected to the third input port is connected,
# it is unzipped under ".\Script Bundle". This directory is added
# to sys.path. Therefore, if your zip file contains a Python file
# mymodule.py you can import it using:
# import mymodule

# Return value must be of a sequence of pandas.DataFrame
return SResult,

```

#### D. ALGORITMO EM PYTHON PARA IMPLEMENTAÇÃO DO MODELO MCDM COM MÉTODO TOPSIS PARA CENÁRIOS DIVERSOS COM SELEÇÃO ALEATÓRIA DA OM

```

import pandas as pd
import math
import numpy
import random

def topsis(dataframe1, dataframe2):
    cab = list(dataframe1.columns)
    norm = []
    Maior = []
    Menor = []
    SMAior = []
    SMenor = []
    pesos = dataframe2['pesos'].tolist()

    # Normalizadores
    for index, row in dataframe1.iterrows():
        ac = 0
        for each in cab:
            ac = (row[each] * row[each]) + ac

        norm.append(math.sqrt(ac))

    for index, row in dataframe1.iterrows():
        dataframe1.loc[index, cab] = (row[:] / norm[index]) * pesos[index]
        m1 = dataframe1.loc[index, cab].max()
        m2 = dataframe1.loc[index, cab].min()
        Maior.append(m1)
        Menor.append(m2)

    df3 = dataframe1.copy() # df3 -> valor da matriz adaptada maior
    df4 = dataframe1.copy() # df4 -> valor da matriz adaptada maior

    for index, row in dataframe1.iterrows():
        df3.loc[index, cab] = (row[:] - Maior[index]) ** 2
        df4.loc[index, cab] = (row[:] - Menor[index]) ** 2

    SMAior = df3.sum(axis=0).apply(lambda x: math.sqrt(x))
    SMenor = df4.sum(axis=0).apply(lambda x: math.sqrt(x))

    val = []
    cabs = pd.Series(cab)

    for each in cab:
        val.append(SMAior[each] / (SMenor[each] + SMAior[each]))

```

```

vals = pd.Series(val)
dic = {'Alternativa': cabs, 'Valor': vals}
SResult = pd.DataFrame(dic)
SResult.sort_values(["Valor"], axis=0, ascending=False, inplace=True)
valores = SResult['Valor'].tolist()
Valores_ar = numpy.array(valores)
med = numpy.mean(Valores_ar, axis=0)
desv = numpy.std(Valores_ar, axis=0)
li = [0, 0, 0, 0, 0, 0, 0]
for each in valores[1:]:
    if each >= med:
        if each == med:
            li[3] = li[3] + 1
        elif each < (med + desv):
            li[4] = li[4] + 1
        elif each < (med + 2 * desv):
            li[5] = li[5] + 1
        elif each < (med + 3 * desv):
            li[6] = li[6] + 1
erro = 0
if valores[0] >= med:
    if valores[0] == med:
        if li[3] > 0:
            erro = 1
    elif valores[0] < (med + desv):
        if li[4] > 0:
            erro = 1
    elif valores[0] < (med + 2 * desv):
        if li[5] > 0:
            erro = 1
    elif valores[0] < (med + 3 * desv):
        if li[6] > 0:
            erro = 1
return erro,

```

```

def erro(dataframe1, dataframe2):
    dataframe1.sort_values(["Valor"], axis=0, ascending=False, inplace=True)
    valores = dataframe1["Valor"].tolist()
    Valores_ar = numpy.array(valores)
    med = numpy.mean(Valores_ar, axis=0)
    desv = numpy.std(Valores_ar, axis=0)
    li = [0, 0, 0, 0, 0, 0, 0]
    for each in valores[1:]:
        if each >= med:
            if each == med:
                li[3] = li[3] + 1
            elif each < (med + desv):
                li[4] = li[4] + 1
            elif each < (med + 2 * desv):
                li[5] = li[5] + 1
            elif each < (med + 3 * desv):
                li[6] = li[6] + 1
    erro = 0
    if valores[0] >= med:
        if valores[0] == med:
            if li[3] > 0:
                erro = 1
        elif valores[0] < (med + desv):
            if li[4] > 0:

```

```

        erro = 1
    elif valores[0] < (med + 2 * desv):
        if li[5] > 0:
            erro = 1
    elif valores[0] < (med + 3 * desv):
        if li[6] > 0:
            erro = 1
    return erro,

def pre_proc(dataframe1, dataframe2, dataframe3):
    cab_b = list(dataframe1.columns)
    cab_s = list(dataframe2.columns)
    dictionary2 = {}

    # for each in cab:
    for i in range(0, 5):
        j = i * 2
        limp1 = [x for x in dataframe1[cab_b[j]].values if str(x) != 'nan']
        limp2 = [x for x in dataframe1[cab_b[j + 1]].values if str(x) != 'nan']
        dictionary = dict(zip(limp1, limp2))
        dictionary2[cab_b[j]] = dictionary

    resp = dataframe2.copy()
    resp.drop('employee_id', axis=1)
    maxi = dataframe2['admission_date'].max()
    for index, row in dataframe2.iterrows():
        for each in cab_s:
            if each in cab_b:
                resp[each][index] = dictionary2[each][row[each]]
            if each == 'admission_date':
                resp[each][index] = (dataframe2[each][index] / maxi) * 10
    # resp.index = dataframe2['employee_id']
    resp = resp.drop('employee_id', axis=1)
    resp.columns = list(range(0, 6))
    respT = resp.T
    respT.reset_index(drop=True)
    return respT

dataframe1 = pd.read_csv('Pre_Processamento.csv', sep=',', encoding='latin-1', decimal='.')
dataframe2 = pd.read_csv('Sample_Completo.csv', sep=',', encoding='latin-1', decimal='.')
pesos = pd.read_csv('Pesos.csv', sep=',', encoding='latin-1', decimal='.')
cab_b = list(dataframe1.columns)
cab_s = list(dataframe2.columns)
empl = list(dataframe2.index)
dictionary2 = {}

iter = list(range(3, 60))

#for each in iter:
datacopy = pd.DataFrame()
datacopy_2 = pd.DataFrame()
#predf.to_csv(r'C:\Users\Adonis\PycharmProjects\TCC\venv\pre.csv', index = None, header=True)
li = []

for i in list(range(3, 60)):
    rep = 100
    val = 0
    datacopy = dataframe1.copy()
    columns = ["Eq 1", "Eq 2", "Eq 3", "Eq 4", "Eq 5"]
    for index, row in dataframe1.iterrows():

```

```
for each in columns:
    if datacopy.loc[index][each] != "NaN":
        b = datacopy.loc[index][each]
        datacopy.set_value(index, each, random.randint(1, 10))
        c = datacopy.loc[index][each]

for j in list(range(0, rep)):
    dataframe2 = dataframe2.sample(frac=1)
    datacopy_2 = datacopy_2.iloc[0:i]
    pre = pd.DataFrame(pre_proc(dataframe1, datacopy_2, pesos))
    mcdm = topsis(pre, pesos)
    res = int(mcdm[0])
    if res == 0:
        val = val+1
    li.append([i, val])

print(li)
dic = {}
for each in li:
    dic.__setitem__(each[0], each[1])

print(dic)
outp = pd.DataFrame([dic])
print(outp)
outp.to_csv(r'C:\Users\Adonis\PycharmProjects\TCC\venv\Resultado.csv')
```