

U. PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

**Aplicação de conceitos de manutenção
preditiva com aplicação de ferramentas de
Inteligência Artificial**

Diogo Emanuel Rocha Cardoso

Dissertação submetida a
Faculdade de Engenharia da Universidade do Porto
para o grau de:

Mestre em Engenharia Mecânica

Orientador na FEUP:
Prof. Luís Andrade Ferreira

Mestrado Integrado em Engenharia Mecânica
Faculdade de Engenharia da Universidade do Porto

Porto, Julho 2020

Faculdade de Engenharia da Universidade do Porto

Diogo Cardoso (up201505446@fe.up.pt)

Resumo

A presente dissertação foi desenvolvida no âmbito do Mestrado Integrado em Engenharia Mecânica, ramo de Projeto e Construção Mecânica, da Faculdade de Engenharia da Universidade do Porto.

A crescente competitividade do mercado aliada ao aumento da automatização impulsionado pelo advento da Indústria 4.0 põe em destaque a importância da manutenção no seio das organizações. Simultaneamente, a quantidade de dados capazes de ser extraídos de sistemas industriais aumentou de forma exponencial devido à proliferação de sensores, dispositivos de transmissão e armazenamento de dados via *Internet of Things*. Estes dados, quando processados e analisados, podem fornecer informações e conhecimentos valiosos acerca dos equipamentos, permitindo uma movimentação no sentido de uma manutenção preditiva.

A manutenção é fundamental para a competitividade de uma empresa, uma vez que as ações realizadas a esse nível têm impacto direto em aspetos como o custo e qualidade dos produtos. Assim, falhas nos equipamentos necessitam de ser identificadas e resolvidas. As ferramentas de Inteligência Artificial, em particular o *Machine Learning*, exibem um enorme potencial na análise de grandes quantidades de dados, agora prontamente disponíveis, permitindo assim ambicionar uma melhoria na disponibilidade dos sistemas, reduzindo os custos de manutenção, aumentando o desempenho operacional e ainda a capacidade de apoio na tomada de decisão. Na presente dissertação, aplicam-se ferramentas de Inteligência Artificial, mais especificamente de *Machine Learning*, a um conjunto de dados disponibilizados *online* e procura-se analisar as especificidades desta implementação assim como a definição de metodologias, com vista a permitir fornecer informações e ferramentas à área da manutenção.

Palavras-chave: Manutenção Preditiva, Indústria 4.0, *Internet of Things*, Inteligência Artificial, *Machine Learning*

Abstract

The present dissertation was developed in the context of the Integrated Master's Degree in Mechanical Engineering, specialization in Project and Mechanical Construction, in the Faculty of Engineering of the University of Porto.

The growing competitiveness of the market, coupled with the increase in automation driven by the advent of the Industry 4.0, highlights the importance of maintenance within organizations. At the same time, the amount of data capable of being extracted from industrial systems has increased exponentially due to the proliferation of sensors, transmission devices and data storage via Internet of Things. These data, when processed and analyzed, can provide valuable information and knowledge about the equipment, allowing a move towards predictive maintenance.

Maintenance is fundamental to a company's competitiveness, since actions taken at this level have a direct impact on aspects such as cost and quality of products. Hence, equipment failures need to be identified and resolved. Artificial Intelligence tools, in particular Machine Learning, exhibit enormous potential in the analysis of large amounts of data, now readily available, thus aiming to improve the availability of systems, reducing maintenance costs, increasing operational performance and support in decision making. In this dissertation, Artificial Intelligence tools, more specifically Machine Learning, are applied to a set of data made available online and the specifics of this implementation are analyzed as well as the definition of methodologies, in order to provide information and tools to the maintenance area.

Keywords: *Predictive Maintenance, Industry 4.0, Internet of Things, Artificial intelligence, Machine Learning*

Data is the new Oil. It is only useful when refined.

Agradecimentos

Em primeiro lugar, gostaria de agradecer ao meu orientador, o Professor Luís Andrade Ferreira pelo apoio, orientação e disponibilidade ao longo deste percurso.

Não poderia deixar de agradecer à minha família, amigos e colegas de curso, pelo apoio incondicional ao longo destes anos.

Por fim, gostaria de agradecer a todas as pessoas que direta ou indiretamente tornaram possível a realização deste trabalho e me ajudaram a dar um passo importante na caminhada da minha formação pessoal e académica.

Conteúdo

Lista de Figuras	xv
Lista de Tabelas	xviii
Lista de acrónimos e siglas	xix
1 Introdução	1
1.1 Enquadramento	1
1.2 Objetivos	3
1.3 Estrutura da dissertação	3
2 Estado da Arte	5
2.1 Introdução	5
2.2 Definição de Manutenção	5
2.3 Tipos de Manutenção	7
2.3.1 Manutenção Corretiva (<i>Run-to-failure</i>)	7
2.3.2 Manutenção Preventiva	8
2.3.3 Manutenção Preditiva	9
2.3.4 Manutenção Prescritiva	12
2.4 Falha, Avaria e Modo de Falha	12
2.5 Normas Portuguesas da Manutenção	13
2.6 Indicadores de desempenho da manutenção	14
2.6.1 Manutibilidade	14

2.6.2	Fiabilidade e Disponibilidade	14
2.7	Gestão da Manutenção	17
2.8	Custo do Ciclo de Vida	18
3	Indústria 4.0 e Inteligência Artificial	21
3.1	Introdução	21
3.2	Revoluções Industriais	21
3.3	Indústria 4.0	22
3.4	Inteligência Artificial	24
3.5	Breve distinção entre Inteligência Artificial, <i>Machine Learning</i> e <i>Deep Learning</i>	25
3.6	Impacto e Desafios da Indústria 4.0 na Manutenção	26
4	<i>Machine Learning</i>	29
4.1	O que é <i>Machine Learning</i> ?	29
4.1.1	Porque usar <i>Machine Learning</i> ?	29
4.2	Tipos de <i>Machine Learning</i>	30
4.2.1	<i>Supervised, unsupervised, reinforcement learning</i> e <i>Semisupervised learning</i>	30
4.2.2	<i>Batch</i> e <i>online learning</i>	33
4.2.3	<i>Instance-based</i> e <i>model-based learning</i>	34
4.3	<i>Workflow</i> de um projeto de <i>Machine Learning</i>	34
4.4	Treino e métricas de avaliação	37
4.5	Métodos de <i>Machine Learning</i> para manutenção preditiva	41
4.5.1	Revisão sistemática da literatura científica	42
5	Aplicação de modelos de <i>Machine Learning</i>	45
5.1	Introdução	45
5.2	Origem dos Dados	46

5.2.1	Telemetria em tempo real (<i>Real-time telemetry</i>)	47
5.2.2	Registo de Erros	49
5.2.3	Histórico de Manutenção	50
5.2.4	Histórico de Falhas	51
5.2.5	Informação acerca das máquinas	52
5.3	Definição dos objetivos	53
5.4	<i>Feature Engineering</i>	53
5.4.1	<i>Lag Features</i>	54
5.4.2	<i>Features</i> adicionais	55
5.5	Seleção de <i>Features</i>	56
5.6	Classificação dos dados e construção de <i>Labels</i>	59
5.7	Divisão dos dados	61
5.7.1	Desequilíbrio entre classes em aplicações de problemas de manutenção	61
5.8	Aplicação dos modelos no conjunto de desenvolvimento	63
5.8.1	<i>K-Nearest Neighbors</i>	63
5.8.2	<i>Decision Tree</i>	65
5.8.3	<i>Random Forest</i>	69
5.8.4	<i>Naive Bayes</i>	70
5.8.5	<i>Artificial Neural Networks</i>	70
5.9	Comportamento no Conjunto de Teste	72
5.10	Conclusões	74
6	Conclusões e trabalhos futuros	75
6.1	Conclusões	75
6.2	Trabalhos Futuros	77

Lista de Figuras

2.1	Estratégias de Manutenção	7
2.2	Intervalo de Tempo “P-F” entre a detecção do início da falha (P) e o momento em que existe perda da função (F) [19]	10
2.3	Visão geral das diferentes estratégias de manutenção [16]	11
2.4	Curva da Banheira [23]	15
2.5	Principais Funções da Gestão da Manutenção e as suas Relações [10] .	17
2.6	Comparação do Custo do Ciclo de Vida para dois equipamentos com custo inicial distintos [23]	18
2.7	Iceberg dos Custos da Manutenção [11]	19
3.1	Cronologia das Revoluções Industriais [29]	22
3.2	Hierarquia DIKW - “Data-Information-Knowledge-Wisdom” [30]	24
3.3	Distinção entre Inteligência Artificial, <i>Machine Learning</i> e <i>Deep Learning</i> [33]	26
4.1	<i>Supervised Learning</i> : Classificação e Regressão [6]	31
4.2	<i>Unsupervised Learning</i> [6]	32
4.3	<i>Semisupervised learning</i> [6]	33
4.4	Divisão baseada na capacidade dos algoritmos aprenderem de forma incremental [35]	33
4.5	Divisão baseada na generalização para diferentes modos de aprendizagem [35]	34
4.6	Proposta de <i>workflow</i> de um projeto de <i>Machine Learning</i> [36]	36

4.7	Exemplo de <i>underfitting</i> e <i>overfitting</i> [38]	37
4.8	Ilustração gráfica de <i>Bias</i> e <i>Variance</i> [37]	38
4.9	Técnica de validação <i>Holdout</i> [39]	39
4.10	Número de publicações científicas, por ano, de acordo com os critérios estabelecidos [15]	44
5.1	Evolução dos dados de Telemetria ao longo dos primeiros quinze dias do mês de Janeiro de 2015, para a máquina 1	48
5.2	Representação do número de erros por tipo	49
5.3	Representação do número de componentes substituídos, por tipo	50
5.4	Representação do número de componentes substituídos, por tipo, devido à ocorrência de uma falha	51
5.5	Histograma representativo do número de máquinas e tempo de serviço, por modelo	52
5.6	<i>Feature Engineering</i> [65]	54
5.7	<i>Lag Features</i> [7]	55
5.8	Correlação entre <i>Features</i>	57
5.9	Análise detalhada da correlação entre as <i>features pressuremean_3h</i> e <i>pressuremean_24h</i>	58
5.10	Classificação dos dados e construção de <i>Labels</i> - Adaptado de [7]	59
5.11	Exemplo de <i>K-Nearest Neighbors</i> , com “ <i>n-neighbors</i> ”= 1	63
5.12	Análise da evolução da <i>Accuracy</i> no conjunto de treino e no conjunto de desenvolvimento, para <i>K-Nearest Neighbors</i> com diferentes valores do hiperparâmetro “ <i>n-neighbors</i> ”	64
5.13	Análise da evolução da taxa de erro no conjunto de desenvolvimento, para <i>K-Nearest Neighbors</i> , com diferentes valores do hiperparâmetro “ <i>n-neighbors</i> ”	64
5.14	Análise da evolução da <i>Accuracy</i> no conjunto de treino e no conjunto de desenvolvimento, para <i>Decision Trees</i> com diferentes valores do hiperparâmetro <i>Max_Depth</i>	66

5.15 Matriz de Confusão para <i>Decision Tree</i> no conjunto de desenvolvimento com <i>Max_Depth=6</i>	67
5.16 Matriz de Confusão Normalizada para <i>Decision Tree</i> no conjunto de desenvolvimento com <i>Max_Depth=6</i>	68
5.17 Importância das <i>features</i> para para <i>Decision Tree</i> no conjunto de desenvolvimento com <i>Max_Depth=6</i>	68
5.18 Análise da evolução da <i>Accuracy</i> no conjunto de treino e no conjunto de desenvolvimento, para <i>Random Forest</i> com diferentes valores do hiperparâmetro ' <i>n_estimators</i> '	69
5.19 <i>Artificial Neural Networks</i> com 4 camadas [67]	71

Lista de Tabelas

4.1	Exemplo de uma Matriz de Confusão para um problema de classificação binária	40
4.2	Resumo da revisão sistemática da literatura científica mais recente para manutenção preditiva com métodos de <i>Machine Learning</i> - Adaptado de [15]	44
5.1	Exemplo típico do registo de telemetria em tempo real	47
5.2	Análise estatística aos dados da telemetria em tempo real	47
5.3	Exemplo típico do registo de erros	49
5.4	Exemplo típico do histórico de manutenção	50
5.5	Exemplo típico do histórico de falhas	51
5.6	Exemplo típico das informações de cada máquina	52
5.7	Exemplo de <i>Lag Features</i> para os dados de telemetria em tempo real, com N=3	55
5.8	Exemplo de <i>Lag Features</i> para os dados de telemetria em tempo real, com N=24	55
5.9	Exemplo de <i>Lag Features</i> para o registo de erros	56
5.10	Tempo desde a última substituição, por tipo de componente	56
5.11	Exemplo da representação de falha no componente 2	60
5.12	Quantidade de dados atribuída a cada um dos conjuntos e a percentagem correspondente a falhas	61
5.13	Exemplo do desequilíbrio entre as diferentes classes para a <i>feature 'failure'</i> no conjunto total de dados	62

5.14	Resultado para <i>K-Nearest Neighbors</i> no conjunto de desenvolvimento com “ <i>n-neighbors</i> ”= 13	65
5.15	Resultado para <i>K-Nearest Neighbors</i> no conjunto de desenvolvimento com “ <i>n-neighbors</i> ”= 1	65
5.16	Resultado para <i>Decision Tree</i> no conjunto de desenvolvimento com <i>Max_Depth</i> =6	67
5.17	Resultado para <i>Random Forest</i> no conjunto de desenvolvimento com ‘ <i>n_estimators</i> = 7’	70
5.18	Resultado para <i>naive Bayes</i> no conjunto de desenvolvimento	70
5.19	Resultado para <i>Artificial Neural Network</i> no conjunto de desenvolvimento com 100 camadas ocultas (<i>hidden_layers</i> = 100)	72
5.20	Resultado para <i>Artificial Neural Network</i> no conjunto de desenvolvimento com 100 camadas ocultas (<i>hidden_layers</i> = 100) e normalização <i>min-max scaling</i>	72
5.21	<i>Performance</i> para o modelo <i>Random Forest</i> nos conjuntos de desenvolvimento e teste, com ‘ <i>n_estimators</i> = 7’	73
5.22	<i>Performance</i> para o modelo <i>Artificial Neural Network</i> nos conjuntos de desenvolvimento e teste, com 100 camadas ocultas (<i>hidden_layers</i> = 100) e normalização <i>min-max scaling</i>	73

Lista de acrónimos e siglas

AI *Artificial Intelligence*

ANN *Artificial Neural Network*

AR *Augmented Reality*

BN *Bayesian Network*

CNN *Convolutional Neural Network*

CPS *Cyber-Physical Systems*

DL *Deep Learning*

FN *False Negative*

FP *False Posite*

GPR *Gaussian Process Regression*

IA *Inteligência Artificial*

IoT *Internet of Things*

K-NN *K-Nearest Neighbors*

LCC *Life Cycle Costing*

LR *Linear Regression*

LSTM *Long Short-Term Memory*

MC *Manutenção Preventiva Condicionada*

ML *Machine Learning*

MLP *Multi-Layer Perceptron*

MPS *Manutenção Preventiva Sistemática*

NN *Neural Networks*

PCA *Principal Component Analysis*

RF *Random Forests*

RNN *Recurrent Neural Network*

RTF *Run-To-Failure*

RUL *Remaining Useful Life*

SVM *Support Vector Machine*

TN *True Negative*

TP *True Positive*

VR *Virtual Reality*

1. Introdução

O presente documento foi desenvolvido no âmbito da unidade curricular Dissertação do Mestrado Integrado em Engenharia Mecânica da Faculdade de Engenharia da Universidade do Porto, com o objetivo de apresentar o trabalho desenvolvido durante o segundo semestre do ano letivo 2019/2020.

1.1. Enquadramento

Na economia globalizada atual, o ambiente competitivo, caracterizado pelo aparecimento crescente e cada vez mais rápido de novas técnicas e tecnologias nos processos e produtos, leva a que a sobrevivência das organizações dependa da sua capacidade de inovar, da atualização constante e da procura incessante pela otimização de soluções [1]. Estas mudanças estão relacionadas principalmente com o aumento da variedade de aspetos físicos (equipamentos, sistemas e estruturas) e com a crescente preocupação do seu impacto no ambiente, segurança, mais valias económicas e qualidade dos produtos [1]. A melhoria contínua nos processos e o espírito de inovação é uma realidade que pode ser observada de diversas formas nas áreas que constituem uma empresa [1]. É fundamental procurar uma diferenciação que possibilite uma melhoria na posição estratégica no mercado atual [1]. Assim sendo, um dos setores que pode representar um custo significativo, e que tem despertado as empresas para a implementação de um desenvolvimento contínuo, é o setor de manutenção [1, 2].

A manutenção constitui um fator relevante para a competitividade de uma empresa, uma vez que as ações realizadas a esse nível têm impacto direto em aspetos como o custo, prazo e qualidade dos produtos produzidos ou serviços prestados [2]. A manutenção é um suporte à área operacional de uma empresa, não podendo ser dissociada desta, dada a implicação que tem ao nível da eficiência dos ativos produtivos [2]. Estas duas áreas, operação e manutenção, têm obrigatoriamente de funcionar em paralelo de modo a garantir a disponibilidade e a rápida resposta dos meios humanos e materiais aos problemas operacionais, permitindo assim assegurar o cumprimento dos objetivos com a maximização dos recursos disponíveis [2]. Desta forma, passa a ser importante, não só, atingir os objetivos propostos, mas atingi-los

com o mínimo consumo ou utilização de recursos [2].

É neste contexto de transformação constante, de rápidas mudanças tecnológicas, que surge uma revolução baseada na incorporação de tecnologias de informação e comunicação em processos industriais tradicionais, criando uma conexão entre os diversos sistemas, conhecida como Indústria 4.0 [1].

A Indústria 4.0 é um conceito que surge em novembro de 2011, num artigo publicado pelo governo Alemão [3], relativamente a uma estratégia de desenvolvimento tecnológico. É caracterizada pela fusão conceptual de tecnologias que encurtam a fronteira entre os sistemas físico, digital e o ser humano, através de um conjunto de avanços e desenvolvimentos tecnológicos disruptivos, que afetam a indústria a nível global, tais como a *Internet of Things (IoT)*, *Cyber-Physical Systems (CPS)*, *Virtual/Augmented Reality (VR/AR)*, *Big Data* e *Cloud Computing* [1].

A Indústria 4.0 implementa as ferramentas fornecidas pelos avanços nas tecnologias de informação e comunicação de modo a aumentar os níveis de automação e digitalização nos processos industriais e de produção [1]. O objetivo é gerir todo o processo da cadeia de valor, melhorando a eficiência da produção e criação de produtos e serviços de qualidade superior [4]. Um dos pontos chave desta revolução são os dados, que agora são lidos, processados, armazenados, analisados e partilhados muito mais facilmente entre máquinas e o ser humano [4].

A IoT é definida como um ecossistema no qual os objetos e equipamentos nele inseridos estão equipados com sensores e outros dispositivos digitais, sendo assim capazes de reunir e trocar informações uns com os outros, num sistema interligado em rede [1]. Esta tecnologia permitirá avanços significativos na execução e gestão de tarefas em tempo real [1].

Nos últimos anos, uma queda no custo e um aumento da fiabilidade de sensores, dispositivos de transmissão e armazenamento de dados promoveram o aparecimento de sistemas de monitorização de condições [1, 5]. Simultaneamente, a IoT permite uma transmissão em tempo real destas informações acerca das condições dos sistemas capturadas por diversos dispositivos de monitorização [5]. Este desenvolvimento oferece uma excelente oportunidade para usar os dados de monitorização de condições de forma inteligente dentro dos regimes de manutenção preditiva, aliando a capacidade de recolha de dados a uma análise efetiva e integrada dos mesmos [5]. Neste sentido, o potencial das ferramentas de Inteligência Artificial, mais concretamente o *Machine Learning*¹, permitirá ambicionar uma melhoria na disponibilidade dos sistemas, reduzindo os custos de manutenção, aumentando o desempenho operacional e ainda a capacidade de apoio na tomada de decisão em relação ao ponto

¹Capítulo 4

ideal no tempo e a ação ideal para a realização da intervenção de manutenção [4, 5].

A Inteligência Artificial em alguma das suas vertentes, como o Machine Learning, está frequentemente associada à análise de dados [5]. O *Machine Learning* pode ser definido como “o campo de estudo que dá ao computador a habilidade de aprender sem ser explicitamente programado” [6]. Pode dizer-se que “os algoritmos de *Machine Learning* usam métodos computacionais para aprender informação diretamente dos dados sem recurso a equações pré-definidas como modelo” [6].

1.2. Objetivos

O objetivo principal da presente dissertação é aplicar ferramentas de Inteligência Artificial, mais especificamente de *Machine Learning*, a um conjunto de dados, provenientes de diversas fontes, disponibilizados *online* [7]. Mais ainda, procurar analisar as especificidades desta implementação e a definição de metodologias, com vista a permitir fornecer informações e ferramentas à área da manutenção.

1.3. Estrutura da dissertação

A presente dissertação encontra-se dividida em seis capítulos.

Neste primeiro capítulo (Capítulo 1) é realizada uma pequena introdução e enquadramento ao tema, bem como os objetivos propostos.

No Capítulo 2 é realizada uma revisão bibliográfica, onde são apresentados os conceitos inerentes à manutenção, tipos de manutenção, Normas Portuguesas da Manutenção, custo do ciclo de vida e indicadores de desempenho.

No Capítulo 3 começa por abordar-se, muito brevemente, a história das revoluções industriais de modo a enquadrar o tema da Indústria 4.0, onde são abordados os conceitos de *Internet of Things* (IoT), *Cyber-Physical Systems* (CPS) e *Big Data*. É introduzido o tema da Inteligência Artificial e realizada uma breve distinção entre Inteligência Artificial, *Machine Learning* e *Deep Learning*. Por fim, é apresentado o impacto e principais desafios da Indústria 4.0 na manutenção.

No Capítulo 4, relacionado com *Machine Learning*, apresenta-se a sua definição, os tipos de *Machine Learning* existentes, o *Workflow* de um projeto, as métricas de avaliação dos modelos e uma primeira abordagem à aplicação de métodos de *Machine Learning* para a manutenção preditiva, auxiliada por uma revisão sistemática da lite-

ratura científica.

No Capítulo 5 são apresentados os resultados da aplicação de modelos de *Machine Learning* a um conjunto de dados disponibilizado *online*. É realizado um estudo das diferentes origens dos dados, divisão e classificação, e ainda como estes se podem conjugar, através de *feature engineering*, de modo a aplicar os modelos de *Machine Learning*. Estuda-se ainda o problema do desequilíbrio entre classes comum em aplicação destas ferramentas para problemas de manutenção.

No Capítulo 6 são apresentadas as conclusões finais do projeto e propostas de trabalho futuro.

2. Estado da Arte

2.1. Introdução

No presente capítulo apresentam-se conceitos relacionados com a manutenção, tipos de manutenção, Normas Portuguesas da Manutenção, custo do ciclo de vida e indicadores de desempenho. Este capítulo é de extrema relevância para o desenvolvimento da presente dissertação, uma vez que permite refletir sobre os conceitos adquiridos.

Recentemente, é possível observar um processo de mudança de mentalidade relativamente à forma como a manutenção é percecionada, sendo-lhe atribuído um papel cada vez de maior relevância, pelo que, faz todo o sentido fazer o seu enquadramento, mas também apresentar novas abordagens e políticas de manutenção.

2.2. Definição de Manutenção

Devido aos elevados padrões de qualidade e inspeção em vigor atualmente, um equipamento, quando adquirido e colocado em funcionamento opera, regra geral, no limiar da sua eficiência máxima [2]. Salvo raras exceções, provenientes de possíveis defeitos de fabrico ou montagem, um sistema que possua componentes móveis acaba inevitavelmente por se desgastar [2, 8]. Assim sendo, será necessária, por exemplo, a realização de trocas dos componentes que apresentam desgaste, de modo a restabelecer a capacidade máxima do equipamento em questão [2]. É essencial para as empresas detetar estes acontecimentos e desenvolver um plano para combater o inevitável desgaste [2].

De acordo com Giagi (2007) [9], “para que uma instalação assegure a função para que foi concebida, é necessário que os seus equipamentos e máquinas sejam mantidos em boas condições de funcionamento. Isto requer que sejam efetuadas reparações, inspeções, rotinas preventivas, substituição de componentes ou peças, mudanças de óleo, limpezas, correções de defeitos e pinturas para que se possa repor os níveis de operacionalidade. Este conjunto de ações forma o leque de atividades da Função de Manutenção”.

É possível encontrar várias definições para a manutenção propostas por diversos autores. Para Pinto [10], a manutenção é o ato de “combinar ações de gestão, técnicas e económicas, aplicadas aos bens para otimização do seu ciclo de vida”. Já para Cabral [11], a manutenção pode ser definida como “o conjunto das ações destinadas a assegurar o bom funcionamento das máquinas e das instalações, garantindo que estas são intervencionadas nas oportunidades e com o alcance certo, de forma a evitar que avariem ou baixem de rendimento e, no caso de tal acontecer, que sejam repostas em boas condições de operacionalidade com a maior brevidade, tudo a um custo global otimizado”.

No entanto, a definição adotada na presente dissertação é a encontrada na Norma Portuguesa NP EN 13306:2007 [12], onde a manutenção é definida como a “combinação de todas as ações técnicas, administrativas e de gestão, durante o ciclo de vida de um bem, destinadas a mantê-lo ou repô-lo num estado em que ele pode desempenhar a função requerida”.

De notar que, apesar das suas diferenças, os objetivos da manutenção mantêm-se transversais a todas estas definições.

Independentemente do objetivo considerado primordial, as ações de manutenção devem orientar-se pelos seguintes conceitos [2, 10, 11]:

- **Segurança:** deve ser garantida a segurança de todos os elementos intervenientes no processo, desde os equipamentos às pessoas;
- **Qualidade:** o número de defeitos deve ser minimizado, visando-se a obtenção de um rendimento cada vez maior. Esta aquisição deve ser conseguida sempre respeitando as normas estabelecidas;
- **Criação de Valor:** devem ser objetivados custos globais mínimos. Esses valores dependem dos custos de produção, bem como das intervenções ou não de manutenção;
- **Disponibilidade:** visa garantir a maior operacionalidade dos equipamentos e sistemas, permitindo a regularização da produção e cumprimento de prazos.

A harmonização dos parâmetros supramencionados é muitas vezes de extrema dificuldade, uma vez que melhorias implementadas num dos parâmetros não conduzem invariavelmente a progressos nos outros [2, 10, 11]. Por exemplo, a redução do número de defeitos e posterior aumento de qualidade podem encaminhar a aumentos exponenciais de custos, o que pode não ser rentável para a empresa [2, 10, 11]. Esta interdependência das variáveis constitui um desafio permanente, a que os profissionais de manutenção estão sujeitos [10, 11].

Atualmente, é possível observar uma nova fase de mudança no conceito de manutenção, com recurso à implementação de novas tecnologias, cada vez mais presentes na

indústria, que abrem a possibilidade de monitorizações dos equipamentos de forma mais eficaz, tornando, no entanto, o processo mais complexo [13].

Algumas das principais limitações presentes atualmente na manutenção são [2, 14]:

- Escolha das práticas que conduzam a um desempenho superior (*benchmarking*);
- Falta de comunicação e informação;
- Incapacidade de medição e controlo dos parâmetros de *performance*, razão de qualidade e disponibilidade do sistema;
- Aumento da complexidade dos sistemas;
- Inexistência de uma proposta e plano estratégico e a sua implementação;
- Falta de preocupações com a segurança e saúde.

2.3. Tipos de Manutenção

É fundamental para as empresas que os equipamentos críticos operem no pico de eficiência e utilização, de modo a permitir um retorno sobre os investimentos de capital [15].

De forma geral, a manutenção de ativos pode dividir-se em quatro grupos distintos de estratégias de manutenção (Figura 2.1): manutenção corretiva ou *Run-to-failure* (RTF), manutenção preventiva, manutenção preditiva e manutenção prescritiva [2].

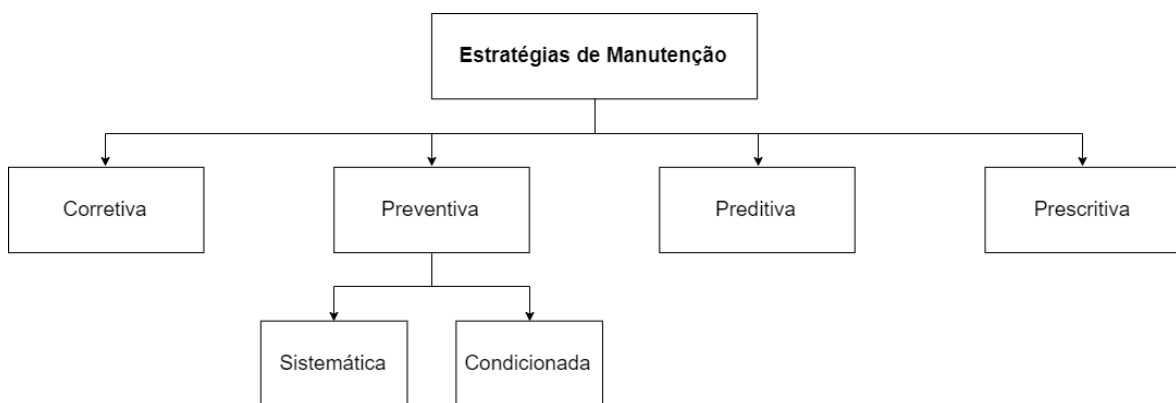


Figura 2.1: Estratégias de Manutenção

2.3.1. Manutenção Corretiva (*Run-to-failure*)

A manutenção corretiva ou *Run-to-failure* (RTF) é a “manutenção efetuada depois da deteção de uma avaria e destinada a repor um bem num estado em que este é capaz de realizar uma função requerida” [11]. As falhas dos equipamentos ocorrem sem

qualquer aviso e, portanto, não existe qualquer planeamento prévio da manutenção [11]. A manutenção corretiva garante que as peças são completamente utilizadas, isto é, não existe desperdício da vida útil dos componentes [14].

Esta abordagem é de fácil implementação, uma vez que não necessita de grandes esforços de organização e planeamento o que, conseqüentemente, conduz a uma diminuição dos custos indiretos da manutenção, isto é, custos associados às atividades de gestão, organização e planeamento da manutenção [11, 14]. No entanto, a manutenção corretiva apresenta diversas desvantagens: uma paragem não planeada leva a um aumento do custo e perdas na produção; possível aumento no número de defeitos; diminuição do tempo útil de vida dos equipamentos; aumento do risco de acidente e conseqüente redução da segurança dos utilizadores do equipamento [11, 14].

2.3.2. Manutenção Preventiva

A manutenção preventiva “é a manutenção efetuada em intervalos de tempo pré-determinados com a finalidade de reduzir a probabilidade de avaria ou de degradação do funcionamento de um bem” [11].

A manutenção preventiva pode-se dividir em duas vertentes: a manutenção preventiva sistemática (MPS) e a manutenção preventiva condicionada (MC) [11].

A manutenção preventiva sistemática consiste na realização de operações de manutenção em intervalos de tempo fixos com o objetivo de detetar e prevenir possíveis falhas, possibilitando assim prolongar o tempo de vida de um equipamento [14]. O agendamento das intervenções de MPS deve ser otimizado de modo a diminuir os riscos de falha e assim permitir minimizar os custos totais de manutenção [11, 14].

Na MPS, é assumida a existência de uma relação entre a perda de fiabilidade¹ e o tempo de funcionamento, com padrões de avaria constantes e que as ações de manutenção realizadas repõem a fiabilidade dos equipamentos [11].

De notar que na MPS não existe o recurso à monitorização nem à análise do estado atual dos equipamentos e dos seus componentes [11] e, portanto, podem ser realizadas ações de manutenção desnecessárias, que contribuem para um aumento do custo total da manutenção e afetam o estado de equilíbrio entre os vários componentes do equipamento [10, 11].

A manutenção preventiva condicionada (MC) é uma técnica de manutenção que, recorrendo à monitorização de parâmetros de funcionamento dos equipamentos (como

¹Secção 2.6.2

por exemplo, temperatura, pressão, vibrações) e à análise de tendências [11], processa essa informações e, quando estes parâmetros se encontram próximos ou tendem rapidamente para níveis de alarme previamente definidos, é realizada a intervenção de manutenção de imediato (ou pouco tempo depois) [16], permitindo assim uma redução dos custos associados a avarias e, conseqüentemente, os custos associados à perda de produtividade [11]. No entanto, isto leva a um aumento dos custos indiretos de manutenção, uma vez que, devido ao curto tempo de planeamento da intervenção necessária, o momento selecionado pode não ser o ideal do ponto de vista operacional [11, 16]. Mais ainda, esta estratégia de manutenção pode conduzir a um possível subaproveitamento da vida útil do componente [11].

2.3.3. Manutenção Preditiva

De acordo com a norma NP EN 13306:2010 [17], a manutenção preditiva é “baseada na condição dos equipamentos, sendo realizada de acordo com a previsão obtida pela análise e avaliação repetitiva dos parâmetros de funcionamento ou outras características conhecidas do equipamento”.

Ao contrário da manutenção preventiva condicionada, a manutenção preditiva vai muito além da simples monitorização de parâmetros de funcionamento, utilizando dados da monitorização para realizar previsões relacionadas com o estado futuro do equipamento [11], podendo ainda permitir estimar o tempo útil de vida restante até ocorrer uma falha - *Remaining Useful Lifetime* (RUL) [16] e, assim, agir de forma proativa [16].

Ao aumentar o tempo de vida dos componentes (em comparação com a manutenção preventiva) e reduzir os custos não programados de manutenção e mão de obra (em comparação com a manutenção corretiva), permite às empresas uma redução de custos e ainda vantagens competitivas [14].

De acordo com Schmidt (2017) [18], a manutenção preditiva, de modo a ser capaz de estimar a saúde e o estado de degradação dos equipamentos, deve utilizar vários tipos de informação:

- Dados relacionados com os equipamentos: tipo de máquina, localização e constituição;
- Ordens de trabalho: ações de manutenção realizadas no equipamento, descrição das falhas ou avarias, lista de peças de substituição utilizadas, apresentação das suas causas e métodos utilizados para as resolver;
- Monitorização de parâmetros de funcionamento do equipamento: temperatura, pressão, vibrações, medidas geométricas, entre outros;

- Sistemas de supervisão e aquisição de dados: número de ciclos do equipamento, tipos de variantes produzidas.

De modo a ser exequível utilizar efetivamente a condição dos equipamentos na manutenção é necessário ser possível definir um intervalo de tempo “P-F” entre a deteção do início da falha (P) e o momento em que existe perda da função (F) - Figura 2.2 - suficientemente lato para que o processo produtivo possa ser interrompido com um mínimo de perdas económicas e a intervenção de manutenção possa ser planeada de forma efetiva [19].

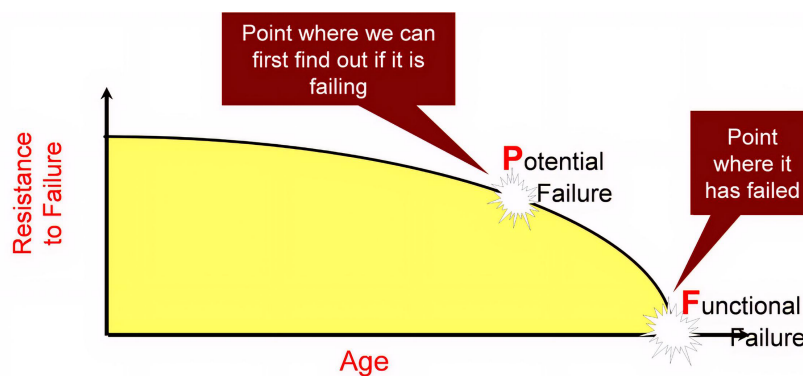


Figura 2.2: Intervalo de Tempo “P-F” entre a deteção do início da falha (P) e o momento em que existe perda da função (F) [19]

Com o advento da *Internet of Things* (IoT), cada vez mais parâmetros relacionados com a condição dos equipamentos estão disponíveis em tempo real, possibilitando assim estimar a duração do intervalo de tempo “P-F” com maior precisão, permitindo uma abordagem “just in time” à prevenção de falhas, prolongando a vida útil dos componentes [19]. Além disso, *Big Data*² pode permitir determinar fatores adicionais com possível influência neste intervalo, para um determinado modo de falha numa peça específica, sob condições específicas [5, 19].

No entanto, um problema comum em todas as formas de análise preditiva, quando se trata de prever a vida útil restante de um equipamento, é a necessidade de ter dados de falhas anteriores para analisar [15, 20]. A maioria das organizações, principalmente quando se trata de ativos críticos, tende a não permitir que esses recursos falhem - por razões óbvias [15]. Dados relacionados com falhas “suspensas”, isto é, o ativo foi sujeito a uma substituição e/ou reparação antes que falhasse, permitem saber que ele durou até esse ponto no tempo sem sofrer uma falha funcional - mas não é possível determinar por quanto tempo continuaria a funcionar caso não tivesse sido submetido à intervenção [15]. Assim, uma limitação que necessita de ser superada está relacionada com uma escassez de dados precisos (para momentos perto de

² Secção 3.3

uma falha) de modo a poder treinar e testar modelos³ que permitem prever falhas nos equipamentos [15].

Os objetivos típicos da aplicação de métodos de manutenção preditiva são [2, 14]:

- Reduzir o risco operacional de um equipamento crítico;
- Aumentar a taxa de rentabilidade dos ativos, prevendo falhas antes de estas ocorrerem;
- Controlar o custo da manutenção, permitindo operações de manutenção “*just-in-time*”;
- Reduzir os custos de inventário reduzindo os níveis de inventário, prevendo o ponto de encomenda;
- Descobrir padrões ligados a vários problemas de manutenção;
- Fornecer indicadores-chave de desempenho para as condições do ativo;
- Estimar o tempo de vida remanescente dos ativos;
- Recomendar atividades de manutenção atempadamente.

Na Figura 2.3 está representada uma visão geral das diferentes estratégias de manutenção abordadas até aqui, num intervalo de tempo $[t_0, t]$, no que diz respeito à deteção e possível ocorrência de uma falha e ainda o posicionamento e duração temporal dos custo e indisponibilidade dos equipamentos [16].

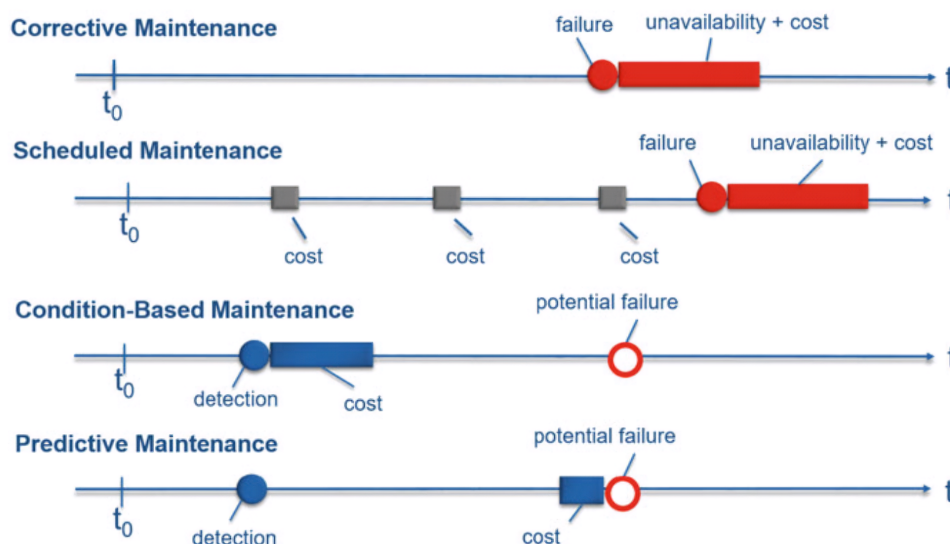


Figura 2.3: Visão geral das diferentes estratégias de manutenção [16]

³ Capítulo 4

2.3.4. Manutenção Prescritiva

A manutenção prescritiva (por vezes também chamada de manutenção proativa) é uma estratégia de manutenção que se baseia na aquisição e análise de dados acerca da condição de um equipamento para apresentar recomendações especializadas e correspondentes resultados dessas recomendações, de modo a reduzir os riscos operacionais [2, 21].

Como referido anteriormente, a manutenção preditiva recorre a vários tipos de informações que descrevem a condição de um ativo e o seu estado operacional geral [2, 18]. Estes dados podem ser analisados para prever quando os eventos de falha poderão ocorrer [2]. Contudo, a manutenção prescritiva leva essa análise a um nível superior, não apenas prevendo eventos de falha, mas também recomendando ações a serem tomadas [2, 21]. Os resultados potenciais da execução dessas recomendações são calculados e antecipados [2, 21]. A manutenção prescritiva permite calcular os efeitos da variação das condições operacionais até o momento da falha [2, 21].

Por exemplo, dado que um determinado equipamento opera com uma temperatura variável nos seus rolamentos, os métodos preditivos informam quando é provável a ocorrência de uma falha devido ao seu perfil de temperatura [21]. Por outro lado, os métodos prescritivos informam que a redução da velocidade do equipamento numa certa quantidade pode permitir duplicar o tempo até ocorrer uma falha [21].

À medida que a capacidade de aquisição de dados aumenta, aumenta também a complexidade da análise que pode ser feita para interpretar estes dados. Ferramentas de Inteligência Artificial exibem um enorme potencial na aplicação destes conceitos [2, 21]. No entanto, atualmente, o número de aplicações reais de estratégias de manutenção prescritiva é baixo [21].

2.4. Falha, Avaria e Modo de Falha

De acordo com a norma NP EN 13306:2007 [12], define-se “falha da função requerida” ou simplesmente “falha”, como sendo o “estado de um bem inapto para cumprir uma função requerida, excluindo a inaptidão devida à manutenção preventiva ou outras ações programadas, ou devida à falta de recursos externos”, ou seja, é não só a cessação de funcionamento mas também a degradação de um parâmetro de funcionamento até um nível que se considere insatisfatório. Ainda segundo NP EN 13306:2007 [12] define-se “avaria” como sendo a “cessação da aptidão de um bem para cumprir uma função requerida”. A avaria representa um caso particular de falha, ou seja, o termo avaria refere-se não só à ocorrência de uma falha mas engloba

igualmente a consequência, isto é, o bem imobiliza-se devido à função desempenho se reduzir a zero. Assim sendo, após a declaração da avaria o bem poderá estar em falha, total ou parcial. “Avaria” é então um acontecimento, enquanto que “em falha” ou “avariado” é um estado.

Novamente de acordo com a norma NP EN 13306:2007 [12], modo de falha pode ser definido como a “maneira pela qual é verificada a incapacidade de um bem para cumprir uma função requerida”. Pode igualmente ser entendido como o sintoma através do qual uma falha de um componente ou equipamento se pode manifestar, originando uma falha funcional no próprio equipamento ou sistema. Assim sendo, o modo de falha é uma propriedade inerente a cada equipamento, dado que cada um tem as suas próprias características.

2.5. Normas Portuguesas da Manutenção

Tendo em conta a diversidade de domínios técnicos relacionados com a execução da intervenções de manutenção, é recomendável que se recorra a um conjunto de normas que deverão ser acessíveis pelas organizações e recursos humanos afetos ao departamento [22]. Recentemente, é possível observar uma atividade significativa na uniformização de terminologias, conceitos, definições e formas de atuação, através de normas internacionais [22]. Esta tendência representa inequivocamente um fator positivo, no entanto, a sua eficácia depende do realismo no número de normas que se elaboram, do rigor dos seus conteúdos e da sua utilização intensiva pelos interessados [22]. São de seguida listadas as normas mais significativas [22]:

- NP EN 13306:2007: “Terminologia da Manutenção”;
- NP EN 15341:2009: “Manutenção – Indicadores de desempenho da manutenção (*KPI*)”;
- NP EN 13269:2007: “Manutenção. Instruções para a preparação de contratos de manutenção”;
- NP EN 13460:2009: “Manutenção – Documentação para a manutenção”;
- NP 4483:2009: “Guia para a implementação do sistema de gestão de manutenção”;
- NP 4492:2010 “Requisitos para a prestação de serviços de manutenção”.

2.6. Indicadores de desempenho da manutenção

Só é possível quantificar a importância da manutenção através da monitorização de indicadores de desempenho dos equipamentos e do processo de manutenção [2]. A análise dos parâmetros contidos nestes conceitos ajudará a compreender as anomalias dos equipamentos, podendo assim ser desenvolvido um plano de manutenção mais adequado para cada equipamento [2].

2.6.1. Manutibilidade

De acordo com a norma NP EN 13306:2007 [12], a manutibilidade é definida como a “aptidão de um bem, sob condições de utilização definidas, para ser mantido ou restaurado, de tal modo que possa cumprir uma função requerida, quando a manutenção é realizada em condições definidas, utilizando procedimentos e recursos prescritos”.

É quantificada pelo parâmetro *Mean Time To Repair (MTTR)* (Tempo Médio de Reparação) - Equação 2.1. Representa a média de tempo que leva a reparação dos equipamentos, na qual o intervalo de tempo considerado é entre a deteção de uma falha até à retoma de trabalho.

$$MTTR = \frac{\sum TTR}{N} \quad (2.1)$$

Onde:

- *MTTR* - *Mean Time to Repair* (Tempo Médio de Reparação);
- *TTR* - *Time to Repair* (Tempo de Reparação);
- N, número de reparações no período de tempo considerado.

2.6.2. Fiabilidade e Disponibilidade

A norma NP EN 13306:2007 [12] define fiabilidade como a “aptidão de um bem para cumprir uma função requerida sob determinadas condições, durante um dado intervalo de tempo”.

A fiabilidade também se pode aplicar como uma medida de desempenho da fiabilidade, podendo, nesse caso, definir-se como uma probabilidade [14]. O processo de resolução de problemas de fiabilidade deve começar por um conhecimento profundo

dos equipamentos que se pretende analisar e respectivas condições de funcionamento [14].

De modo a quantificar a fiabilidade, é utilizado o parâmetro *Mean Time Between Failures* (*MTBF*) ou Tempo Médio entre Falhas - Equação 2.2.

$$MTBF = \frac{\sum TBF}{N} \quad (2.2)$$

Onde:

- *MTBF* - *Mean Time Between Failures* (Tempo Médio entre Falhas);
- *TBF* - *Time Between Failures* (Tempo entre Falhas), o intervalo de tempo que decorre entre duas falhas consecutivas num determinado equipamento;
- *N*, número de falhas verificadas no período.

Outro parâmetro associado ao *MTBF* é a taxa de falhas (λ) - Equação 2.3 - que pode ser definida como a razão entre o número de falhas, n , e o produto entre um determinado intervalo de tempo, τ , e o número de equipamentos sobre o qual incide a análise. Este parâmetro quantifica a frequência média de falhas por equipamento. Desta forma, tem-se que:

$$\lambda = \frac{n}{m \tau} \quad (2.3)$$

A análise da fiabilidade envolve não só a análise do *MTBF* dos equipamentos, mas também o modo de degradação dos mesmos. Esta degradação influencia a evolução da taxa de falhas ao longo do tempo, normalmente representada pela denominada curva da banheira - Figura 2.4.

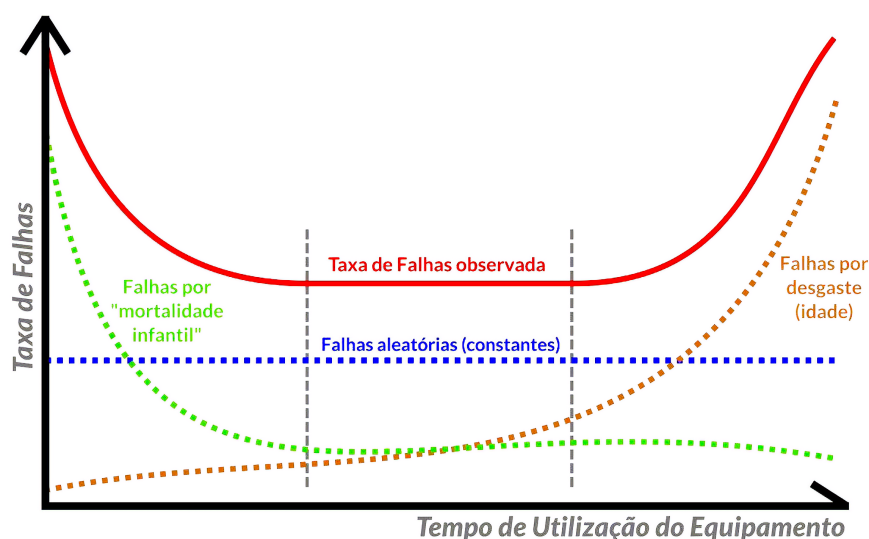


Figura 2.4: Curva da Banheira [23]

A curva da banheira (Figura 2.4) representa três períodos distintos da vida de um equipamento [11]:

- **Juventude (Fase Inicial):** a taxa de falhas é mais elevada, sendo que depois decresce e tende a estabilizar;
- **Maturidade:** as falhas são aleatórias, com taxa de falhas baixa e aproximadamente constante;
- **Velhice (Fase Final):** a taxa de falhas aumenta de uma forma acentuada, devido a factores como a acumulação de danos, fadiga ou degradação.

Pela norma NP EN 13306:2007 [12], a disponibilidade (D) - Equação 2.4 - representa a “aptidão de um bem para cumprir uma função requerida sob determinadas condições, num dado instante ou durante um dado intervalo de tempo, assumindo que é assegurado o fornecimento dos necessários recursos externos”. A disponibilidade é uma função directa da fiabilidade e da de manutibilidade.

$$D = \frac{MTBF}{MTBF + MTTR} \times 100 \% \quad (2.4)$$

Onde:

- D – Disponibilidade, geralmente expressa em percentagem;
- *MTBF* - *Mean Time Between Failures* (Tempo Médio entre Falhas);
- *MTTR* - *Mean Time to Repair* (Tempo Médio de Reparação);

Apesar de ser importante definir fiabilidade e disponibilidade de forma bastante precisa, o processo de avaliação do desempenho e do cumprimento de metas impostas na manutenção, envolve necessariamente a definição de métricas que permitam avaliar quantitativamente a fiabilidade e a disponibilidade dos equipamentos, “quantificar o que se faz, estabelecer os objetivos do que se pretende fazer” e ainda “quantificar o que se conseguiu de facto fazer” [12]. A definição de indicadores apropriados, que medem os resultados atuais e esperados desta área, permite perceber o valor criado pelos processos de manutenção e ainda quantificar se o investimento realizado está a produzir mais valias para a organização [11].

Mais ainda, em manutenção, não é possível definir um número de indicadores necessários para medir ou fundamentar decisões da gestão universais, sendo que, em cada caso será importante perceber quais os indicadores que podem ser utilizados, adaptados ou até mesmo criados especificamente para uma determinada aplicação [2].

2.7. Gestão da Manutenção

A gestão da manutenção é uma atividade dinâmica que envolve funções como o planeamento, a coordenação, a monitorização e o controlo de recursos [10]. Neste sentido, a manutenção pode ser considerada como uma atividade de apoio à produção, uma vez que contribui para uma melhoria da operacionalidade dos equipamentos [10]. Ainda assim, é muitas vezes designada como um “mal necessário” para a empresa, pois possui sempre um determinado custo associado, além de forçar a paragem de equipamentos e, conseqüentemente, da produção. No entanto, atualmente, a manutenção pode ser um fator estratégico para garantir alta produtividade de sistemas industriais [10]. É, portanto, fundamental desenvolver estratégias de gestão de manutenção [10].

De acordo com a norma NP EN 13306:2007 [12], a gestão da manutenção define-se como “o conjunto das atividades de gestão que estabelecem os objetivos, a estratégia e as responsabilidades da manutenção, e que procedem à sua implementação através do planeamento, controlo e supervisão desta e da melhoria contínua dos métodos da organização, incluindo os fatores económico”. Na Figura 2.5 estão representadas as principais funções da gestão da manutenção.

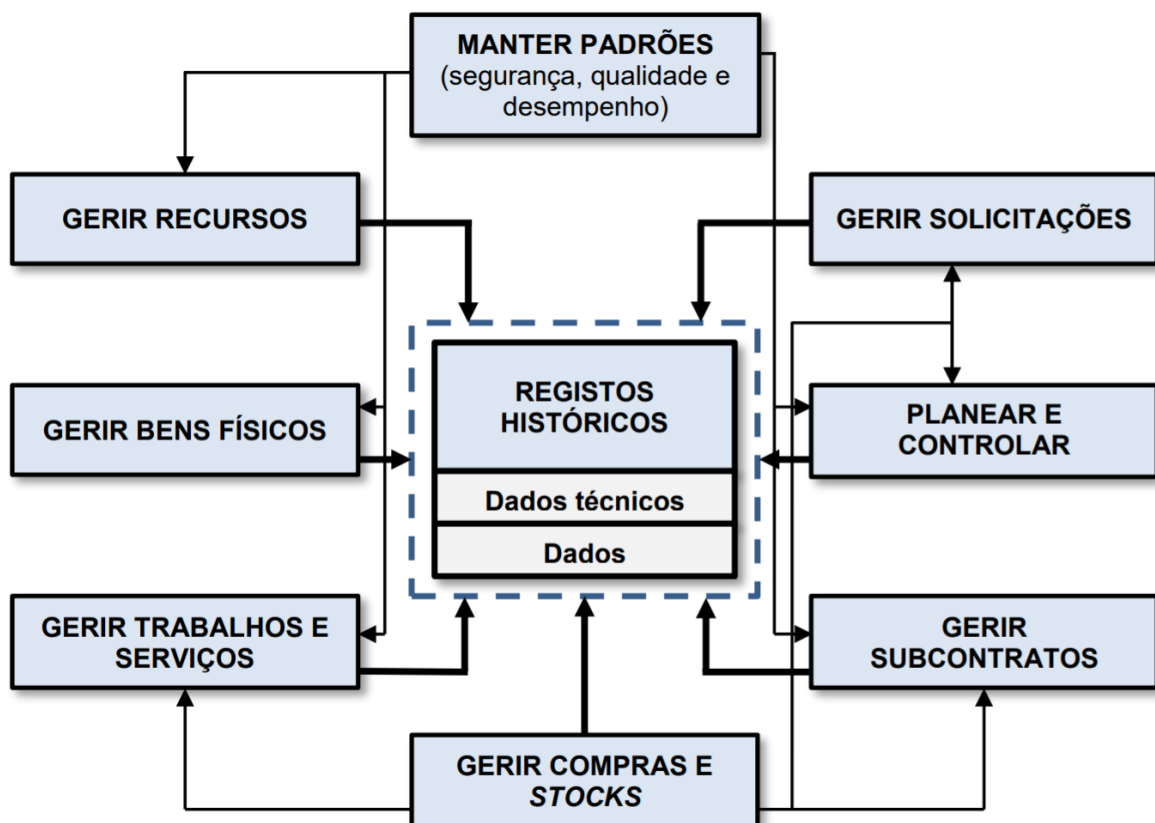


Figura 2.5: Principais Funções da Gestão da Manutenção e as suas Relações [10]

A gestão da manutenção é fundamental para a produtividade de uma empresa, otimização dos custos e, principalmente, preservação dos equipamentos [10].

2.8. Custo do Ciclo de Vida

O desenvolvimento de novos contextos tecnológicos conduziu a uma evolução natural dos conceitos básicos dos sistema de produção. De acordo com a norma IEC 60300-3-3:2004 [24], o Custo de Ciclo de Vida (*Life Cycle Costing - LCC*) pode ser definido como “o custo acumulado de um produto ao longo do seu ciclo de vida, considerando ciclo de vida como o intervalo de tempo entre a conceção de um produto e a sua retirada de serviço e abate seguindo as Normas ambientais”.

Esta definição impõe uma visão global da gestão de ativos físicos, desde a sua fase inicial de projeto até à sua desativação e abate [25]. Este é um processo de análise económica com fim a avaliar o custo do ciclo de vida de aquisição, posse e alienação de um equipamento [25]. Esta análise pode ser útil em qualquer etapa da vida de um equipamento, no entanto, quando realizada na fase de conceção e/ou aquisição, revela-se de maior utilidade dado que permite conhecer as diferentes alternativas que podem surgir nesta fase e compreender qual será o custo total dos equipamentos ao longo da sua vida, podendo assim auxiliar a tomada de decisão com base não apenas nos custos iniciais, mas em todos os custo envolvidos [25]. A título de exemplo, na Figura 2.6 está representada a evolução dos custos gerados para dois equipamentos distintos A e B. Apesar do equipamento A apresentar um custo inicial mais elevado (equipamento de qualidade superior), o custo de ciclo de vida é superior para o equipamento B (de menor qualidade e menor vida útil) [23].

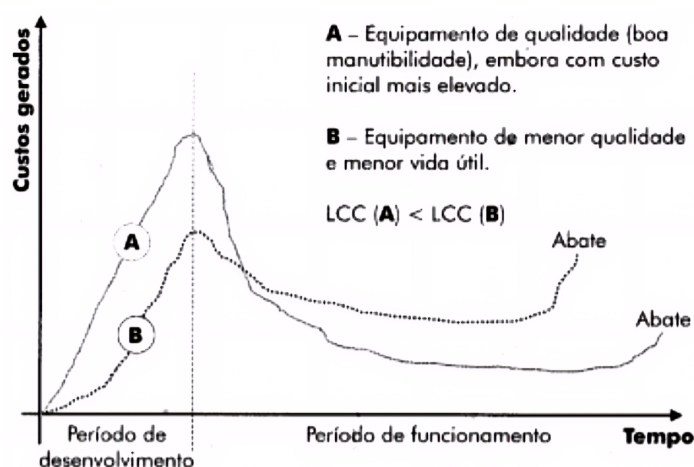


Figura 2.6: Comparação do Custo do Ciclo de Vida para dois equipamentos com custo inicial distintos [23]

O setor da manutenção representa um custo intrínseco para as empresas, no entanto,

essencial para os resultados globais da mesma [23]. Os métodos tradicionais de avaliação de custos permitem determinar facilmente os custos diretos da manutenção, sendo que o mesmo não se verifica para a estimativa dos custos indiretos [23].

A fiabilidade tem um grande impacto no custo do ciclo de vida, sendo que, em muito sistemas, pode até ser um dos fatores de custo mais importantes nesta análise [10]. As faltas de fiabilidade, ou seja, as falhas dos equipamentos, podem originar diferentes tipos de custos, desde custos diretos, resultantes das intervenções de manutenção como materiais e peças de substituição, assim como custos indiretos como a perda de produção, mão-de-obra de produção parada, perdas financeiras devido à paragem dos equipamentos, entre outros [10]. Estes fatores são normalmente representados com recurso ao “iceberg” dos custos de manutenção - Figura 2.7.

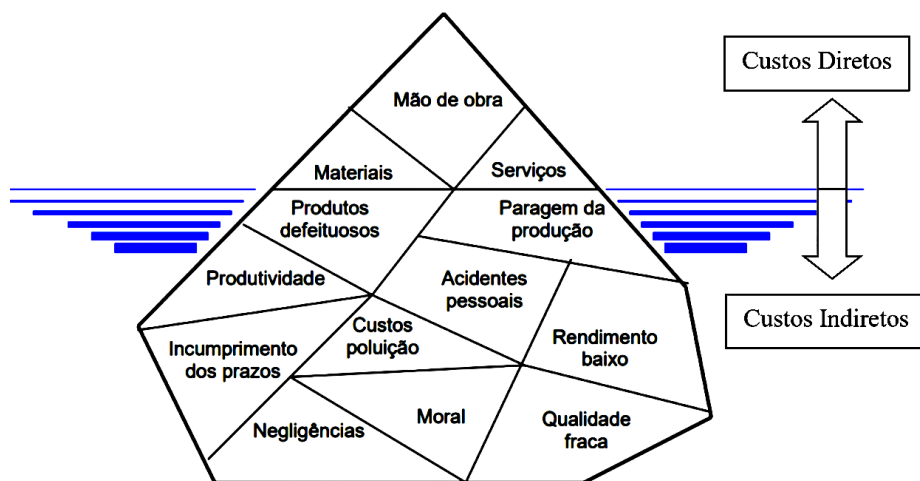


Figura 2.7: Iceberg dos Custos da Manutenção [11]

O impacto no custo do ciclo de vida das intervenções de manutenção e das perdas de produção devido a falhas imprevistas pode ser muito importante e deve ser avaliado corretamente [25].

Existem diferentes métodos para o calcular o custo do ciclo de vida, dependendo dos objetivos pretendidos. No entanto, todos eles reconhecem que, para equipamentos de capital intensivo com uma vida produtiva longa, os custos operacionais e os custos de manutenção são, normalmente, mais elevados do que os custos de investimento inicial [23]. Assim, a estimativa inicial destes custos é fundamental, uma vez que, não sendo estes conhecidos desde o início, podem vir a ser cruciais para a sobrevivência das organizações [23].

A manutenção contribui efetivamente para o sucesso das organizações e esta deve ser encarada como uma componente estratégica do sistema empresarial, deixando de ser vista como uma área geradora de custos e evoluindo para uma área que deve acrescentar valor aos processos de trabalho e negócio [25].

3. Indústria 4.0 e Inteligência Artificial

3.1. Introdução

Na sociedade atual, a informação e o conhecimento têm um papel crescente na criação de riqueza, de bem-estar e melhoria da qualidade de vida das pessoas [1, 26]. A revolução digital está a conduzir a mudanças profundas nos hábitos de consumo dos clientes, devido a, entre outros fatores, um maior acesso a dados e a um crescente desenvolvimento de novas tecnologias [1, 26]. Uma alavanca fundamental para a transformação dos modelos de negócio é a *Data Science*, que se baseia no uso combinado de técnicas de Inteligência Artificial, *Machine Learning*, matemática, estatística e bases de dados [26]. A crescente utilização destas técnicas é impulsionada por diversos fatores: um aumento sem precedentes do volume e da tipologia dos dados disponíveis; a conectividade e o acesso a estes dados; a melhoria dos algoritmos utilizados e ainda o aumento da capacidade computacional dos sistemas [26, 27].

3.2. Revoluções Industriais

A indústria é um setor da economia responsável pela produção de bens materiais através de, regra geral, processos mecanizados e automatizados [28]. Desde o início da industrialização de processos, os grandes avanços tecnológicos conduziram a mudanças de paradigma, referenciadas como “revoluções industriais” [28].

Uma revolução industrial é um processo caracterizado por mudanças abruptas e radicais, associadas à incorporação de novas tecnologias, que se vão materializar e implicar alterações na sociedade, em particular ao nível político, económico, e social [28].

A primeira revolução industrial, que ocorreu em finais do século XVIII, ficou caracterizada pela mecanização da produção, usando água e energia a vapor como fontes desse movimento [29]. Na segunda revolução industrial, final do século XIX e início

do século XX, surge a produção em larga escala de produtos, de uma forma rápida e barata, com a energia elétrica como o fator de inovação [29]. Na terceira revolução industrial, durante os anos 70 do século passado, surge o advento das tecnologias de informação e comunicação [29]. Com a revolução digital, a que ela dá origem, surge o uso intensivo dos computadores pela sociedade, a Internet e as plataformas digitais [28]. Atualmente, estamos perante a Indústria 4.0, como uma quarta revolução industrial, que assenta no desenvolvimento de Sistemas Ciber-Físicos inteligentes com capacidade de resposta rápida e assertiva a alterações na produção e ainda supervisão contínua de equipamentos e processos [26, 28]. Na Figura 3.1 está representada a cronologia das revoluções industriais.

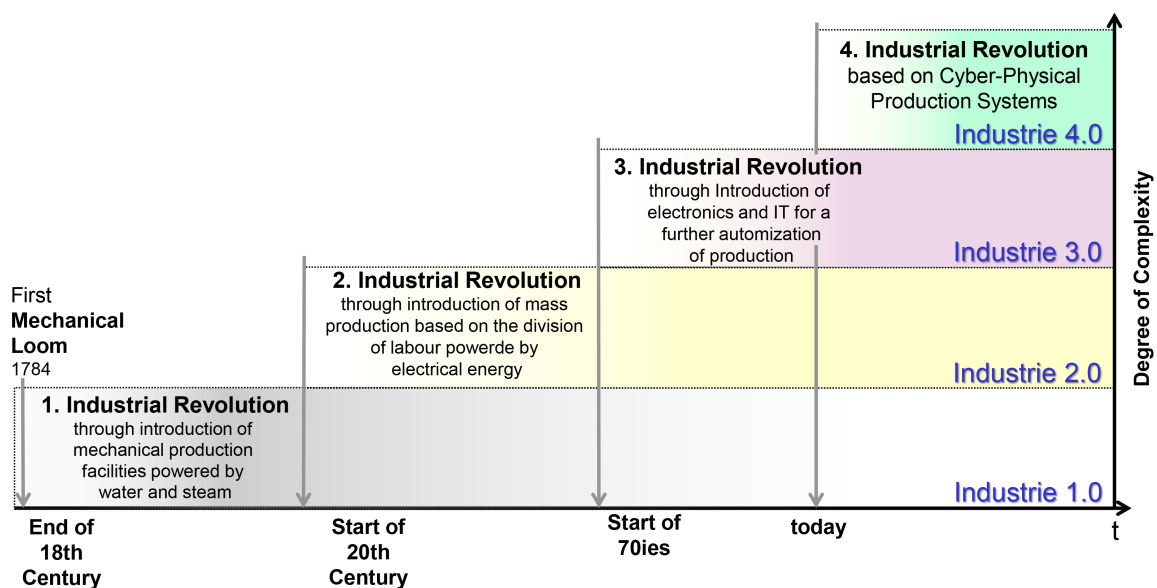


Figura 3.1: Cronologia das Revoluções Industriais [29]

3.3. Indústria 4.0

A Indústria 4.0 é um conceito que surge em novembro de 2011, num artigo publicado pelo governo Alemão [3], relativamente a uma estratégia de desenvolvimento tecnológico. Caracteriza-se essencialmente por um conjunto de avanços e desenvolvimentos tecnológicos disruptivos, que afetam a indústria a nível global, tais como a *Internet of Things* (IoT), os *Cyber-Physical Systems* (CPS), a *Virtual/Augmented Reality* (VR/AR), o *Big Data* e o *Cloud Computing* [26, 27].

A Internet das Coisas (*Internet of Things* - IoT) pode ser definida como um ecossistema no qual os objetos e equipamentos nele inseridos estão equipados com sensores e outros dispositivos digitais, sendo assim capazes de reunir e trocar informações uns com os outros, num sistema interligado em rede [1]. Esta tecnologia permitirá

avanços significativos na execução e gestão de tarefas em tempo real [1].

A Indústria 4.0, como uma quarta revolução industrial, assenta no desenvolvimento de Sistemas Ciber-Físicos (*Cyber-Physical Systems* - CPS) inteligentes com capacidade de resposta rápida e assertiva a alterações na produção e ainda supervisão contínua de equipamentos e processos [26]. Com a introdução destes sistemas, as tecnologias preditivas conseguem prever a degradação no desempenho do produto, equipamento ou recurso e otimizar, de forma autónoma, as necessidades do mesmo [1].

Os sistemas Ciber-Físicos resultam da fusão do nível físico com o nível digital, integrando computação e redes de comunicação que interagem entre si e influenciam-se mutuamente [26]. No caso da manutenção preditiva, por exemplo, os dados relativos a parâmetros de processo de componentes mecânicos (elementos físicos) são armazenados em sistemas digitais e, assim sendo, a condição real do sistema resulta da interação dos seus elementos físicos com os parâmetros digitais implementados [26].

Os dados recolhidos via IoT necessitam de ser analisados e processados [27]. O conceito *Big Data* está relacionado com esta grande quantidade de dados que são gerados e armazenados a cada instante resultantes da existência de um elevado número de sistemas em rede via IoT [1]. Por ser um volume exponencial, torna-se impossível a sua gestão manual. “Dados” podem ser números, palavras ou outros sinais e representam factos discretos sobre uma realidade objetiva [1]. Podem ser verificados e validados, contudo não tem qualquer significado se não forem interpretados e contextualizados, dando origem a informação passível de ser utilizada como auxílio à tomada de decisão. A capacidade de processar estes dados é fundamental, de modo a permitir que estes possam ter significado e que as organizações sejam capazes de melhorar as operações com decisões mais rápidas e inteligentes [1]. Na Figura 3.2 está representada a Hierarquia DIKW, de “*Data-Information-Knowledge-Wisdom*”, que é por vezes utilizada no campo da ciência dos dados e procura refletir a relação entre os vários níveis de informação [30].

O *Cloud Computing* tem oferecido um outro tipo de vantagens para a indústria, nomeadamente, tarefas de computação com maior capacidade e rapidez, armazenamento de dados em grande escala e outros recursos computacionais, incluindo servidores, armazenamento, bases de dados, rede, software e análises, através da Internet (“*a cloud*”) para disponibilizar mais rapidamente recursos flexíveis [1].

Devido à grande quantidade de dados a serem gerados continuamente são necessárias ferramentas de análise poderosas para lhe dar significado [1, 5]. Assim sendo, um dos grandes desafios da Indústria 4.0 é a sua capacidade de colecionar todos estes dados, processá-los e transformá-los em conhecimento [1, 5]. Para isso, são necessá-

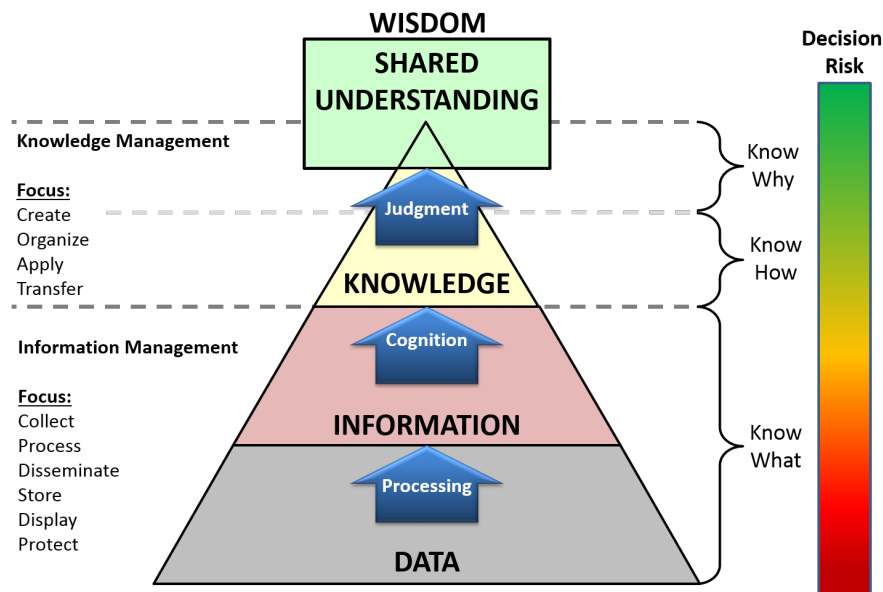


Figura 3.2: Hierarquia DIKW - “Data-Information-Knowledge-Wisdom” [30]

rios sistemas tecnologicamente evoluídos, capazes de processar grandes quantidades de dados, com recurso a algoritmos sofisticados [1, 5]. A Inteligência Artificial ou, mais especificamente, o *Machine Learning* destacam-se neste campo [1, 5].

3.4. Inteligência Artificial

O campo da Inteligência Artificial (IA) sempre despertou um fascínio no homem. Desde os primórdios que tentamos criar uma máquina que pudesse demonstrar as mesmas características que nos definem como “inteligentes” [31]. No entanto, existem ainda diversos obstáculos a serem superados. A representação do conhecimento é um dos aspetos mais complexos para a resolução de problemas de IA e a ciência continua a procurar uma forma eficiente de representar o conhecimento de maneira mais genérica em sistemas computacionais [31].

A IA é a ciência que procura estudar e compreender o fenómeno da inteligência e, ao mesmo tempo, um ramo da engenharia, na medida em que procura construir instrumentos para apoiar a inteligência humana [1].

Luger [31] define IA como o ramo da ciência da computação que se ocupa da automação do comportamento inteligente, focado no desenvolvimento de sistemas que exibem características que estão associadas à inteligência no comportamento humano, como a compreensão da linguagem, raciocínio e resolução de problemas.

Independentemente da abordagem adotada, a IA trata de uma variedade de problemas cujas características são [1]:

- Uso do computador para executar raciocínio, reconhecimento de padrões ou aprendizagem;
- Problemas que não podem ser resolvidos utilizando soluções algorítmicas tradicionais;
- Solução de problemas utilizando informação não exata, em falta ou pobremente definida;
- Respostas nem sempre exatas ou ótimas, mas suficientes em determinado contexto;
- Uso de grandes quantidades de conhecimento específico num domínio particular.

3.5. Breve distinção entre Inteligência Artificial, *Machine Learning* e *Deep Learning*

Inteligência artificial (IA), *Machine Learning* (ML) e *Deep Learning* (DL) são conceitos que estão relacionados, sendo que, habitualmente existe dificuldades para diferenciá-los. Com o objetivo de ilustrar as suas diferenças (Figura 3.3), são apresentadas as seguintes definições [6]:

A Inteligência Artificial é o conceito mais amplo, cujo objetivo é o desenvolvimento de sistemas que exibem características que estão associadas à inteligência no comportamento humano. Na conferência de Inteligência Artificial de Dartmouth (1956)[32] foi definida como “todos os aspetos da aprendizagem ou qualquer outra característica da inteligência que pode, em princípio, ser precisamente descrita de tal forma que uma máquina possa realizá-la”.

Por outro lado, o *Machine Learning* pode ser considerado como uma vertente da IA, sendo definido como “o conjunto de métodos capazes de detetar padrões automaticamente num conjunto de dados e usá-los para fazer previsões acerca de dados futuros, ou para tomar outro tipo de decisões num ambiente de incerteza”. O tema *Machine Learning* será analisado com detalhe no Capítulo 4 [6].

Por último, o *Deep Learning* é um ramo do *Machine Learning* que, de uma forma simples, procura produzir representações hierárquicas de alto nível dos dados de entrada, por meio de camadas de processamento sequencial numa rede neuronal artificial [6].

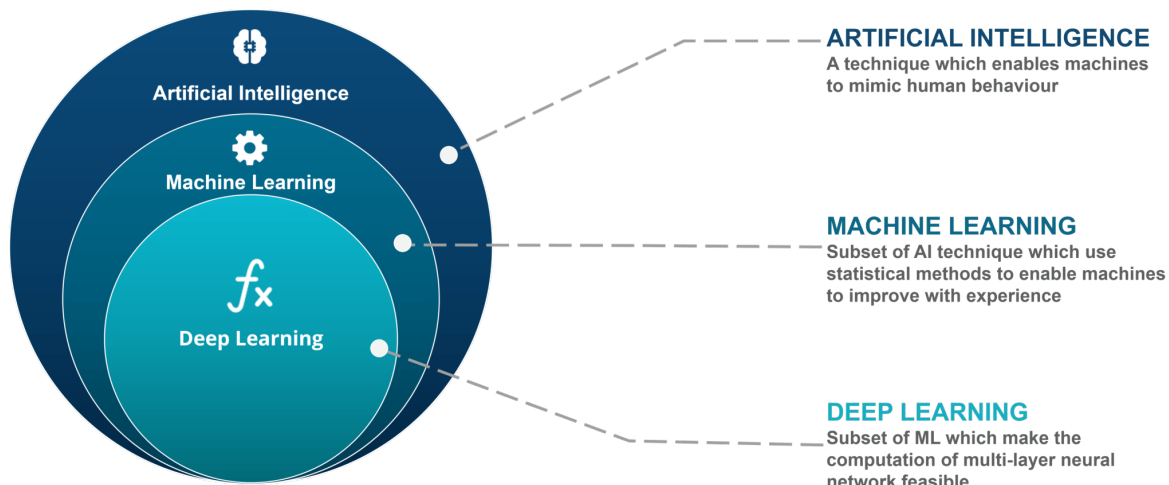


Figura 3.3: Distinção entre Inteligência Artificial, *Machine Learning* e *Deep Learning* [33]

3.6. Impacto e Desafios da Indústria 4.0 na Manutenção

As tecnologias de IoT irão desempenhar um papel fundamental na transição da área da manutenção de uma empresa para a filosofia da Indústria 4.0, baseada na sensorização de ativos físicos, assim capazes de gerar informações associadas ao seu funcionamento [1].

A implementação de sensores nos equipamentos e a análise e armazenamento da informação produzida relativa ao seu funcionamento permite uma movimentação no sentido de uma manutenção preditiva e prescritiva, já que possibilita a utilização das informações recolhidas e de métodos de IA para estudar e prever o comportamento dos equipamentos [1, 5]. A análise de dados permite igualmente a deteção de anomalias e, por outro lado, encontrar padrões e condições de funcionamento ótimas ou de maior eficiência ou prever tendências de necessidades futuras e planear as operações de uma linha de produção de forma otimizada [2].

A transformação de uma indústria tem ser um processo contínuo e abrangente, que além da melhoria de tecnologias, deve igualmente afetar toda a cadeia envolvente e sequencial de processos, modelos de negócio, aspetos organizacionais e de gestão, criando por sua vez desafios significativos para as empresas [4].

Alguns dos desafios mais relevantes são a pouca familiarização com uma visão e estratégia digital, desconhecimento dos benefícios tecnológicos, recursos financeiros, segurança de dados, falta de qualificação de recursos humanos, falta de normas e legislação, ou insuficiências na infraestrutura tecnológica [4]. A criação de uma ar-

quitetura integrativa de conectividade entre equipamentos para eficazmente gerir a recolha, armazenamento e processamento dos dados é essencial para tornar estes acessíveis e visualizáveis, possibilitando a análise e utilização de dados para promover operações mais eficientes, com aplicação de diversas técnicas baseadas em Inteligência Artificial [4].

No entanto, na área da manutenção, é importante perceber o motivo subjacente que leva um sistema de IA a recomendar uma determinada ação [34]. Um dos benefícios de aumentar a explicabilidade dos sistemas de IA é um aumento da confiança no sistema. Se o utilizador for capaz de entender o que levou a uma determinada recomendação gerada pela IA, o nível de confiança nessa decisão é superior. No entanto, a conexão entre *explicação* e *confiança* é, por vezes, complexa. Se um sistema produzir explicações convincentes, mas ilusórias, o utilizador pode desenvolver um falso sentido de confiança ou compreensão, isto é, pode ter confiança na eficácia ou segurança dos sistemas, sem que essa confiança seja justificada. Explicações podem ajudar a aumentar a confiança a curto prazo mas não ajudam necessariamente a criar sistemas que geram resultados confiáveis [34].

4. *Machine Learning*

4.1. O que é *Machine Learning*?

Na maioria dos casos, quando se deseja resolver um problema com o auxílio de um computador, é necessário um algoritmo [6]. Um algoritmo consiste na programação de uma dada sequência de instruções com as quais é possível transformar uma entrada numa saída desejada [6].

O *Machine Learning* pode ser definido como “o campo de estudo que dá ao computador a habilidade de aprender sem ser explicitamente programado”. De forma mais simples, pode dizer-se que “os algoritmos de *Machine Learning* usam métodos computacionais para aprender informação diretamente dos dados sem recurso a equações pré-definidas como modelo” [6].

4.1.1. Porque usar *Machine Learning*?

O *Machine Learning* certamente não será capaz de resolver todos os problemas para os quais ainda não há resposta, no entanto, o seu potencial não deve ser desprezado [35]. De entre as várias tarefas para as quais há uma elevada probabilidade das ferramentas de *Machine Learning* serem usadas com sucesso, destacam-se:

1. Procurar soluções para problemas complexos para os quais as abordagens tradicionais não encontraram nenhuma ou, tendo encontrado, requeiram demasiadas afinações de parâmetros ou uma longa lista de regras [35].
2. Auxiliar o ganho de “intuições” acerca de grandes quantidades de dados [35].

No caso da manutenção, o *Machine Learning* permite integrar informação e aproveitar o potencial da grande quantidade de dados que, tipicamente, são armazenados [8].

Os modelos preditivos possuem a capacidade de aprender padrões a partir de dados históricos e assim prever resultados futuros, com uma determinada probabilidade associada, com base nestes padrões observados. A precisão preditiva de um modelo

depende da relevância, quantidade e qualidade dos dados [8, 35].

4.2. Tipos de *Machine Learning*

Os sistemas de *Machine Learning* podem dividir-se em várias categorias, de acordo com vários aspetos [35]:

1. Se são ou não treinados com supervisão (Subsecção 4.2.1):
 - (a) *Supervised learning*
 - (b) *Unsupervised learning*
 - (c) *Reinforcement learning*
 - (d) *Semisupervised learning*
2. Se podem aprender de forma incremental (Subsecção 4.2.2):
 - (a) *Batch learning*
 - (b) *Online learning*
3. Se fazem previsões comparando os novos dados com pontos conhecidos ou se constroem novos modelos preditivos (Subsecção 4.2.3):
 - (a) *Instance-based learning*
 - (b) *Model-based learning*

4.2.1. *Supervised, unsupervised, reinforcement learning e Semisupervised learning*

Os sistemas de *Machine Learning* podem ser divididos de acordo com a quantidade e tipo de supervisão que recebem durante a fase de treino [35]. Este aspeto permite a divisão dos modelos *Machine Learning* em três categorias principais (*supervised*, *unsupervised* e *reinforcement learning*) e, ainda, numa quarta categoria que consiste simplesmente na combinação de modelos das categorias anteriores (*semisupervised learning*).

Supervised learning

Supervised learning é um dos modos de aprendizagem mais comum e também com maior sucesso [35]. Neste, os dados fornecidos aos algoritmos durante a fase de treino incluem a solução desejada (*label*), isto é, a variável dependente resultante das variáveis independentes observadas [6]. É assim possível desenvolver modelos preditivos capazes de gerar respostas para novos dados [35].

Existem dois principais tipos de problemas de *supervised learning*: classificação e regressão - Figura 4.1.

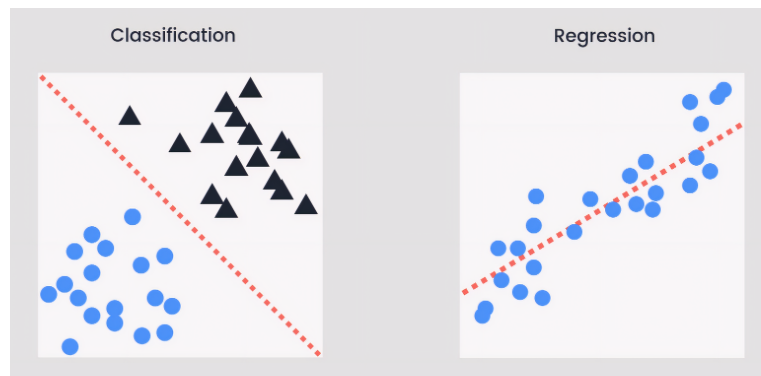


Figura 4.1: *Supervised Learning*: Classificação e Regressão [6]

As técnicas de classificação são utilizadas para prever a resposta, isto é, a classificação (*class label*), a partir de um conjunto de valores de entrada. As respostas são discretas (os modelos classificam os dados em categorias) [6]. Os problemas de classificação podem ainda ser separados em: classificação binária (*binary classification*), que consiste na distinção entre duas classes, e problemas de classificação multi-classe (*multiclass classification*), quando o número de classes é superior a dois [6]. Classificar um e-mail como *spam* ou *não spam* é um exemplo de um problema típico de *binary classification* enquanto que, prever qual a língua de um *website* com base no texto nele presente é um exemplo de um problema de *multiclass classification*, onde, neste caso, as classes seriam uma lista predefinida de possíveis línguas [35].

Por outro lado, as técnicas de regressão prevêem respostas contínuas [35]. Um exemplo de um problema deste tipo é, dado um conjunto de dados sobre o tamanho de casas no mercado imobiliário, tentar prever o seu preço (variável contínua) [35].

As variáveis utilizadas em modelos de *Machine Learning* que, espera-se, contêm informação relevante acerca do problema e permitem ao modelo aprender com sucesso são denominadas por *features* (ou atributos) [35]. Uma *feature* pode ser diretamente medida, calculada a partir dos sinais medidos ou, simplesmente, gerada pelo programador (por exemplo, a identificação de um equipamento) [35].

Os algoritmos mais populares de *supervised learning* são: *k-nearest neighbors (kNN)*, *linear regression*, *logistic regression*, *support vector machines (SVMs)*, *decision trees*, *random forests*, *naive Bayes* e *neural networks* [6, 35].

Unsupervised learning

Em alguns casos, conseguir dados com a solução desejada (*label*) pode ser caro ou até mesmo impossível [35].

Em *unsupervised learning* um algoritmo procura aprender padrões e relações existentes num conjunto de dados sem categorização (sem *labels*) fornecidos [35]. Os sistemas de *unsupervised learning* procuram encontrar padrões escondidos e estruturas intrínsecas nos dados - Figura 4.2 [6].

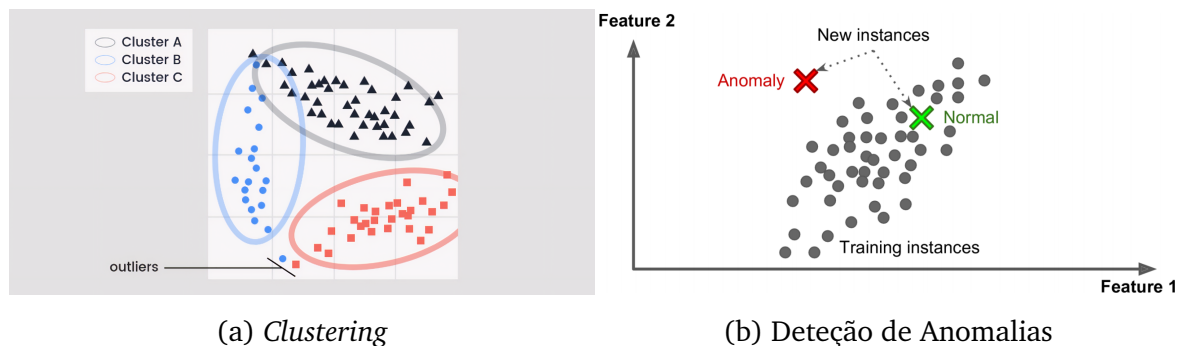


Figura 4.2: *Unsupervised Learning* [6]

Dentro do *unsupervised learning* destacam-se quatro principais tarefas [35]: *clustering*, que consiste na divisão dos dados em grupos (*clusters*) semelhantes (Figura 4.2a); visualização e redução de dimensionalidade, que procura simplificar os dados sem perda significativa de informação; deteção de anomalias, uma vez que o sistema é treinado com exemplos normais, perante um novo ponto é capaz determinar se este se encontra fora do normal (Figura 4.2b); e *association rule learning*, que consiste em encontrar relações entre *features* [35].

Reinforcement learning

Em *reinforcement learning*, o sistema tenta aprender qual a melhor ação a ser tomada, dependendo das circunstâncias na qual essa ação será executada [35]. Como, *a priori*, não sabe o que irá acontecer no futuro, é desejável uma abordagem que leve em consideração essa incerteza, capaz de incorporar as eventuais mudanças no processo de tomada da melhor decisão [35]. Uma recompensa ou punição é dada ao sistema de aprendizagem, dependendo da decisão tomada; ao longo do tempo e com a repetição do processo, espera-se que este seja capaz de associar as ações que geram maior recompensa para cada situação que o ambiente apresenta, e passe a evitar as ações que geram punição ou recompensa menor [35].

Semisupervised learning

Alguns algoritmos são capazes de lidar, em simultâneo, com uma combinação de dados de treino categorizados (normalmente em pequena quantidade) e não categorizados (em grande quantidade) - Figura 4.3 [35]. Esta combinação de um grande número de dados não categorizados, regra geral mais fáceis de obter, com alguns

dados categorizados, geralmente dispendiosos de obter, pode levar a um aumento da *performance* do sistema de aprendizagem [35].

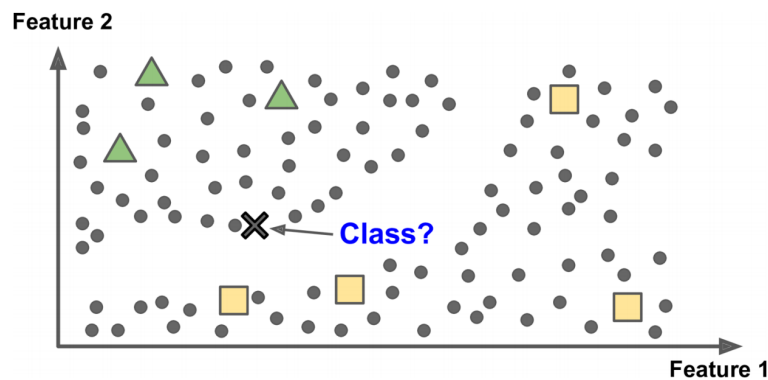


Figura 4.3: *Semisupervised learning* [6]

4.2.2. *Batch e online learning*

Outro critério usado para classificar os sistemas de *Machine Learning* é a sua capacidade de aprender de forma incremental (Figura 4.4), a partir de um fluxo de dados recebidos [35]. Em *batch learning* (Figura 4.4a), o sistema é incapaz de aprender incrementalmente, sendo que necessita de acesso à totalidade dos dados para ser treinado [35]. De modo a fornecer novos dados ao sistema, é necessário treinar uma nova versão com o conjunto de dados completo (não apenas os novos dados, mas também os antigos) [35].

Por outro lado, em *online learning* (Figura 4.4b), a fase de treino é realizada de forma incremental, isto é, o sistema é capaz de treinar continuamente ao ser alimentado com exemplos [35]. Este modo de treino é ideal para sistemas que recebem um fluxo de dados contínuo e necessitam de se adaptar de forma rápida ou autónoma [35].

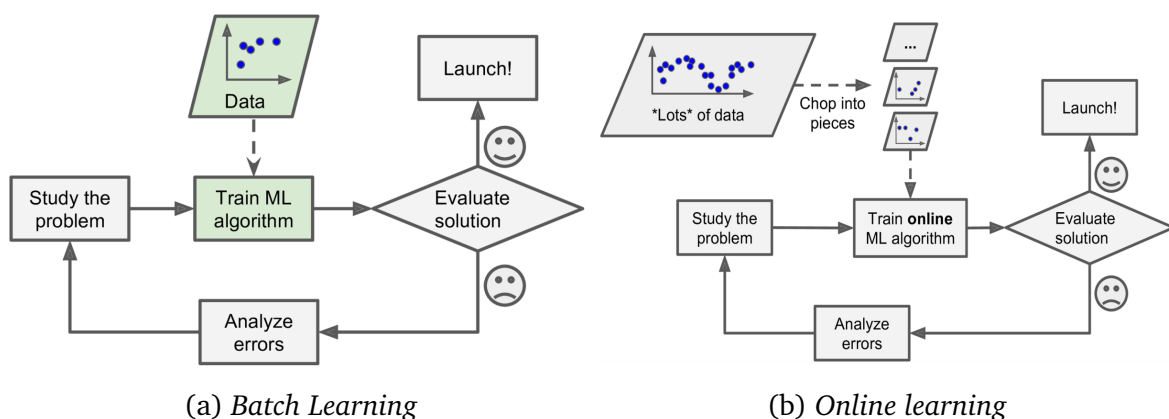


Figura 4.4: Divisão baseada na capacidade dos algoritmos aprenderem de forma incremental [35]

4.2.3. Instance-based e model-based learning

A última divisão dos sistemas de *Machine Learning* baseia-se no modo como as previsões são efetuadas, ou seja, na forma como os modelos generalizam (Figura 4.5). Dado um conjunto de exemplos para treino, o modelo tem de ser capaz de generalizar para exemplos desconhecidos. Esta generalização pode basear-se em medidas de similaridade (distância) aos exemplos conhecidos - *instance-based learning* (Figura 4.5a) - ou então pode basear-se num modelo matemático associado a estes sistemas - *model-based learning* (Figura 4.5b) [35].

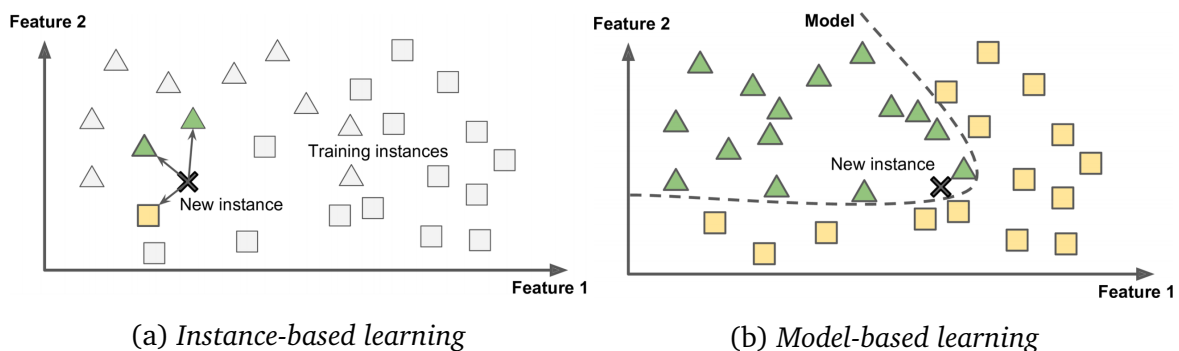


Figura 4.5: Divisão baseada na generalização para diferentes modos de aprendizagem [35]

4.3. *Workflow* de um projeto de *Machine Learning*

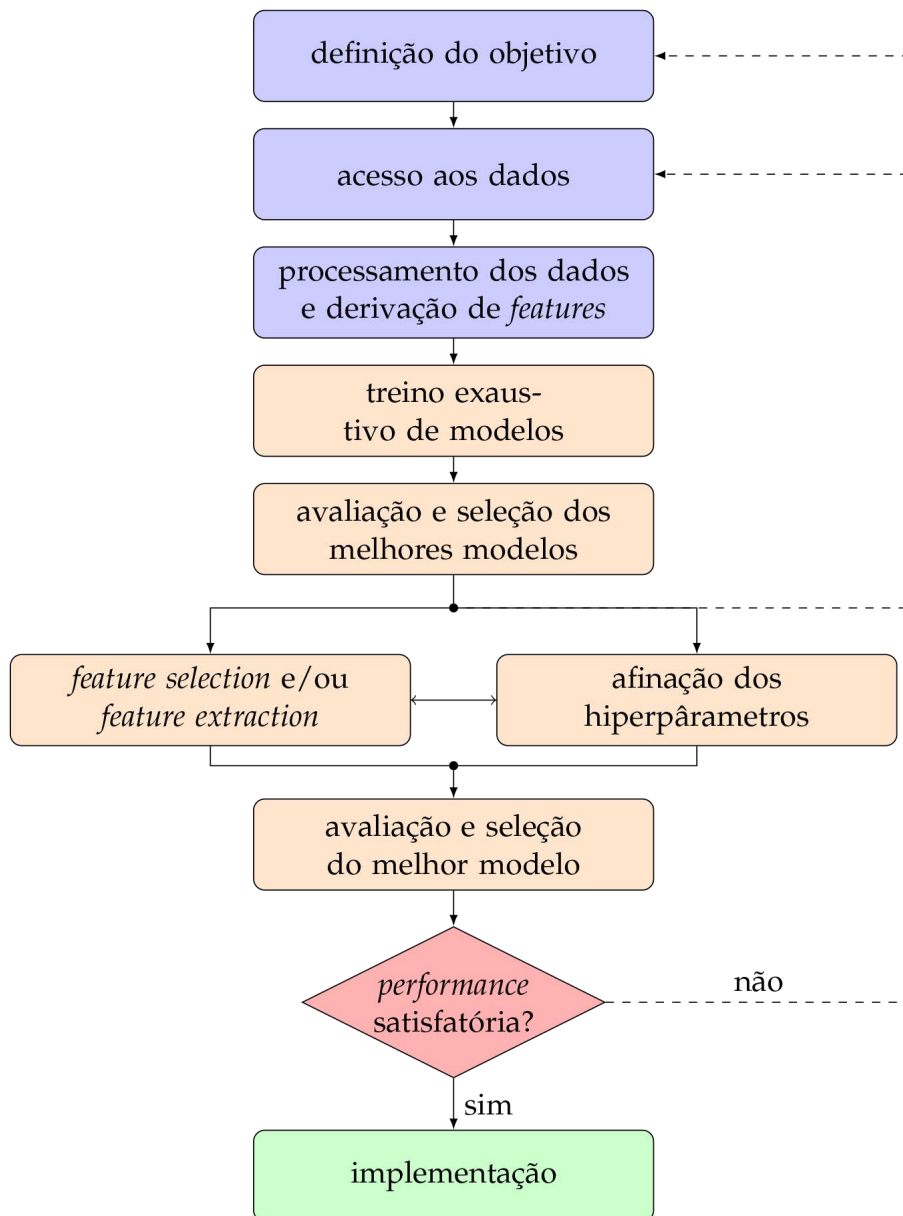
No presente projeto será seguido o *workflow* apresentado em [36], representado na Figura 4.6. Um projeto de *Machine Learning* deve começar, sempre, pelo estabelecimento rigoroso e claro dos objetivos, uma vez que um sistema de *Machine Learning* cumpre uma tarefa muito específica e a definição de objetivos muito vagos pode levar a que o que o modelo desenvolvido não seja capaz de prever exatamente o que se pretende 4.6.

Muito possivelmente, a parte mais importante de um projeto de *Machine Learning* é a capacidade de entender os dados utilizados e como estes se relacionam com a tarefa que desejamos resolver [13]. Não será eficaz escolher aleatoriamente um algoritmo, usar o conjunto de dados que temos disponível e esperar bons resultados [13]. É necessário entender o que está a acontecer no conjunto de dados antes de começar a construir um modelo. Quando construimos uma solução de *Machine Learning* devemos responder ou pelo menos ter em mente as seguintes questões:

- A que perguntas estamos a tentar responder? O conjunto de dados disponível permite responder a estas perguntas?

- Qual a melhor forma de parafrasear a minha questão como um problema de *Machine Learning*?
- O conjunto de dados disponível é suficiente para representar o problema que estamos a tentar resolver?
- Que *features* (ou atributos) foram extraídos e serão estes capazes de conduzir às corretas previsões?
- Como medir o sucesso da aplicação de *Machine Learning*?
- Como irá a solução de *Machine Learning* interagir com o resto do processo?

Os algoritmos e métodos de *Machine Learning* são apenas uma parte de um processo maior para resolver um problema específico, e é importante manter um panorama geral em mente [20]. Por vezes, muito tempo é despendido a construir soluções complexas de *Machine learning*, apenas para descobrir no final que elas não resolvem o problema que estaríamos à espera [20]. Ao aprofundar os aspetos técnicos do *Machine Learning*, é fácil perder de vista os objetivos finais. É importante manter em mente todas as suposições criadas, de forma explícita ou implícita, quando são construídos modelos de *Machine Learning*.

Figura 4.6: Proposta de *workflow* de um projeto de *Machine Learning* [36]

4.4. Treino e métricas de avaliação

A única forma de determinar a capacidade do modelo em generalizar para novos casos é mesmo testá-lo com novos casos, nunca antes vistos [6, 35]. Para isso, regra geral, o *dataset* é dividido em dois conjuntos: o conjunto de treino e o conjunto de teste [35]. Como o próprio nome indica, o conjunto de treino será utilizado para treinar o modelo. Durante esta fase de treino, o modelo não tem acesso aos dados atribuídos ao conjunto de teste [35]. Ao avaliar o modelo no conjunto de teste, é possível obter uma estimativa da taxa de erro em novos casos, chamada de erro de generalização. Este valor permite perceber a *performance* do modelo em novos registos desconhecidos [37].

Caso o erro cometido pelo modelo seja baixo no conjunto de treino mas alto no conjunto de teste significa que o modelo pode apresentar *overfitting*, isto é, o modelo é capaz de explicar muito bem os dados de treino, mas não de generalizar [35]. Por outro lado, o *underfitting* acontece quando o modelo "não aprendeu o suficiente", ou seja, não é capaz de capturar a tendência subjacente dos dados de treino [35]. Na Figura 4.7 estão representados estes comportamentos. Dependendo do modelo em questão, é importante procurar um desempenho que se situe entre o *underfitting* e o *overfitting* [37]. Estes dois termos estão também ligados aos conceitos de *bias* e *variance* (Figura 4.8) [35]. O conceito de *bias* está relacionado com o *underfitting* e significa que o algoritmo tende a aprender consistentemente a mesma coisa errada [35]. O conceito de *variance*, relacionado com o *overfitting*, ocorre quando não existe um padrão nos erros de previsão [35].

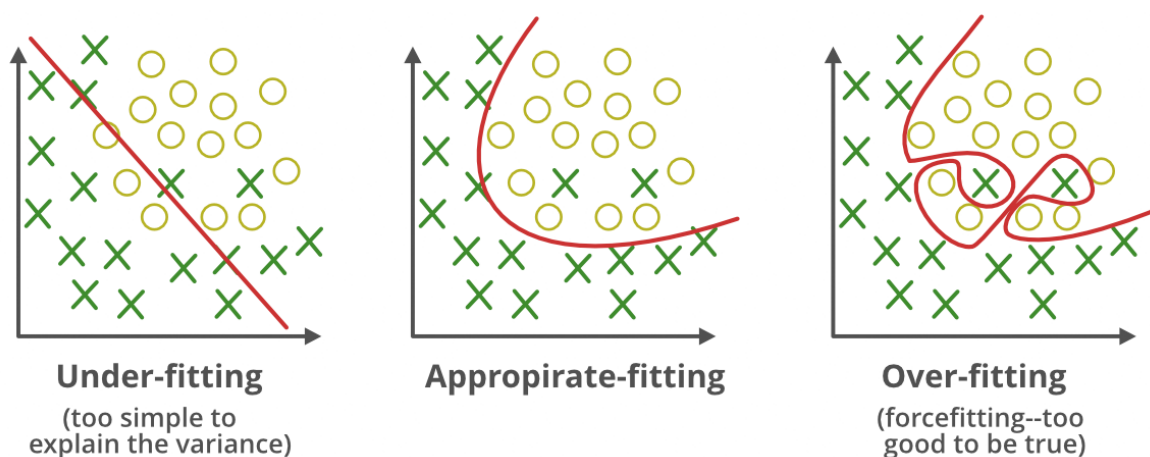


Figura 4.7: Exemplo de *underfitting* e *overfitting* [38]

Cada algoritmo de *Machine Learning* possui diversos parâmetros que podem ser afinados de modo a procurar obter uma melhor *performance* para cada *dataset* específico [6]. A estes parâmetros intrínsecos dos algoritmos dá-se o nome de hiperparâmetros

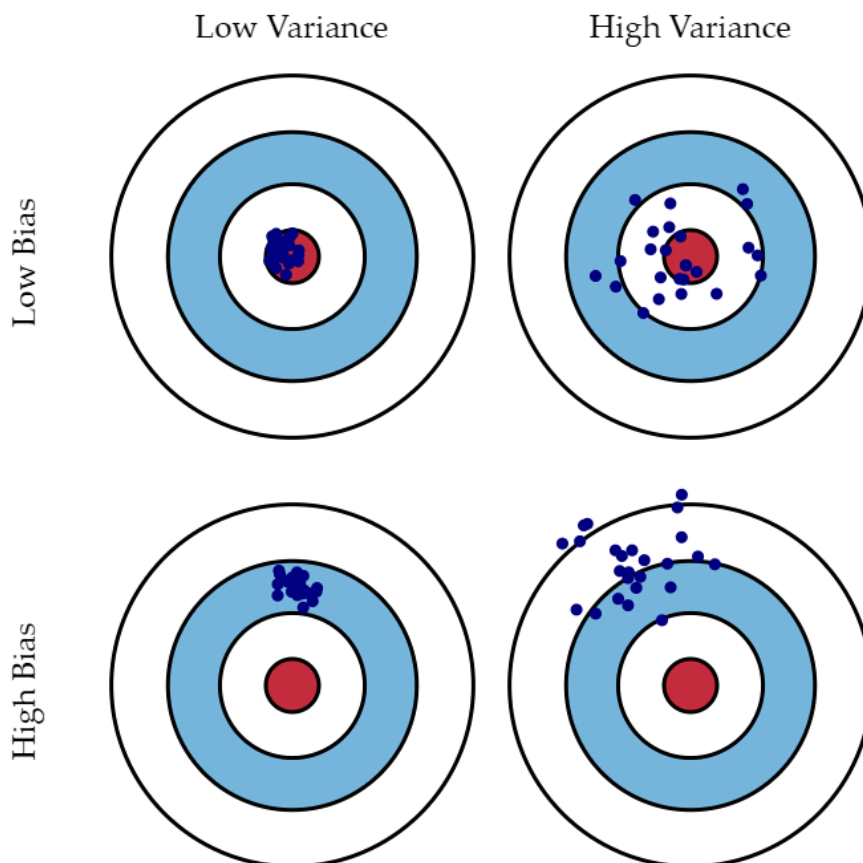


Figura 4.8: Ilustração gráfica de *Bias* e *Variance* [37]

[6].

Um problema comum neste fase é, ao afinar os hiperparâmetros de modo a procurar diminuir o erro de generalização no conjunto de teste, a *performance* do modelo para novos dados pode piorar [6]. Ora, ao realizar uma segunda divisão do conjunto de treino em dois subconjuntos: subconjunto de treino e subconjunto de desenvolvimento (Figura 4.9), é possível resolver este problema. A esta técnica de validação dá-se o nome de *Holdout* [39]. O processo de treino dos algoritmos é realizado no subconjunto de treino e o novo subconjunto de desenvolvimento fornece uma avaliação imparcial do ajuste do modelo aos dados de treino permitindo simultaneamente afinar o modelo e hiperparâmetros de modo a procurar uma melhor *performance* [35, 39]. Por fim, o conjunto de teste é utilizado para obter uma estimativa do erro de generalização para o modelo. Caso fosse apenas considerada a avaliação feita sobre o subconjunto de desenvolvimento corria-se o risco de se lançar um sistema com uma *performance* inferior à esperada, uma vez que os hiperparâmetros são afinados para melhorar as métricas de avaliação nesse subconjunto [35, 39]. O conjunto de teste contém dados que o modelo nunca “viu” anteriormente, permitindo assim simular o comportamento do modelo com dados futuros [6]. Esta avaliação poderá ser generalizada para o desempenho operacional do modelo [6, 8, 13].

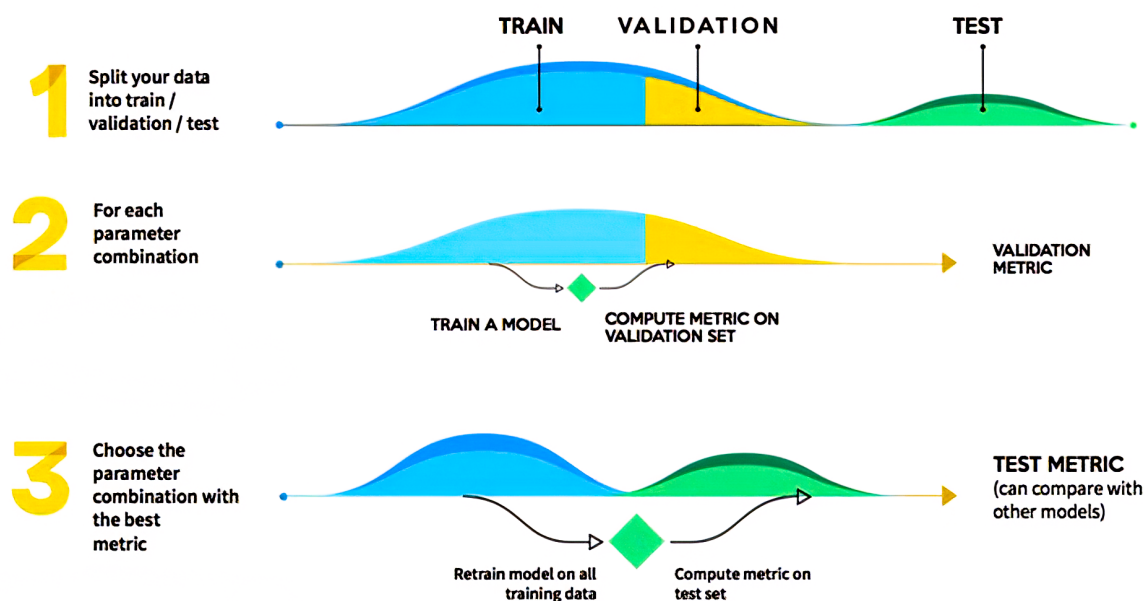


Figura 4.9: Técnica de validação *Holdout* [39]

Métricas de Avaliação

O processo de avaliação dos modelos é essencial em qualquer projeto de *Machine Learning* [6, 35]. Existem várias métricas de avaliação distintas, sendo que, é necessário escolher com cuidado quais usar, uma vez que a *performance* dos modelos será inteiramente dependente da métrica escolhida [5, 6, 35].

A métrica mais simples é a *Accuracy* (Equação 4.1), que representa a razão entre o número de previsões efetuadas com sucesso e o número total de previsões efetuadas [6]. No entanto, quando existe desequilíbrio entre classes, esta métrica pode não ser a mais indicada [6].

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4.1)$$

A matriz de confusão (*Confusion Matrix*) - (Tabela 4.1) - permite determinar como o modelo se comporta na previsão de cada categoria [6]. Regra geral, cada linha da matriz de confusão representa a categoria real, enquanto que as colunas representam as categorias previstas pelo modelo [6]. Na primeira linha são representados os verdadeiros-positivos (*True Positive* - TP) e os falsos-negativos (*False Negative* - FN) e na segunda linha são representados os falsos-positivos (*False Positive* - FP) e os verdadeiros-negativos (*True Negative* - TN) [6].

No entanto, ao avaliar modelos de *Machine Learning* é conveniente utilizar métricas mais concisas, baseadas na matriz de confusão: *Precision* e *Recall* [6, 35].

A *Precision* representa a razão entre o número de verdadeiros-positivos e a soma de

Tabela 4.1: Exemplo de uma Matriz de Confusão para um problema de classificação binária

		Previsão	
		Positivo	Negativo
Real	Positivo	TP	FN
	Negativo	FP	TN

verdadeiros-positivos com falsos-positivos (Equação 4.2), medindo assim a percentagem de previsões positivas que o são realmente [6].

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

O *Recall* define-se como a razão entre o número de verdadeiros-positivos e a soma de verdadeiros-positivos com falsos negativos (Equação 4.3) e representa a capacidade do modelo em detetar os exemplos positivos [6].

$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

Contudo, as métricas *Precision* e *Recall* não são independentes: o aumento de uma implica a diminuição da outra [6].

O F_1 score - Equação 4.4 - é uma média harmónica destas duas métricas que dá mais peso a valores mais baixos, pelo que um modelo com um elevado F_1 score tem *Precision* e *Recall* elevados [6].

$$F_1 \text{ Score} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FN + FP} \quad (4.4)$$

Estas métricas são, regra geral, mais importantes para problemas de classificação binária [5, 6], no entanto, permitem também tirar conclusões importantes em problemas de classificação multi-classe [40]. No Capítulo 5 será realizada uma aplicação destas métricas de avaliação num problema de classificação multi-classe. A *Accuracy*, para classificação multi-classe, define-se novamente como a fração de exemplos corretamente classificados. Uma vez mais, quando existe um desequilíbrio entre classes, a *Accuracy* pode não a métrica de avaliação mais adequada [6]. Em relação às métricas *Precision*, *Recall* e F_1 Score para problemas multi-classe, estas são calculadas, para cada uma das classes, de forma semelhante aos problemas binários, obtendo assim, para cada uma das classes, um valor de *Precision*, *Recall* e F_1 Score. De seguida, é possível realizar uma média sob todas as classes [6]. Esta média pode ser realizada de acordo com uma das seguintes estratégias [6]:

- “*Macro Averaging*”: É realizada uma média não ponderada com igual peso para todas as classes, independentemente do seu tamanho relativo;
- “*Weighted Averaging*”: É realizada uma média ponderada tendo em consideração o tamanho relativo de cada classe;
- “*Micro Averaging*”: É calculado o número total de falsos positivos, falsos negativos e verdadeiros positivos em todas as classes e, de seguida, utilizando estes valores, as métricas de avaliação multi-classe são calculadas.

4.5. Métodos de *Machine Learning* para manutenção preditiva

Nesta secção serão apresentadas algumas considerações acerca da aplicação de métodos de *Machine Learning* para manutenção preditiva.

No que diz respeito às métricas de avaliação, regra geral, o mais importante é o número de falhas reais que o modelo é capaz de prever, ou seja, o *Recall* - Equação 4.3 - do modelo [41]. Este parâmetro torna-se ainda mais importante à medida que as consequências de falsos negativos, isto é, falhas verdadeiras que o modelo não foi capaz de prever, excedem as consequências de falsos positivos, ou seja, uma previsão falsa de uma falha [40, 41].

Relativamente aos dados utilizados, é necessário que estes sejam relevantes para o problema. Geralmente, é recomendado conceber sistemas de previsão sobre componentes específicos e não subsistemas maiores, uma vez que estes terão dados mais dispersos [41].

Em relação à quantidade de dados necessários, são geralmente colocadas duas questões: quantos eventos de falha são necessários para treinar um modelo e quantos registos são considerados “suficientes”. A resposta a estas duas perguntas irá depender do sistema, mas, regra geral, quanto maior o número de eventos de falha, melhor a capacidade preditiva do modelo, sendo que, este número depende da relevância dos dados e do contexto do problema que esta a ser resolvido [41]. Por outro lado, se um componente falhar demasiadas vezes, este acabará provavelmente por ser substituído, o que reduz o número de eventos de falha. Será necessário adotar métodos que permitam lidar com “acontecimentos raros” [2, 40].

Em sistemas críticos, eventos de falha são raros. Ainda assim, ao construir modelos de previsão, os algoritmos necessitam de aprender acerca do padrão operacional normal de um componente, assim como os seus padrões de falha [41]. Os dados de treino devem conter um número suficiente de exemplos de ambas as categorias.

Registos de manutenção e histórico de substituição de peças são boas fontes para encontrar eventos de falha [41]. O histórico de manutenção de um ativo pode conter informações acerca de componentes substituídos e atividades de reparação realizadas, registando assim padrões de degradação. A ausência desta informação crucial nos dados de treino pode levar a resultados insatisfatórios [41].

Assim, no que diz respeito à origem dos dados, as fontes relevantes para a manutenção preditiva incluem, mas não se limitam a: histórico de falhas, histórico de manutenção e reparação, condições de funcionamento do sistema e informações acerca dos equipamentos [41].

Os dados baseados em sensores do equipamento em funcionamento são fundamentais para a aplicação de modelos de *Machine Learning*. É de esperar que estes dados contêm características que variam com tempo e que permitem capturar padrões de envelhecimento e quaisquer anomalias que possam conduzir à degradação dos ativos [40, 41].

No entanto, para aplicação dos conceitos de manutenção preditiva, é fundamental ter em consideração os seguintes pressupostos [42]:

- A conexão entre as condições dos sistemas de engenharia, envelhecimento e a longevidade é múltipla e variada;
- Os sistemas de engenharia não são afetados apenas pela passagem do tempo, mas também por fatores como o uso e as condições ambientais;
- O envelhecimento e a degradação de um sistema são um processo possivelmente não linear que se manifesta de várias maneiras;
- Existem sinais precoces de envelhecimento que podem ser detetados antes do final da vida do sistema, ou seja, antes da perda de função;
- É possível modelar o processo de desgaste do equipamento;
- O desempenho dos prognósticos melhora à medida que mais dados se tornam disponíveis. Por outras palavras, a incerteza da previsão pode ser reduzida com o passar do tempo, dado que mais informações estarão disponíveis.

Sem estas considerações, as estratégias de manutenção preditiva não poderiam existir [42].

4.5.1. Revisão sistemática da literatura científica

Em [15], é realizada uma revisão sistemática da literatura científica de métodos de *Machine Learning* aplicados a manutenção preditiva, de modo a tentar responder às seguintes questões:

- Q1. Quais os algoritmos de *Machine Learning* mais utilizados para a manutenção preditiva?
- Q2. Em que equipamentos são aplicados este métodos?
- Q3. Quais os conjuntos de dados utilizados?
- Q4. O conjunto de dados é real ou sintético?
- Q5. Como é que os algoritmos de *Machine Learning* são aplicados no contexto da manutenção preditiva?

A pesquisa das publicações científicas em [15] foi realizada a 18 de outubro de 2018, em duas bases de dados de referência: IEEEExplore Digital Library¹ e ScienceDirect². A escolha das palavras-chave para criar a sequência de pesquisa foi baseada em termos comuns encontrados na literatura relacionados com o tema e é apresentada de seguida:

- **IEEE Xplore:** (“predictive maintenance” OR “PdM”) AND (“Machine Learning” or “Machine Learning technique”)
- **ScienceDirect:** title-abstr-key((“predictive maintenance” OR “PdM”) AND (“Machine Learning” or “Machine Learning technique”))

Três critérios de exclusão foram definidos [15]:

- E1. Trabalhos não relacionados com manutenção preditiva e *Machine Learning*
- E2. Trabalhos que não apresentam nenhum tipo de experimentação ou resultados de comparações.
- E3. Trabalhos publicados antes de 2009.

De acordo com os critérios de exclusão definidos, das 54 publicações científicas encontradas, 36 foram selecionadas para análise [15]. Na Figura 4.10 está representado o número de publicações científicas, por ano, de acordo com os critério estabelecidos [15]. É possível verificar uma tendência de aumento do número de publicações nesta área.

Contudo, segundo [43], uma razão que pode explicar o ainda baixo número de publicações científicas é a complexidade de implementação de estratégias de manutenção preditiva em ambientes industriais. Mais ainda, o recurso a algoritmos de *Machine Learning* em manutenção preditiva necessita de profissionais especialista em Inteligência Artificial e/ou *Data Science*.

Na Tabela 4.2 é apresentado um resumo da revisão sistemática da literatura científica realizada em [15], da qual é possível retirar diversas conclusões. As estratégias de manutenção preditiva estão a ser aplicadas aos mais diversos equipamentos, em múltiplos

¹<http://ieeexplore.ieee.org>

²<http://www.sciencedirect.com>

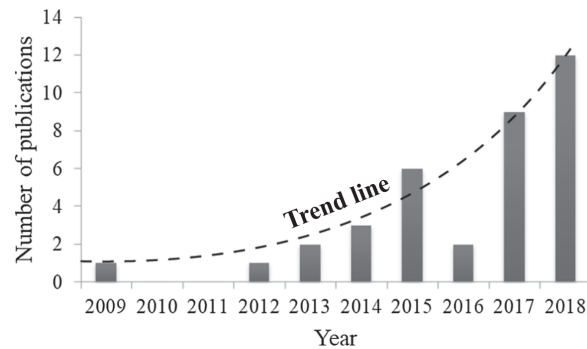


Figura 4.10: Número de publicações científicas, por ano, de acordo com os critério estabelecidos [15]

tipias áreas [15]. Os equipamentos onde estes métodos são aplicados incluem, mas não se limitam a, turbinas, motores, compressores, bombas [15]. Cerca de 89% dos artigos utiliza um conjunto de dados reais, sendo que 11% recorre a dados sintéticos [15]. Em relação à utilização dos algoritmos de *Machine Learning* nas publicações científicas analisadas em [15], o mais utilizado é *Random Forest (RF)* - 33%, seguido de métodos baseados em *Neural Networks (NN)*, como por exemplo *Artificial NN*, *Convolution NN*, *Long Short-Term Memory Network (LSTM)* e *Deep Learning* - 27%, *Support Vector Machine (SVM)* - 25% e *k-means* - 13% [15]. Foi também verificado uma maior tendência de uso de sinais de vibração [15].

Tabela 4.2: Resumo da revisão sistemática da literatura científica mais recente para manutenção preditiva com métodos de *Machine Learning* - Adaptado de [15]

Referência	Algoritmo de ML	Equipamento	Conjunto de Dados
(Onanena et al., 2009) [44]	LR	Célula de combustível	Medições de espectroscopia de impedância eletroquímica
(Hong and Zhou, 2012) [45]	GPR	Rolamentos	Dados de vibração
(Susto et al., 2013) [46]	SVM	Filamento de Tungsténio	Ciclos históricos de manutenção
(Schopka et al., 2013) [47]	LR, RF, BN	Filamento	Dados de processo, equipamentos e logística de avaria
(Li et al., 2014) [48]	SVM	Rede Ferroviária	Dados de falhas, manutenção, inspeção, tipo e clima
(Praveenkumar et al., 2014) [49]	SVM	Caixa de Velocidades Automóvel	Sinais de vibração
(Prytz et al., 2015) [50]	RF	Compressores de ar em caminhões e autocarros	Dados On-board e registos de serviços
(Biswal and Sabareesh, 2015) [51]	ANN	Turbina Eólica	Dados de acelerómetros
(Machado and Mota, 2015) [52]	ANN and SVM	Sistemas de energia elétrica	Sinais elétricos
(Durbhaka and Selvaraj, 2016) [53]	k-NN, SVM, k-means	Rolamentos	Sinais de vibração
(Aydin and Guldamlasioglu, 2017) [54]	LSTM	Motor	Dados operacionais e de medições de sensores
(Canizo et al., 2017) [55]	RF	Turbina Eólica	Dados operacionais e de estado
(Santos et al., 2017) [56]	RF	Motores de Indução	Formas de onda de corrente e tensão
(Su and Huang, 2018) [57]	RF	Disco Rígido	Dados históricos (vibração, temperatura)
(Uhlmann et al., 2018) [58]	k-means	Fusão a Laser	Dados do sensor da máquina-ferramenta
(Amihai et al., 2018) [59]	RF	Bombas Industriais	Sinais de vibração
(Huuhtanen and Jung, 2018) [60]	CNN	Painéis fotovoltaicos	Sinal diário de energia elétrica
(Kulkarni et al., 2018) [61]	RF	Sistemas de refrigeração para supermercados	Dados de sensor de temperatura
(Luo et al., 2018) [62]	Deep Learning	Máquina de Comando Numérico Computadorizado	Sinais de vibração

5. Aplicação de modelos de *Machine Learning*

5.1. Introdução

Ao longo do presente capítulo, o processo de implementação dos algoritmos de *Machine Learning* será detalhado e explicado.

Em primeiro lugar, o conjunto de dados é apresentado, é justificada a sua escolha, analisado e visualizado. De seguida, é realizado o estabelecimento rigoroso e claro dos objetivos da presente aplicação de *Machine Learning*. Posteriormente, o conjunto de dados é processado através de *feature engineering*, criando novas *features* de modo a procurar um melhor desempenho dos modelos.

O conjunto de dados utilizado é chave para a resolução de problemas de *Machine Learning*. Uma escolha sensata de quais os dados a usar e como tratá-los é crucial para melhorar o desempenho dos algoritmos. De acordo com Domingos [35], o *feature engineering* é a chave dos projetos de *Machine Learning* e que, não raras vezes, os sinais medidos não são adequados à aprendizagem, sendo necessário construir *features* a partir deles que o são.

De seguida, o conjunto de dados é dividido em subconjuntos de treino, desenvolvimento e teste e procedeu-se à primeira aplicação de modelos de *Machine Learning*, onde uma variedade de algoritmos foi treinada e avaliada. Como referido anteriormente, o processo de treino dos algoritmos é realizado no subconjunto de treino e o subconjunto de desenvolvimento fornece uma avaliação imparcial do ajuste dos modelos aos dados de treino permitindo simultaneamente afinar o modelo e respetivos hiperparâmetros de modo a procurar uma melhor *performance*. Por fim, o conjunto de teste é utilizado para obter uma estimativa da *performance* do modelo, simulando o seu comportamento para dados futuros.

A implementação descrita neste capítulo será realizada com recurso à linguagem de programação *Python*, com a utilização das *packages Matplotlib, Numpy, Pandas e Scikit-Learn* [63].

5.2. Origem dos Dados

Apesar do crescimento desta área, devido à competitividade empresarial, a partilha de informação sensível desta natureza é rara, o que leva a que o número de *datasets* (relevantes para esta aplicação) disponíveis publicamente seja muito escasso.

Um conjunto de dados com alguma reputação neste campo, disponibilizado pelo *Prognostic Center of Excellence* da NASA em [64], intitulado “*Turbofan Engine Degradation Simulation Data Set*”, é composto por dados obtidos através de simulação com o *software* C-MAPSS, de cenários *run-to-failure* de degradação de turbinas de aviões. No entanto, no âmbito do presente documento, é considerado que uma aplicação mais relevante é conseguida recorrendo ao *dataset* introduzido de seguida, uma vez que este permite demonstrar como é possível combinar dados de fontes distintas, muito comum em manutenção.

Dentro de um ambiente industrial, existe um conjunto de dados bastante completo, disponibilizado pela *Microsoft*, publicado em [7], de relevância para o presente projeto.

Este *dataset* contém dados de cinco fontes distintas:

- Telemetria em tempo real (*Real-time telemetry*)
- Registo de erros
- Histórico de manutenção
- Histórico de falhas
- Informação acerca das máquinas

Os dados foram adquiridos ao longo de um ano (2015) para cem máquinas, à exceção do histórico de manutenção, que contém também registos do ano de 2014 [7].

Para um total de cem máquinas, de quatro modelos distintos, o conjunto de dados contém 876100 registos de telemetria de hora em hora, isto é, 8761 registo por máquina [7]. O registo de erros contém 3919 entradas e o histórico de manutenção 3286 [7]. O histórico de falhas possui 761 registos, ou seja, em média, cerca de 8 registo de falha, por máquina, ao longo do ano de 2015 [7]. Cada máquina possui 4 componentes de interesse para a análise e também 4 sensores, que medem a tensão, a pressão, a vibração e a rotação [7]. Um controlador monitoriza o sistema e é capaz de alertar para a ocorrência de 5 tipos de erro [7].

Após o acesso aos dados, e antes de iniciar o pré-processamento, é importante proceder à exploração e visualização destes.

5.2.1. Telemetria em tempo real (*Real-time telemetry*)

Os dados da telemetria em tempo real consistem em medições provenientes de diferentes sensores (4 por máquina), com a respetiva data e hora associadas. As medições de tensão (“*volt*”), rotação (“*rotate*”), pressão (“*pressure*”) e vibração (“*vibration*”) são adquiridas em tempo real e a média destas medições durante uma hora é registada - Tabela 5.1.

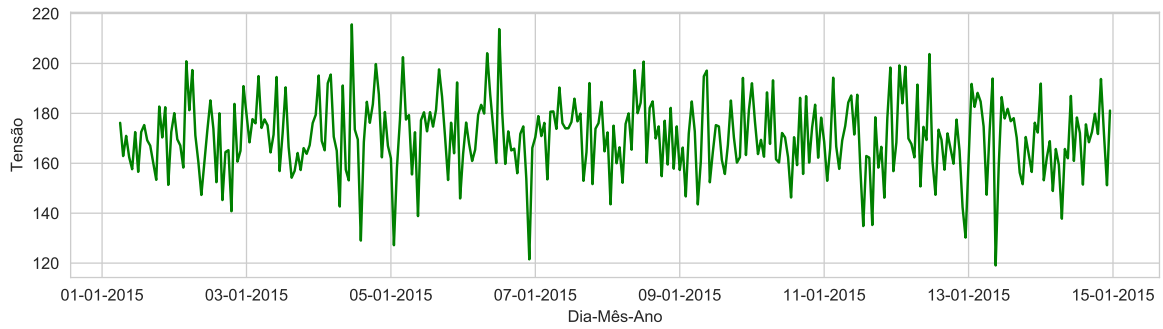
Tabela 5.1: Exemplo típico do registo de telemetria em tempo real

	datetime	machineID	volt	rotate	pressure	vibration
0	2015-01-01 06:00:00	1	176.217853	418.504078	113.077935	45.087686
1	2015-01-01 07:00:00	1	162.879223	402.747490	95.460525	43.413973
2	2015-01-01 08:00:00	1	170.989902	527.349825	75.237905	34.178847
3	2015-01-01 09:00:00	1	162.462833	346.149335	109.248561	41.122144
4	2015-01-01 10:00:00	1	157.610021	435.376873	111.886648	25.990511

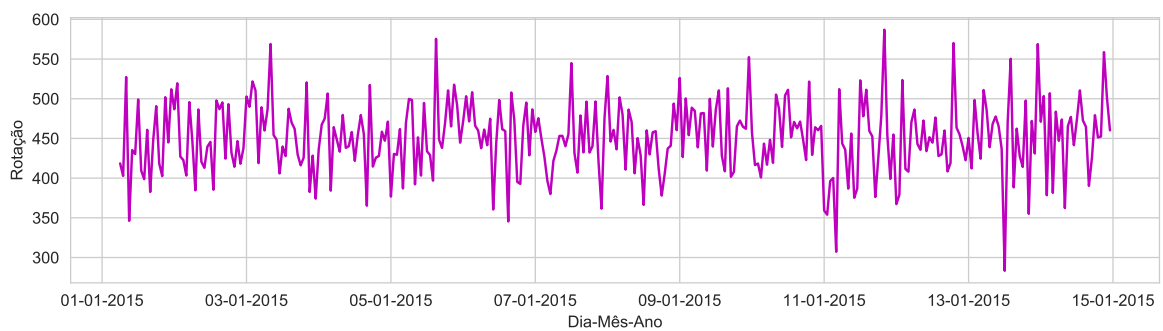
Para melhor compreender o comportamento de cada sensor, na Tabela 5.2 é realizada uma análise estatística simples, onde a média, o desvio-padrão, o valor mínimo e máximo são calculados para os parâmetros tensão (“*volt*”), rotação (“*rotate*”), pressão (“*pressure*”) e vibração (“*vibration*”) durante o ano de 2015. A título de exemplo, na Figura 5.1 estão representadas as evoluções gráficas dos sinais de Tensão (Figura 5.1a), Rotação (Figura 5.1b), Pressão (Figura 5.1c) e Vibração (Figura 5.1d), ao longo do primeiros quinze dias do mês de janeiro de 2015, para a máquina 1 (*machineID=1*).

Tabela 5.2: Análise estatística aos dados da telemetria em tempo real

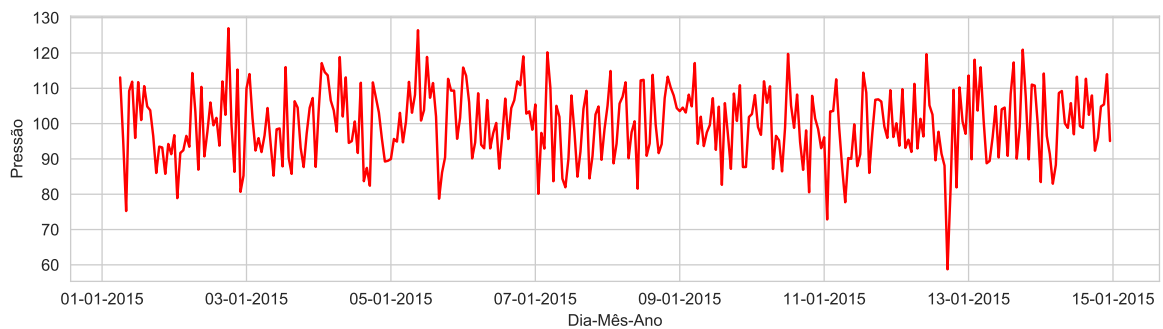
	<i>volt</i>	<i>rotate</i>	<i>pressure</i>	<i>vibration</i>
count	876100	876100	876100	876100
mean	170.777736	446.605119	100.858668	40.385007
std	15.509114	52.673886	11.048679	5.370361
min	97.333604	138.432075	51.237106	14.877054
25%	160.304927	412.305714	93.498181	36.777299
50%	170.607338	447.558150	100.425559	40.237247
75%	181.004493	482.176600	107.555231	43.784938
max	255.124717	695.020984	185.951998	76.791072



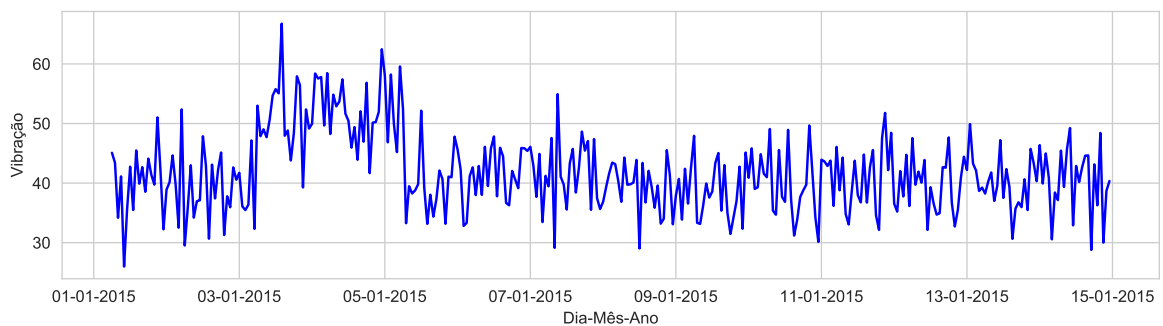
(a) Sinal de Tensão



(b) Sinal de Rotação



(c) Sinal de Pressão



(d) Sinal de Vibração

Figura 5.1: Evolução dos dados de Telemetria ao longo dos primeiros quinze dias do mês de Janeiro de 2015, para a máquina 1

5.2.2. Registo de Erros

A segunda fonte de informação é o registo de erros. Estes são erros que não conduziram de imediato a uma falha, sendo que a máquina permaneceu operacional. Existem 5 tipos de erros: *error1*, *error2*, *error3*, *error4*, *error5*. A data e hora é arredondada para a hora mais próxima. Cada registo consiste numa data/hora, máquina e tipo de erro - Tabela 5.3. O número total de registos de erros ao longo do ano de 2015 é de 3919. Na Figura 5.2 é possível observar o número de erros por tipo ao longo do ano de 2015.

Tabela 5.3: Exemplo típico do registo de erros

	datetime	machineID	errorID
0	2015-01-03 07:00:00	1	error1
1	2015-01-03 20:00:00	1	error3
2	2015-01-04 06:00:00	1	error5
3	2015-01-10 15:00:00	1	error4
4	2015-01-22 10:00:00	1	error4

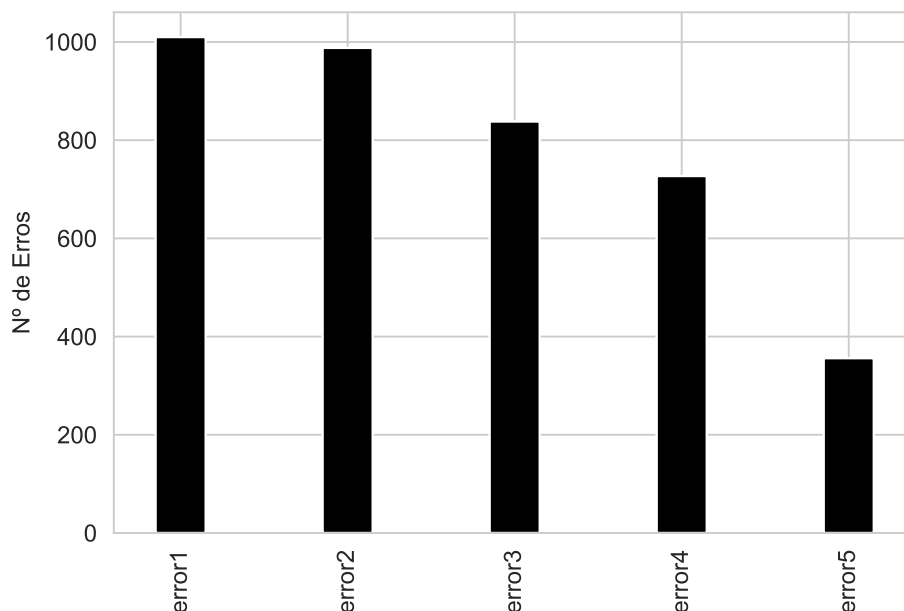


Figura 5.2: Representação do número de erros por tipo

5.2.3. Histórico de Manutenção

O histórico de manutenção contém registos de substituição de componentes fruto de uma intervenção de manutenção quer programada quer não-programada como, por exemplo, inspeções periódicas, degradação de *performance* ou falha de um componente. Em caso de intervenção de manutenção devido à ocorrência de uma falha num componente, é gerado igualmente um registo de falha - subsecção 5.2.4.

Para cada máquina, este conjunto de dados contém informação acerca de 4 tipos de componentes: *comp1*, *comp2*, *comp3*, *comp4*. A data e hora é arredondada para a hora mais próxima. Cada registo consiste numa data/hora, máquina e o tipo de componente substituído - Tabela 5.4. O número total de registos de manutenção ao longo do ano de 2015 é de 3286. Como referido anteriormente, o histórico de manutenção contém também entradas de 2014. Na Figura 5.3 é representado o número de componentes substituídos, por tipo. É possível observar que o número de substituições é próximo para os 4 tipos de componentes.

Tabela 5.4: Exemplo típico do histórico de manutenção

	datetime	machineID	comp
0	2014-06-01 06:00:00	1	comp2
1	2014-07-16 06:00:00	1	comp4
2	2014-07-31 06:00:00	1	comp3
3	2014-12-13 06:00:00	1	comp1
4	2015-01-05 06:00:00	1	comp4

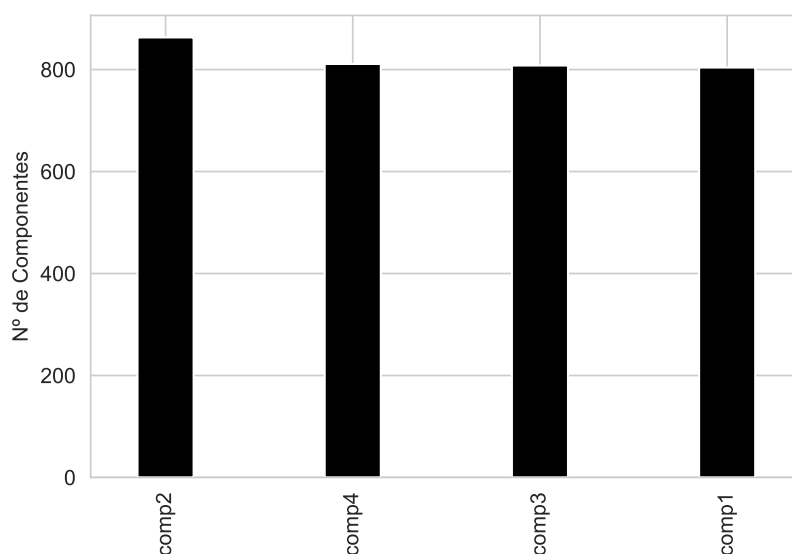


Figura 5.3: Representação do número de componentes substituídos, por tipo

5.2.4. Histórico de Falhas

O histórico de falhas contém os registos de substituição de componentes, resultado da intervenção de manutenção, devido à ocorrência de uma falha. Existem 4 tipos de componentes: *comp1*, *comp2*, *comp3*, *comp4*. A data e hora é arredondada para a hora mais próxima. Cada registo consiste numa data/hora, máquina e o tipo de componente substituído - Tabela 5.5. O número total de registos de falhas ao longo do ano de 2015 é de 761. Na Figura 5.4 é possível observar o número de substituições, por tipo de componente, ao longo do ano de 2015, devido à ocorrência de uma falha.

Tabela 5.5: Exemplo típico do histórico de falhas

	datetime	machineID	failure
0	2015-01-05 06:00:00	1	comp4
1	2015-03-06 06:00:00	1	comp1
2	2015-04-20 06:00:00	1	comp2
3	2015-06-19 06:00:00	1	comp4
4	2015-09-02 06:00:00	1	comp4

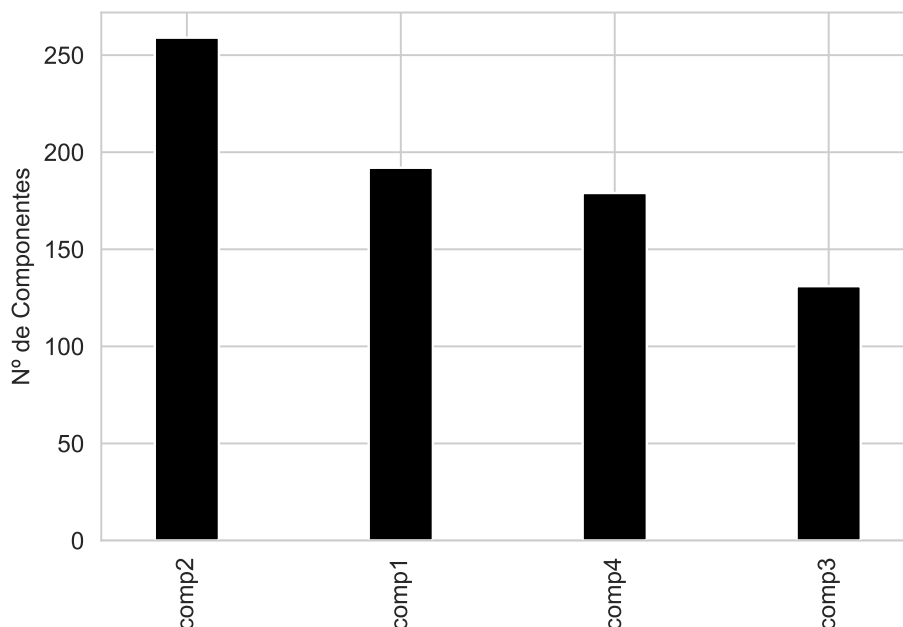


Figura 5.4: Representação do número de componentes substituídos, por tipo, devido à ocorrência de uma falha

5.2.5. Informação acerca das máquinas

Por fim, este conjunto de dados contém informações acerca do modelo e número de anos de serviço de cada uma das 100 máquinas - Tabela 5.6. Na Figura 5.5 é representado um histograma que mostra a distribuição do número de máquinas e tempo de serviço, por modelo.

Tabela 5.6: Exemplo típico das informações de cada máquina

	machineID	model	age
0	1	model3	18
1	2	model4	7
2	3	model3	8
3	4	model3	7
4	5	model3	2
5	6	model3	7
6	7	model3	20
7	8	model3	16

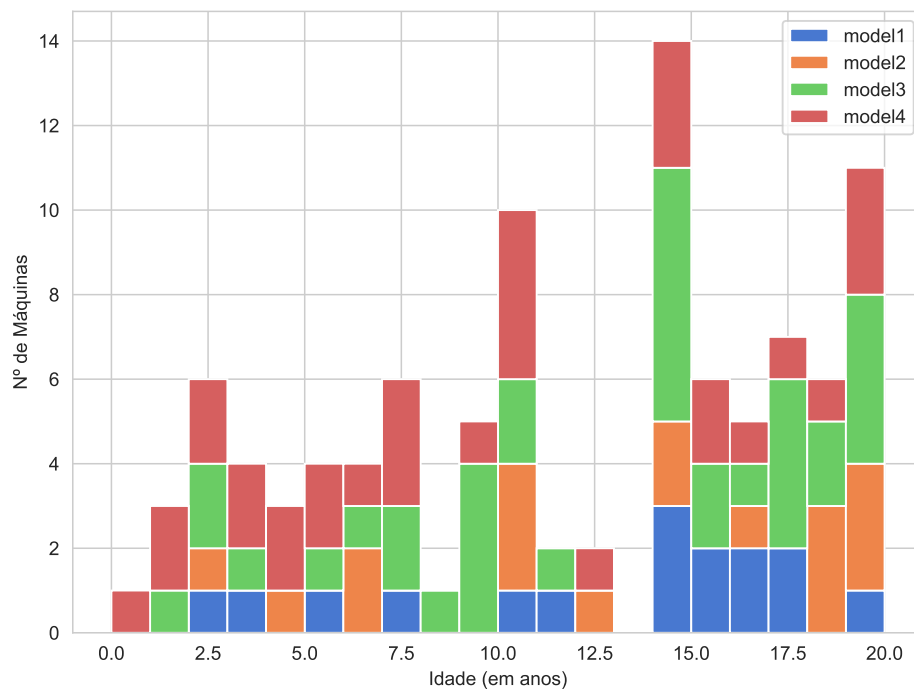


Figura 5.5: Histograma representativo do número de máquinas e tempo de serviço, por modelo

5.3. Definição dos objetivos

Como já referido, um projeto de *Machine Learning* deve começar pelo estabelecimento rigoroso e claro dos objetivos. Neste caso, o principal objetivo dos modelos utilizados será prever a probabilidade de ocorrência de uma falha dentro da janela temporal definida. Mais especificamente, a probabilidade de ocorrência de uma falha numa máquina nas próximas 24 horas (duração da janela temporal escolhida para esta aplicação) relacionada com um dos componentes (componentes 1,2,3 ou 4).

De seguida, dado que já se estabeleceu um objetivo particular e claro, pode começar a fazer-se perguntas mais específicas quanto ao *Machine Learning* propriamente dito¹: 1) deve optar-se por modelos de *supervised*, *unsupervised*, *reinforcement learning* ou, eventualmente, por combinações de modos de aprendizagem? 2) se *supervised learning*, classificação ou regressão? 3) pretende-se que os modelos treinem imediatamente à medida que novos dados são obtidos (*batch learning* ou *online learning*)?

Após análise do problema, e tendo presente o objetivo proposto, optou-se por *supervised learning* e, em particular, classificação. Mais ainda, tendo em conta a existência de 4 componentes distintos em análise, o problema será de classificação multi-classe (“*Multiclass Classification*”). Considerou-se igualmente que não será necessário, dado o âmbito do problema e a natureza dos dados, que os modelos treinem imediatamente à medida que novos dados são obtidos. Assim sendo, estamos perante um problema de *batch learning*.

5.4. *Feature Engineering*

Uma *feature* é um atributo preditivo para o modelo [35]. O objetivo do *feature engineering* é procurar aumentar o poder preditivo dos algoritmos de *Machine Learning*, criando novas *features* a partir dos dados disponíveis (Figura 5.6) [6, 35]. Regra geral, o *feature engineering* é realizado em primeiro lugar sendo que, de seguida, ocorre a seleção de *features*, eliminando *features* irrelevantes, redundantes ou com alta correlação [6, 35]. Tendo como ponto de partida as diferentes fontes de informação apresentadas na secção anterior, será criado um único *dataset*, que será usado para a aplicação dos modelos preditivos.

¹De acordo com a Secção 4.2

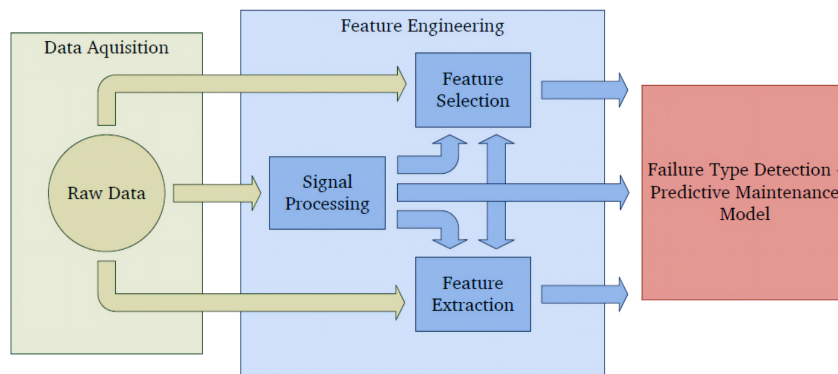


Figura 5.6: Feature Engineering [65]

5.4.1. Lag Features

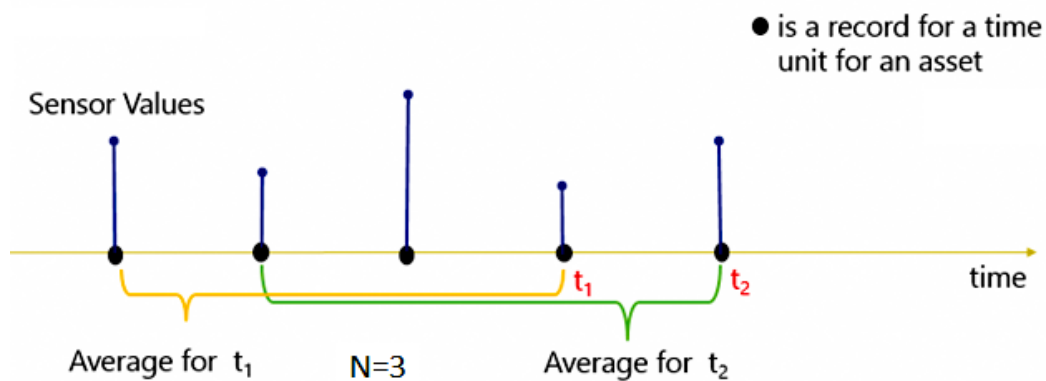
Os dados históricos a que os modelos têm acesso são momentos individuais no passado. Em particular para os dados de telemetria, é possível a existência de perturbações resultantes das medições, como ruído, dificultando assim a tarefa preditiva. Deste modo, os dados podem ser agregados em janelas de tempo, permitindo assim “suavizar” os valores, minimizando os efeitos do ruído nas *features* utilizadas pelos modelos.

Tendo em consideração até que ponto no futuro o modelo deverá ser capaz de prever, de acordo com os requisitos do projeto, é importante definir até onde deve “olhar” para realizar estas previsões. A este intervalo de tempo passado para onde o modelo deve “olhar para trás” dá-se o nome de *lag*. Diversas *features* podem ser extraídas destes intervalos de tempo - *lag features*. O conjunto de dados utilizado para gerar *lag features* tem normalmente associado uma data/hora.

Para cada registo, uma janela temporal de dimensão N é criada e as *lag features* são calculadas para o período N anterior à data/hora desse registo. A Figura 5.7 mostra um exemplo desta aplicação para uma medição t_i com $N=3$. O valor de N é tipicamente em minutos ou horas, dependendo da natureza dos dados.

Assim, duas janelas temporais foram criadas. A primeira, de 3 horas, com fim a permitir retratar o comportamento dos dados de telemetria a curto prazo (Tabela 5.7) e a segunda, de 24 horas, de modo a representar a evolução a longo prazo (Tabela 5.8). Em cada um destes intervalos de tempo foram calculados, de 3 em 3 horas, dois novos parâmetros para cada uma das *features* dos dados de telemetria: a média móvel e o desvio-padrão. De notar que, no caso de $N=24$ horas (Tabela 5.8), naturalmente os dois novos parâmetros não estão disponíveis para os instantes iniciais (primeiras 24 horas).

Assim como os dados de telemetria, o registo de erros também possui associado uma

Figura 5.7: *Lag Features* [7]Tabela 5.7: Exemplo de *Lag Features* para os dados de telemetria em tempo real, com $N=3$

machineID	datetime	voltmean_3h	rotatemean_3h	pressuremean_3h	vibrationmean_3h	
0	1	2015-01-01 09:00:00	170.028993	449.533798	94.592122	40.893502
1	1	2015-01-01 12:00:00	164.192565	403.949857	105.687417	34.255891
2	1	2015-01-01 15:00:00	168.134445	435.781707	107.793709	41.239405
3	1	2015-01-01 18:00:00	165.514453	430.472823	101.703289	40.373739
4	1	2015-01-01 21:00:00	168.809347	437.111120	90.911060	41.738542

Tabela 5.8: Exemplo de *Lag Features* para os dados de telemetria em tempo real, com $N=24$

machineID	datetime	voltmean_24h	rotatemean_24h	pressuremean_24h	vibrationmean_24h	
7	1	2015-01-02 06:00:00	169.733809	445.179865	96.797113	40.385160
8	1	2015-01-02 09:00:00	170.614862	446.364859	96.849785	39.736826
9	1	2015-01-02 12:00:00	169.893965	447.009407	97.715600	39.498374
10	1	2015-01-02 15:00:00	171.243444	444.233563	96.666060	40.229370
11	1	2015-01-02 18:00:00	170.792486	448.440437	95.766838	40.055214

data/hora. No entanto, estes dados são categóricos e não numéricos. Neste caso, é realizada a soma do número de erros de cada tipo, de 3 em 3 horas, para a janela temporal $N=24$ (Tabela 5.9). Cada linha da Tabela 5.9 representa a soma do número de erros de cada tipo nas 24 horas anteriores ao *datetime* assinalado.

5.4.2. *Features* adicionais

O registo de manutenção, que contém a informação relacionada com a substituição de componentes, permite gerar novas *features* potencialmente importantes, como, por exemplo, há quanto tempo ocorreu a última substituição de um determinado

Tabela 5.9: Exemplo de *Lag Features* para o registo de erros

	machineID	datetime	error1count	error2count	error3count	error4count	error5count
15	1	2015-01-03 06:00:00	0.0	0.0	0.0	0.0	0.0
16	1	2015-01-03 09:00:00	0.0	0.0	0.0	0.0	0.0
17	1	2015-01-03 12:00:00	1.0	0.0	0.0	0.0	0.0
18	1	2015-01-03 15:00:00	1.0	0.0	0.0	0.0	0.0
19	1	2015-01-03 18:00:00	1.0	0.0	0.0	0.0	0.0
20	1	2015-01-03 21:00:00	1.0	0.0	0.0	0.0	0.0
21	1	2015-01-04 00:00:00	1.0	0.0	1.0	0.0	0.0
22	1	2015-01-04 03:00:00	1.0	0.0	1.0	0.0	0.0
23	1	2015-01-04 06:00:00	1.0	0.0	1.0	0.0	0.0
24	1	2015-01-04 09:00:00	1.0	0.0	1.0	0.0	1.0

componente - Tabela 5.10. É de esperar que esta *feature* se relacione bem com as possíveis falhas dos componentes, dado que, quanto maior o tempo de utilização de um componente, maior será a degradação esperada.

É relevante notar que, a criação de *features* com base em dados de manutenção não é tão linear como nos casos anteriores. No entanto, este tipo de *feature engineering* específico caso a caso é muito comum na manutenção preditiva, onde o conhecimento do domínio e a experiência cumprem um papel crucial na compreensão e criação de *features* com relevância.

Tabela 5.10: Tempo desde a última substituição, por tipo de componente

	datetime	machineID	comp1	comp2	comp3	comp4
0	2015-01-01 06:00:00	1	19.000000	214.000000	154.000000	169.000000
1	2015-01-01 07:00:00	1	19.041667	214.041667	154.041667	169.041667
2	2015-01-01 08:00:00	1	19.083333	214.083333	154.083333	169.083333
3	2015-01-01 09:00:00	1	19.125000	214.125000	154.125000	169.125000
4	2015-01-01 10:00:00	1	19.166667	214.166667	154.166667	169.166667

Por último, a informação acerca das máquinas pode ser utilizada sem modificações adicionais, isto é, a informação relacionada com o modelo e número de anos em serviço de cada uma das máquinas - Tabela 5.6.

5.5. Seleção de *Features*

Foi realizada uma análise da correlação linear entre as variáveis (Figura 5.8). O coeficiente de correlação varia entre -1 e 1 . Este coeficiente permite perceber se uma variável justifica a variação linear de outra. Quando está próximo de 1 , significa que existe uma correlação positiva acentuada, isto é, se determinada *feature A* aumenta,

então, a *feature B* também aumenta e se *A* diminui *B* também diminuirá.

Neste caso, verifica-se que a correlação entre as *features* é maioritariamente baixa ou inexistente (coeficiente de correlação próximo de zero). Ainda assim, no caso das *features* *pressuremean_3h* e *pressuremean_24h*, o valor do coeficiente de correlação é aproximadamente 0.5 e será relevante uma análise mais detalhada.

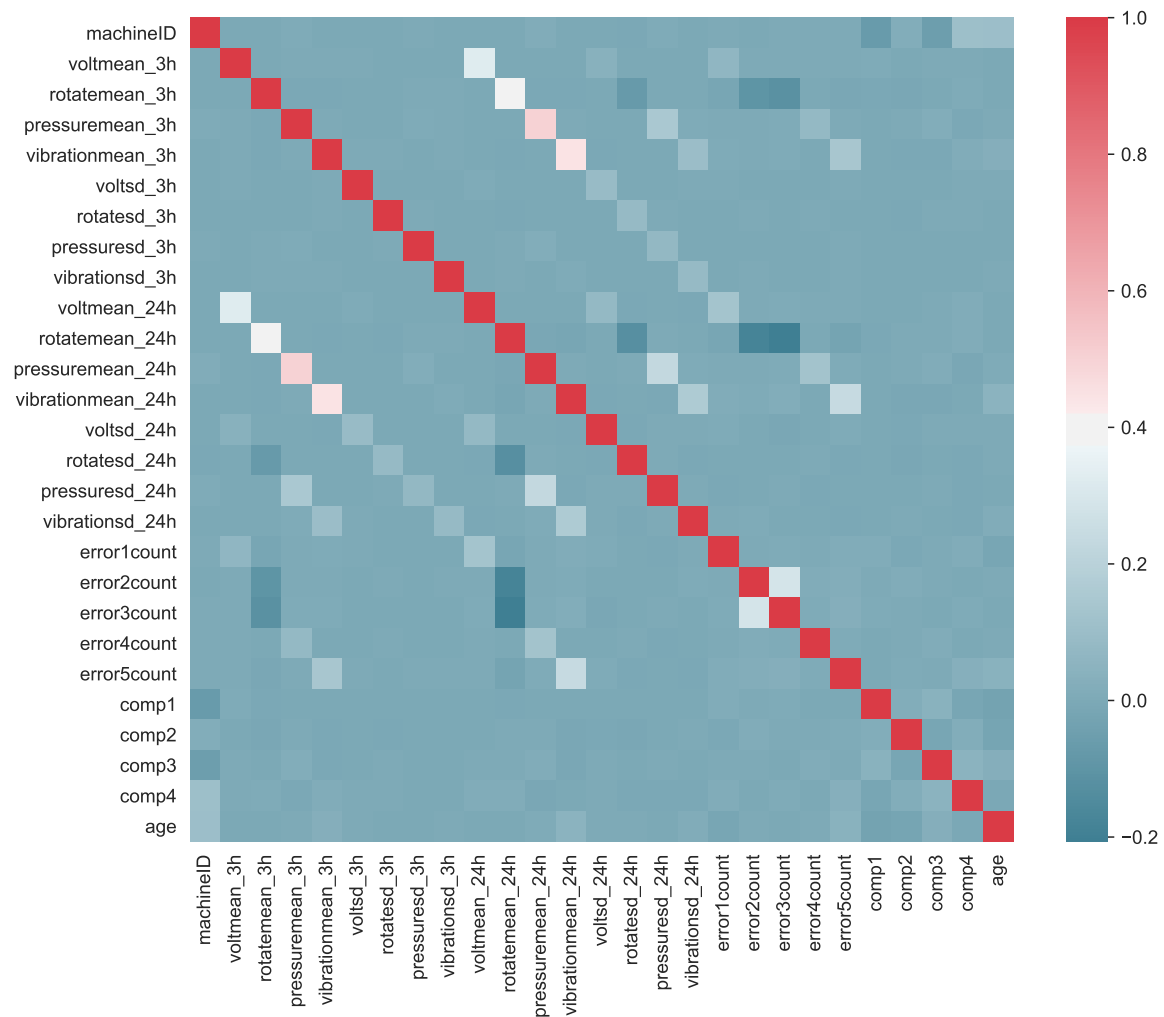


Figura 5.8: Correlação entre *Features*

Assim, na Figura 5.9 estão representadas as falhas, por tipo de componentes, em função da evolução das *features* *pressuremean_3h* e *pressuremean_24h*. É possível observar que, para os componentes 1 e 3, existem aglomerados (*clusters*) de pontos. No entanto, o mesmo não se verifica para os componentes 2 e 4. É, de igual modo, possível observar que a maioria das falhas para o componente 3 ocorre para valores de *pressuremean_3h* e *pressuremean_24h* mais elevados, quando comparado com os outros componentes. Deste modo, decidiu-se manter ambas as *features*, uma vez que existe um clara relação entre estas e a ocorrência de falhas em, pelo menos, alguns dos componentes.

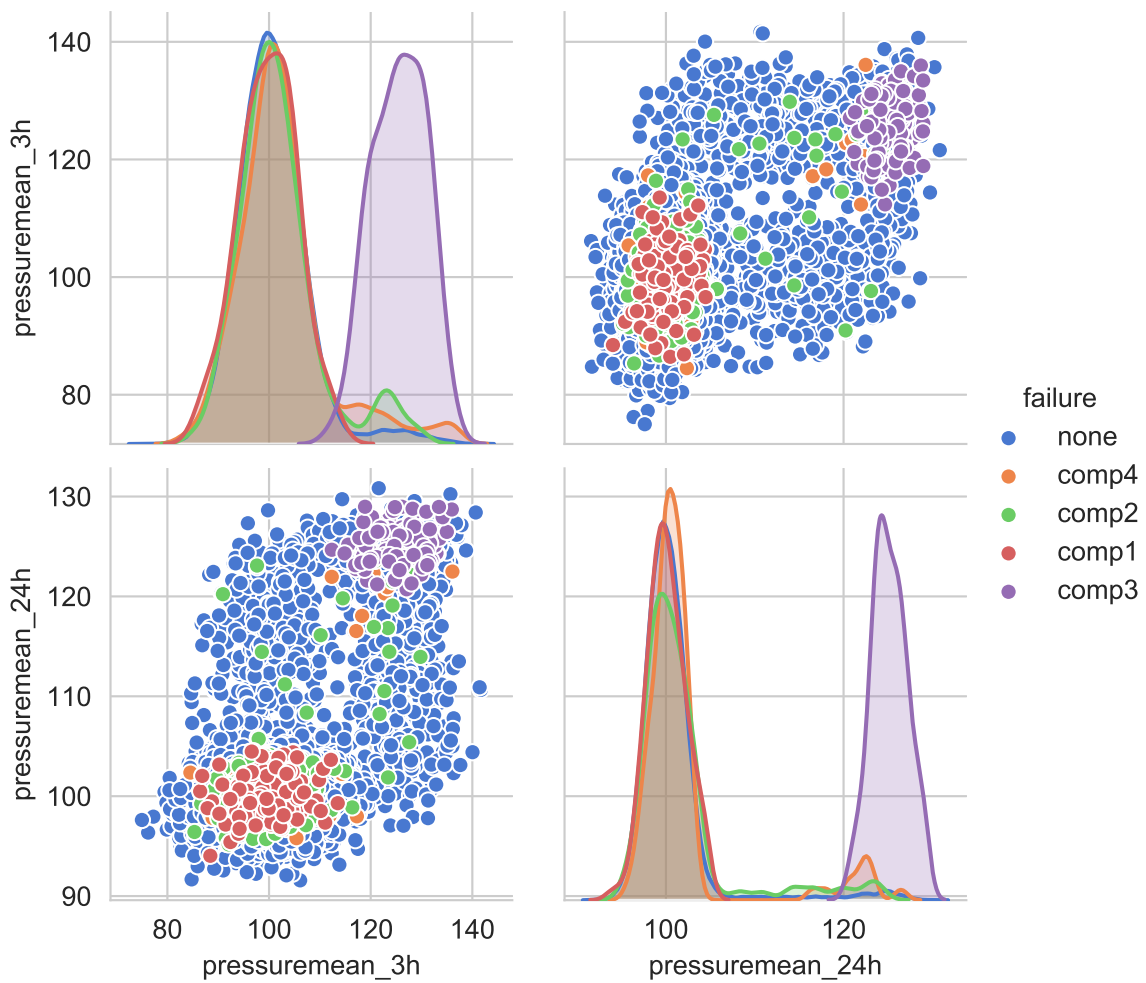


Figura 5.9: Análise detalhada da correlação entre as *features* *pressuremean_3h* e *pressuremean_24h*

5.6. Classificação dos dados e construção de *Labels*

Como referido anteriormente, o problema de manutenção preditiva em análise é um caso de *Supervised Learning*². De forma a treinar um modelo para prever falhas, é necessário não só exemplos de falha como também uma série temporal de observações que conduziram a essa falha. Mais ainda, o modelo precisa de exemplos de períodos de operação “normal” de modo a ser capaz de perceber a diferença entre os dois. A classificação entre estes dois estados é binária (estável ou sem falha/instável ou com falha).

Com esta informação disponível (estável/instável), o modelo apenas é útil caso seja capaz de alertar de forma atempada para a iminência de ocorrência de uma falha. De modo a cumprir este critério de aviso prévio, é necessário modificar a definição do rótulo de *evento de falha*, que ocorre num momento específico de tempo, para um intervalo temporal onde o evento de falha poderá ocorrer. O tempo até a ocorrência da falha, que delimita a fronteira entre as duas categorias, deve ser escolhido de acordo com critérios operacionais. Será o conhecimento de que uma falha irá ocorrer dentro de 12 horas suficiente para a impedir? E 24 horas? E duas semanas? A capacidade do modelo em prever uma falha dependerá também da duração desta janela temporal.

Na Figura 5.10 é ilustrado este processo. De modo a alcançar a redefinição de *instável* para *pré-instável*, as observações dentro da janela temporal (representada por “X” na Figura 5.10) antes da ocorrência de uma falha foram rotuladas como *pré-instável*, enquanto que os registos fora deste intervalo de tempo X foram rotulados como *estável*.

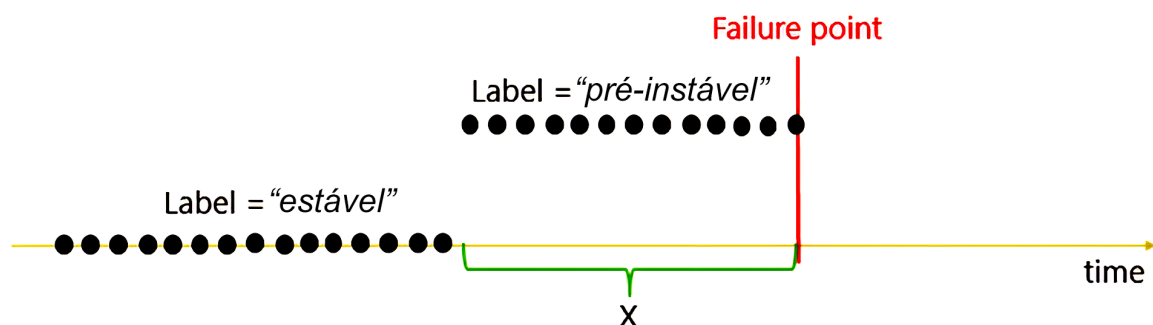


Figura 5.10: Classificação dos dados e construção de *Labels* - Adaptado de [7]

O principal objetivo dos modelos de *Machine Learning* utilizados será prever a probabilidade de ocorrência de uma falha dentro desta janela temporal. Neste caso, mais especificamente, a probabilidade de ocorrência de uma falha numa máquina

²Secção 4.2.1

nas próximas 24 horas (duração da janela temporal escolhida para esta aplicação) relacionada com um dos componentes (componentes 1,2,3 ou 4). Assim, foi criada uma nova *feature* categórica “*failure*”, onde todos os registos nas 24 horas anteriores à ocorrência de uma falha no componente 1 têm o valor *failure=comp1* e assim sucessivamente para os componentes 2,3 e 4. Os registos que não verificam estas condições têm o valor de *failure=none*. Isto leva a que o problema se transforme de um problema binário (estável/pré-instável) para um problema de classificação multi-classe (estável/componente n pré-instável). Salientar ainda que, doravante, devido a esta redefinição de *evento de falha*, quando se menciona que um determinado algoritmo prevê uma falha, de facto, o que se está a referir é que o algoritmo prevê a ocorrência de uma falha dentro desta janela temporal.

Na Tabela 5.11 são representados exemplos de falha no componente 2. De notar que os primeiros 8 registos ocorrem nas 24 horas anteriores à ocorrência da primeira falha do componente 2. Os próximos 8 registos nas 24 horas anteriores a outra falha do componente 2.

Tabela 5.11: Exemplo da representação de falha no componente 2

	machineID	datetime	model	age	failure
857	1	2015-04-19 09:00:00	model3	18	comp2
858	1	2015-04-19 12:00:00	model3	18	comp2
859	1	2015-04-19 15:00:00	model3	18	comp2
860	1	2015-04-19 18:00:00	model3	18	comp2
861	1	2015-04-19 21:00:00	model3	18	comp2
862	1	2015-04-20 00:00:00	model3	18	comp2
863	1	2015-04-20 03:00:00	model3	18	comp2
864	1	2015-04-20 06:00:00	model3	18	comp2
2297	1	2015-10-16 09:00:00	model3	18	comp2
2298	1	2015-10-16 12:00:00	model3	18	comp2
2299	1	2015-10-16 15:00:00	model3	18	comp2
2300	1	2015-10-16 18:00:00	model3	18	comp2
2301	1	2015-10-16 21:00:00	model3	18	comp2
2302	1	2015-10-17 00:00:00	model3	18	comp2
2303	1	2015-10-17 03:00:00	model3	18	comp2
2304	1	2015-10-17 06:00:00	model3	18	comp2

5.7. Divisão dos dados

Ao trabalhar com dados com registo de data e hora associados, como é o caso, a divisão entre os conjunto de treino, desenvolvimento e teste deve ser realizada com cuidado, de modo a garantir que as avaliações obtidas correspondem à *performance* real que se deve esperar dos modelos, uma vez que existe uma correlação temporal inerente entre observações (elevada similaridade entre dados temporalmente próximos). Como referido anteriormente, a esta técnica de validação dá-se o nome de *Holdout* - Figura 4.9. De notar que em [7], não se optou pela aplicação desta técnica.

Em problemas de manutenção preditiva, regra geral, a melhor opção é realizar uma divisão com base no tempo, isto é, escolher um ponto no tempo, treinar o modelo com todos os registos anteriores a esse ponto, utilizando os registos posteriores para validar o modelo. Esta metodologia permite também simular como o modelo se irá comportar realmente na prática.

Assim, na presente aplicação, os registos até 31-08-2015 1:00:00 foram atribuídos ao conjunto de teste, os registos entre 01-09-2015 1:00:00 e 31-10-2015 1:00:00 ao conjunto de desenvolvimento e os registos a partir de 01-11-2015 1:00:00 ao conjunto de teste. De modo a garantir que os dados em diferentes conjuntos não partilham janelas temporais, foram removidos os registos nas fronteiras, isto é, os registos das 24 horas que antecedem a data da divisão. Desta forma, apresenta-se na Tabela 5.12 a quantidade de dados que foi atribuída a cada um dos conjuntos e a percentagem que corresponde a falhas.

Tabela 5.12: Quantidade de dados atribuída a cada um dos conjuntos e a percentagem correspondente a falhas

	Quantidade /%	Falha /%
Treino	66.52	2.02
Desenvolvimento	16.57	1.89
Teste	16.91	1.92

5.7.1. Desequilíbrio entre classes em aplicações de problemas de manutenção

Algo em ter em consideração em manutenção preditiva é o facto de que a ocorrência de falhas é rara durante o ciclo de vida de uma determinada máquina, quando comparado com o funcionamento normal. Isto leva a que exista um desequilíbrio entre classes (Tabela 5.13), o que normalmente conduz a uma *performance* ilusória por

parte dos algoritmos, que tendem a classificar o exemplo mais comum mais frequentemente à custa dos menos comuns, uma vez que o número total de classificações incorretas é assim menor. Como consequência, os valores de *Recall* e *Precision* podem ser baixos, apesar de o valor de *Accuracy* ser elevado. Um exemplo claro deste fenómeno é, no conjunto de desenvolvimento (onde serão calculadas a maioria das métricas de avaliação), 98.11% (Tabela 5.12) dos dados correspondem à categoria Estável (*failure=none*), ou seja, um modelo (sem qualquer utilidade) que prevê funcionamento estável em todos os instantes teria uma *Accuracy* de 98.11%. É assim fundamental olhar para outras métricas de avaliação.

Para um número considerável de aplicações em equipamentos críticos, a incapacidade do modelo em prever uma falha pode vir a ser exorbitantemente cara. Na manutenção preditiva, regra geral, o mais importante é o número de falhas reais que o modelo é capaz de prever, ou seja, o *Recall*³ do modelo. Este parâmetro torna-se ainda mais importante à medida que as consequências de falsos negativos, isto é, falhas verdadeiras que o modelo não foi capaz de prever, excedem as consequências de falsos positivos, ou seja, uma previsão falsa de uma falha. Este fenómeno é conhecido por “custo de classificação incorreta” e pode ser estimado pelas empresas de acordo com os custos de reparação, das peças e da mão-de-obra. Geralmente, é preferível que o modelo erre por precaução, uma vez que, será mais económico a realização de uma verificação de manutenção do que uma interrupção parcial ou total da operação. No entanto, a previsão errada de uma falha, isto é, um falso positivo, pode igualmente conduzir a uma perda de tempo e recursos. Nesse caso, o modelo deverá ser ajustado para uma *Precision* elevada. Contudo, como mencionado anteriormente, as métricas *Recall* e *Precision* não são independentes: o aumento de uma implica a diminuição da outra.

Tabela 5.13: Exemplo do desequilíbrio entre as diferentes classes para a *feature* ‘*failure*’ no conjunto total de dados

	<i>failure</i>	%
none	285684	98.06
comp2	1985	0.68
comp1	1464	0.50
comp4	1240	0.43
comp3	968	0.33

³Secção 4.4

5.8. Aplicação dos modelos no conjunto de desenvolvimento

Nesta primeira aplicação, é utilizado o conjunto de desenvolvimento de modo a perceber como uma grande variedade de modelos se comporta, assim como procurar a afinação dos hiperparâmetros de certos modelos. Tal abordagem deve-se ao facto de não ser possível, à partida, determinar qual o algoritmo mais adequado para um determinado problema. Ao treinar e avaliar uma grande variedade de modelos numa fase inicial é possível perceber quais têm um maior potencial, sendo que, no entanto, para que esta etapa seja bem sucedida, é necessário que as métricas de avaliação dos modelos tenham sido escolhidas em concordância com os objetivos estabelecidos.

Entender o princípio de funcionamento de um algoritmo de *Machine Learning* é diferente de compreender quais as etapas que este realiza entre a entrada de um dado conjunto de valores e a saída de um resultado: muitos algoritmos são caixas-negras. No entanto, embora desconfortável, a utilização destes algoritmos é inevitável, dado que o potencial destes é, tipicamente, superior ao daqueles para os quais é possível perceber o processo de decisão. Contudo, a seleção de uma boa métrica de avaliação pode permitir diminuir esse desconforto.

5.8.1. *K-Nearest Neighbors*

O algoritmo *K-Nearest Neighbors* (*K-NN*) é um dos mais simples algoritmos de *Machine Learning*. Dado um conjunto de dados, a previsão para um novo registo consiste em encontrar os K pontos mais próximos no conjunto de treino (Figura 5.11), isto é, os seus “vizinhos mais próximos” (“*nearest neighbors*”) [6].

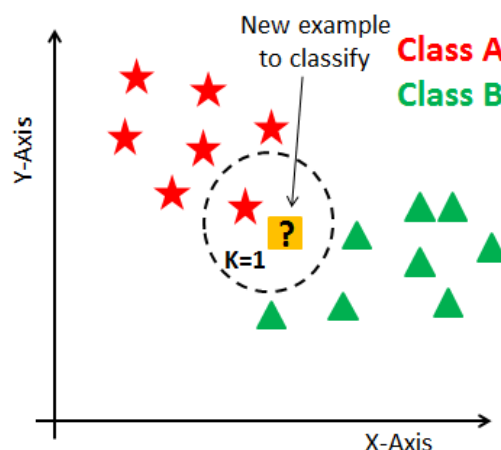


Figura 5.11: Exemplo de *K-Nearest Neighbors*, com “*n-neighbors*” = 1

Na Figura 5.12 está representada a evolução da *Accuracy* no conjunto de treino e no conjunto de desenvolvimento, para *K-Nearest Neighbors* com diferentes valores do hiperparâmetro “*n-neighbors*”. De igual modo, na Figura 5.13 está representada a evolução da taxa de erro no conjunto de desenvolvimento, para *K-Nearest Neighbors* com diferentes valores do hiperparâmetro “*n-neighbors*”.

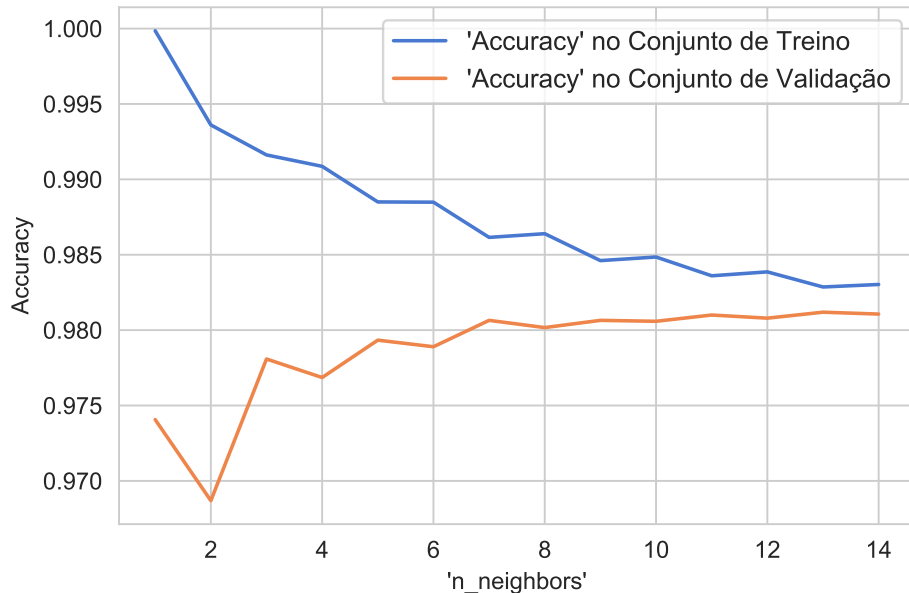


Figura 5.12: Análise da evolução da *Accuracy* no conjunto de treino e no conjunto de desenvolvimento, para *K-Nearest Neighbors* com diferentes valores do hiperparâmetro “*n-neighbors*”

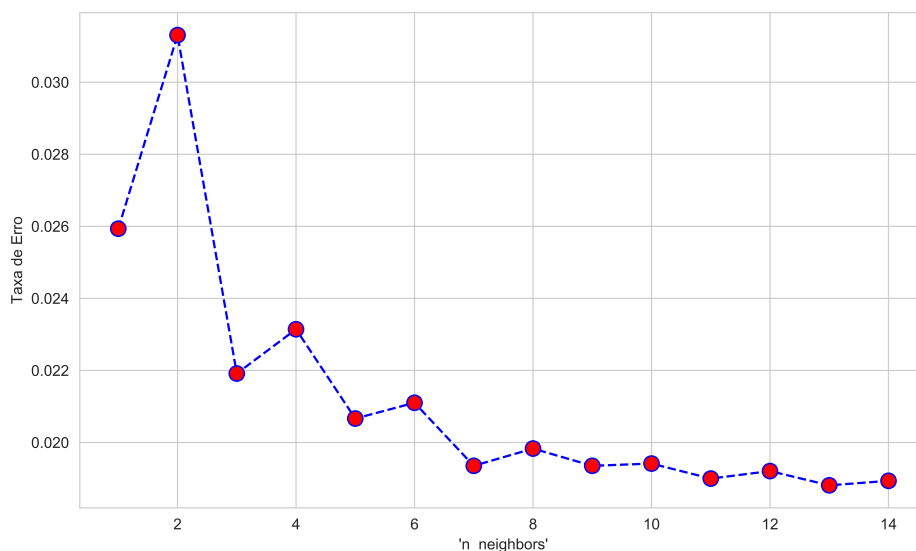


Figura 5.13: Análise da evolução da taxa de erro no conjunto de desenvolvimento, para *K-Nearest Neighbors*, com diferentes valores do hiperparâmetro “*n-neighbors*”

Os resultados obtidos para *Precision*, *Recall* e F_1 *Score* para cada uma das classes, no conjunto de desenvolvimento, com “*n-neighbors*” = 13 (hiperparâmetro selecionado

com base nas Figuras 5.12 e 5.13), são apresentados na Tabela 5.14. É, de imediato, possível verificar que os resultados não são satisfatórios. Inclusive, para o componente 4, o algoritmo não é capaz de prever qualquer falha. Uma possível explicação para os maus resultados obtidos na aplicação deste algoritmo é o facto de existir um desequilíbrio entre classes (Tabela 5.13), sendo que, por norma, este algoritmo tem dificuldades nesses casos [6]. Para um conjunto de treino desequilibrado, a probabilidade de os k vizinhos mais próximos de qualquer ponto aleatório pertencer à classe com mais exemplos é elevada. Para os registos das classes menos representadas, mesmo que o vizinho mais próximo desse ponto pertença à classe correta, existe uma grande probabilidade dos restantes pontos ($(k - 1)$ pontos) pertencerem à classe com maior densidade no espaço e, deste modo, o ponto será classificado incorretamente. Assim, uma abordagem de $k = 1$ (“*n-neighbors*”= 1) pode, eventualmente, conduzir a um melhor resultado.

Tabela 5.14: Resultado para *K-Nearest Neighbors* no conjunto de desenvolvimento com “*n-neighbors*”= 13

	none	comp1	comp2	comp3	comp4
<i>Precision</i>	0.9830	1.0000	0.4662	0.1667	0.0000
<i>Recall</i>	0.9982	0.0120	0.1917	0.0139	0.0000
<i>F₁ Score</i>	0.9905	0.0238	0.2717	0.0256	—

Esta hipótese é testada. Os resultados obtidos para “*n-neighbors*”= 1 são apresentados na Tabela 5.15. É possível observar que, para as classes menos representadas, apesar de os resultados obtidos serem ligeiramente melhores, mantêm-se em níveis insatisfatórios.

Tabela 5.15: Resultado para *K-Nearest Neighbors* no conjunto de desenvolvimento com “*n-neighbors*”= 1

	none	comp1	comp2	comp3	comp4
<i>Precision</i>	0.9857	0.1917	0.3639	0.2252	0.0667
<i>Recall</i>	0.9883	0.1386	0.3639	0.2361	0.0411
<i>F₁ Score</i>	0.9870	0.1608	0.3639	0.2305	0.0508

5.8.2. *Decision Tree*

O próximo algoritmo aplicado foi *Decision Trees*. *Decision Trees* são amplamente utilizadas para tarefas de classificação e regressão [6]. De acordo com [6], essencialmente, estes modelos aprendem através de uma hierarquia de perguntas *if/else*, que

conduz a uma decisão. *Decision Trees* apresentam duas principais vantagens: o modelo resultante pode ser facilmente visualizado e compreendido (para “árvores” de pequenas dimensões) e os algoritmos são capazes de lidar com dados em diferentes escalas, isto é, uma vez que cada *feature* é processada separadamente, e as possíveis divisões dos dados não dependem da sua escala, não são necessárias operações de pré-processamento, como a normalização. Em particular, *Decision Trees* apresentam bom desempenho quando as *features* estão em escalas completamente diferentes assim como quando existe uma mistura de *features* binárias com *features* contínuas. A principal desvantagem das *Decision Trees* é a sua tendência a apresentar *overfitting*, conduzindo a uma baixa capacidade de generalização. No entanto, a afinação de hiperparâmetros como *Max_Depth* procura limitar o *overfitting*. Ainda assim, este facto leva a que, em muitas aplicações, métodos de *Ensemble*, como *Random Forests*, sejam utilizados em detrimento das *Decision Trees*.

Na Figura 5.14 está representada a evolução da *Accuracy* no conjunto de treino e no conjunto de desenvolvimento, para diferentes valores do hiperparâmetro '*Max_Depth*'. É possível observar que, não só os valores de *Accuracy* são elevados, como não se verifica a existência de *overfitting*. Para *Max_Depth*=6 (hiperparâmetro seleccionado com base na Figura 5.14), são apresentados na Tabela 5.16 os valores obtidos para *Precision*, *Recall* e *F₁ Score* para cada uma das classes, no conjunto de desenvolvimento.

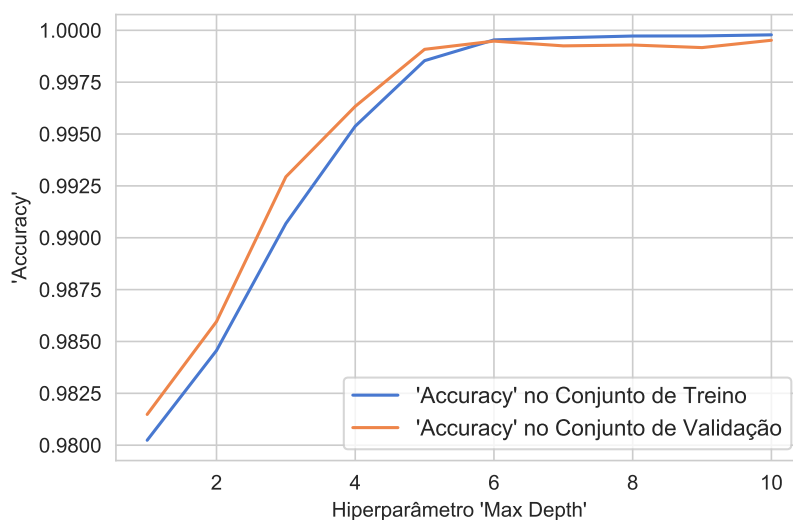


Figura 5.14: Análise da evolução da *Accuracy* no conjunto de treino e no conjunto de desenvolvimento, para *Decision Trees* com diferentes valores do hiperparâmetro *Max_Depth*

Na Figura 5.15 está representada a matriz de confusão não normalizada e na Figura 5.16 a matriz de confusão normalizada. Para o caso em análise, a matriz de confusão lista, em cada linha, as falhas reais dos componentes e em cada coluna a previsão de falha pelo modelo. A título de exemplo, o segundo número da linha superior indica o

Tabela 5.16: Resultado para *Decision Tree* no conjunto de desenvolvimento com $Max_Depth=6$

	none	comp1	comp2	comp3	comp4
<i>Precision</i>	0.9999	0.9390	0.9917	1.0000	0.9167
<i>Recall</i>	0.9999	0.9277	1.0000	0.9167	0.9543
<i>F₁ Score</i>	0.9999	0.9333	0.9959	0.9565	0.9351

número de vezes em que o modelo previu que o componente 1 iria falhar, quando na realidade nenhum componente falhou. Na matriz de confusão, os valores da diagonal correspondem a uma correta classificação de falha de um certo componente por parte do modelo.



Figura 5.15: Matriz de Confusão para *Decision Tree* no conjunto de desenvolvimento com $Max_Depth=6$

Na Figura 5.17 está representada a importância das *features* para este modelo. O facto de uma *feature* apresentar um baixo coeficiente de importância não significa necessariamente que a *feature* é pouco informativa, apenas que esta não foi escolhida pela árvore, provavelmente porque outra *feature* codifica a mesma informação. De notar que, os resultados obtidos para a aplicação deste modelo são satisfatórios.



Figura 5.16: Matriz de Confusão Normalizada para *Decision Tree* no conjunto de desenvolvimento com *Max_Depth=6*

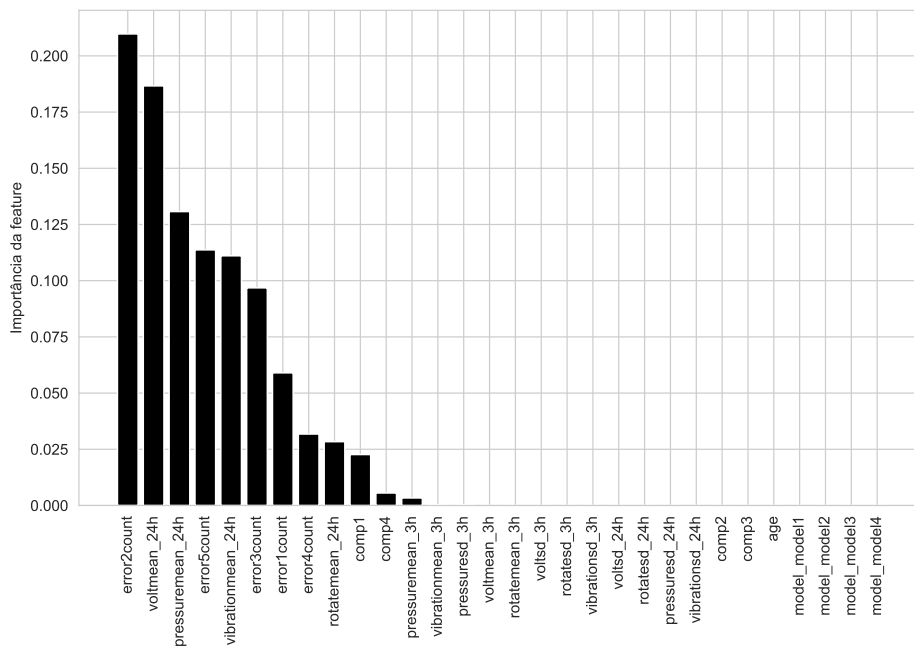


Figura 5.17: Importância das *features* para para *Decision Tree* no conjunto de desenvolvimento com *Max_Depth=6*

5.8.3. *Random Forest*

Como referido anteriormente, a principal desvantagem das *Decision Trees* é a tendência para existência de *overfitting* (algo que, no entanto, não se verificou na presente aplicação). Os modelos *Random Forest* procuram resolver esse problema. De acordo com [6], uma *Random Forest* é essencialmente uma coleção de *Decision Trees*, onde cada “árvore” é ligeiramente diferente das outras. A ideia é que cada “árvore” é capaz de fazer um bom trabalho de previsão, mas provavelmente com tendência a existir *overfitting* de parte dos dados. Ao construir várias árvores, com bom desempenho e que apresentam *overfitting* de maneiras diferentes, podemos reduzir a quantidade de *overfitting* total calculando a média dos seus resultados. O hiperparâmetro *'n_estimators'* define o número de *Decision Trees* utilizadas pelo modelo *Random Forest*. Na Figura 5.18 está representada a evolução da *Accuracy* no conjunto de treino e no conjunto de desenvolvimento, para *Random Forest* com diferentes valores do hiperparâmetro *'n_estimators'*.

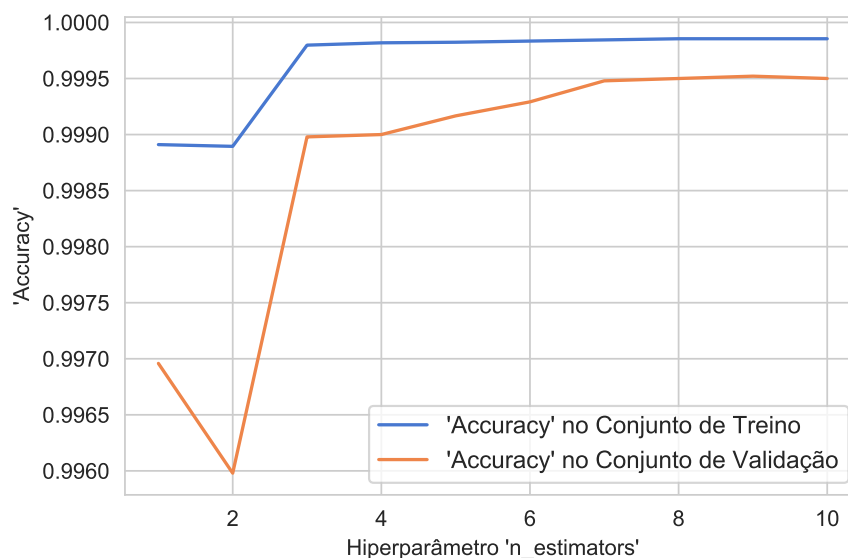


Figura 5.18: Análise da evolução da *Accuracy* no conjunto de treino e no conjunto de desenvolvimento, para *Random Forest* com diferentes valores do hiperparâmetro *'n_estimators'*

Para *'n_estimators'=7'* (hiperparâmetro selecionado com base na Figura 5.18), são apresentados na Tabela 5.17 os valores obtidos para *Precision*, *Recall* e *F₁ Score* para cada uma das classes, no conjunto de desenvolvimento.

De notar que, os resultados obtidos para a aplicação do modelo *Random Forest* no conjunto de desenvolvimento são bastante satisfatórios, sendo que, para todas as classes, os valores das métricas de avaliação encontram-se acima dos 90%.

Tabela 5.17: Resultado para *Random Forest* no conjunto de desenvolvimento com $n_estimators = 7$

	none	comp1	comp2	comp3	comp4
<i>Precision</i>	0.9997	0.9244	0.9916	1.0000	0.9812
<i>Recall</i>	0.9999	0.9578	0.9861	0.9514	0.9543
<i>F₁ Score</i>	0.9998	0.9408	0.9889	0.9751	0.9676

5.8.4. Naive Bayes

Os algoritmos *naive Bayes* são uma família de modelos de classificação bastante semelhante aos modelos lineares [6]. No entanto, eles tendem a ser ainda mais rápidos durante a fase de treino [6]. O preço a pagar pela rapidez de treino é que estes modelos geralmente apresentam desempenhos de generalização um pouco piores do que o de classificadores lineares como *Logistic Regression* e *Linear SVC* [6]. De notar que, na manutenção tradicional, a análise bayesiana é muito utilizada [14] e, por isso, considerou-se interessante verificar os resultados para a aplicação deste algoritmo.

Na Tabela 5.18 são apresentados os valores obtidos para *Precision*, *Recall* e *F₁ Score* para cada uma das classes, no conjunto de desenvolvimento.

Tabela 5.18: Resultado para *naive Bayes* no conjunto de desenvolvimento

	none	comp1	comp2	comp3	comp4
<i>Precision</i>	1.0000	0.2120	0.8946	0.3318	0.4255
<i>Recall</i>	0.9759	0.9157	0.9194	0.9861	0.9909
<i>F₁ Score</i>	0.9878	0.3443	0.9068	0.4965	0.5953

É possível verificar que, para as classes correspondentes aos componentes 1,3 e 4, o valor do parâmetro *Precision* (e, conseqüentemente, *F₁ Score*) é baixo e insatisfatório. No entanto, o mesmo não se verifica para o *Recall*.

5.8.5. Artificial Neural Networks

As *Artificial Neural Networks* (ANN), ou Redes Neurais Artificiais, são um conjunto de técnicas computacionais que representam um modelo matemático inspirado no sistema nervoso central, em particular o cérebro [6, 35].

As *Artificial Neural Networks* são modeladas de forma análoga ao cérebro humano, sendo compostas por vários neurónios artificiais (o número de neurónios e de con-

xões entre estes é, no entanto, significativamente menor do que o cérebro humano) [66, 67]. Na Figura 5.19 está representada uma *Artificial Neural Networks* com 4 camadas: camada de entrada, duas camadas intermédias e camada de saída.

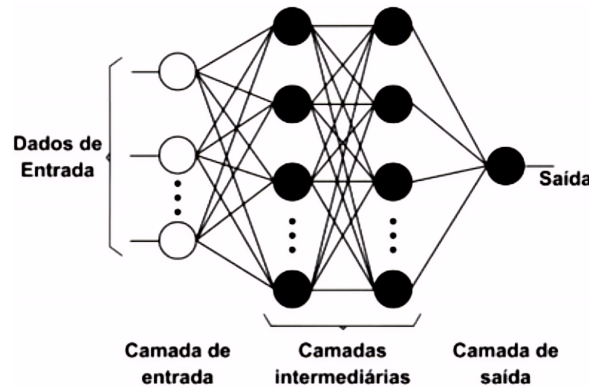


Figura 5.19: *Artificial Neural Networks* com 4 camadas [67]

As redes neuronais artificiais têm sido aplicadas com sucesso nas mais diversas áreas, sendo que, a sua utilização tem vindo a aumentar progressivamente, uma vez que um número cada vez maior de problemas mostra-se adequado à resolução através desta técnica e as soluções obtidas apresentam um melhor desempenho quando comparadas com outros algoritmos [6, 35, 49].

Neste caso, será aplicado o algoritmo *Multi-Layer Perceptron (MLP)*, que consiste numa *Artificial Neural Networks* onde a aprendizagem é realizada através de retropropagação do erro [6, 35]. A flexibilidade das Redes Neuronais Artificiais é também uma das suas principais desvantagens: existem múltiplos hiperparâmetros ajustáveis, como por exemplo, é possível utilizar topologias de rede variadas, mesmo para um MLP simples. É ainda possível alterar o número de camadas, o número de neurónios por camada e o tipo de função de ativação a ser usada em cada camada. À medida que a complexidade da rede aumenta, é verificado igualmente um aumento significativo no tempo de treino [6].

Na Tabela 5.19 estão representados os valores obtidos para *Precision*, *Recall* e *F₁ Score* para *Artificial Neural Network* no conjunto de desenvolvimento com 100 camadas ocultas (*hidden_layers* = 100).

No caso das *Artificial Neural Network*, a *performance* dos modelos é, regra geral, superior quando o conjunto de dados é homogéneo [6, 35]. Assim, todas as *features* foram reduzidas à mesma escala (*feature scaling*), isto é, recorreu-se ao método *min-max scaling*, que consiste em dividir o conjunto de valores ao qual foi subtraído o seu valor mínimo pela diferença entre os valores máximo e mínimo, garantindo assim que todos os valores ficam compreendidos entre 0 e 1 [6, 35]. Os resultados obtidos após realização de *min-max scaling* são apresentados na Tabela 5.20.

Tabela 5.19: Resultado para *Artificial Neural Network* no conjunto de desenvolvimento com 100 camadas ocultas (*hidden_layers* = 100)

	none	comp1	comp2	comp3	comp4
<i>Precision</i>	0.9994	0.9286	0.9890	0.9860	0.9231
<i>Recall</i>	0.9996	0.7831	0.9972	0.9792	0.9863
<i>F₁ Score</i>	0.9995	0.8497	0.9931	0.9826	0.9536

Tabela 5.20: Resultado para *Artificial Neural Network* no conjunto de desenvolvimento com 100 camadas ocultas (*hidden_layers* = 100) e normalização *min-max scaling*

	none	comp1	comp2	comp3	comp4
<i>Precision</i>	0.9997	0.9451	0.9917	1.0000	0.9520
<i>Recall</i>	0.9998	0.9337	0.9972	0.9167	0.9954
<i>F₁ Score</i>	0.9997	0.9394	0.9945	0.9565	0.9732

É possível verificar que, após recorrer ao método *min-max scaling*, os resultados obtidos são, de facto, superiores e bastante satisfatórios. De notar que, para os modelos de *Artificial Neural Networks*, na presente aplicação, não foi procurada uma otimização exaustiva dos hiperparâmetros. Ainda assim, os resultados obtidos estão ao nível de outros algoritmos anteriormente aplicados, tornando evidente a potencialidade destes modelos.

5.9. Comportamento no Conjunto de Teste

O conjunto de desenvolvimento foi utilizado, até o momento, para afinar os modelos e respetivos hiperparâmetros, de modo a procurar uma melhor *performance*. É agora importante verificar como os modelos se comportam no conjunto de teste. Apesar de, num caso real, ser aconselhável avaliar apenas o modelo que se pretende implementar [6], na presente secção apresentam-se os resultados obtidos para a avaliação dos dois melhores modelos (no conjunto de desenvolvimento), neste caso, *Random Forest* e *Artificial Neural Networks*, com normalização *min-max scaling*.

Na Tabela 5.21 e Tabela 5.22 são apresentados os valores obtidos para *Precision*, *Recall* e *F₁ Score*, para o modelo *Random Forest* e *Artificial Neural Network*, respetivamente, nos conjuntos de desenvolvimento e teste.

Como seria de esperar, verifica-se uma diminuição de *performance* generalizada das

Tabela 5.21: *Performance* para o modelo *Random Forest* nos conjuntos de desenvolvimento e teste, com ' $n_estimators = 7$ '

		none	comp1	comp2	comp3	comp4
Conj. Desenvol.	<i>Precision</i>	0.9997	0.9244	0.9916	1.0000	0.9812
	<i>Recall</i>	0.9999	0.9578	0.9861	0.9514	0.9543
	F_1 Score	0.9998	0.9408	0.9889	0.9751	0.9676
Conj. Teste	<i>Precision</i>	0.9988	0.9718	0.9711	0.9855	0.9830
	<i>Recall</i>	0.9998	0.8150	0.9882	0.9189	0.9611
	F_1 Score	0.9993	0.8865	0.9796	0.9510	0.9719

Tabela 5.22: *Performance* para o modelo *Artificial Neural Network* nos conjuntos de desenvolvimento e teste, com 100 camadas ocultas ($hidden_layers = 100$) e normalização *min-max scaling*

		none	comp1	comp2	comp3	comp4
Conj. Desenvol.	<i>Precision</i>	0.9997	0.9451	0.9917	1.0000	0.9520
	<i>Recall</i>	0.9998	0.9337	0.9972	0.9167	0.9954
	F_1 Score	0.9997	0.9394	0.9945	0.9565	0.9732
Conj. Teste	<i>Precision</i>	0.9990	0.9030	0.9941	0.9858	0.9725
	<i>Recall</i>	0.9995	0.8425	0.9853	0.9392	0.9833
	F_1 Score	0.9993	0.8717	0.9897	0.9619	0.9779

métricas de avaliação no conjunto de teste. Ainda assim, os resultados mantêm-se satisfatórios. Como referido anteriormente, na manutenção preditiva, regra geral, o mais importante é o número de falhas reais que o modelo é capaz de prever, ou seja, o valor do parâmetro *Recall* do modelo [41]. Este parâmetro torna-se ainda mais importante à medida que as consequências de falsos negativos, isto é, falhas verdadeiras que o modelo não foi capaz de prever, excedem as consequências de falsos positivos, ou seja, uma previsão falsa de uma falha [40, 41]. Para ambos os modelos, verifica-se uma queda no valor do *Recall* (e, conseqüentemente, do F_1 Score) para valores inferiores a 90% para o componente 1 no conjunto de teste. Na presente aplicação, os quatro componentes foram considerados de igual importância. Numa aplicação real, onde poderá ser possível conhecer mais informação acerca de cada um deles (como por exemplo custo, importância no processo, localização no equipamento, facilidade de substituição), a análise pode passar por tentar otimizar certas métricas que se considerem de maior relevância.

5.10. Conclusões

No presente capítulo foram aplicados modelos de *Machine Learning* a um *dataset* disponibilizado *online*. O conjunto de dados utilizado foi publicado pela *Microsoft*, em [7], num *Notebook* de Manutenção Preditiva e *Machine Learning*. A utilização deste conjunto de dados foi justificada. Na implementação realizada no presente projeto, até a fase final do *feature engineering* (Secção 5.4), foram seguidos os passos apresentados no *Notebook* [7]. No entanto, a partir desse momento, por ser considerado que a abordagem apresentada em [7] é demasiado simplista (não é utilizada nenhuma técnica de validação e apenas é aplicado um único modelo), decidiu-se aprofundar a análise com implementação da técnica de validação *Holdout*, que divide o conjunto de dados em três subconjuntos (Treino, Desenvolvimento e Teste), assim como variados modelos de *Machine Learning*, de acordo com a estrutura apresentada no Capítulo 4, sendo assim possível demonstrar como pode ser realizada a afinação dos modelos e respetivos hiperparâmetros com recurso ao conjunto de desenvolvimento. O facto de se tratar de um problema de classificação multi-classe acrescentou complexidade à análise e, porventura, começar por um problema de classificação binário pode ser aconselhável para uma melhor compreensão de conceitos basilares do *Machine Learning*, fundamentais para o sucesso de qualquer aplicação. Foi também possível abordar o desequilíbrio entre classes, muito comum em aplicações de manutenção, uma vez que a ocorrência de falhas é rara durante o ciclo de vida de uma determinada máquina, quando comparado com o seu funcionamento normal.

À partida, e sabendo que uma escolha sensata de quais os dados a usar e como tratá-los é crucial para o desempenho dos algoritmos de *Machine Learning*, seriam de esperar bons resultados com base no resultado obtido, embora de forma diferente, em [7]. Este facto veio a concretizar-se. No entanto, mais importante do que qualquer resultado foi a demonstração de uma metodologia que permitiu mostrar, partindo de dados de diferentes tipos e fontes (muito comum em aplicações de manutenção), como é possível visualizá-los e tratá-los com fim a aplicar ferramentas de Inteligência Artificial, neste caso, *Machine Learning*.

6. Conclusões e trabalhos futuros

6.1. Conclusões

A presente dissertação permitiu retirar um conjunto de conclusões sobre o trabalho realizado. Foi possível perceber de que modo ferramentas de Inteligência Artificial, mais concretamente o *Machine Learning*, podem ser aplicadas na área da manutenção, com o advento da Indústria 4.0, permitindo uma movimentação no sentido de uma manutenção preditiva e prescritiva, sob pena de perda de competitividade. Espera-se que a necessidade de um engenheiro mecânico conhecer estas ferramentas (ainda que do ponto de vista de utilizador) tenha sido evidenciada.

O conjunto de dados utilizado é chave para a resolução de problemas de *Machine Learning*. Uma escolha sensata de quais os dados a usar e como tratá-los é crucial para melhorar o desempenho dos algoritmos. A divisão dos dados em conjuntos de treino, desenvolvimento e teste é fundamental para que a *performance* real dos sistemas de *Machine Learning* não seja inferior às expectativas geradas por uma avaliação incorreta dos resultados dos modelos.

Demonstrou-se uma metodologia que permitiu mostrar, partindo de dados de diferentes tipos e fontes (muito comum em aplicações de manutenção), como é possível visualizá-los e tratá-los com fim a aplicar ferramentas de Inteligência Artificial, neste caso, *Machine Learning*. Abordou-se o desequilíbrio entre classes, muito comum em aplicações destas ferramentas em manutenção, uma vez que a ocorrência de falhas é rara durante o ciclo de vida de uma determinada máquina, quando comparado com o seu funcionamento normal.

A implementação foi realizada com recurso à linguagem de programação *Python*. Esta linguagem de programação contém numerosas bibliotecas disponíveis (*Matplotlib*, *Numpy*, *Pandas*, *Scikit-Learn* [63]) e uma extensa bibliografia [6, 35] que demonstra como estas podem ser aplicadas em projetos de *Machine Learning*. Assim, o *Python* apresenta-se com uma ótima ferramenta para a aplicação de todos os conceitos apresentados ao longo da dissertação.

Não se deve esperar que o *Machine Learning* resolva todos os problemas da área

da manutenção, no entanto, o seu potencial não deve ser desprezado. Os modelos de *Machine Learning* são apenas uma parte de um processo maior para resolver um problema específico, e é importante manter um panorama geral em mente. Ao aprofundar os aspetos técnicos do *Machine Learning*, é fácil perder de vista os objetivos finais. É importante manter em mente todas as suposições criadas, de forma explícita ou implícita, quando são construídos modelos de *Machine Learning*.

Casos de estudo podem ter um papel importante nesta área, ainda que, o acesso a conjuntos de dados reais é frequentemente uma barreira. Casos de estudo permitem perceber como um determinado fenómeno é influenciado pelo contexto em que se encontra assim como obter informações detalhadas acerca de um cenário específico, possibilitando uma análise mais profunda dessa situação. Um dos objetivos é compreender a peculiaridade do caso de modo a extrapolar conclusões para outras situações e compreender a relação entre as variáveis e o modelo. Mesmo em contextos altamente específicos, algumas das conclusões podem ser extrapoladas para situações mais amplas.

Apesar do progresso nos métodos de aquisição de dados e ferramentas de Inteligência Artificial, a plena utilização dos recursos fornecidos por estes é, ainda, incipiente. Os sistemas estão longe de atingir o grau de desempenho e robustez necessários para automação total na maior parte dos casos. O fator humano é de importância central. Sem a ajuda de mão de obra especializada não seria possível rastrear com eficiência problemas em sistemas críticos. Contudo, é recomendável que as duas fontes de conhecimento sejam combinadas de modo a produzir programas de manutenção mais confiáveis e precisos. Ainda, a existência de uma visão no sentido de desenvolver sistemas que possam trazer maior produtividade ao elemento humano. Um sistema de prognóstico será eventualmente capaz de reduzir a carga de trabalho desnecessária e aliviar pequenos erros humanos, sendo que, no entanto, poderá igualmente criar lacunas induzidas pela automação, produzir alarmes falsos e, talvez mais importante, aumentar a carga de trabalho mental necessária para executar a monitorização adicional do sistema. De modo a alcançar o equilíbrio correto entre a automação e o fator humano, é importante garantir que a tecnologia esteja presente para ajudar o especialista naturalmente, deixando-o ser a autoridade final na determinação de uma possível falha. O operador deve poder substituir as recomendações do sistema e usar suas próprias habilidades e experiência quando necessário.

O difícil equilíbrio entre intervenção humana e automação é mais facilmente alcançado se os sistemas estiverem equipados com modelos fáceis de interpretar e entender. No entanto, a falta de transparência do processo de decisão inerente a muitos dos modelos de *Machine Learning* é inevitável e, portanto, o desconforto que possa advir deste facto tem de ser ultrapassado com, por exemplo, o uso de métricas de avaliação poderosas e demonstrações práticas de sucesso preditivo.

Para que a utilização de ferramentas de Inteligência Artificial na área da manutenção seja viável, as consequências provenientes dessa aplicação devem ser significativas, para que o custo de desenvolvimento, implementação e utilização seja justificado. Deve existir uma lacuna significativa entre o desempenho de qualquer estratégia existente e aquele que pode ser alcançado através da aplicação de modelos de *Machine Learning*. As tecnologias preditivas avançadas necessitam de ser direcionadas para as oportunidades mais significativas de modo a permitir uma melhoria no desempenho geral da organização. Estes métodos devem ser capazes de atender não só às necessidades técnicas mas também às necessidades económicas.

A transformação de uma indústria deve ser um processo contínuo e abrangente, que vai além da melhoria de tecnologias, tendo que afetar toda a cadeia envolvente e sequencial de processos, modelos de negócio, aspetos organizacionais e de gestão. Deve-se reter que o investimento na digitalização poderá trazer melhorias em produtividade e eficiência, mas os benefícios virão apenas quando acompanhados da melhoria de desempenho organizacional e visão nas empresas.

6.2. Trabalhos Futuros

Em relação ao trabalho realizado e possíveis trabalhos futuros, propõe-se a exploração do impacto nos modelos de algumas escolhas efetuadas, como a data de divisão dos dados em conjunto de treino, desenvolvimento e teste e o tempo de fronteira definido entre as categorias estável/pré-estável. Igualmente, o facto de se tratar de um problema de classificação multi-classe acrescentou complexidade à análise e, porventura, começar por um problema de classificação binário pode ser aconselhável para uma melhor compreensão de conceitos basilares do *Machine Learning*, fundamentais para o sucesso de qualquer aplicação.

De um modo mais geral, em relação à aplicação de ferramentas de Inteligência Artificial na área da manutenção, geralmente, dada a escassez de dados rotulados para aplicações de *Machine Learning* supervisionadas, abordagens não supervisionadas e semi-supervisionadas são particularmente promissoras e sugere-se uma exploração destas técnicas.

Como foi visto, nos métodos tradicionais de *Machine Learning*, um número de *features* necessita de ser identificado e extraído por especialistas no domínio, de modo a reduzir a complexidade dos dados e tornar os padrões mais visíveis para que os algoritmos sejam capazes de apresentar bom desempenho. Este *feature engineering* específico para cada aplicação concreta impossibilita, de igual modo, a utilização de um modelo desenvolvido em diversas aplicações e equipamentos distintos. A utiliza-

ção de algoritmos de *Deep Learning* elimina esta necessidade de extração de *features ad hoc*, uma vez que estes aprendam *features* de alto nível a partir do conjunto de dados completo, de maneira incremental. Sugere-se a exploração destes algoritmos que, embora mais complexos, apresentam enorme potencial.

Bibliografia

- [1] Alasdair Gilchrist. *Industry 4.0: the industrial internet of things*. Springer, 2016. ISBN: 978-1-4842-2047-4.
- [2] Khairy Ahmed Helmy Kobbacy e DN Prabhakar Murthy. *Complex system maintenance handbook*. Springer Science & Business Media, 2008. ISBN: 978-1849967006.
- [3] Henning Kagermann, Wolf-Dieter Lukas e Wolfgang Wahlster. «Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution». Em: *VDI nachrichten* 13.11 (2011), p. 2.
- [4] João Ribau. *Afinal, o que é isto da Indústria 4.0?* URL: <https://visao.sapo.pt/opiniao/ponto-de-vista/2019-10-28-afinal-o-que-e-isto-da-industria-4-0> (acedido em 07/06/2020).
- [5] Marina Paolanti et al. «Machine learning approach for predictive maintenance in industry 4.0». Em: *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. IEEE, pp. 1–6. ISBN: 1538646439.
- [6] Andreas C Müller e Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists*. "O'Reilly Media, Inc.", 2016. ISBN: 1449369901.
- [7] Microsoft. *Predictive Maintenance Modelling Guide*. 2018. URL: <https://notebooks.azure.com/Microsoft/projects/PredictiveMaintenance> (acedido em 01/05/2020).
- [8] Nikolaos Kolokas et al. «Forecasting faults of industrial equipment using machine learning classifiers». Em: *2018 Innovations in Intelligent Systems and Applications (INISTA)*. IEEE, pp. 1–6. ISBN: 1538651505.
- [9] GIAGI. *Gestão da Manutenção e Disponibilidade dos Equipamentos - Manual Formando*. Aveiro: GIAGI – Consultores em Gestão Industrial, Lda., 2007.
- [10] Carlos Varela Pinto. *Organização e Gestão da Manutenção*. Monitor—Projecto e Edições, Lda: Lisbon, Portugal, 2002. ISBN: 978-9729413391.
- [11] José Paulo Saraiva Cabral. *Organização e Gestão da Manutenção: dos conceitos à prática*. 5ª ed. Lidel, 2006. ISBN: 978-9727574407.
- [12] Norma Portuguesa NP EN 13306:2007. «Terminologia da Manutenção». Em: *Instituto Português da Qualidade* (2007).

- [13] Ameeth Kanawaday e Aditya Sane. «Machine learning for predictive maintenance of industrial machines using IoT sensor data». Em: *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, pp. 87–90. ISBN: 1538604973.
- [14] R Keith Mobley. *An introduction to predictive maintenance*. Elsevier, 2002. ISBN: 978-0-7506-7531-4.
- [15] Thyago P Carvalho et al. «A systematic literature review of machine learning methods applied to predictive maintenance». Em: *Computers & Industrial Engineering* 137 (2019), p. 106024. ISSN: 0360-8352.
- [16] Olga Fink. «Data-Driven Intelligent Predictive Maintenance of Industrial Assets». Em: *Women in Industrial and Systems Engineering: Key Advances and Perspectives on Emerging Topics*. Ed. por Alice E. Smith. Cham: Springer International Publishing, 2020, pp. 589–605. ISBN: 978-3-030-11866-2. DOI: 10.1007/978-3-030-11866-2_25.
- [17] Norma Portuguesa NP EN 13306: 2010. «Terminologia da Manutenção». Em: *Instituto Português da Qualidade* (2010).
- [18] Bernard Schmidt, Lihui Wang e Diego Galar. «Semantic Framework for Predictive Maintenance in a Cloud Environment». Em: *Procedia CIRP* 62 (2017). 10th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '16, pp. 583–588. ISSN: 2212-8271.
- [19] Sandy Dunn. *Big Data, Predictive Analytics and Maintenance*. 2018. URL: <https://www.assetivity.com.au/article/maintenance-management/big-data-predictive-analytics-and-maintenance.html> (acedido em 01/06/2020).
- [20] Yasir Hassan Ali. «Artificial Intelligence Application in Machine Condition Monitoring and Fault Diagnosis». Em: *Artificial Intelligence: Emerging Trends and Applications* (2018), p. 275. ISSN: 178923364X.
- [21] Ryan Chan. *What is prescriptive maintenance and how does it differ from predictive maintenance?* 2019. URL: <https://www.onupkeep.com/answers/predictive-maintenance/what-is-prescriptive-maintenance-and-how-does-it-differ-from-predictive-maintenance> (acedido em 15/06/2020).
- [22] A. J. Marques Cardoso C. Pereira Cabrita. «Conceitos e definições de falha e avaria nas normas portuguesas de manutenção NP EN 13306:2007 e NP EN 15341:2009». Em: *Revista Manutenção* 125 (2015).
- [23] João Paulo Pinto. *Manutenção Lean*. LIDEL - Edições Técnicas, Lda., 2013. ISBN: 978-972-757-877-1.
- [24] IEC. «Dependability management - Part 3-3: Application guide - Life cycle costing». Em: *International Electrotechnical Commission* (2004).

- [25] Luís Andrade Ferreira; José Sobral; José Torres Farinha. «LCC – Custo do Ciclo de Vida, sua relevância para a Gestão de Ativos Físicos». Em: *Revista Manutenção* 118-119 (2013).
- [26] Jay Lee, Hung-An Kao e Shanhu Yang. «Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment». Em: *Procedia CIRP* 16 (2014). Product Services Systems and Value Creation. Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems, pp. 3–8. ISSN: 2212-8271.
- [27] Jay Lee, Behrad Bagheri e Hung-An Kao. «A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems». Em: *Manufacturing Letters* 3 (2015), pp. 18–23. ISSN: 2213-8463.
- [28] Paulo Novais e Pedro Miguel Freitas. «Inteligência Artificial e Regulação de algoritmos». Em: *Inteligência Artificial, Diálogos União Europeia* 1 (2018).
- [29] Wolfgang Wahlster. *Industrie 4.0: Cyber-Physical Production Systems for Mass Customization*. 2016. URL: http://www.dfki.de/wdata/German-Czech_Workshop_on_Industrie_4.0_Prague_11_04_16/Industrie_4_0_Cyber-Physical_Production_Systems_for_Mass_Customizations.pdf (acedido em 20/05/2020).
- [30] US Army. «Techniques for Effective Knowledge Management». Em: *Techniques Publication (ATP) 6-01.1* (2015).
- [31] George F Luger. *Artificial intelligence: structures and strategies for complex problem solving*. Pearson education, 2005. ISBN: 978-0321545893.
- [32] J McCarthy et al. «Dartmouth Summer Research Conference on Artificial Intelligence». Em: (1956).
- [33] Noelia González Rodríguez. *How Deep Learning is Different from Machine Learning?* 2019. URL: <https://noeliagorod.com/2019/11/14/how-deep-learning-is-different-from-machine-learning> (acedido em 15/06/2020).
- [34] Uday Kumar; Diego Galar; Ramin Karim. *Industrial AI in maintenance: false hopes or real achievements?* 2020. URL: <https://www.maintworld.com/R-D/Industrial-AI-in-maintenance-false-hopes-or-real-achievements> (acedido em 01/05/2020).
- [35] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. 2019. ISBN: 978-1491962299.
- [36] L. Filipe Gomes Pereira. «Previsão de Falhas em Empanques Mecânicos da Refinaria de Matosinhos Usando Modelos de Machine Learning». Tese de mestrado. FEUP, 2018.

- [37] Scott Fortmann-Roe. *Understanding the Bias-Variance Tradeoff*. 2012. URL: <http://scott.fortmann-roe.com/docs/BiasVariance.html> (acedido em 01/06/2020).
- [38] Dewang Nautiyal. *Underfitting and Overfitting in Machine Learning*. 2018. URL: <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/> (acedido em 01/06/2020).
- [39] Robert Kelley. *Making Predictive Models Robust: Holdout vs Cross-Validation*. 2017. URL: <https://www.kdnuggets.com/2017/08/dataiku-predictive-model-holdout-cross-validation.html> (acedido em 01/06/2020).
- [40] Vijayaramaraju Poosapati et al. «Enabling Cognitive Predictive Maintenance Using Machine Learning: Approaches and Design Methodologies». Em: *Soft Computing and Signal Processing*. Springer, 2019, pp. 37–45.
- [41] Microsoft. *Azure AI guide for predictive maintenance solutions*. 2020. URL: <https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/predictive-maintenance-playbook> (acedido em 20/04/2020).
- [42] Márcia Baptista. «Machine Learning and Deep Learning for Prognostics and Predictive Maintenance of Aeronautical Equipment». Tese de doutoramento. IST, 2018.
- [43] H. Hashemian e Wendell Bean. «State-of-the-Art Predictive Maintenance Techniques*». Em: *IEEE Transactions on Instrumentation and Measurement* 60 (out. de 2011), pp. 3480–3492. DOI: 10.1109/TIM.2009.2036347.
- [44] R. Onanena et al. «Estimation of Fuel Cell Life Time Using Latent Variables in Regression Context». Em: *2009 International Conference on Machine Learning and Applications*. 2009, pp. 632–637.
- [45] Sheng Hong e Zheng Ou Zhou. «Application of Gaussian Process Regression for bearing degradation assessment». Em: *2012 6th International Conference on New Trends in Information Science, Service Science and Data Mining (IS-SDM2012)* (2012), pp. 644–648.
- [46] G. A. Susto et al. «Prediction of integral type failures in semiconductor manufacturing through classification methods». Em: *2013 IEEE 18th Conference on Emerging Technologies Factory Automation (ETFA)*. 2013, pp. 1–4.
- [47] U. Schopka et al. «Practical aspects of virtual metrology and predictive maintenance model development and optimization». Em: mai. de 2013, pp. 180–185. ISBN: 978-1-4673-5006-8. DOI: 10.1109/ASMC.2013.6552793.
- [48] Hongfei Li et al. «Improving rail network velocity: A machine learning approach to predictive maintenance». Em: *Transportation Research Part C: Emerging Technologies* 45 (2014). *Advances in Computing and Communications and their Impact on Transportation Science and Technologies*, pp. 17–26. ISSN: 0968-090X.

- [49] T. Praveenkumar et al. «Fault Diagnosis of Automobile Gearbox Based on Machine Learning Techniques». Em: *Procedia Engineering* 97 (2014), pp. 2092–2098. ISSN: 1877-7058.
- [50] Rune Prytz et al. «Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data». Em: *Engineering Applications of Artificial Intelligence* 41 (2015), pp. 139–150. ISSN: 0952-1976.
- [51] S. Biswal e G. R. Sabareesh. «Design and development of a wind turbine test rig for condition monitoring studies». Em: *2015 International Conference on Industrial Instrumentation and Control (ICIC)*. 2015, pp. 891–896.
- [52] Rubem Geraldo Vasconcelos Machado e Hilton de Oliveira Mota. «Simple self-scalable grid classifier for signal denoising in digital processing systems». Em: *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2015, pp. 1–6.
- [53] G. K. Durbhaka e B. Selvaraj. «Predictive maintenance for wind turbine diagnostics using vibration signal analysis based on collaborative recommendation approach». Em: *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2016, pp. 1839–1842.
- [54] O. Aydin e S. Guldamlasioglu. «Using LSTM networks to predict engine condition on large scale data processing framework». Em: *2017 4th International Conference on Electrical and Electronic Engineering (ICEEE)*. 2017, pp. 281–285.
- [55] Mikel Canizo et al. «Real-time predictive maintenance for wind turbines using Big Data frameworks». Em: *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE. 2017, pp. 70–77.
- [56] Tiago dos Santos et al. «Stator winding short-circuit fault diagnosis in induction motors using random forest». Em: *2017 IEEE International Electric Machines and Drives Conference (IEMDC)*. IEEE. 2017, pp. 1–8.
- [57] Chuan-Jun Su e Shi-Feng Huang. «Real-time big data analytics for hard disk drive predictive maintenance». Em: *Computers & Electrical Engineering* 71 (2018), pp. 93–101.
- [58] Eckart Uhlmann et al. «Cluster identification of sensor data for predictive maintenance in a Selective Laser Melting machine tool». Em: *Procedia Manufacturing* 24 (2018), pp. 60–65.
- [59] I. Amihai et al. «An Industrial Case Study Using Vibration Data and Machine Learning to Predict Asset Health». Em: *2018 IEEE 20th Conference on Business Informatics (CBI)*. Vol. 01. 2018, pp. 178–185.
- [60] T. Huuhtanen e A. Jung. «Predictive Maintenance of Photovoltaic Panels via Deep Learning». Em: *2018 IEEE Data Science Workshop (DSW)*. 2018, pp. 66–70.

- [61] K. Kulkarni et al. «Predictive Maintenance for Supermarket Refrigeration Systems Using Only Case Temperature Data». Em: *2018 Annual American Control Conference (ACC)*. 2018, pp. 4640–4645.
- [62] B. Luo et al. «Early Fault Detection of Machine Tools Based on Deep Learning and Dynamic Identification». Em: *IEEE Transactions on Industrial Electronics* 66.1 (2019), pp. 509–518.
- [63] F. Pedregosa et al. «Scikit-learn: Machine Learning in Python». Em: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [64] Abhinav Saxena e Kai Goebel. «Turbofan engine degradation simulation data set». Em: *NASA Ames Prognostics Data Repository* (2008).
- [65] Patrick Janhke. «Machine learning approaches for failure type detection and predictive maintenance». Em: *Master's Thesis, Technische Universitat Darmstadt* (2015).
- [66] John Templeton. *The Humble Approach Rev Ed: Scientist Discover God*. Templeton Foundation Press, 1998. ISBN: 978-1890151171.
- [67] Carl Sagan. *Dragons of Eden: Speculations on the evolution of human intelligence*. Ballantine Books, 2012. ISBN: 978-0345346292.