Computational Models for Behavioral Analysis of an Aircraft Turbine

Matheus Henrique Menezes Pinto

Thesis presented to the School of Technology and Management in the scope of the Master Degree in Mechanical Engineering.

Supervisors:
Prof. Ana Isabel Pinheiro Nunes Pereira
Prof. Arthur Hirata Bertachi

Bragança
2024

Computational Models for Behavioral Analysis of an Aircraft Turbine

Matheus Henrique Menezes Pinto

Thesis presented to the School of Technology and Management in the scope of the Master in Mechanical Engineering.

Supervisors:
Prof. Ana Isabel Pinheiro Nunes Pereira
Prof. Arthur Hirata Bertachi

Bragança
2024

Dedication

I dedicate this master's thesis, first and foremost, to God, who granted me the grace to wake up every morning with health and strength to pursue my goals. My gratitude extends to my family: my mother, Theresa; my father, Marcos; and my girlfriend, Kézya. Their presence, support, and encouragement have been essential on my journey. I also express my gratitude to my advisors, key figures in this work, Ana Isabel Pinheiro Nunes Pereira and Arthur Hirata Bertachi.

Abstract

In the era of globalization, emphasis on economic advancement becomes pivotal, intensifying market competition and necessitating novel strategies and technologies to meet the demands of society. Within the aviation domain, alongside the pursuit of cost reduction and process efficiency, safety remains paramount. In response to these challenges, this study introduces the application of artificial intelligence to evaluate the degradation level of a pivotal aviation component: the gas turbine.

This study is founded on a detailed case study utilizing data from the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) provided by the National Aeronautics and Space Administration (NASA). By integrating predictive maintenance principles and artificial intelligence models, the core objective is to proactively anticipate equipment operational conditions, enabling planned and precise repairs or replacements.

Our study stands out for implementing techniques for converting and categorizing the equipment's state in relation to its remaining useful life. By translating data, we have achieved potentially useful results for practical use. This approach not only allows for an assessment of gas turbine wear but also facilitates informed decision-making regarding maintenance and replacement. The ability to anticipate problems before they occur not only reduces the costs associated with unscheduled maintenance but also significantly increases passenger safety, demonstrating the positive impact of artificial intelligence in the aerospace industry. The results obtained are promising when compared to the existing literature, validating the potential practical application of the developed techniques.

Keywords: predictive maintenance, aeronautics, machine learning.

Resumo

Com o avanço da globalização, a economia torna-se cada vez mais competitiva, exigindo a implementação de novas estratégias e tecnologias para atender às demandas da sociedade. Na indústria aeronáutica, onde a segurança é fundamental, além da necessidade de reduzir custos e aprimorar a eficiência dos processos, surge a importância de inovar. Este trabalho se concentra na aplicação de inteligência artificial para avaliar o desgaste de um componente crítico da aviação - a turbina a gás.

A pesquisa é baseada em um estudo de caso que utiliza dados do C-MAPSS da NASA. Através da combinação de técnicas de manutenção preditiva e modelos de inteligência artificial, nosso objetivo é antecipar o momento em que o equipamento precisa de reparo ou substituição, permitindo um planejamento adequado.

Nosso estudo se destaca pela implementação de técnicas de conversão e categorização do estado do equipamento em relação à sua vida útil restante. Ao traduzir dados conseguimos atingir resultados potencialmente úteis para uso na prática. Essa abordagem não apenas permite uma avaliação do desgaste da turbina a gás, mas também facilita a tomada de decisões informadas sobre manutenção e substituição. A capacidade de antecipar problemas antes que ocorram não apenas reduz os custos associados à manutenção não programada, mas também aumenta significativamente a segurança dos passageiros, demonstrando assim o impacto positivo da inteligência artificial na indústria aeronáutica. Os resultados obtidos mostram-se promissores quando comparados à literatura existente, validando a possível aplicação prática das técnicas desenvolvidas.

Palavras-chave: : manutenção preditiva, aeronáutica, aprendizado de máquina.

Contents

List of Figures

Chapter 1

Introduction

The aircraft maintenance management is a discipline that combines engineering precision with administrative meticulousness, requiring a delicate balance between operational efficiency and uncompromising safety.

In the introductory section, an overview of the use of Machine Learning (ML) in aircraft maintenance management is provided, including the motivation behind this study and the objectives to be achieved.

1.1 Context

Equipment preservation is crucial in the industry, and maintenance management plays a fundamental role [45]. Furthermore, the pursuit of greater efficiency in the production process and increased profitability renders maintenance even more essential [33].

These activities often involve equipment downtime until they are restored to an acceptable operational state [33]. However, Industry 4.0 is reducing this impact by enhancing productivity through technologies such as Internet of Things (IoT), Artificial Intelligence (AI), and Big Data Analytics.

For this purpose, the advent of Predictive Maintenance (PdM), one of the main pillars of Industry 4.0, provides significant productivity gains in maintenance tasks with the aim of defining equipment tolerances, detecting faults, and irregularities. PdM fulfills its role by reducing the number of unexpected shutdowns [70].

IoT is defined as responsible for obtaining, organizing, and transmitting equipment data through sensors, generating Big Data. This machine history is used as input data in AI models which, when processed, generate predictions of possible failures [37].

AI tools are based on human behavior seeking to perform tasks and challenges in an enhanced manner, surpassing the conventional method of automation. ML, a subfield of AI, leverages computational processing to analyze past experiences and Big Data, identifying patterns and generating predictions about future data [11].

1.2 Motivation

Maintenance provides greater reliability and effectiveness throughout the industry, as well as safety [7]. For aviation, despite being highlighted as one of the safest means of transportation, accidents still result in fatalities [67].

When relating aircraft accidents to maintenance, they are 6.5 times more likely to be fatal. Moreover, about 10% of accidents have maintenance attributed responsibility [64]. In this context, engines and their components are paramount for maintenance, underscoring the need for stringent and regular procedures to ensure operational safety [52].

In seeking to mitigate maintenance errors, the advantage of using AI over human action lies in its processing power, ranging from response speed, error suppression to the search for correlations imperceptible to the human eye [11].

AI is already consolidated in the aeronautical sector; besides being efficient, it has a wide range of applications such as ticket price variation, advertising, chatbots. However, maintenance control is highlighted due to significant revenue losses caused by unplanned maintenance pauses, which are addressed through predictive maintenance [59].

From a global perspective, Industry 4.0 is advancing rapidly, becoming necessary from a competitive standpoint [58]. This highlights the need to explore the integration of AI and predictive maintenance to further improve safety and efficiency in the aviation industry and other sectors.

1.3 Goals

In this study, the main objective is to estimate the time of aircraft turbine shutdown, aiming to mitigate air accidents and reduce costs associated with unscheduled maintenance. To achieve this purpose, the following steps will be followed:

- Identify weaknesses in the aeronautical sector.
- Select a robust database related to the identified problem.
- Analyze the variables present in the chosen database.
- Conduct a literature review to identify previous studies that used the same database.
- Apply mathematical methods for data processing.
- Use training algorithms to create custom models.
- Validate the generated models and interpret the obtained numerical results.
- Evaluate and discuss the feasibility of implementing the proposed solutions based on previous studies found in the literature.

These steps will guide the research in pursuit of the central objective of estimating the time of aircraft turbine shutdown, with an emphasis on safety and operational efficiency.

1.4 Framework and Content

This work is organized into six chapters.

In the first chapter, the objectives and structure of the research are established, introducing the topic of predictive maintenance in aircraft turbines.

The second chapter consists of a literature review, identifying gaps in existing studies on predictive maintenance in aircraft turbines, exploring previous works and relevant research.

The third chapter explores the essential concepts and methods for the application of ML in predictive maintenance of aircraft turbines, addressing topics such as Industry 4.0, IoT, Big Data, and AI.

In the fourth chapter, the methods, algorithms, and techniques used to develop the predictive model of aircraft turbine failure are described, including data collection and preparation, as well as model training.

The fifth chapter presents the results obtained with the developed Machine Learning models, including analyses of the model's effectiveness in predicting failures, performance metrics.

The sixth and final chapter concludes the work, highlighting the main insights and contributions to the field of predictive maintenance in aircraft turbines, and providing suggestions for future research.

Chapter 2

Systematic Literature Review

Systematic Literature Review (SLR) is a method employed to evaluate scientific information related to a specific topic. Its main advantages are the rigorous assessment of study quality, extraction of essential information, and consolidation of relevant findings. In essence, SLR synthesizes relevant studies on a given subject [32].

To conduct a SLR, it is crucial to follow rigorous and systematic protocols, which promote clarity and comprehension of the obtained results. Essential steps include the precise definition of research questions, conducting a thorough literature search, establishing selection criteria, among other procedures [66].

Although not strictly followed, the methodology adopted for the implementation of the systematic review is the PRISMA. This tool aims to ensure that pertinent information is presented in a clear and objective manner, enabling other researchers to evaluate and replicate the results of the review [47]. The rationale for selecting this methodology stems from its extensive adoption and the existence of a published study in area [12].

This review cited was undertaken to scrutinize the usage of ML techniques in the domain of Predictive Maintenance (PdM), expounding upon the methods of ML and showcasing their deployment in this approach.

The primary aim of this review is to comprehend the ML techniques utilized in the PdM, along with the equipment being monitored by these methods. Additionally, this review strives to ascertain the nature of the data, i.e., whether it is real or synthetic and how the ML methods are being implemented in the PdM.

The review presented in [12] was published in 2019 and analyzed articles published before this date. The analysis revealed an increasing number of publications over the years, as shown in Figure 2.1. This demonstrates a growing interest and advancement in this field, providing further validation for the current study.

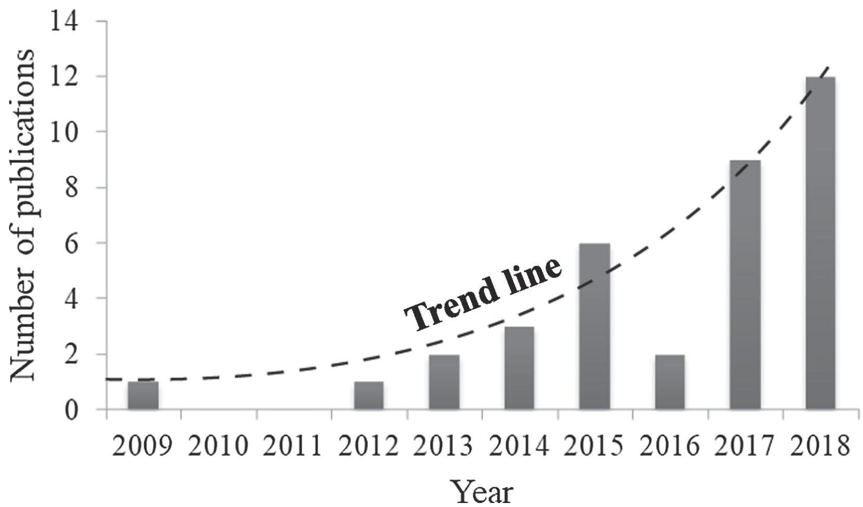

Figure 2.1: Number of papers per year with a trend line until 2018 [12]

2.1 Literature Review Process

The procedures followed to elaborate the review include, first of all, defining the research questions. This is the starting point for the review, as it is the questions that determine the objective and purpose of the research, ensuring that all relevant aspects are considered.

- Q_1. How is ML applied in PdM?
- Q_2. What are the ML methods used to predict the Remaining Useful Life (RUL) of equipment?
- Q_3. On which equipment are these methods applied?
- Q_4. What are the dataset used?

- Q_5. Are the data used real or synthetic?

Continuing with the analysis, it is essential to determine the sources where literature will be searched, such as important databases and scientific journals. For this study, three widely recognized databases in the scientific field were used:

- IEEEXplore[1]
- Scopus[2]
- Web of Science[3]

To perform the search for studies related to the subject in question, the study conducted by [12] was used as a basis. This study used keywords directly related to the scope of the research, "machine learning" and "predictive maintenance," both in their full and abbreviated forms. Furthermore, given that this is a topic in significant growth, as can be seen in Figure 2.1, a third keyword was included that further limits and directs the search to the research objective, the term "remaining useful life", according to Table 2.1. This was carried out with the aim of ensuring that the majority of the studies address the concept of time to failure, as well as reducing the large quantity of responses.

Table 2.1: Keywords searched

Full form	**Abbreviated form/Synonym**
Predictive maintenance	PdM
Machine learning	Artificial intelligence
Remaining useful life	RUL

The search process differs for each database, thus the script of keywords used is described below:

- **IEEEXplore**: ("predictive maintenance" OR "PdM") AND ("machine learning" or "artificial intelligence") AND ("RUL" or "remaining useful life")

[1]ieeexplore.ieee.org
[2]scopus.com
[3]webofscience.com

- **Scopus**: TITLE-ABS-KEY (("predictive maintenance" OR "PdM") AND ("machine learning" OR "artificial intelligence") AND ("RUL" OR "remaining useful life"))

- **Web of Science**: ALL = (("predictive maintenance" OR "PdM") AND ("machine learning" or "artificial intelligence") AND ("RUL" or "remaining useful life"))

It is worth noting that, in all searches, a period limit filter was applied, with the timeframe set from 2018 to the present day (2018-2023).

In this stage of the research, the selection parameters of the study are defined, which determine the criteria for inclusion and exclusion of the studies. However, there is a deviation from the PRISMA guidelines in the exclusion criteria, as there is a bias on the part of the author on the third item. The exclusion criteria defined are:

- E_1. Works published in more than one database

- E_2. Publications that do not pertain to PdM and ML.

- E_3. Publications that do not present results related to RUL.

- E_4. Publications that do not present any comparison means.

The first stage of exclusion involved the removal of duplicated studies, as many publications were found in more than one database. In the same phase, studies that were not available in English were also excluded. In the subsequent stage, studies that did not contain keywords related to PdM and ML were eliminated. For the next two stages, a complete reading of the documents was required. In the third stage, studies that did not intend to predict the RUL were removed, and finally, works that met all criteria but lacked a comparison mode in the literature were excluded.

At the end, relevant characteristics of each selected study are listed. These characteristics were used to compile a table that provides an overview of the studies, enabling a more precise comparative analysis.

Data extraction fields:

- D_1. Year of publication.
- D_2. Machine learning method used.
- D_3. Equipment, case, or scenario in which the PdM techniques were applied
- D_4. Origin of the data, whether it was synthetic or real.

2.2 Results and Discussion

Similar to the study described in [12], a timeline graph, was produced to illustrate the quantity of documents found each year, Figure 2.2. Although the study was conducted in March 2023 and therefore only includes a limited number of documents for that year, the trend line demonstrates an upward trajectory, indicating growth in the study.

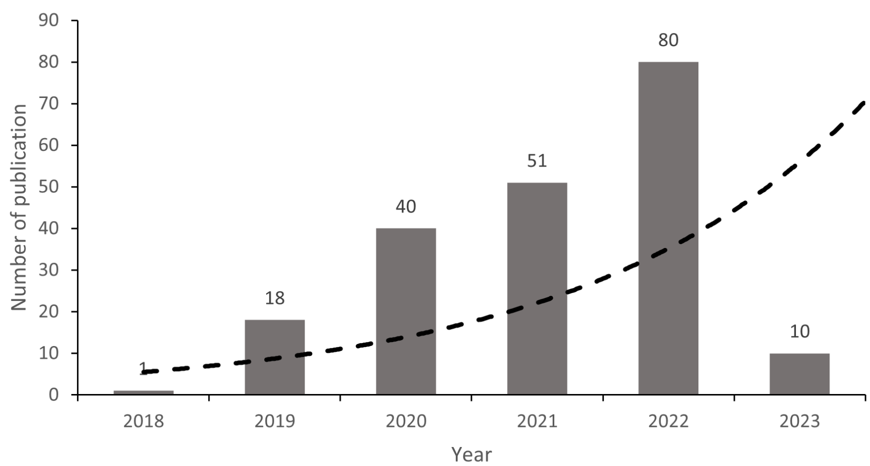

Figure 2.2: Number of papers per year with a trend line from 2018

A total of 355 documents were identified, of which 148 were duplicates. After applying exclusion criteria to the identified scientific publications, only 12 were selected for analysis. The flow diagram of the study, which follow to the guidelines presented in section 2.1, is exhibited in Figure 2.3. This diagram displays the number of excluded papers and can be accessed on the website[4].

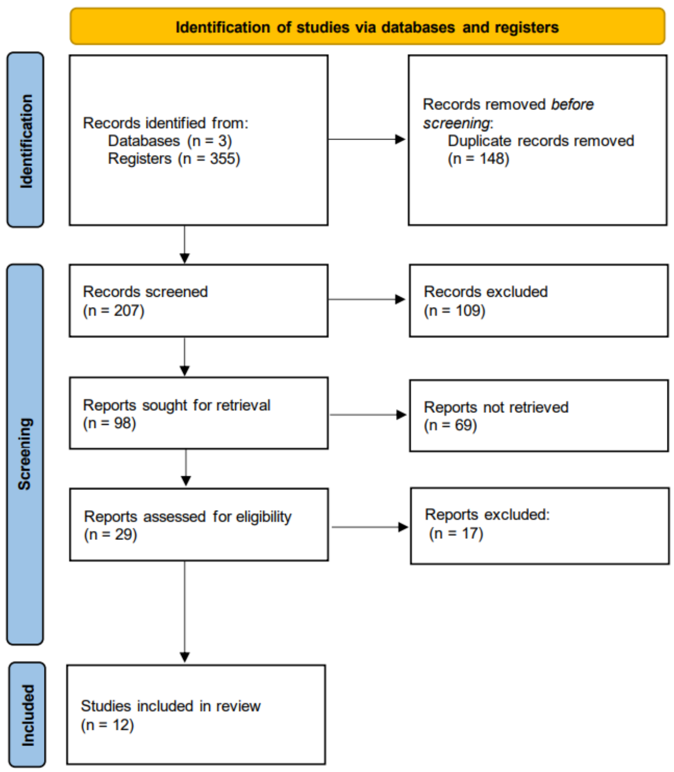

Figure 2.3: PRISMA Flow Diagram

[4] prisma-statement.org/PRISMAStatement/FlowDiagram

Upon finishing the review, a summary of the analysis of selected works is presented, from which it is possible to draw various conclusions. This information is presented through the Table 2.2.

It can be observed a wide variety of equipment applied to PdM, especially when it comes to data from real sensors, which represented 89% of the selected papers in [12]. In this research, however, these data represent less than half of the analyzed documents. Thus, it is possible to conclude that, for RUL prediction, researchers give more preference to synthetic data. Furthermore, the part that represents the synthetic data belongs to the same dataset, the CMAPSS, which suggests its great relevance and credibility.

Regarding the algorithms used, there is no unanimity, as various methods are applied with their respective conclusions. In comparison with [12], which categorized 33% Random Forest as the predominant algorithm, it can be noted that, even with the advancement of studies, ML architectures have not shown bias, and in general, various methodologies are tested for each study.

Table 2.2: Summary of the selected publications on predictive maintenance.

Reference	**ML method(s)**	**Case**	**Data type**[5]
[2]	SVR, ANN	Aircraft engine	SD
[56]	ANN, SVM	Bearings in grinding machines	RD
[31]	ANN	Aircraft engine	SD
[62]	k-NN, NB , RF, SVM	BLDC motor	RD
[19]	ANN	Wheel-bearing component of a railcar	RD
[54]	LR	Aircraft engine	SD
[57]	SVM, ANN	Wind turbine shaft bearings	RD
[40]	LR, DT, SVM, RF, k-NN, ANN	Aircraft engine	SD
[29]	ANN	Aircraft engine	SD
[1]	SVM	Aircraft engine	SD
[68]	LR, ANN, DT, RF	Aircraft engine	SD
[4]	RF	Rolling Bearings	RD

Chapter 3

Concepts and Methods

With the motivation of the work already established, this chapter will provide the essential theoretical foundation for the development and a deeper understanding of this work. This encompasses the study of maintenance, the utilization of AI, and its applications in the aeronautical context.

3.1 Maintenance

The concept of maintenance in the industry is fundamental to ensure the proper functioning of machines, equipment, and parts over time. Maintenance comprises a set of actions performed periodically, such as replacement, repair, or conservation [46]. Additionally, as defined in [3], maintenance is a combination of technical and administrative activities, including supervision, with the aim of keeping or restoring an item to suitable conditions to perform its function.

The history of maintenance can be understood in four major periods [30] .These periods began before the Industrial Revolution, when humans started developing tools and machines to facilitate their daily tasks, giving rise to mechanical mills and hammers. With the invention of the steam engine by James Watt during the Industrial Revolution, there was a revolution in work automation and mechanization; however, these early designs were prone to failures and degradation. This need to keep the machines operational gave birth to the concept of maintenance [27].

The first and second eras are marked by World War II, where machines and equipment, in general, were built to withstand greater loads than they were actually subjected to, exceeding their useful life. Additionally, mechanization was limited, necessitating manual labor. Thus, repairs were carried out only after breakdowns, which today are termed corrective maintenance. This model, although common at the time, proved to be inefficient and resulted in prolonged downtimes and high repair costs [30].

Right after this period, there was a rise in these procedures to meet the post-war market demand. Productivity began to take priority in the industrial environment due to the lack of qualified labor, which further propelled the study and application of maintenance to be done effectively [42]. It is emphasized in [30] that this progress is driven by necessity rather than opportunity.

The second generation of maintenance, in the late 19th and early 20th centuries, saw a gradual shift in planning and management being carried out more rigorously, with the introduction of equipment inspection techniques. Seeking greater utilization of machines, both in lifespan and efficiency, the idea of anticipating failures began to emerge, now known as preventive and predictive maintenance. This was a significant milestone in the evolution of maintenance practice as it helped reduce operational costs and improve factory efficiency [30].

By the 1970s, companies began to notice the high cost of maintenance compared to operation. Procedures were carried out at predetermined intervals and required production shutdowns, exacerbating the profit difference curve, as spending money on maintenance decreased production. The Just-in-Time management system, whose concept was to produce at the last possible moment according to demand, intensified this problem [30].

The high demand for perfect sizing, machine availability, and improvement of machine lifespan aiming for maximum profit gave rise to the third generation of maintenance. During this period, technologies such as the use of intelligent systems and new methods of predictive maintenance were introduced [30].

In the modern era, with the advent of the Fourth Industrial Revolution, maintenance has evolved even further. Technologies such as IoT, Big Data, and AI have revolutionized how organizations manage and perform maintenance on their assets. Now, it is possible to monitor machine performance in real-time, predict failures in advance, and even automate maintenance processes, making them more efficient and less costly.

At the moment, maintenance is divided into preventive and corrective, as shown in Figure 3.1 [46].

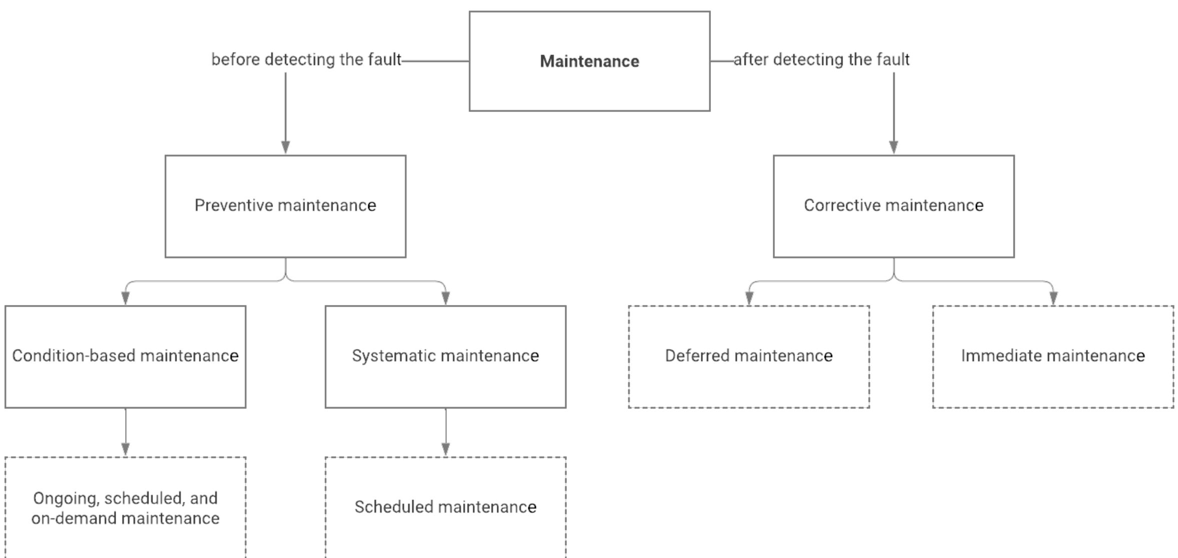

Figure 3.1: Maintenance classification [46]

With corrective maintenance defined as "maintenance carried out after a breakdown to restore an item to a condition in which it can perform a required function," in other words, this type of maintenance aims to correct problems that have already occurred [3].

It is also possible to differentiate corrective maintenance into two types: scheduled, where the element is not functioning properly, performing its function ineffectively, and the more critical, unscheduled, where the component is already in failure mode and requires full replacement, often occurring in an emergency manner [28].

This type of maintenance is associated with the first generation, that is, a backward and less envisaged management in today's context. Because it generates production interruptions and machine unavailability, it is related to the lack of maintenance management in the industry [30].

On the other hand, preventive maintenance seeks to anticipate corrective maintenance by proposing to avoid or even prevent equipment failures, with the definition being "maintenance carried out at predetermined intervals, or according to prescribed criteria, intended to reduce the probability of failures or degradation of the functioning of an item" [3].

Originating from the second generation, actions taken in preventive maintenance include, for example, replacing worn elements, cleaning, lubricating, adjusting, inspecting, and any activity that prevents machine unavailability [44].

These concepts, although they seem most appropriate, suffer from some negative points, such as the possibility of human errors, lack of replacement components, and the lack of premise related to the time interval between maintenance. These factors are corrected with the introduction of predictive maintenance [30].

3.1.1 Predictive Maintenance

Predictive maintenance is established as "maintenance that allows ensuring a desired service quality, based on the systematic application of analysis techniques, using centralized supervision means or sampling, to minimize maintenance and reduce corrective maintenance to a minimum" [3]. In other words, predictive maintenance seeks to minimize corrective maintenance. By reducing the number of unnecessary downtimes, predictive maintenance aims to perform maintenance only when it is indispensable for the machine [61].

The predictive technique is carried out by monitoring equipment data collected through monitoring and inspections, which brings a different characteristic from other types of maintenance, condition-based maintenance, as shown in Figure 3.2 [41].

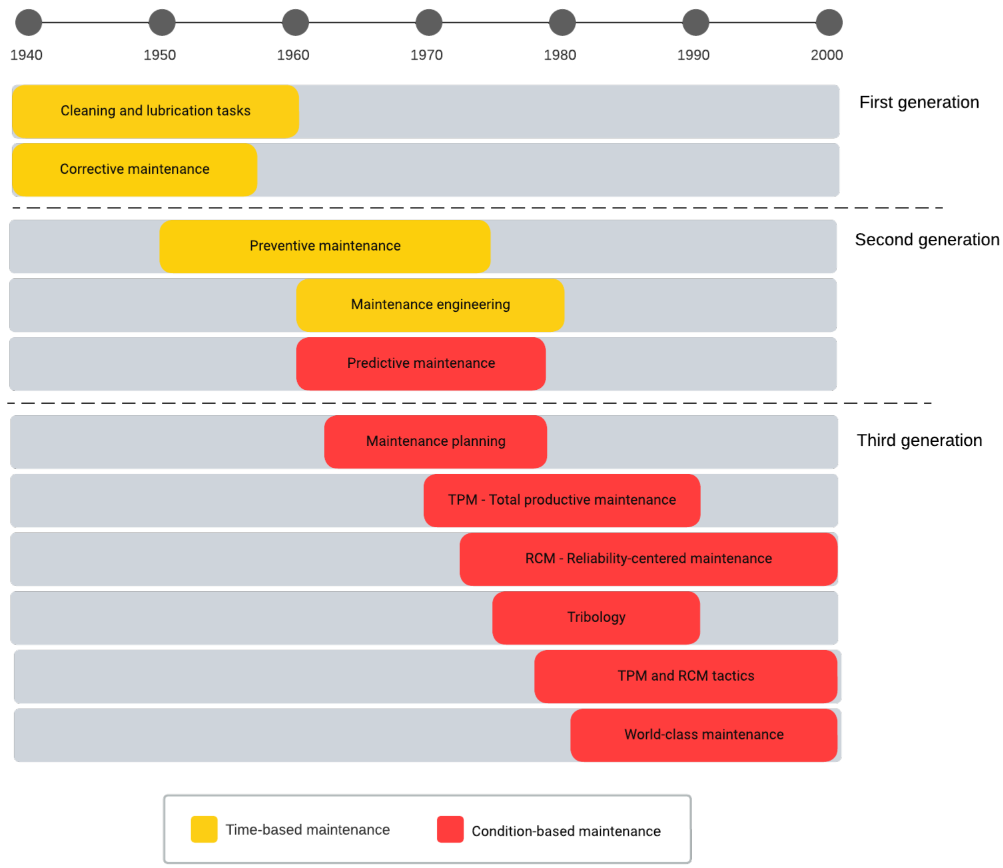

Figure 3.2: Technical development of maintenance over the decades [41]

Unlike the first and second generations marked by corrective and preventive maintenance, the techniques used were based on the equipment's usage time, which were performed at predetermined intervals, often resulting in exceeded corrective maintenance or even the need for corrective maintenance [42].

At the end of the second and the beginning of the third generation, the concept of predictive maintenance focuses on machine usage conditions. Based on studies of the current state and how the equipment is operated, it is possible to predict the time required for maintenance [41].

For all of this to be possible, there is a need for adequate monitoring, which incurs significant costs; therefore, the equipment must be studied to allow the installation of a monitoring system that justifies its maintenance cost. With all these mechanisms in place, what is now called Industry 4.0 begins [8].

Industry 4.0 is the terminology used to characterize the use of modern technology in industry, such as IoT, which allows machine communication to the internet, facilitating the use, transmission, and reception of data. This and other elements are facilitators of predictive maintenance [5].

One of the concepts of IoT is Production Monitoring, which records total performance on the production line in real-time. The data collected generates Big Data, which are input elements for predictive maintenance forecasting and decision-making [5].

The goal of an effective maintenance organization is to provide the necessary system performance at the lowest cost. Therefore, the maintenance approach should be based on a clear understanding of failure [44].

The curve that relates machine condition to the lifespan is described in Figure 3.3 [44].

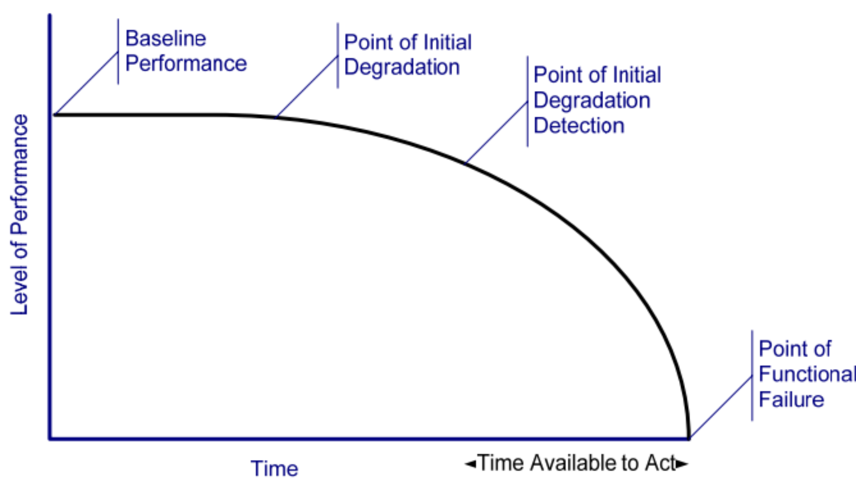

Figure 3.3: Generic equipment degradation over its lifespan [44]

Representing the temporal evolution of a system's or equipment's performance, highlighting crucial points such as the onset of degradation, detection of this degradation, and the occurrence of functional failure. This graph helps to understand how the machine's degradation state behaves over time [44].

Predictive maintenance acts by predicting the time when the machine's performance stage is reduced, thus implementing an appropriate maintenance plan. This type of approach is widely used in aircraft maintenance management [44].

3.1.2 Aircraft Maintenance

Flight safety is of critical importance in aviation, aiming to mitigate accident occurrences. Aircraft maintenance can be divided into two essential parts: maintenance involving the aircraft as a whole and maintenance performed on each of its components. One approach treats the aircraft as a single unit, while the second addresses each component individually [14].

An example of this is the classes, presented in Figure 3.4, developed by the National Civil Aviation Agency (ANAC) to certify the capability of each workshop in Brazil [10]. Although these classes may seem similar, they have significant differences. Aircraft maintenance as a whole, or its cell, cannot undergo tests after maintenance, unlike individual components, which often undergo approval before installation. However, these distinctions do not mean that components can tolerate errors, but indicate study guidelines to be followed [14].

About 75% of aviation accidents are attributed to human error, which can also be related to maintenance. According to Article 87 of the Brazilian Aeronautical Code, the prevention of aviation accidents is the responsibility of all individuals, physical or legal, involved in the manufacture, maintenance, operation, and circulation of aircraft [9].

Therefore, the goal of aircraft maintenance is to ensure flight safety, preventing and correcting failures so that they do not become accidents, aiming for the lowest cost and shortest time, so that the aircraft can return to operate at full capacity as quickly as possible [53].

For this purpose, predictive maintenance is used in aeronautical components, considered the most promising technology for maintenance management. Through it, it is possible to obtain precise information about failures and degradation levels, meeting all the needs of aeronautical maintenance, including time, cost, and safety issues [53].

Category	Class
Cell – Maintenance, modifications and repairs on cells	Class 1 - Aircraft made with composite material, approved up to 12500 lbf (5670 kgf) for airplanes and 6018 lbf (2730 kgf) for helicopters. Class 2 - Aircraft made with composite material, approved up to 12500 lbf (5670 kgf) for airplanes and 6018 lbf (2730 kgf) for helicopters. Class 3 - Aircraft made with metallic structure, approved up to 12500 lbf (5670 kgf) for airplanes and 6018 lbf (2730 kgf) for helicopters. Class 4 - Aircraft made with metallic structure, approved up to 12500 lbf (5670 kgf) for airplanes and 6018 lbf (2730 kgf) for helicopters.
Engine – Maintenance, modifications and repairs on aircraft engines	Class 1 - Conventional engines up to 400 hp (298 kW). Class 2 - Conventional engines over 400 hp (298 kW). Class 3 - Turbine engines.

Figure 3.4: Permission classes for aeronautical maintenance companies [10]

About 30% of delays in US airports are motivated by unscheduled aircraft maintenance [48]. Artificial intelligence operates in this sector by predicting the ideal downtime, based on sensor data [5].

An aircraft has thousands of parts; for example, a Boeing 737 has about 367,000 replaceable components [35]. This enormous quantity consequently implies a larger number of sensors and equipment to be monitored. Artificial intelligence benefits from having the capacity to process vast databases, providing adequate maintenance management for each aircraft individually [26].

3.2 Industry 4.0

All this methodology is made possible through Industry 4.0, also known as the fourth industrial revolution, a technological revolution that is changing the way factories operate worldwide. As mentioned in Section 3.1, it is a technological trend that is radically reshaping the production and operation processes of companies worldwide. The essence of Industry 4.0 lies in integrating digital, physical, and biological technologies, which together enable a high level of automation, connectivity, and intelligence [13].

At its core, Industry 4.0 seeks to integrate physical and digital systems to create highly efficient and adaptable production environments. This is achieved through the use of technologies such as IoT, cloud computing, Big Data, AI, augmented reality, among others. These technologies are applied in various areas, from production and logistics to quality control and maintenance [50].

One of the main characteristics of Industry 4.0 is the creation of smart factories, where machines, equipment, and production systems are interconnected and capable of communicating and cooperating autonomously. This allows for greater flexibility in production, enabling rapid adaptation to changes in market demand and mass customization of products [50].

Automation plays a significant role in Industry 4.0, allowing for the efficient and safe execution of repetitive and hazardous tasks. Robots and autonomous systems are widely used in industrial environments to perform operations such as assembly, welding, painting, packaging, among others. Automation not only increases productivity but also improves working conditions and reduces the risk of accidents [5].

Furthermore, mass customization is a growing trend in Industry 4.0, driven by consumer demand for increasingly personalized and tailor-made products. Thanks to digital technologies, companies can customize products according to individual customer preferences, offering a unique and differentiated experience [13].

Another important aspect of Industry 4.0 is the use of artificial intelligence to optimize processes and develop decisions. Machine learning algorithms and data analysis are employed in various applications, from production planning to demand forecasting, through optimizing delivery routes and inventory control. This allows for more efficient resource management and greater responsiveness to market changes [70].

Moreover, Industry 4.0 promotes a proactive approach to industrial maintenance. Using sensors and algorithms to continuously monitor the state of machines and equipment, identifying potential failures early and allowing intervention before unplanned production stoppages occur. This is an integration of IoT and the massive exploitation of Big

Data. IoT is providing comprehensive connectivity between devices and assets. At the same time, Big Data is empowering the analysis and interpretation of this data in search of valuable insights. Both technologies play an essential role in optimizing predictive maintenance in an increasingly complex industrial scenario [5], [13].

3.2.1 IoT and Big Data

The Internet of Things (IoT) is a network of interconnected devices, sensors, and machines that collect, communicate, and share data in real-time. In Industry 4.0, IoT provides the foundation for gathering detailed information about industrial assets' performance. Sensors embedded in machines, equipment, and products can collect a wide range of data, including temperature, pressure, vibration, humidity, and more. These devices constantly transmit information about the assets' state and operating conditions, creating a real-time view of operations [5].

The connectivity offered by IoT allows data to be transmitted to centralized management systems, where it can be processed and analyzed. This capability is crucial for predictive maintenance. For example, sensors on a machine can be used to detect an anomaly in operation before a complete failure occurs. This information is then transmitted to an analytics system that uses AI algorithms to predict when maintenance is needed, enabling proactive interventions before a serious failure occurs [5].

Big Data plays an important role in analyzing the massive volumes of information generated by IoT. Big Data is an approach that involves collecting, storing, and analyzing large datasets to identify trends, patterns, and hidden insights. In Industry 4.0, Big Data deals with the enormous streams of information generated by IoT, allowing for deeper and richer analysis [37].

Together, these technologies transform predictive maintenance into a highly effective process. IoT collects real-time data, Big Data processes it at scale, and AI interprets these data to make accurate predictions of failures [70].

3.2.2 Artificial Intelligence

Artificial Intelligence (AI) emerged with the idea of developing a machine capable of being programmed to perform any calculation it was presented with. Intelligence is not related to the ability to solve complex problems, but to adapt to new circumstances. Therefore, to perform this function, it is necessary for a machine dedicated to a specific task to have the freedom to change itself [17].

Defined as the ability of a computer to perform tasks that approximate those of intelligent beings, artificial intelligence attempts to replicate human ability [17]. In a modern approach, AI is established as a branch of computer science that focuses on automating processes through intelligence [36]. Regardless of how it is defined, AI has striking characteristics, described in [26], such as:

- Uses a processing unit for learning.
- Solves problems that cannot be solved by conventional computational approaches.
- Ability to process a huge amount of data in a short time;
- Capability to solve problems with data that do not follow patterns perceptible by the human mind.
- Although never achieving 100% accuracy, it provides sufficiently adequate answers for the established objectives.

These systems are capable of learning from data, identifying patterns, making decisions, and solving problems autonomously. Since then, significant advancements have been made, driven by increased computational capacity and the availability of large datasets. The use of this technology is particularly special in the aerospace industry [26].

Application of Artificial Intelligence in the Aeronautics Industry

Artificial intelligence (AI) is revolutionizing aviation [60]. In the commercial sector, models are developed to control ticket prices. Operating at extremely low profit margins, airlines require any tool that offers revenue improvement [20].

For operations, route planning is defined through the use of AI, allocating aircraft at the correct times and reducing idle time. In this way, fleet utilization is optimized with financial benefits [20].

In customer relations, automation comes through chatbots. AI models are developed to assist customers, making reservations, planning trips, and answering queries, all under the self-service concept [20].

Machine learning proves useful in aircraft design, acting on the concept of AI, capable of evaluating all possible variables and creating solutions unimaginable to humans. This kind of enhancement in aeronautics leads to optimized structural designs, predictive maintenance, and precise flight path calculations, resulting in significant advancements in efficiency and safety [20].

As seen in subsection 3.1.2, human error accounts for a significant portion of aviation problems [65]. To minimize this issue, autonomous aircraft techniques are developed, based on AI, which automate the aircraft so that it can fly and make decisions on its own. In commercial aviation, this methodology is already partially applied, with modern fleets capable of flying most of the route on autopilot, but human supervision is required due to regulations. However, in cargo and military aircraft, this process is fully autonomous [20].

Another way to assist in piloting is the use of "Intelligent" Digital Cockpit Assistants, a system that combines route planning with weather monitoring and establishes the best trajectory profiles. Another benefit of AI is within the cockpit with the Global Telecommunication Network responsible for reading wind forecasts, informing position, changing radio channels, among other functions that were previously attributed to the co-pilot [39].

Many aeronautical accidents result from aircraft maintenance failures, making predictive maintenance essential [45]. By extracting data from the thousands of sensors scattered throughout the aircraft, it is possible, with AI models, to predict the moment of failure to take proactive measures, such as strategic maintenance planning that reduces the chances of accidents, delays, parts unavailability for replacement, among others [20].

In the context of AI, there are other specialties, such as machine learning (ML) and deep learning (DL). While AI encompasses the broad field of automation through data science, ML and DL are branches that enable its application [43].

3.3 Machine Learning

ML is defined as the area of study that empowers computers to solve problems through algorithms. On the other hand, Deep Learning (DL) is a more advanced method that utilizes high levels of chained hierarchies to analyze data and thus assign more specific and complex weights to features [43].

The main difference between these concepts is the feature extraction. DL algorithms are capable of extracting special characteristics from data, especially in Big Data scenarios; DL is particularly useful for handling redundant data that carries little information with it [22].

Intelligences based on ML are categorized according to their learning: supervised, unsupervised, semi-supervised, and reinforcement learning [25].

Supervised learning deals with pre-labeled information. In this case, the training data provided to the algorithm contains the solution to the problem; the machine uses this data, along with its respective responses, to understand its behavior and create parameters for future predictions [43].

Among the possible solutions, there are two types: classification and regression. For classification problems, the algorithm is trained to identify a model which result belong to a discrete set, meaning the model classifies the data into categories, usually used for binary categories. On the other hand, in regression, the algorithm generates a functional relationship of the training parameters, meaning the output will be a function of the provided data [43].

Unsupervised learning, on the other hand, is fed with unlabeled data; the model finds patterns and similarities to understand the best way to classify. This type of approach is used when data acquisition is very costly or sometimes impossible. Thus, the machine learns without having a concrete result, only an analysis of the data [25].

In some cases, it is necessary to use both techniques, with and without supervision. Unsupervised intelligence is used to categorize the data, offering supervised learning better performance; in these cases, the algorithm is called semi-supervised [25].

Finally, there is a way to optimize the results of unsupervised intelligence through reinforcement learning. With this system, it is possible to create penalty or reward metrics for each prediction made, increasing the accuracy of your solution [25].

Among these, supervised learning is the most popular and common, as validated in chapter 2, with the most cited methods being: Decision Tree (DT), Random Forests (RF), Logistic Regression (LR), Support Vector Machine (SVM), and others [25].

3.3.1 Splitting Data

The reason for the existence of various AI methods lies in their characteristics; each intelligence is developed for a specific type of need. One of the most notable characteristics of an algorithm is its measure [26].

Although never fully achieved, the measure of a model is essential for it to become viable. The only way to determine this parameter is to use the data itself to perform tests after the model is trained; however, this information must be novel, meaning that when training the model, the model will not have access to test data [26].

For this reason, dataset is divided into two sets: one that will serve to perform training, called training data, and another used to validate and determine its behavior, called test data [25].

This concept of dividing a dataset into distinct sets is called split data. This separation is fundamental for evaluating the performance of a model, as it allows checking how well it generalizes to unseen data during training [43].

Cross-validation is a technique used in machine learning to evaluate the generalization ability of a model. Instead of relying on a single training and test set, cross-validation divides the data into multiple parts and performs multiple iterations of training and testing, providing a more robust estimate of the model's performance [24].

Comparing the traditional method of data splitting with k-fold cross-validation, we notice that the former tends to be more susceptible to variations in training and test sets, which can lead to biased performance estimates. On the other hand, k-fold cross-validation uses all samples from the dataset for training and testing, reducing bias and providing a more accurate assessment of the model [69].

K-fold cross-validation works by dividing the data into k parts (or folds), as illustrated in Figure 3.5, using each part as a test set once while the remaining are used for training. This process is repeated k times, ensuring that each part of the data is used as a test set in one iteration [69].

Figure 3.5: Graphical representat1ion of K-Fold Cross Validation [24]

There are several variations of k-fold cross-validation, such as stratified k-fold, nested k-fold, and repeated k-fold. These variations are useful in different contexts, such as dealing with imbalanced datasets, evaluating nested models, and increasing result reliability, respectively [24].

However, it is necessary to consider some factors, such as determining the ideal number of folds for a specific dataset, the influence of dataset size on the choice of k, and the proper interpretation of cross-validation results [24].

The advantages of k-fold cross-validation include a more reliable estimate of model performance, better utilization of available data, and identification of overfitting or underfitting issues. However, this can increase computational time, especially for very large datasets [25].

3.3.2 Feature Engineering

One of the reasons why AI methods are not perfectly accurate is their generalist nature. One problem that can occur when trying to improve accuracy is overfitting or underfitting, as shown in Figure 3.6 [25].

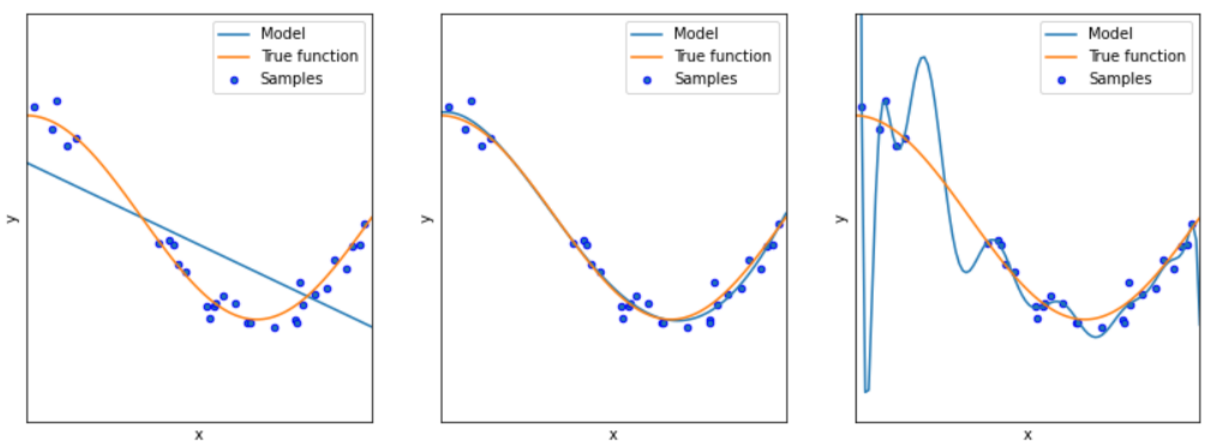

Figure 3.6: Graphical representation of overfitting and underfitting [63]

The first plot presents a case of underfitting, where the trained model exhibits much lower accuracy than expected. This problem occurs because either insufficient data was provided for the intelligence to learn, or it failed to capture the necessary patterns to make good predictions [63].

Overfitting, represented by the third plot, is another case as bad as underfitting. In this situation, the model incorporates noise, making the training data well reproduced. However, it creates a non-generalized intelligence, resulting in weak accuracy when using new information. Therefore, something close to the second plot is sought, a model capable of reproducing any data within the context of its training [63].

To solve these problems, there are three possibilities: the most obvious is to change the model; improve the data; or refine the hyperparameters. Hyperparameters are adjustable factors before training that control the behavior of the model aiming to improve its performance. The behavior of the model is highly dependent on these settings, and this process is usually manual, making it computationally expensive [43].

Improving the data is always possible and necessary, referred to as Feature Engineering. It involves preparing data for analysis, highlighting relevant aspects for machine learning methods. This not only improves model performance but also reduces training time and makes results more understandable. Removing duplicates is a crucial step in this process. Duplicates in datasets can distort results and impair prediction quality. Methods such as identifying duplicate records based on specific criteria are used to mitigate this problem. However, it is important to ensure that data representativeness is not compromised during this process [49].

Feature correlation is also an important aspect to consider. Highly correlated features can introduce bias into machine learning methods. To deal with this, techniques for calculating and visualizing feature correlations, such as correlation matrices, are employed. Strategies such as selective feature removal or aggregation can be adopted to handle this correlation, ensuring that only the most relevant information is considered by the model [49].

Furthermore, another technique that can be used is SVM, PCA, correlation, FS a widely used algorithm for feature selection. It uses support vectors and maximizing margins to define decision boundaries. Methods such as Recursive Feature Elimination (RFE) and weight coefficient analysis are commonly employed with SVM to select the most relevant features for the model [34].

It is important to highlight that the benefit of performing feature engineering is directly related to the ability to improve model performance. By carefully selecting and transforming features, it is possible to reduce noise in the data, increase prediction accuracy, and even facilitate model interpretation. This is especially important in complex and high-dimensional datasets, where feature quality can have a significant impact on the final result [34].

However, it is important to recognize that the feature engineering process also has some drawbacks. For example, it can be time-consuming and require deep knowledge of the problem domain and statistical and machine learning techniques. Additionally, there is a risk of introducing bias into the model by manually selecting features, which can lead to biased or unrepresentative results [34].

In terms of efficiency, performing feature engineering can be highly beneficial when done correctly. By creating more relevant and informative features, it is possible to significantly improve model performance, reducing overfitting and increasing the ability to generalize to new data [25].

With the two approaches defined for improving the accuracy of a machine learning model: refining hyperparameters and improving data through feature engineering techniques. As mentioned, it is important to consider the possibility of changing the model method. Different machine learning model methods have varied capabilities to handle different types of data, structures, and relationships between variables. Therefore, changing the model method can be a viable option to improve accuracy, as long as it is done judiciously, based on analysis and experimentation [43].

3.3.3 Machine Learning Methods

As mentioned, Machine Learning methods are algorithms and techniques that enable computers to learn from data, identifying patterns and making predictions or classification without being explicitly programmed to do so. They constitute a fundamental area of artificial intelligence and play an important role in predictive maintenance.

A variety of methods are employed in machine learning, each method having its own characteristics and approaches, offering different advantages and suitability to different types of data. In this section, we explore six of these methods: Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression (LR), Multi-Layer Perceptron (MLP), and Naive Bayes, outlining their strategies and underlying processes.

Decision Tree are a flexible approach that stands out for its simplicity. By splitting the data based on split conditions, trees can capture nonlinear patterns in the data. Each node of the tree represents a split in the data based on a specific feature, and the leaves contain the predicted values. A significant advantage of decision trees is the ability to handle different types of data, including categorical and continuous variables, without the need for extensive preprocessing. However, decision trees do not tend to have the same accuracy as other more complex methods and can be prone to overfitting if not properly controlled through regularization techniques, such as limiting the depth of the tree [21].

Random Forest a powerful extension of decision trees that overcomes many of the limitations associated with individual trees. By constructing multiple trees with random samples of the data and independent variables, Random Forests reduce variance and overfitting. The bagging technique combined with aggregation of predictions from individual trees results in a robust and stable model capable of handling a variety of datasets. Additionally, the randomness introduced in the tree construction process helps reduce correlation between them, which contributes to the model's diversity and generalization ability. Random Forests are highly scalable and suitable for large and complex datasets, making them a popular choice in a wide range of regression applications [38].

Support Vector Machine are primarily known for their effectiveness in classification problems but are also applicable to regression. By utilizing the concept of maximizing the margin between data points and the regression line, SVM seeks to find the best fit line for the data. This is done by optimizing the parameters of the regression function, aiming to minimize an appropriate loss function. Although SVMs may be more complex to interpret compared to other methods, they have the ability to effectively handle high-dimensional data and are robust to outliers [38].

Logistic Regression it is a fundamental technique that offers a straightforward understanding of the relationship between independent and dependent variable. Modeling this relationship as a straight line, linear regression is widely used due to its simplicity and interpretability. Although linear regression is sensitive to assumptions such as linearity and independence of residuals, it can be extended to handle nonlinear relationships and multicollinearity, making it highly versatile [43].

Multi-Layer Perceptron is another interesting approach, which represent a powerful class of artificial neural network methods for regression tasks. Comprising multiple layers of neurons, MLPs are capable of learning complex data representations, allowing for the modeling of nonlinear relationships. The training process involves optimizing the weights of connections through algorithms such as backpropagation. Although MLPs may require more hyperparameter tuning and training time compared to other methods, they are able to capture complex relationships in the data and offer competitive performance in a variety of scenarios [51].

The **k-Nearest Neighbors** algorithm is a simple and intuitive technique for regression problems. By computing the average or weighted average of the values of the nearest neighbors, KNN makes continuous predictions based on the training data. Although k-Nearest Neighbors Algorithm (KNN) is sensitive to the value of k and the chosen distance metrics, it is often used due to its ease of implementation and interpretation. However, KNN can be computationally expensive for large datasets and may not be suitable for high-dimensional data [38].

Finally, the **Naive Bayes** method is known for its computational efficiency. It relies on Bayes' theorem to calculate the probability of a class given a set of observed features. One of the distinctive features of Naive Bayes is its assumption of independence between attributes, which simplifies the calculation of conditional probabilities. This approach is especially useful in high-dimensional datasets, although the assumption of independence may not be realistic in all situations. Despite this, Naive Bayes is widely applied in classification problems, such as text analysis and spam filtering, due to its simplicity and satisfactory performance in most cases [18].

Each of these methods has its own advantages and limitations, and the choice of the appropriate method depends on the specific problem and the characteristics of the available data.

3.3.4 Performance Metrics

Validation is one of the most important parts in the development of an AI model. There are several ways to validate the performance of the model, for that it is necessary to define measures [25].

Regression Metrics in Machine Learning Methods

When evaluating regression methods in Machine Learning, it is important to employ appropriate metrics to understand their performance. Among the various available metrics, some stand out for their utility and interpretability. In this section, five of these metrics will be presented: Mean Squared Error (MSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), Variance, and Maximum Error, which offer valuable insights into the accuracy of the regression model predictions [6].

The MSE is a fundamental metric in evaluating regression methods as it provides a quantitative measure of the model's performance relative to the actual values. By squaring the differences between predictions and actual values and calculating their average, MSE penalizes larger errors more significantly. This means that the model is evaluated based on the magnitude of errors, providing a detailed understanding of its ability to make accurate predictions. However, it is important to be aware that MSE can be sensitive to outliers, which may affect its interpretation. Therefore, when using MSE, it is essential to consider the specific context of the problem and analyze whether the results are influenced by extreme values in the data [6]. The MSE is expressed by the formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.1}$$

Where:

- n is the total number of observations;

- y_i is the actual value of the variable of interest;
- $\hat{y}_i$ is the value predicted by the model.

On the other hand, MAE offers a different approach to evaluating the accuracy of the regression model. By calculating the average of the absolute differences between predictions and actual values, MAE provides a direct measure of the average prediction error, without considering the direction of errors. This characteristic makes MAE especially useful in situations where the magnitude of errors is more relevant than their direction. Additionally, MAE is robust to outliers, meaning it is not as influenced by extreme values in the data as MSE. This makes MAE a reliable metric for evaluating model performance in a variety of contexts, especially when an intuitive interpretation of prediction error is desired [6]. The MAE is formulated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.2}$$

Another essential metric in evaluating regression methods is the coefficient of determination, commonly known as R^2. This metric provides a more comprehensive view of the model's performance, providing information about the proportion of variability in the dependent variable that is explained by the independent variables in the model. R^2 ranges from 0 to 1, where a value closer to 1 indicates that the model is able to explain a large portion of the variability in the data, while a value closer to 0 indicates that the model is not capturing variability well [18]. Given R^2:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.3}$$

One of the main advantages of R^2 is its direct interpretability. By providing a measure of the proportion of variability explained by the model, R^2 allows for a clear understanding of the model's explanatory power relative to the observed data. However, it is important to note that R^2 does not indicate the overall quality of the model, as it can be influenced by various factors such as overfitting or inclusion of irrelevant variables. Therefore, it is essential to complement model evaluation with other metrics and validation techniques to obtain a comprehensive assessment of performance [18].

Explained variance is another useful metric in evaluating regression methods, offering a direct measure of the proportion of the total variability in the data that is explained by the model. This metric complements R^2, providing a more detailed view of the model's explanatory power relative to the observed data. By calculating the difference between the total variance of the data and the variance of the model residuals, explained variance highlights the amount of variability that is explained by the independent variables included in the model [18].

Like R^2, explained variance offers an intuitive interpretation of the model's explanatory power, allowing for a clear understanding of how much of the variability in the data is captured by the independent variables. However, it is important to remember that explained variance does not provide a measure of the overall quality of the model but rather how well it is able to explain the variability in the observed data. Therefore, it is crucial to consider other metrics and validation techniques for a comprehensive assessment of regression model performance [18].

In addition to the mentioned metrics, Maximum Error, or Max Error, plays an important role in evaluating regression methods. While metrics like MSE and MAE offer an average view of model performance, Max Error highlights the worst-case error, revealing the largest deviation between the model prediction and the actual value across the entire dataset [6].

The interpretation of Max Error depends on the specific context of the problem. In situations where accurate predictions are essential in all cases, a high Max Error may indicate critical failures of the model to correctly predict certain scenarios. On the other hand, in problems where significant errors in extreme cases are acceptable, a higher Max Error may be tolerable [6].

By considering Max Error in the evaluation of regression methods, it is possible to identify specific areas where the model is failing more significantly and direct efforts towards improvements in those areas. This metric complements other performance metrics, offering a more comprehensive understanding of the model's behavior in different situations and scenarios [6].

Regression Metrics for Remaining Useful Life in Machine Learning

In the context of predicting remaining useful life, it is important not only to assess how accurate predictions are overall but also to pay special attention to the lower values of the distribution of the variable of interest, as these values often have critical significance. In addition to traditional metrics such as MAE and MSE, which are commonly used in prediction problems, this section explores more specific and customized metrics [6], [61].

In this work it is explored the development of a customized metric called Weighted Absolute Percentage Error (WAPE). This metric goes beyond traditional metrics, aiming to enhance the evaluation of model performance in predicting the remaining useful life of turbines. As detailed in the workings of MAE, WAPE is calculated using its principles but for specific application in this context. This metric plays a fundamental role in providing a more comprehensive and refined analysis of predictions at critical values of the variable of interest, allowing for a more precise and targeted assessment of model performance [15].

The use of WAPE is an effective strategy for evaluating prediction methods, especially when one wants to give more weight to errors in predictions of lower values of the distribution of the variable of interest, as is the case with predicting the remaining useful life of turbines [15].

Its ability to not be affected by extreme values makes it an appropriate choice in many contexts. However, in situations where it is crucial to more heavily penalize errors that affect the lower values of the variable of interest, the WAPE approach proves valuable [16].

In this approach, weights are calculated based on the statistical order of the values of the variable of interest. For this, we use z-scores, which are calculated by normalizing the values of the variable of interest relative to the first quartile (Q1) and the third quartile (Q3) of the distribution. Thus, weights are assigned proportionally to the deviations from these quartiles, reflecting the importance of accurately predicting the lower values.

The formula for WAPE is as follows:

$$WAPE = \frac{1}{n}\sum_{i=1}^{n}\left(\left|\frac{y_i - Q1}{Q3 - Q1}\right| \cdot |y_i - \hat{y}_i|\right)$$

Where:

- n is the total number of observations;
- y_i is the actual value of the variable of interest;
- $\hat{y}_i$ is the value predicted by the model;
- $Q1$ is the first quartile of the distribution of the variable of interest;
- $Q3$ is the third quartile of the distribution of the variable of interest.

By more heavily penalizing errors that occur in the lower values of the distribution of the variable of interest, WAPE stands out as a valuable metric for cases where it is essential to obtain accurate predictions in this segment of the distribution. This approach offers an effective way to assess model performance, ensuring that they are sensitive to the specific needs of predicting the remaining useful life of turbines [16].

Classification Metrics in Machine Learning

Just like in the evaluation of regression methods, evaluating classification methods in Machine Learning also requires the use of appropriate metrics to understand their performance. Among the various available metrics, some stand out for their relevance and interpretability. In this section, five of these metrics will be presented: accuracy, precision, recall, F1-score, and Receiver Operating Characteristic (ROC).

The accuracy metric is a fundamental measure for evaluating the performance of classification methods. It provides an overall view of how well the model is correctly classifying instances relative to the total number of evaluated instances. In simple terms, accuracy is the proportion of correct predictions relative to the total predictions made by the model [23].

To understand how accuracy works, it is important to first understand some key concepts:

1. True Positive (TP): Cases where the model correctly classified an instance as positive.

2. False Positive (FP): Cases where the model incorrectly classified an instance as positive when it was actually negative.

3. False Negative (FN): Positively labeled cases incorrectly classified by the model as negative, i.e., cases the model should have identified as positive but did not.

4. True Negative (TN): Cases where the model correctly classified an instance as negative.

Simply put, we first count the number of correct predictions made by the model. Then, we divide this number by the total number of instances evaluated. The formula for accuracy is:

$$Accuracy = \frac{\sum(TP + TN)}{\sum(TP + FN + TN + FP)} \tag{3.4}$$

Although accuracy is a useful and easy-to-understand metric, it has some limitations. For example, in imbalanced datasets, where one class is much more prevalent than others, accuracy can be misleading. A model could achieve high accuracy simply by always predicting the majority class, without actually learning anything about the other classes [23].

Moreover, in problems where the costs of false positives and false negatives are very different, accuracy may not be the most appropriate metric. In such cases, metrics such as precision, recall, or F1-score may be more informative [23].

Despite its limitations, accuracy is still an important measure, especially when all classes are equally important, and the dataset is well-balanced. However, for a more complete assessment of model performance, it is recommended to consider other metrics in conjunction with accuracy [23].

Another metric used in the evaluation of classification methods is the precision metric, often used in problems where the relevance of positive predictions is crucial. It provides an indication of the proportion of positive predictions correctly made by the model relative to the total positive predictions made [18].

Precision is calculated by dividing the number of true positives by the total predicted positives by the model, i.e.:

$$Precision = \frac{\sum TP}{\sum(TP + FP)} \tag{3.5}$$

In simpler terms, precision answers the question: "Of all instances that my model classified as positive, how many were actually positive?" [18]

High precision indicates that the model has a low rate of false positives, meaning when it predicts an instance as positive, it is generally correct. On the other hand, low precision may indicate that the model is classifying many negative instances as positive, which can be problematic in scenarios where false positives are costly or undesirable [18].

However, precision alone can be misleading, especially in imbalanced datasets, where one class is much more prevalent than the other. In such cases, a model can achieve high precision simply by predicting the majority class all the time, completely ignoring the minority class. Therefore, precision should be interpreted in conjunction with other evaluation metrics, such as recall, F1-score, and confusion matrix, to get a complete understanding of the performance of the classification model [18].

An alternative is the recall metric, particularly used in contexts where identifying all positive cases is essential, such as in fraud detection systems, medical diagnoses, or spam detection. Recall, also known as sensitivity or true positive rate, quantifies the model's ability to correctly identify all positive examples relative to the total real positive examples present in the data [23].

The formula for recall is given by:

$$Recall = \frac{\sum TP}{\sum(TP + FN)} \tag{3.6}$$

Directly, recall measures the proportion of positive examples that the model correctly classified relative to the total positive examples existing in the data. High recall indicates that the model is able to identify most positive cases, while low recall suggests that the model is failing to detect many positive examples [23].

In addition to these, the F1-score metric, typically used in binary machine learning problems, where there are two classes to be predicted, such as yes or no, positive or negative, true or false, among others. It combines two other important metrics, precision and recall, into a single value, providing a comprehensive view of the model's performance [23].

Now, the F1-score metric is a harmonic mean between precision and recall. Harmonic mean is used because it heavily penalizes low values in both metrics, ensuring that the F1-score value is high only if both precision and recall are high [23]. The formula for calculating F1-score is given by:

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.7}$$

Therefore, the F1-score value will be high if the model has both high precision and high recall. This means that the model not only minimizes false positives but also is able to identify most true positives. On the other hand, a low F1-score value indicates that the model is failing in one or both metrics, meaning it is having trouble correctly identifying positive instances or is incorrectly classifying negative instances as positive [23].

To get an overview of all these metrics and better understand the behavior of the model, the confusion matrix is used, represented in Table 3.1. Here, the rows are instantiated for the actual values and the columns for the predicted ones. In the main diagonal, the values that the model predicts successfully are recorded, the TP and TN, while in the secondary diagonal, the errors are pointed out, being FP and FN [43].

Table 3.1: Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Finally, the ROC metric is a fundamental tool in evaluating binary classification methods, especially in problems where the goal is to distinguish between two classes. It offers a way to visualize and assess the performance of a binary classifier at different decision thresholds [18].

The functioning of the ROC metric can be understood from a graph, exemplified in Figure 3.7, which represents the TP rate against the FP rate for all possible threshold values. Here, "positive" refers to the class we are trying to detect or predict [18].

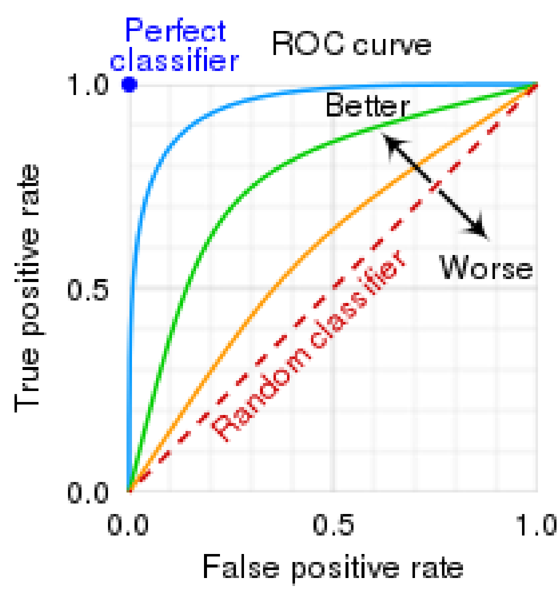

Figure 3.7: Graphical representation of the ROC curve [18]

The ROC curve is then generated by plotting the TP rate on the y-axis and the FP rate on the x-axis, for different threshold values. The closer the point is to the upper-left corner of the graph, the better the classifier's performance, indicating a high true positive rate and a low false positive rate [18].

Furthermore, the area under the ROC curve, Area Under the Curve (AUC), is often calculated as a summary measure of the classifier's performance. The higher the AUC, the better the classifier's performance in distinguishing between the two classes. An AUC of 0.5 indicates a classifier making random predictions, while an AUC of 1.0 indicates a perfect classifier [18].

Chapter 4

Case Study

In this chapter, we will address the detailed development of the implementation process of Machine Learning algorithms, outlining each essential step that contributes to the creation of effective models. We will begin by presenting the dataset chosen for this study, carefully justifying the reason behind its selection. Additionally, we will conduct an in-depth analysis and visualization of the dataset, seeking crucial insights for the development of Machine Learning models.

The clear and precise establishment of the objectives of this Machine Learning application is a fundamental step that we will address attentively. Clearly defining the goals we intend to achieve with our models is essential for the project's success.

Finally, the implementation of this chapter will be conducted using the Python programming language, leveraging the Matplotlib, Numpy, Pandas, and Scikit-Learn libraries. These tools will play a fundamental role in executing the steps of developing and evaluating Machine Learning models.

4.1 Workflow

In this section, we will detail how the application of ML techniques in the maintenance scenario will be implemented in practice. Executing this workflow will be essential to ensure the quality and effectiveness of the ML models developed, from the beginning of the project to its completion and deployment. For this purpose, a flowchart has been developed to serve as a guide for performing the tasks, as shown in Figure 4.1:

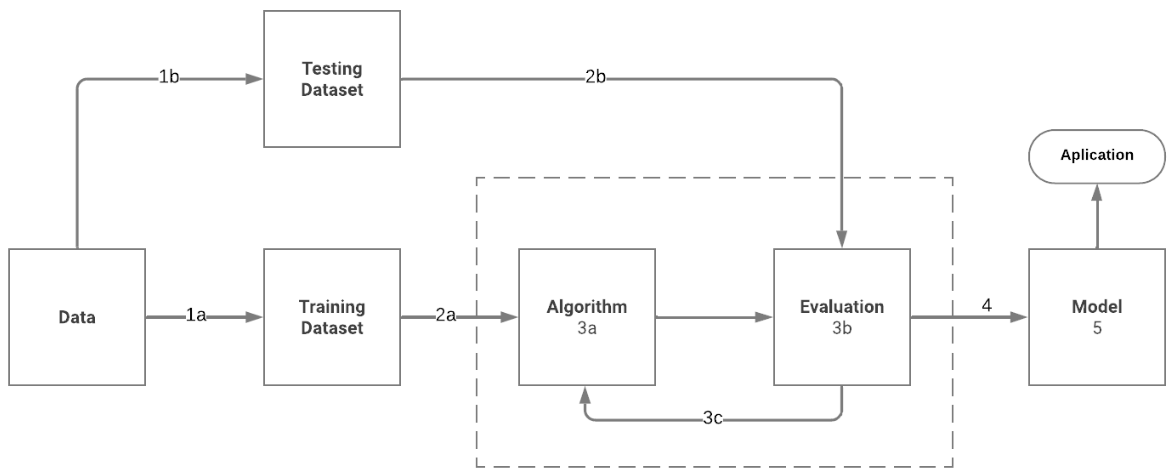

Figure 4.1: Machine Learning workflow

Where:

1. Gathering data;
2. Data pre-processing;
3. Researching the model that will be best for the type of data;
4. Training and testing the model;
5. Evaluation

Initially, we will begin with a clear definition of the objectives of the ML project and the precise delineation of the scope of work. This step will allow us to fully understand the demands of the problem at hand and establish relevant performance metrics to evaluate the model's success.

Next, we will move on to data acquisition and preparation. This will involve collecting raw data from reliable sources, followed by a preprocessing phase for cleaning, integration, and transformation of the data into a suitable format for analysis.

Subsequently, we will proceed to explore and analyze the data, where we will apply visualization techniques and statistical analysis to better understand the characteristics of the data and identify possible patterns and relationships between variables.

Based on the analysis performed, we will proceed to feature selection and engineering, choosing the most relevant features for the ML model and, if necessary, creating new features to enhance its predictive performance.

Next, we will select the most appropriate ML algorithms for the problem at hand and train the models using the prepared data. During training, we will conduct cross-validation and hyperparameter tuning to ensure the model performs optimally and avoids overfitting.

After training, we will evaluate the model's performance using independent test datasets, utilizing metrics appropriate to the problem at hand. If necessary, we will make adjustments to the model, returning to earlier stages of the workflow to improve its performance.

Finally, once the model has been evaluated and approved, we will discuss its deployment in a production environment, potential changes for improvement in accuracy, performance, and scalability. During this phase, we will conduct a detailed analysis of the model's performance in relation to production requirements, identifying bottlenecks or areas for improvement. Additionally, it is crucial to ensure that the model is robust enough to handle edge cases or unforeseen situations. Once this phase is completed, we will be ready for the deployment of the model into production, ensuring it meets the defined operational expectations and requirements.

4.2 Data Description

Although the field of predictive aircraft maintenance is on the rise, it is extremely rare to find information regarding sensors on aircraft. Even though one of the most established cloud-based aircraft data systems holds about 12 Petabytes, this data is limited to large companies [60].

Therefore, for this work, a dataset provided by NASA was used. A study of damage propagation in aircraft with a run-to-failure concept, which examines the operation of the component until its failure. The data is obtained through the C-MAPSS software, containing information about different flight conditions and sensors installed in the aircraft turbine [55].

C-MAPSS is a turbofan engine simulation system that can monitor engine health and its parameters through the graphical interface, established by the *US Army Research Laboratory*.

Its central purpose is to provide a platform for the development and validation of prognosis and predictive maintenance algorithms at an extremely low cost compared to real sensors, as well as to produce data at an unmatched speed. These algorithms are essential for predicting the degradation and future performance of aircraft turbines, allowing for more effective maintenance management and increasing the safety and efficiency of these critical air transport systems.

The considerable volume of data contained in the database reflects the complexity and diversity of aircraft turbine systems, thus providing a robust and comprehensive dataset for analysis.

This dataset is already widely used in studies and has proven to be reliable, as shown in chapter 2. Four datasets representing different conditions to which the turbines are subjected are provided:

- Altitudes ranging from sea level to 40,000 feet;
- Mach number from 0 to 0.90;
- Sea level temperature from -60 to 103°F.

There are 26 signals recorded, of which: 21 are sensor recordings contaminated by noise, as shown in Table 4.1; 3 represent the linked conditions; and the remaining 2 present the engine identifications and the number of cycles.

Comprised of four individualized datasets, each one encompasses approximately 20,000 operational cycles. This extensive amount of data enables a detailed investigation into the

Table 4.1: Measurement parameters of sensors generated by C-MAPSS

Index	Description	Units
1	Total temperature at fan inlet	ºR
2	Total temperature at low-pressure compressor outlet	ºR
3	Total temperature at high-pressure compressor outlet	ºR
4	Total temperature at low-pressure turbine outlet	ºR
5	Fan inlet pressure	psia
6	Total pressure at bypass duct	psia
7	Total pressure at high-pressure compressor outlet	psia
8	Physical fan speed	RPM
9	Physical core speed	RPM
10	Motor pressure ratio	RPM
11	Static pressure at high-pressure compressor outlet	-
12	Fuel flow ratio to Ps30	-
13	Corrected fan speed	-
14	Corrected core speed	-
15	Deviation rate	-
16	Burner fuel-air ratio	-
17	Bleed enthalpy	-
18	Required fan speed	RPM
19	Required corrected fan speed	RPM
20	High-pressure turbine bleed	lb/s
21	Low-pressure turbine bleed	lb/s

behavior and performance of the engines across multiple operating cycles. Each dataset provides a unique insight into the operational characteristics of the engines, allowing for comprehensive analyses of wear, degradation, and failure prediction.

To make the dataset more realistic, the initial level of wear of each turbine is different. Considering it normal in a real scenario, this information is not quantified in the documentation. Additionally, the exact moment of failure cycle, noise amplitude are not explicitly delivered.

Furthermore, it was necessary to work on the concept of Time to Failure, determining the RUL of the turbines. In the context of the study of aircraft damage propagation, cycles refer to the number of operations or operating cycles that an aircraft turbine performs from the beginning of its life cycle to the moment of failure. These cycles are continuous and incrementally increased with each operation of the aircraft. Each cycle represents a unit of measurement for assessing the progressive degradation of turbine components over time and use.

The life cycle of an aircraft turbine begins with the first cycle, which is recorded when the aircraft enters operation for the first time. From that point on, each time the aircraft is used, whether for takeoff, flight, or landing, the number of cycles is incremented by one. These cycles continue to accumulate until the turbine reaches a state where it can no longer operate effectively, i.e., until the component fails.

The RUL is calculated based on the total number of remaining cycles until failure, which decreases as the system degrades. This remaining cycle count becomes our target for prediction, as it represents the time remaining until component failure.

4.3 Data Analysis

Exploratory Data Analysis concerns the initial understanding of a dataset, allowing us to explore its main characteristics and identify patterns, trends, and anomalies. In this chapter, we present a detailed analysis of the collected data, using a variety of visual and statistical techniques. Through data visualization and interpretation, we aim to reveal significant insights that can guide further analysis and inform decisions.

Statistical Summary

The statistical summary is a synthesis of the main descriptive measures of the numerical variables in a dataset. By analyzing the statistical summary presented in the table below, we can obtain a comprehensive understanding of the central characteristics, dispersion, and distribution of the numerical variables.

In this table, each row represents a variable of our study, being selected: the IDs, the cycles, target of our study, the altitude, the total temperature at the low-pressure turbine exit, the total pressure at the high-pressure compressor exit, and the physical core speed, and each column presents a specific statistical measure.

Upon examining the statistical summary, we observe the central characteristics and

Table 4.2: Statistical Summary

Variables	**Mean**	**%Std**	**Min**	**25%**	**50%**	**75%**	**Max**
ID	51.05	-	1.00	26.00	52.00	77.00	100
Cycle	108.80	-	1.00	52.00	104.00	156.00	362.00
Temperature turbine [°R]	1408.93	0.08	1382.25	1402.36	1408.04	1414.55	1441.49
Pressure at HPC [psia]	553.37	0.16	549.85	552.81	553.44	554.01	556.06
Core speed [RPM]	8443.75	0.24	8099.94	8133.24	8140.54	8148.31	8293.72

dispersion of the data in each variable. In the total temperature at the low-pressure turbine outlet, we observe that the mean is close to the median, suggesting an approximately symmetric distribution. Furthermore, the standard deviation is relatively low, indicating a relatively small dispersion of the data around the mean.

In the physical core speed, we observe a mean slightly higher than the median, suggesting a slightly right-skewed distribution. Additionally, its standard deviation is slightly higher than that of temperature, indicating a slightly greater dispersion of the data around the mean.

Finally, regarding the cycles, it is important to note that they are incremental values, suggesting a high concentration at low values and a large dispersion at higher values, impacting their standard deviation. For this analysis, we highlight the parameter that brings the maximum value of this variable, with 362 being the maximum cycles that the turbine is capable of enduring, for the studied turbines.

Histogram: Cycle Distribution

A histogram is a powerful visual tool for understanding the distribution of a numerical variable, in this case, cycles. By analyzing the graph presented in Figure 4.2, we can observe several important characteristics of the cycle distribution.

The horizontal axis of the histogram represents the possible values of cycles, while the vertical axis represents the frequency of occurrence of each value.

Generally, the histogram reveals that most failures occur after a large number of cycles, with the lowest frequency concentrated in the higher cycle ranges, while there are

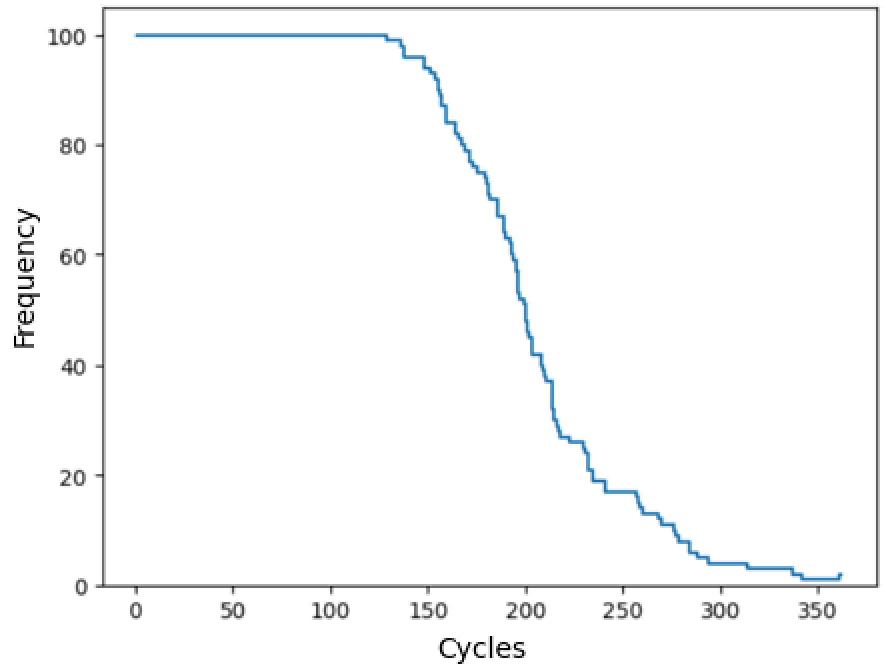

Figure 4.2: Histogram: Cycle Distribution

few failures in the initial cycles. This analysis shows that 100% of the equipment lasts at least about 140 cycles.

The cycle-to-failure distribution is not symmetrical and exhibits a slowly decaying shape, indicating that few failures occur at low cycle intervals. At the beginning of the graph (cycle 0), the frequency is high, indicating that there are many components that have not yet failed. As the number of cycles increases, the frequency of failures begins to rise gradually, declining as the cycles progress further. This is evidenced by the height of the histogram bars, which start high, remain stable for a while, and then decrease.

Most of the failure observations are concentrated around 150 to 200 cycles, with decreasing density as we move to higher cycles. We also observe that the line becomes almost vertical in this range, indicating a trend of machine failure within this interval, as validated in Table 4.2.

This characteristic of the distribution indicates that cycles until failure have a central tendency at higher values, with few cases of failures occurring in the initial cycles. The long tail on the right side of the histogram suggests the presence of some extremely high cycle values, which are less frequent.

The presence of this long tail may indicate outliers or rare events that influence the data distribution. With only about 20% of the equipment managing to exceed 200 cycles, these cases are exceptional. This can also be seen in Figure 4.3, where the total number of cycles for each tested turbine is presented. Identifying and investigating these outliers is important to better understand the cycles until failure and the factors contributing to the durability of the analyzed components.

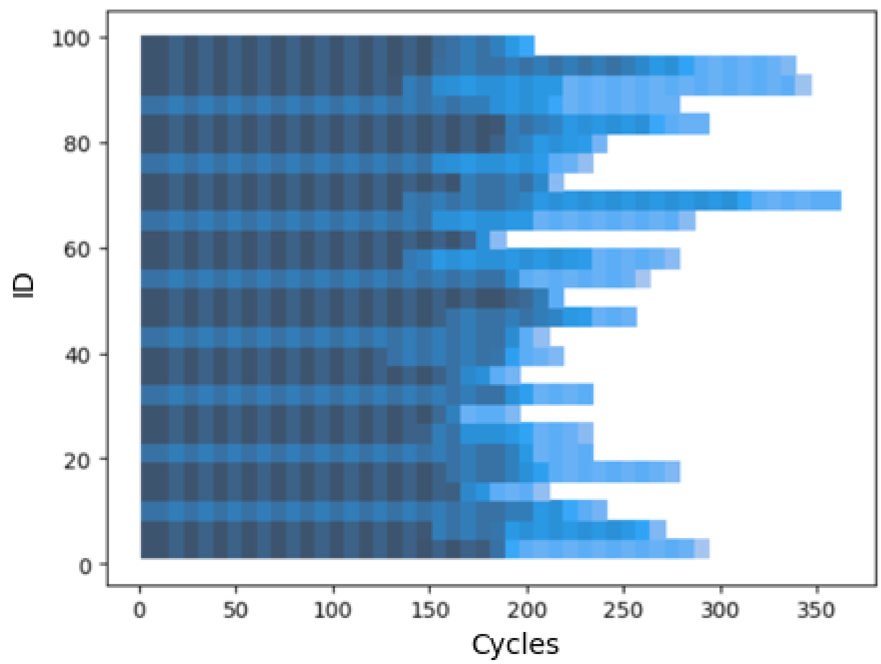

Figure 4.3: Bar Chart: Cycles x ID

Box Plot: Distribution of Corrected Core Speed

A box plot is an effective visual tool for exploring the distribution of a numerical variable, as in the case of Corrected Core Speed. By analyzing the box plot presented in Figure 4.4, we can extract various crucial information about this distribution.

The vertical axis of the box plot represents the possible values of Corrected Core Speed. The rectangle in the center of the box plot, called the "box," represents the interquartile range (IQR), which is the difference between the third and first quartiles (Q3 - Q1). The

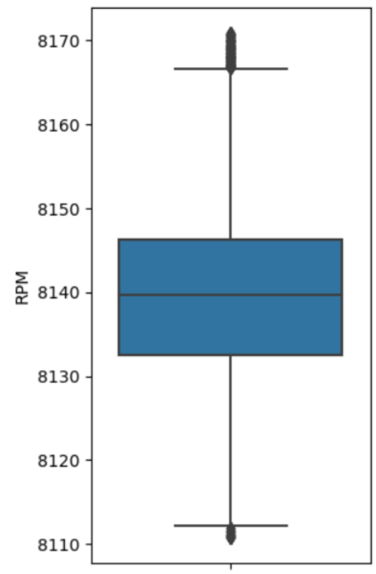

Figure 4.4: Box Plot: Distribution of Corrected Core Speed

line inside the box represents the median of the data.

By examining the box plot of Corrected Core Speed, we can identify if the distribution is symmetric and if there is the presence of outliers. If the box is centered within the data range and the whiskers, which extend from the box, are not too asymmetric, this suggests a more symmetric distribution.

Additionally, the presence of outliers, represented by points outside the whisker lines, may indicate extreme values that are influencing the distribution of the data.

In the specific case of the box plot of Corrected Core Speed, we notice the presence of some outliers above the upper limit of the whisker, which may indicate the existence of rare events or extreme values of Corrected Core Speed.

These outliers deserve further investigation, as they may provide valuable insights into possible issues or exceptional conditions affecting the distribution of Corrected Core Speed.

By analyzing the lower and upper limits of the box plot, we can conclude that none of the values fall below the minimum limit of 8100. This indicates that the minimum values of Corrected Core Speed are well-defined and may influence operation or maintenance strategies related to system performance.

Scatter Plot: Relationship between Compressor Temperature and RUL

The scatter plot is a fundamental visual tool for exploring the relationship between two numerical variables, in this case, total temperature at the low-pressure compressor outlet and RUL. Additionally, the number of machines was limited to facilitate interpretation. By analyzing the scatter plot presented in Figure 4.5, we can identify patterns and relationships between these variables.

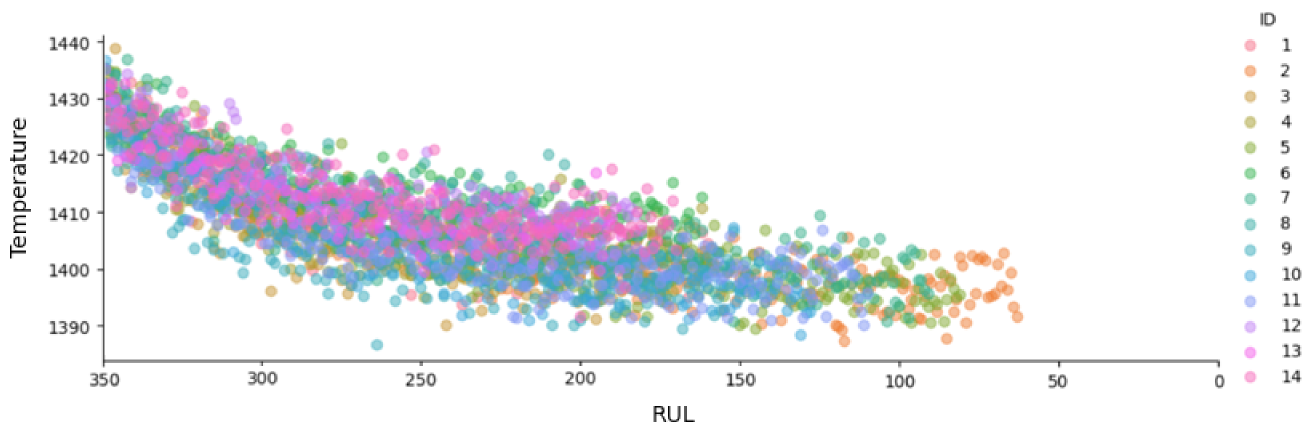

Figure 4.5: Scatter Plot: Relationship between Compressor Temperature and RUL

In the figure, each point represents an individual observation, where the horizontal axis (X) represents the RUL values and the vertical axis (Y) represents the temperature values, considering 14 different pieces of equipment. The arrangement of points in the plot reveals the relationship between the variables: the points tend to align within less than 50 cycles. Above 200 cycles, the points are scattered randomly, suggesting a lack of relationship between them.

Upon examining the scatter plot, we observe a linear relationship between the two variables. The linear relationship is evidenced by the points that roughly align in a descending straight line. This suggests that as the values of one variable increase, the values of the other variable also increase.

Additionally, the dispersion of the points indicates the variability of the data and the strength of the relationship between the variables: the closer the points are to failure, the more concentrated they are, indicating a stronger relationship between temperature and

RUL, indicating a good feature to be used as a prediction parameter.

On the other hand, at turbine's healthy levels, the data is more scattered, indicating a weaker relationship, which for our prediction may result in low accuracy.

Correlation Matrix

The correlation matrix is a statistical tool that allows us to examine the relationship between different numerical variables in a dataset. By analyzing the correlation matrix presented in Figure 4.6, we can identify association patterns between the variables and understand the strength and direction of these relationships.

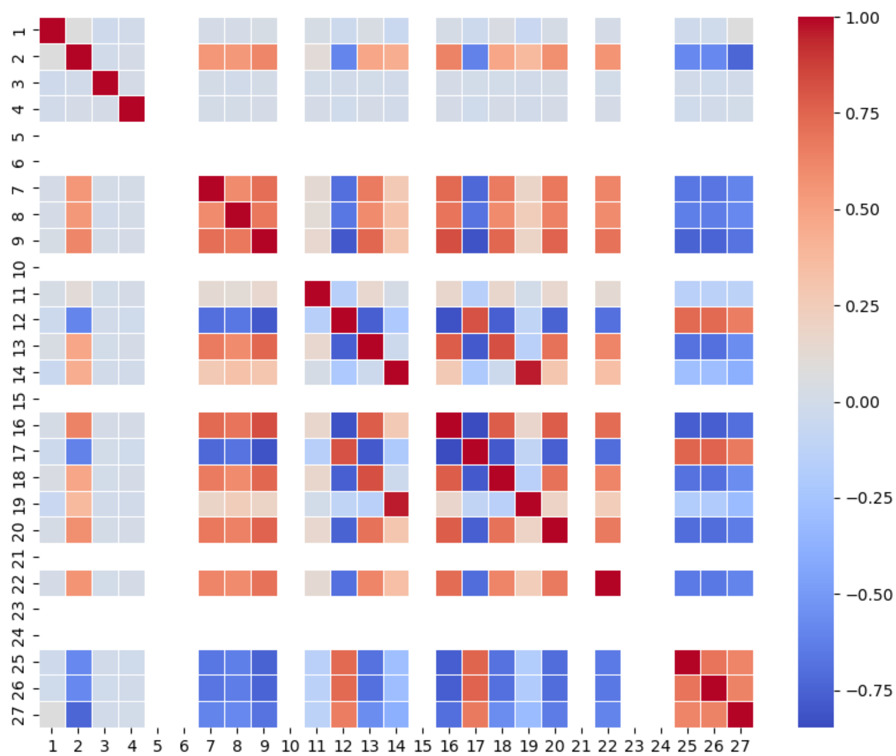

Figure 4.6: Correlation Matrix

In the correlation matrix, each cell represents the correlation coefficient between two variables. The correlation coefficient ranges from -1 to 1, where -1 indicates a perfect negative correlation, 0 indicates no correlation, and 1 indicates a perfect positive correlation. A positive correlation means that the variables tend to increase or decrease together, while a negative correlation indicates an inverse relationship, where one variable increases while the other decreases.

By examining the correlation matrix, we can identify pairs of variables with high positive or negative correlation, suggesting a strong relationship between these variables. This may indicate that the variables are related in some way and may influence each other in a predictive model or exploratory analysis.

Additionally, we can also observe variables with low or no correlation with each other, suggesting that they vary independently of each other. This is useful for identifying variables that can be considered independent in a statistical analysis or predictive model.

In the specific case of the correlation matrix presented, we can observe that the variables Total Pressure at HPC Outlet and Physical Fan Speed have a moderate positive correlation, indicating that they tend to increase or decrease together. On the other hand, the variables Total Temperature at Low Pressure Compressor Outlet and High Pressure have a weak negative correlation, suggesting an inverse relationship between them.

By analyzing the correlation matrix, we identify patterns of association between numerical variables and better understand the data structure. This allows us to select relevant variables for further analysis and predictive model construction.

4.4 Data Preprocessing

In this section, we address the data preprocessing that will serve to analize the data to feed our ML model. The goal is to prepare our data by removing irrelevant variables or those that may introduce unnecessary noise. This includes excluding variables with weak correlations, as well as those that do not vary across the entire dataset. This preparation is essential to ensure that our model is built on clean and relevant data, thus optimizing its efficiency and accuracy in the modeling phase.

The application of correlation analysis between variables, as presented in Figure 4.6, provides us with an understanding of the relationships existing in our dataset.

This analysis is particularly important in ML modeling contexts, where the quality of models significantly depends on the quality of input data. Strong correlations between variables can result in multicollinearity, which can impair the accuracy and interpretability of models. By identifying and understanding these correlations, we can take steps to mitigate their effects, such as selecting more relevant variables or applying dimensionality reduction techniques [63].

When interpreting the results of correlation analysis, it is crucial to consider not only the values of the coefficients but also the specific context of the problem under study. For example, a positive correlation between two variables may indicate a direct relationship. Similarly, a correlation close to zero does not necessarily imply the absence of a relationship but may suggest a nonlinear relationship or other types of association.

During the correlation analysis, we identified that some variables exhibited very high correlations with all other variables within the dataset. Consequently, we decided to exclude these variables that showed correlation coefficients close to 1 with respect to all other variables, as presented in Figure 4.6. The choice of this cutoff value was based on criteria of relevance and impact on the ML model.

By excluding variables with high correlations, we aim to simplify and optimize the model, reducing the dimensionality of the data without losing significant information. Variables with very high correlations tend to contribute little to the predictive capacity of the model and may even introduce unnecessary noise. Therefore, removing them can improve the computational efficiency and interpretability of the model without compromising its accuracy.

Furthermore, by establishing a specific cutoff value (0.8 in this case), we ensure consistency and transparency in the variable selection process. This helps to avoid arbitrary and subjective decisions, providing an objective and replicable approach to variable selection.

At the end of the process, all variables that did not reach the minimum correlation value were excluded from the dataset. This approach resulted in a more concise and relevant set of variables for the ML model, contributing to its effectiveness and interpretability.

Therefore, to ensure the efficiency and effectiveness of the model, all constant variables were identified and excluded from the dataset. In all of these processes, the removed variables were:

1. Throttle resolver angle
2. Total temperature at fan inlet [ºR]
3. Pressure at fan inlet [psia]
4. Engine pressure ratio (P50/P2)
5. Burner fuel-air ratio
6. Required fan speed [RPM]
7. Required corrected fan speed [RPM]

This data manipulation remained unchanged throughout the study. However, aiming to improve the obtained results, Recursive Feature Elimination (RFE) was employed in conjunction with Support Vector Machine (SVM). This approach provides a systematic methodology for selecting the most relevant features. One of the primary advantages of RFE lies in its ability to efficiently handle high-dimensional datasets, where manual identification of significant features would be unfeasible.

During its application, several aspects were considered to ensure robust and reliable results. For example, it was crucial to appropriately choose the machine learning algorithm to train the model in each iteration of the elimination process. We opted to use flexible and well-established algorithms, such as SVM, which are known for their ability to deal with different types of problems and datasets, as mentioned in subsection 3.3.2.

Additionally, it was necessary to determine the ideal number of features to be selected. This required a careful evaluation of the trade-off between model complexity and performance. Through iterative experimentation, it was possible to find a balance that resulted in an effective yet not overly complex model.

During the feature elimination process, it was also important to closely monitor the model's performance metrics. This included not only traditional evaluation metrics but also problem-specific metrics tailored to the data characteristics and analysis objectives, as discussed in subsection 3.3.4.

One of the most significant advantages of RFE was its ability to provide insights into the relative importance of features in the context of the problem under study. By identifying which features contributed most to the model's predictive capacity, we could gain a deeper understanding of the underlying data mechanisms and relationships between variables of interest.

Ultimately, the careful application of RFE resulted in a compact and informative set of features, which were used to build more accurate models. These selected features not only simplified result interpretation but also enhanced the model's generalization capability to new data, thus improving its usefulness and applicability in practical contexts.

Chapter 5

Results

In this section, we present the results of the evaluation of the models' behavior when subjected to the test set. This critical stage of analysis aims not only to validate the models' ability to generalize learned patterns during training but also to offer a clear insight into their effectiveness in real-world situations. Metrics presented in Section 3.3.4 will be carefully examined to provide a comprehensive assessment of the models' predictive power concerning the classes of interest.

Additionally, we will discuss the iterations carried out for the refinement of the models' hyperparameters. These iterations were essential to adjust the models in order to optimize their performance and generalization capability. Through careful experimentation and iterative analysis, it was possible to find hyperparameter configurations that resulted in significant improvements in the models' performance, thus ensuring a more precise and effective approach to data analysis.

By combining the evaluation of the models' behavior on the test set with the refinement of hyperparameters, this section offers complete results to validate the usefulness of the developed models and provide valuable insights for future research and practical applications.

5.1 Machine Learning Regression Models

Logistic Regression

After applying the Linear Regression model, we conducted a comprehensive analysis to evaluate the model's performance and the effects of the implemented changes. Initially, when subjecting the model to the test set, we recorded an R^2 value of approximately 36.4%. However, a deeper analysis of the residuals revealed non-random patterns, indicating that the model may not be fully capturing the underlying structure of the data. Given this observation, we decided to revise the model training process.

To improve the model's predictive capability, we chose to explore the Stochastic Gradient Descent (SGD) algorithm. SGD is known for its efficiency in optimizing linear regression models, especially on large datasets like ours. By updating the model parameters incrementally for each training sample, SGD allows for faster and more effective convergence.

Implementing SGD involved careful adjustments to hyperparameters, including the learning rate and batch size. These adjustments were crucial to ensure stable and rapid model convergence. Additionally, we regularly monitored the training progress and evaluated the model's performance on a validation set to avoid overfitting or underfitting.

To visualize these results, we generated Figure 5.1, which illustrates the relationship between actual and predicted values in a regression model. In the plot, each point represents a pair of actual and predicted values, while the black line represents the ideal relationship between these values (45-degree line). This allows us to visually assess the model's performance by observing how close the predicted values are to the actual values.

Throughout the iterations, we expected to observe a decrease in both training error and validation error. However, it is crucial to monitor any signs of overfitting, where the validation error starts to increase while the training error continues to decrease. This is an indicator that the model is fitting too closely to the training data but failing to generalize to new data.

At the end of the optimization process, we achieved an R^2 value of approximately

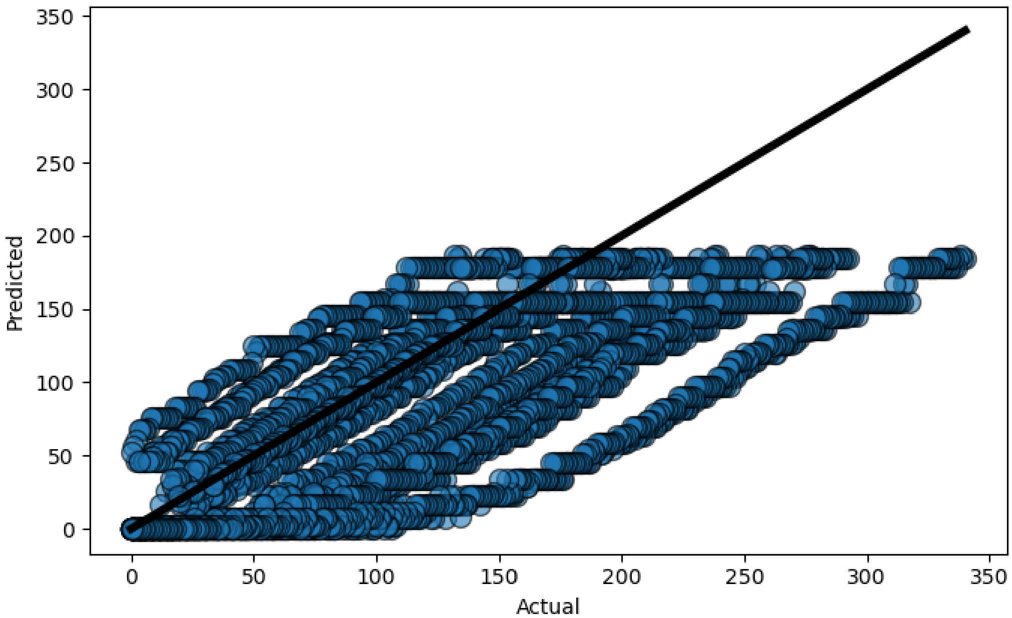

Figure 5.1: Evolution of R^2 over iterations for Random Forest Regressor

57.7%, and we used this result as a reference to compare with other models. This metric provides us with a measure of the quality of predictions from our Logistic Regression model concerning the observed data.

Random Forest Regressor

The results of applying the Random Forest Regressor algorithm reveal a remarkable performance concerning data prediction. During the iterations, a consistent behavior of the model was observed, reflected in stable R^2 values obtained on different training and test datasets. On average, the R^2 value found throughout the iterations was approximately 77.1%, indicating a good ability of the model to explain the variation in the data. The best R^2 value obtained during the iterations was 78.0%, demonstrating the consistent and promising performance of the Random Forest Regressor in data prediction.

In addition to the R^2 values, other aspects of the Random Forest Regressor results are noteworthy. The analysis of residuals revealed a relatively homogeneous distribution around zero, suggesting that the model effectively captured the variation in the data.

Evaluation of the relative importance of features also provided insights into which variables had a greater influence on the model predictions, enriching the understanding of underlying data patterns.

During the iterations, a variety of hyperparameter configurations were explored to optimize the model performance. These efforts resulted in gradual improvements in predictive performance, showcasing the flexibility and adaptability of the Random Forest Regressor to handle a variety of modeling scenarios.

To visualize the progression of the model performance throughout the iterations, Figure 5.2 presents a line graph showing the evolution of R^2 values as parameters were adjusted. This graph offers a clear visual representation of the influence of different configurations on the model outcomes.

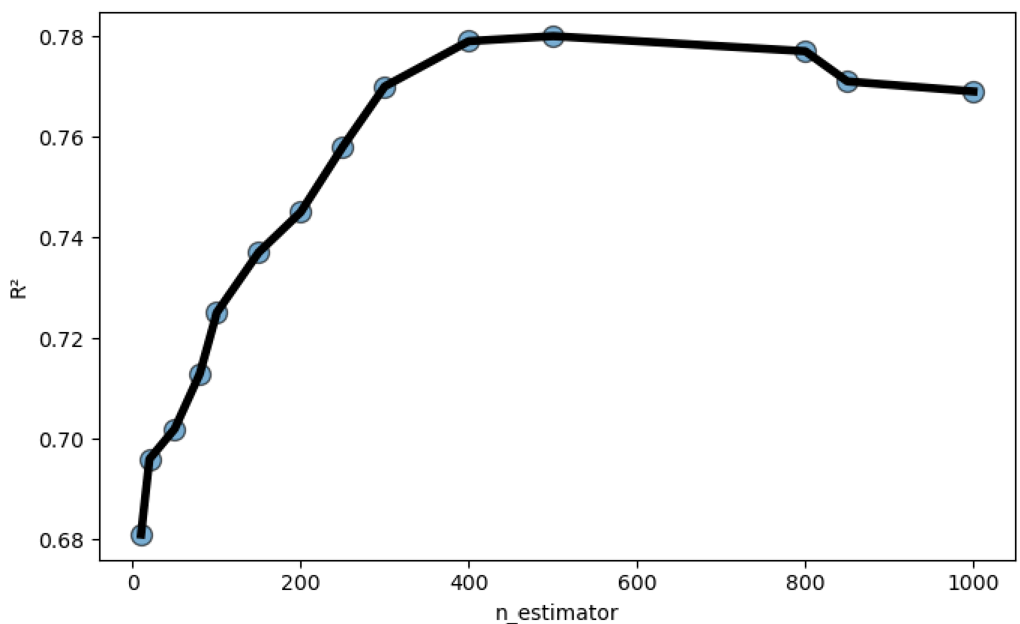

Figure 5.2: Evolution of R^2 over iterations for Random Forest Regressor

Another crucial aspect is the model's ability to generalize to new data. Tests on separate validation datasets confirmed that the model maintained consistent and robust performance even on previously unseen data, highlighting its generalization capability.

Support Vector Regression

For result analysis with the SVR algorithm, our challenge lay in modeling hyperparameter iterations that were computationally feasible within our computational power capacity. Given that several parameters need to be iteratively adjusted to converge to the best result, it was essential to find an approach that was efficient in terms of time and resources.

During the iterations, variables such as kernel types, regularization parameters (C), and kernel bandwidths (gamma) were adjusted to find the most suitable combinations that maximized the model's performance. Each iteration was conducted using cross-validation techniques, as described in Section 3.3.2, to ensure model robustness and generalization.

As a first iteration, we adjusted the regularization parameter C to control the trade-off between model complexity and fitting error. Decreasing the C value helped avoid overfitting, resulting in an improvement in residual uniformity, but there was still room for enhancements.

To address non-random patterns, we conducted a second iteration by adjusting the kernel and gamma parameter of the SVR. We explored different kernels such as linear, polynomial, and radial to find the best kernel function for our data. Additionally, we adjusted the bandwidth parameter to control the influence of each training point on the final model. These adjustments further reduced patterns in residuals and improved the model's generalization capability.

After implementing these iterations, we observed a significant improvement in the SVR performance. Below, we present the results of the most representative iterations:

Kernel Types	**Parameters C**	**Gamma**	**R^2**
RBF	10	0.1	0.494
Linear	1	N/A	0.456
Polynomial	100	N/A	0.483
RBF	1	0.01	0.531

Table 5.1: Results of iterations with different kernel types

These results illustrate the influence of SVR parameters on model performance. Iteration 4, for example, demonstrated the highest R^2, indicating that the combination of an RBF kernel with a C parameter of 1 and a gamma of 0.01 provided the best data prediction capability.

K-Nearest Neighbors

During the iterations conducted to analyze the performance of the K-Nearest Neighbors (KNN) algorithm for regression, various considerations were made to optimize its prediction effectiveness. One of the main areas of focus was determining the ideal number of neighbors (K) to consider during the prediction phase.

Starting with a broad range of K values, ranging from 1 to 20, we examined how adjusting this parameter affected the model's ability to capture patterns in the data and avoid both overfitting and underfitting. We observed that initially, with very low K values, the model tended to be overly sensitive to noise in the data, resulting in instability in predictions. On the other hand, with very high K values, the model lost important details of underlying patterns, leading to a decrease in prediction accuracy. This meticulous exploration of the K value allowed us to identify an ideal range that balanced the model's ability to capture complex patterns with its generalization capability.

During the iterative analysis process, a crucial component was visualizing the results obtained through graphs. One of the most informative graphs was the cross-validation plot for different K values, as shown in Figure 5.3. In this plot, the x-axis represented the tested K values, while the y-axis showed the R^2 metric.

Observing this plot allowed us to identify overfitting or underfitting trends as the K value varied. Overall, we observed an inverted "U" shaped curve, indicating that for moderate K values, the model's performance was optimized, while very low or very high values resulted in inferior performance.

Additionally, the analysis of the plot allowed us to select a K value that maximized the model's accuracy without compromising its generalization capability. This selected K value served as the basis for subsequent iterations, where other parameters were refined

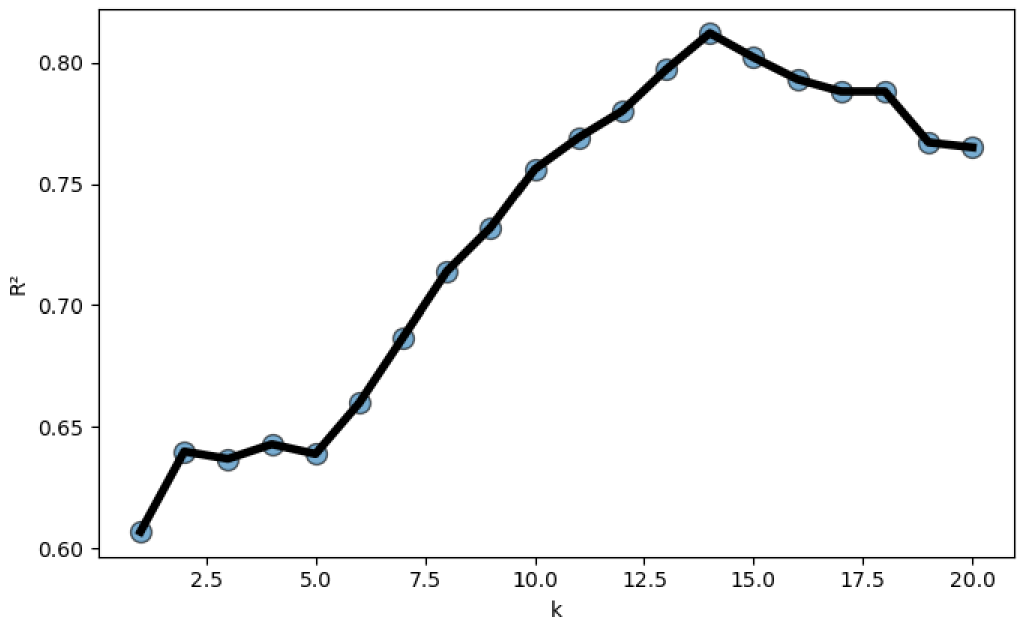

Figure 5.3: Evolution of R^2 over iterations for K-Nearest Neighbors

to further optimize the model performance.

Furthermore, we investigated the impact of different distance metrics on determining the proximity between data points. While Euclidean distance is the most commonly used metric in KNN, we also explored Manhattan distance to see if it could provide different or more accurate results, especially in datasets with nonlinear characteristics. We found that the choice of distance metric had a significant impact on the model predictions, especially in datasets with more complex structures.

Another crucial aspect we explored was the use of different weighting schemes for neighbors during the prediction phase. While the traditional KNN method treats all neighbors equally (uniform weights), we also explored weighting schemes where closer neighbors have a greater influence on predictions. This was particularly useful in situations where certain neighbors were more relevant for prediction than others, leading to an overall improvement in model accuracy.

At the end of this iterative process, we were able to identify an optimized configuration of the KNN model that demonstrated robust and accurate performance in the regression task for the given dataset. The R^2 obtained for this configuration was 81.2%, indicating a good ability of the model to explain the variation in the observed data.

Decision Tree

Upon submitting the decision tree for regression to the test set, the results revealed a series of metrics illustrating its performance in detail. Initially, we observed that the model exhibited an R^2 of 64.8% concerning the test data. This value indicates a good ability of the model to predict target values accurately, considering the specific context of the problem at hand.

However, upon closer examination of the error distribution, we noticed that some cases exhibited substantial discrepancies between predictions and actual values. This suggests that, while the overall R^2 is good, there may still be areas where the model is not generalizing properly. Investigating these discrepancies can provide valuable insights for further improving the predictive capability of the model.

Additionally, observing the learning curve, we noted a relatively rapid convergence for both the training and test sets, indicating that the model is efficiently learning from the available data. However, it's important to note that while the error on the test set has stabilized, there's still a small gap between the training and test set errors. This suggests a slight presence of overfitting, although not significant.

During the iterations for hyperparameter refinement, we mainly focused on adjusting the maximum depth of the tree and the minimum number of samples required to split a node, as depicted in Figure 5.4. By varying these hyperparameters and evaluating the model's performance using cross-validation.

For example, increasing the maximum depth of the tree resulted in a gradual decrease in the validation set's R^2. However, beyond a certain point, we noticed that the R^2 started to increase again, indicating that the model was becoming overly complex and prone to overfitting. Similarly, by reducing the minimum number of samples required to split a node, we observed an initial improvement in performance, but again with the risk of overfitting as this value became too low.

Therefore, through these iterations of fine-tuning the hyperparameters, we achieved

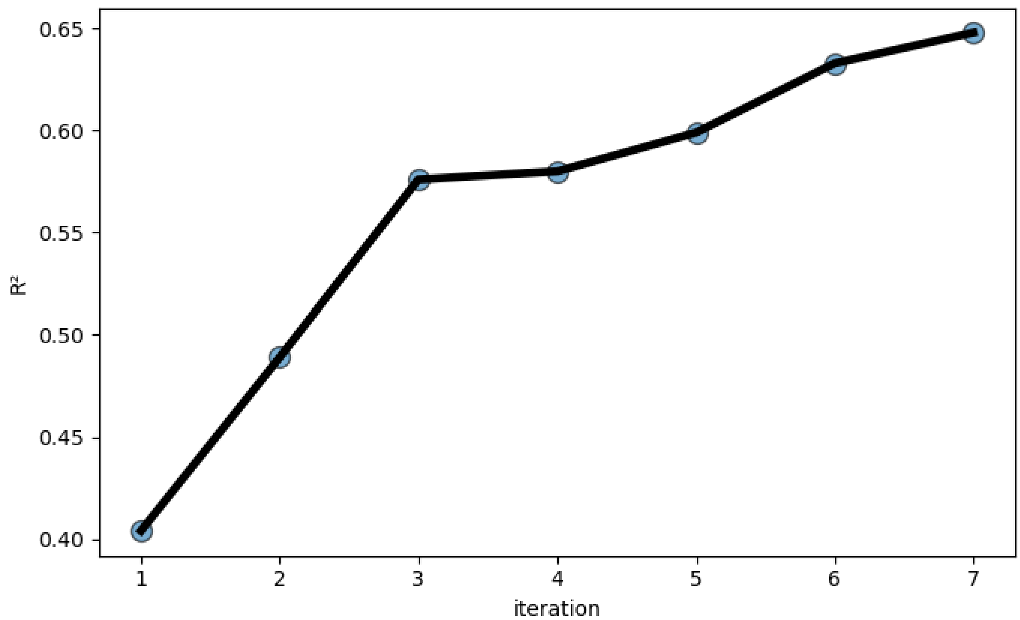

Figure 5.4: Evolution of R^2 over iterations for Decision Tree

a configuration that adequately balanced the model's complexity with its generalization capability, resulting in satisfactory performance concerning the test data.

Multi-Layer Perceptron

We also evaluated the performance of the MLP in relation to the test set. This model, due to its neural nature, offers a different approach than decision tree for regression, exploring the complexity of relationships between input features and the target variable.

Upon examining the results, we observed that the MLP achieved a slightly lower R^2 compared to the decision tree, recording a value of 62.4% on the test data. This suggests an equally good, if not better, ability to predict target values. However, it's worth noting that the MLP is more sensitive to the random initialization of weights and the network size, which can result in performance variations between different model runs.

Analyzing the learning curve of the MLP, we observed a smoother convergence pattern compared to the decision tree. The error on the training set consistently decreases as the model is trained for more epochs, while the error on the test set also decreases, albeit at a slower rate. This suggests that the model is gradually learning the structure of the data and generalizing well to unseen data.

Similar to what we did with the decision tree, we conducted iterations to adjust the MLP's hyperparameters to optimize its performance. We experimented with different network architectures, including the number of hidden layers, the number of neurons in each layer, and activation functions.

We observed that by increasing the network's complexity by adding more hidden layers or neurons, the model initially exhibited an increase in the validation set's R^2. However, beyond a certain point, we began to observe a decrease in R^2, indicating overfitting.

Furthermore, when testing different activation functions, we noted that some worked better for certain datasets than others. For example, the ReLU function proved to be effective in most cases, providing faster convergence during training.

Therefore, through these iterations of hyperparameter tuning, we were able to configure the MLP to balance the network's complexity with its generalization capability, resulting in an R^2 of 62.4

5.2 Machine Learning Classification Models

During the research, an insight that differs from the applications seen in Chapter 2 was the transformation of the target RUL, which represents the remaining life of the equipment, into suitable values to allow the application of classification models instead of regression. This transformation was necessary to simplify the problem and make it more suitable for the classification approach.

Additionally, this transformation serves as a decision-making tool. By converting the regression problem into a classification problem, it becomes easier to define specific categories for different maintenance actions, facilitating clearer and more objective decisions based on the equipment's classification. This approach simplifies analysis, aids in resource prioritization, and improves operational efficiency and equipment performance.

To perform this transformation, we adopted a categorization-based approach to categorize the equipment's state in relation to its remaining life. Instead of predicting a continuous value for the RUL, the goal now was to classify the equipment into one of three categories: "not ok", "degradation", and "ok". These categories represent different equipment states based on their condition and the amount of remaining life.

The first category, "not ok", refers to equipment that is in a critical state, where the remaining life is considered insufficient for safe or efficient operation. These equipments require immediate intervention or replacement.

The second category, "degradation", indicates that the equipment is in an intermediate state, where its condition is beginning to deteriorate but has not yet reached a critical state. In this state, preventive measures can be taken to avoid severe failures in the future.

The third category, "ok", represents equipment that is in good condition and has a significant amount of remaining life. These equipments are operating within normal parameters and do not require immediate intervention.

To assign the equipments to one of these categories, we initially divided the data into quartiles based on the distribution of the RUL. This provided us with a general idea of how the data was distributed in terms of remaining life. However, simply dividing the data into quartiles did not provide adequate separation between the "not ok", "degradation", and "ok" categories.

Therefore, to find the best data split values and define the boundaries between these categories, we employed a binary search iteration approach. In this approach, we started with the initial cutoff value to separate the data into two categories, such as "not ok" and "degradation + ok". We then evaluated how this separation aligned with our categorization criteria.

If the initial cutoff value did not result in a clear distinction between the categories or was inconsistent with our criteria, we adjusted the cutoff value using binary search. This iterative method allowed us to find optimal split values that best represented the boundaries between the categories.

Once these split values were determined, we could classify the equipments into one of the three categories based on the calculated RUL in relation to these boundaries. This allowed us to simplify the remaining life prediction problem into a classification approach, making it easier to implement suitable models to handle these categorized data.

Decision Tree

The results obtained with the Decision Tree algorithm for classification were analyzed through various iterations. We initiated the process with an implementation of the algorithm using default settings and conducted a preliminary evaluation. The initial results indicated an average accuracy of approximately 0.614 (Precision: 0.659, Recall: 0.639, F1-score: 0.655). Although these results were encouraging, we knew that improvements could be achieved through adjusting the model's parameters.

Therefore, we proceeded with a series of iterations, each focusing on a specific aspect of the decision tree model. The first iteration focused on varying the maximum depth of the decision tree. By testing depths from 1 to 10, we observed a gradual increase in accuracy until depth 6, at which point overfitting started to become a concern. Hence, we established the optimal tree depth at 6, where accuracy reached its peak of 0.643 (Precision: 0.668, Recall: 0.655, F1-score: 0.671).

The table below summarizes the results of the different iterations conducted during the adjustment of decision tree parameters. It can be observed that as we adjusted the maximum tree depth and the splitting criterion, there were gradual improvements in the model's accuracy. Remarkably, precision, recall, and F1-score were also enhanced as we refined these parameters. The final iteration, with a maximum depth of 6, 'entropy' split criterion, and a minimum of 4 samples to split a node, resulted in an accuracy of 0.693 and an F1-score of 0.684 indicating good model performance.

After these iterations of model refinement, we proceeded with the final training using the optimized configuration and evaluated the model's performance on a separate test set. The final results demonstrated an accuracy of 0.693 (Precision: 0.689, Recall: 0.678, F1-score: 0.684), confirming the effectiveness of the decision tree algorithm in classifying the studied data.

Maximum Depth	**Split Criterion**	**Minimum Samples**	**Accuracy**
6	Gini	2	0.614
6	Entropy	2	0.659
6	Entropy	3	0.688
6	Entropy	4	0.693

Table 5.2: Results of iterations with different decision tree configurations

Support Vector Machine

The Support Vector Machine (SVM) approach was also applied for the classification task, aiming to identify patterns in the data and predict the class of new instances. During the implementation of this method, a series of iterations was conducted, with systematic adjustments to the model's parameters to optimize classification performance. The key parameters considered during the iterations were the kernel type, the regularization parameter C, and the kernel coefficient gamma.

In the first iteration, the model was trained using a linear kernel and a C value of 1.0. The results showed an accuracy of approximately 0.604. In the second iteration, a polynomial kernel of degree 2 was employed, keeping the C value at 1.0. The resulting accuracy increased slightly to about 0.623. In the third iteration, the kernel was changed to an RBF (Radial Basis Function) kernel, with gamma set as 0.1 and C kept at 1.0. In this configuration, the accuracy reached approximately 0.650.

Given the promising results obtained with the RBF kernel, additional iterations were conducted to fine-tune the parameters. In the fourth iteration, the gamma value was increased to 0.5, while C was reduced to 0.5. This resulted in a slight decrease in accuracy, with a value close to 0.644. In the fifth and final iteration, the gamma value was kept at 0.5, while C was adjusted to 1.5. This configuration led to an increase in accuracy, reaching approximately 0.691.

In addition to accuracy results obtained in the SVM method iterations, the ROC curve was analyzed to evaluate the model's performance at different cutoff points. The analysis of the ROC curve revealed that the SVM model with an RBF kernel and parameters gamma equal to 0.5 and C set as 1.5 showed solid performance. The ROC curve showed a significantly high area under the curve (AUC), as presented in Figure 5.5, indicating the model's good ability to distinguish between classes. This confirms the effectiveness of the selected configuration for SVM parameters in the classification task.

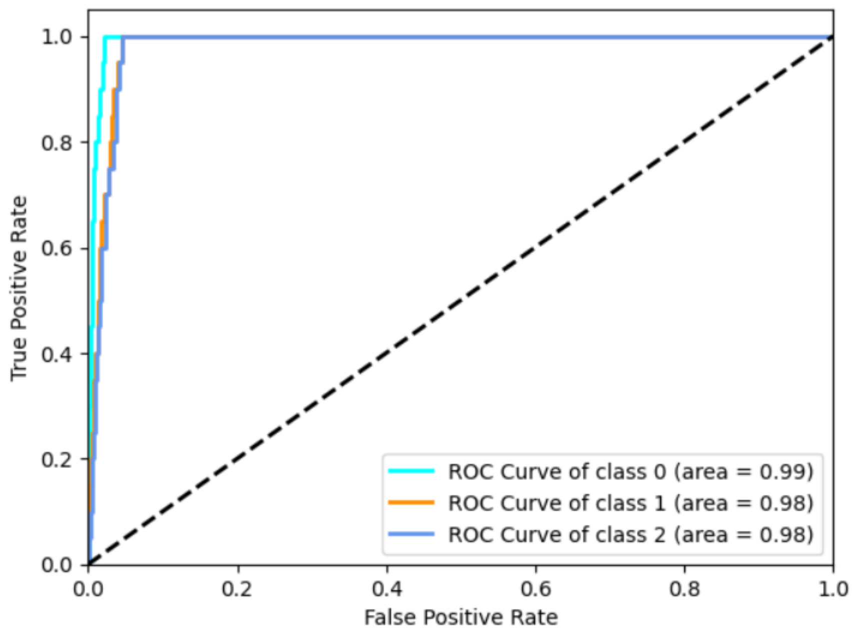

Figure 5.5: ROC Curve for SVM

In conclusion, experiments with SVM revealed that an RBF kernel with gamma equal to 0.5 and C set as 1.5 produced the best classification results, with an accuracy of about 0.691.

Naive Bayes

The analysis of results obtained using the Naive Bayes algorithm for classification revealed significant performance of the model. During the conducted iterations, various parameters were adjusted and tested to optimize prediction accuracy.

Initially, the experiment began with the direct application of the Naive Bayes regression model on a training dataset. In this first iteration, the resulting values showed an accuracy of 0.666, indicating satisfactory explanation of data variance by the model.

Next, a second iteration was conducted, incorporating the $k = 10$ cross-validation technique. This approach allowed for a more robust evaluation of the model on different subsets of the data. The analysis of results showed an average accuracy for the ten folds of 0.645, indicating reasonable consistency in the model's performance.

In the third iteration, the effect of Laplace smoothing on results was explored. The smoothing parameter was varied from $\alpha = 1$ to $\alpha = 0.5$, resulting in a marginal increase in model performance, with an average accuracy of 0.682 for $\alpha = 0.5$.

Subsequently, in the fourth iteration, different distributions for input variables were examined. Naive Bayes traditionally assumes simple probability distributions, such as the Gaussian distribution for continuous variables and the multinomial distribution for categorical variables. However, during the iterations, other distributions were tested, such as the exponential distribution, to determine which one best suited the data under analysis. The results indicated a slight superiority of the Gaussian distribution, with an average accuracy of 0.689, compared to 0.672 for the Multinomial distribution.

After several iterations and parameter adjustments, the final results demonstrated satisfactory performance of the Naive Bayes model in the classification task. The achieved accuracy was approximately 0.691, with a recall of 0.684 and a precision of 0.712. The confusion matrix revealed a balanced distribution of predicted classes, indicating that the model was capable of effectively discriminating between different categories.

K-Nearest Neighbors

The results obtained using the K-Nearest Neighbors (KNN) algorithm for classification were analyzed thoroughly. Initially, several iterations were performed, varying the value of K, which represents the number of nearest neighbors to be considered in classification. The results were documented and compared to determine the impact of different K values on model accuracy.

During the conducted iterations, various nuances were observed that contributed to a deeper understanding of the model's behavior. One of the most significant observations was the relationship between the value of K and the model's complexity.

By varying the value of K, it was possible to observe clearly the impact on the model's flexibility. Smaller values of K resulted in more complex and irregular decision boundaries, often leading to overfitting to the training data. This translated into models with high variance, i.e., sensitive to fluctuations in the training data, which could impair the model's performance when dealing with new data.

On the other hand, larger values of K resulted in smoother and more generalized decision boundaries. This tended to reduce the complexity of the model, making it less sensitive to variations in the training data. However, this reduction in complexity could lead to higher bias, where the model might underestimate the underlying complexity of the data.

The results revealed that the model's performance varied significantly with the change in the value of K. Overall, it was observed that smaller values of K tended to result in models with higher variance and lower generalization capability, while larger values of K led to models with higher bias. Specifically, the value of K = 7 emerged as the most suitable choice, producing a balance between accuracy and generalization capability.

Upon analyzing the performance metrics, it was observed that the KNN model achieved an average precision of approximately 0.712, a recall of 0.704, and an F1-score of 0.706. These metrics were consistent across iterations and suggest that the model is capable of effective classification based on nearest neighbors.

The choice of the ideal K value, therefore, represented a trade-off between these two extremes. During the iterations, the value of K = 7 emerged as a reasonable choice as it provided a balance between bias and variance. This value allowed the model to capture relevant patterns in the data without overfitting, resulting in a precision of 0.717.

5.3 Overall Compilation and Discussion of Results

The compilation of classification and regression results from all models is a step towards comprehensively evaluating the performance and effectiveness of each approach. In this section, we will present and discuss the obtained results, enabling a comparative analysis to identify relevant patterns or trends.

The results from the compilation of classification models presented in Table 5.3 reveal significant information about the performance of each algorithm. Starting with the Decision Tree, we observe an accuracy of 69.3%, reflecting the overall precision of the classifications, while precision and recall were consistent, both around 68.9% and 67.8%, respectively. However, the F1 score, which considers the harmony between precision and recall, stood at 68.4%, indicating a reasonable balance between these metrics.

Table 5.3: Model Results

Classification Models	**Accuracy**	**Precision**	**Recall**	**F1**
Decision Tree	0.693	0.689	0.678	0.684
Support Vector Machine	0.691	0.639	0.653	0.646
Naive Bayes	0.691	0.712	0.684	0.691
K-Nearest Neighbors	0.717	0.721	0.704	0.706

On the other hand, the Support Vector Machine (SVM) model showed similar accuracy, reaching 69.1%, but demonstrated slightly lower precision, achieving 63.9%, indicating a smaller proportion of correctly identified positive instances. Recall remained close to that of the Decision Tree, at 65.3%, and the F1 score was 64.6%, slightly below the first model.

In contrast, Naive Bayes exhibited higher precision, reaching 71.2%, suggesting superior ability in correctly identifying positive instances. Its accuracy remained consistent with the previous models, both around 69.1%. The F1 score remained equal to accurancy, at 69.1%, indicating a balanced harmony between precision and recall.

Finally, the K-Nearest Neighbors (KNN) model stood out with the highest accuracy, reaching 71.7%, and a precision of 72.1%, demonstrating superior ability in accurately identifying positive instances. And the F1-score indicated an effective balance between precision and recall, registering 70.6%.

Considering the results and discussions presented, the choice of which model to prioritize depends on the specific needs of the problem at hand. However, based on the provided metrics and each model's ability to meet different performance requirements, KNN seems to emerge as a promising choice.

KNN demonstrated the highest accuracy among the presented classification models, suggesting a relatively superior ability for correct classification. This can be crucial in situations where prediction accuracy is a priority.

In addition to high accuracy and good balance between precision and recall, KNN offers other advantages that distinguish it as the best model for many classification scenarios.

One of the main advantages of KNN is its conceptual simplicity and straightforward implementation. The algorithm is easy to understand and does not require assumptions about data distribution, making it an attractive choice for beginners and projects where model interpretability is valued.

Moreover, KNN is a non-parametric method, meaning it does not make explicit assumptions about the shape of the underlying data function. This allows the model to adapt well to different types of data and distributions, making it robust and flexible across a variety of scenarios.

Another advantage of KNN is its ability to handle complex and non-linearly separable datasets. Since the algorithm relies on proximity between data points, it can capture non-linear relationships between variables, which is especially useful in problems where relationships between features are not simple or linear.

Furthermore, KNN is a lazy learning model, meaning the training process is quick and simple as the model does not build an internal representation of the data during training. This makes KNN efficient in terms of training time and adaptation to constantly changing datasets.

Therefore, KNN not only demonstrated superior performance in terms of accuracy and balance between precision and recall but also offers additional advantages such as simplicity, robustness, and ability to handle complex data. These characteristics make KNN the ideal choice as the best model for many classification problems.

Regarding the regression analysis results, observing the Mean Squared Error (MSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2), we can evaluate each model's ability to predict the desired values. RandomForestRegressor stands out in this aspect, demonstrating consistent and superior performance compared to other models. Its low MSE and MAE indicate significant prediction capability, while a high R^2 suggests a good ability to explain variation in target values.

It is important to note that the choice of the ideal model depends not only on performance metrics but also on other factors such as model interpretability and computational cost. For example, although RandomForestRegressor demonstrated superior performance, models like KNN and Logistic Regression tend to be more interpretable and simpler to implement.

However, models like Support Vector Machine (SVM) showed inferior performance compared to other regression models, with high MSE and low R^2. This suggests that the model does not fit the data well and has a low prediction capability. Therefore, when prioritizing models for future applications, it is important to consider not only performance but also the efficiency and interpretability of the model.

The discrepancies in results between models can be attributed to various factors, including feature selection, choice of hyperparameters, and dataset peculiarities. A deeper analysis of these aspects can provide valuable insights to enhance models and guide future research in the field.

Considering only performance metrics, RandomForestRegressor emerges as the preferred choice for a regression model. Its proven ability to fit the data and explain variation in target values puts it ahead of other evaluated models. Therefore, for applications prioritizing accuracy and effectiveness in value prediction, RandomForestRegressor is the most recommended choice among the considered models.

When considering additional aspects, RandomForestRegressor emerges not only as the best in terms of performance metrics but also as the most robust and versatile model among those evaluated. Therefore, for an application requiring a reliable, accurate regression model capable of handling a variety of data challenges, RandomForestRegressor is arguably the most solid choice.

Among the two models, the choice of the classification approach has proven not only as a superior option in terms of accuracy and sensitivity to patterns in the data but also as a tool that provides practical and actionable guidance. By identifying specific classes or categories for the data, the KNN not only predicts outcomes but also indicates clear actions to be taken. These actions can range from executing maintenance procedures to performing scheduled stops or component replacements.

This ability to provide actionable insights not only adds immediate value to decision-making processes but also gives the model an additional dimension of practical utility. While the RandomForestRegressor may offer accurate predictions, the classification approach goes further, directly guiding the actions necessary to optimize the performance and efficiency of the systems under study.

For a deeper understanding of the final results achieved with the K-Nearest Neighbors (KNN) method, we generated a confusion matrix, presented in Figure 5.6. This analysis allows a clear view of the model's accuracy, identifying classification patterns, false positives, and false negatives.

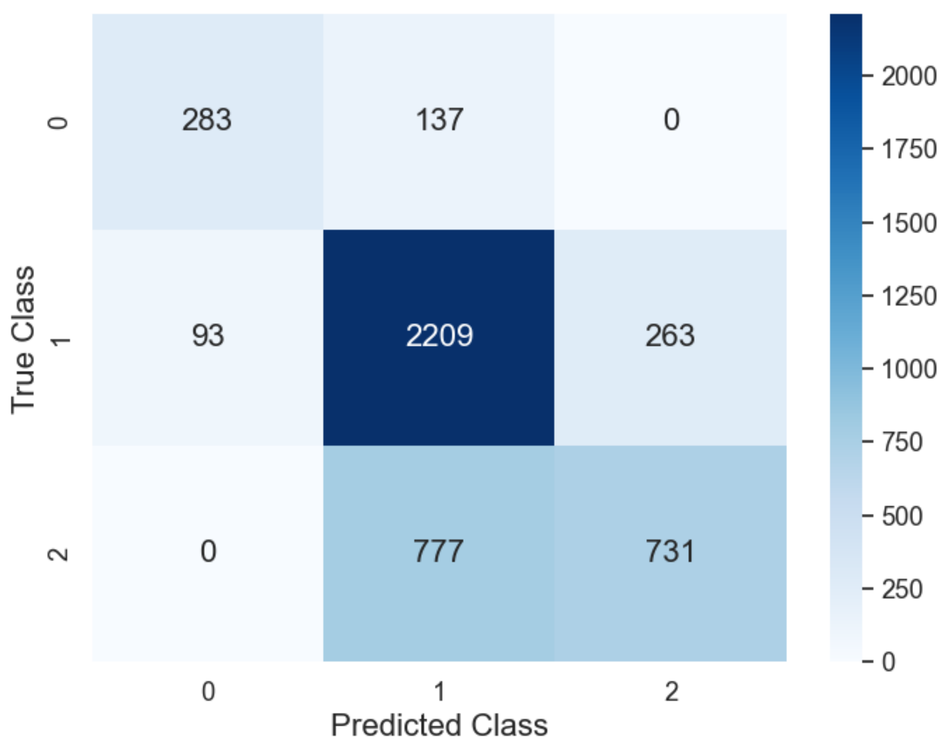

Figure 5.6: Confusion Matrix for K-Nearest Neighbors (KNN) Model

Upon examining the matrix, it is evident that the model presents different levels of accuracy for each class. This analysis highlights the importance of class 0, which represents instances of near failures, in the context of the classification problem. The KNN model demonstrates truly satisfactory performance, especially in identifying instances truly belonging to class 0. Of the cases in this category, the model correctly identified them mostly, indicating its ability to accurately recognize near failures.

However, the analysis also reveals an area for improvement related to the 137 instances that were erroneously classified as class 1 by the model, although they were actually class 0. These cases represent false positives, where the model incorrectly predicted the absence of near failures. While the model has an overall satisfactory performance, these false positives can have significant implications in practical scenarios, especially in this study, where precise identification of near failures is critical for decision-making.

An approach to improving the model would be to further investigate the characteristics of these 137 instances. Understanding what sets them apart from instances correctly classified as class 0 can provide insights to adjust the model and reduce the number of false positives, as detailed in chapter 6. Additionally, considering that the model has already demonstrated a generally good ability to identify near failures, resources can be directed towards specific refinement in this area.

Based on the results obtained, there is a clear indication that the KNN model can be a viable tool for practical application in identifying near failures. Its demonstrated ability to recognize patterns associated with this specific type of event, as evidenced by the majority of class 0 instances being correctly classified, is promising.

Chapter 6

Final Considerations

6.1 Conclusions

As we move into an increasingly globalized era, the demand for innovation and efficiency in the aerospace industry becomes more pressing than ever. Amidst this challenging landscape, the application of artificial intelligence emerges as a crucial tool for addressing the challenges of modernization and safety in this vital sector. This study aimed to precisely explore this frontier, investigating how artificial intelligence can be employed in predictive maintenance, especially in assessing the wear of critical aviation components.

The results thus far point not only to the feasibility but also to the effectiveness of this innovative approach. Although still a relatively recent area of research, our findings clearly demonstrate that, with diligent and careful management of available data, valuable information can be extracted that can be directly applied in practice. The combination of predictive maintenance techniques and artificial intelligence models not only allows for anticipating when equipment needs repair or replacement but also enables strategic and proactive planning, reducing costs associated with unscheduled maintenance and, most crucially, enhancing passenger safety.

However, it is important to recognize that this study is just an initial step in an ongoing journey of discovery and improvement. The complexity and dynamics of the aerospace industry demand a continuous commitment to research and development, as well as a collaborative approach among academics, industry professionals, and regulators. As we progress, it is essential to continue exploring new ways to integrate artificial intelligence and other emerging technologies to address challenges and seize opportunities.

Ultimately, this study not only contributes to the growing body of knowledge in the field of predictive maintenance in aviation but also sheds light on the transformative potential of artificial intelligence in critical sectors of the global economy. By adopting an innovative and collaborative approach, we can not only address current challenges but also shape a safer, more efficient, and sustainable future for the aerospace industry and, by extension, for society as a whole.

6.2 Future Work

Despite the significant advancements achieved in this study, there are several promising directions for future research that can further enrich our understanding and application of artificial intelligence in predictive maintenance in the aerospace industry.

One area of potential interest is the investigation of advanced model enhancement techniques. For instance, exploring different feature selection methods may help identify which variables are most influential in predicting the wear of gas turbine components. Additionally, the application of more sophisticated techniques for converting data into categorical formats may offer additional insights into underlying patterns and trends.

A promising idea with the potential to yield highly positive results is to conduct a group analysis of categorical data. This would involve examining the model's accuracy when it responds multiple times consecutively, particularly considering the cyclical nature of the data. Such an approach would allow for a deeper understanding of the model's consistency and reliability over time, revealing patterns of variation in the data dynamics across different cycles. This in-depth analysis would not only enhance the model's accuracy but also provide a more comprehensive insight into the underlying trends of the studied phenomena.

Another research line is expanding the scope of this study to include real maintenance data. By applying the model developed in this study to maintenance datasets obtained directly from aircraft operators, we can validate its effectiveness in real-world scenarios. This would not only provide a more comprehensive assessment of the practical applicability of the model but also allow for further refinements based on insights derived from real data analysis.

Furthermore, considering the rapid evolution of technologies and practices in the aerospace industry, it is essential to stay updated with the latest developments. Investigating how emerging trends, such as the adoption of new materials or the introduction of new propulsion systems, may affect the effectiveness of artificial intelligence-based predictive maintenance techniques is crucial to ensuring that our work remains relevant and applicable in the long term.

By continuing to explore these and other research areas, we can further strengthen the existing knowledge base and drive the practical application of artificial intelligence in the aerospace industry. In doing so, we not only contribute to safer and more efficient aircraft operation but also open up new possibilities for innovation and progress in this vital sector for the global economy.

Bibliography

[1] S. Abdelghafar, A. Khater, A. Wagdy, A. Darwish, and A. E. Hassanien, "Aero engines remaining useful life prediction based on enhanced adaptive guided differential evolution," *Evolutionary Intelligence*, vol. 1, pp. 1–12, Dec. 2022, ISSN: 18645917. DOI: 10.1007/S12065-022-00805-Z/FIGURES/6. [Online]. Available: https://link.springer.com/article/10.1007/s12065-022-00805-z.

[2] K. Abid, M. Sayed-Mouchaweh, and L. Cornez, "Adaptive data-driven approach for the remaining useful life estimation when few historical degradation sequences are available," *Proceedings - 19th IEEE International Conference on Machine Learning and Applications, ICMLA 2020*, pp. 1145–1152, Dec. 2020. DOI: 10.1109/ICMLA51294.2020.00184.

[3] Associação Brasileira de Normas Técnicas, *NBR 5462, Confiabilidade e mantenabilidade*, Associação Brasileira de Normas Técnicas, Rio de Janeiro, Nov. 1994, p. 37.

[4] C. Bienefeld, E. Kirchner, A. Vogt, and M. Kacmar, "On the importance of temporal information for remaining useful life prediction of rolling bearings using a random forest regressor," *Lubricants 2022, Vol. 10, Page 67*, vol. 10, p. 67, 4 Apr. 2022, ISSN: 2075-4442. DOI: 10.3390/LUBRICANTS10040067. [Online]. Available: https://www.mdpi.com/2075-4442/10/4/67/htm%20https://www.mdpi.com/2075-4442/10/4/67.

[5] D. J. A. Borlido, "Indústria 4.0: Aplicação a sistemas de manutenção," 2017. [Online]. Available: https://hdl.handle.net/10216/102740.

[6] A. Botchkarev, “Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology,” *arXiv preprint arXiv:1809.03006*, 2018.

[7] A. Bousdekis, B. Magoutas, D. Apostolou, and G. Mentzas, “Review, analysis and synthesis of prognostic-based decision support methods for condition based maintenance,” *Journal of Intelligent Manufacturing*, vol. 29, no. 6, pp. 1303–1316, Dec. 2015. DOI: 10.1007/s10845-015-1179-5. [Online]. Available: https://doi.org/10.1007/s10845-015-1179-5.

[8] G. Branco Filho, *A Organização, o planejamento e o controle da manutenção*. Ciência Moderna, 2008.

[9] Brasil, *Código brasileiro de aeronáutica, Lei nº 7.565, de 19 de dezembro de 1986*, Agência Nacional de Aviação Civil, Brasil, 1986.

[10] Brasil, *Regulamento brasileiro da aviação civil rbac nº 145 emenda nº 01 - anac, Organizações de manutenção de produto aeronáutico*, Agência Nacional de Aviação Civil, Brasília, DF, 2014.

[11] E. Brynjolfsson and T. Mitchell, “What can machine learning do? workforce implications,” *Science*, vol. 358, no. 6370, pp. 1530–1534, Dec. 2017. DOI: 10.1126/science.aap8062. [Online]. Available: https://doi.org/10.1126/science.aap8062.

[12] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. da P. Francisco, J. P. Basto, and S. G. S. Alcalá, “A systematic literature review of machine learning methods applied to predictive maintenance,” *Computers Industrial Engineering*, vol. 137, p. 106 024, 2019, ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2019.106024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835219304838.

[13] H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: From big data to big impact," *MIS quarterly*, pp. 1165–1188, 2012. [Online]. Available: https://doi.org/10.2307/41703503.

[14] A. Cheung, W. Ip, and D. Lu, "Expert system for aircraft maintenance services industry," *Journal of Quality in Maintenance Engineering*, vol. 11, no. 4, pp. 348–358, Dec. 2005. DOI: 10.1108/13552510510626972. [Online]. Available: https://doi.org/10.1108/13552510510626972.

[15] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, e623, 2021.

[16] D.-I. Choi and C.-S. Ok, "A new metric for evaluation of forecasting methods: Weighted absolute and cumulative forecast error," *Journal of Society of Korea Industrial and Systems Engineering Vol*, vol. 38, no. 3, pp. 159–168, 2015.

[17] B. J. Copeland and D. Proudfoot, "Artificial intelligence: History, foundations, and philosophical issues," in *Philosophy of Psychology and Cognitive Science*, ser. Handbook of the Philosophy of Science, P. Thagard, Ed., Amsterdam: North-Holland, 2007, pp. 429–482. DOI: https://doi.org/10.1016/B978-044451540-7/50032-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444515407500323.

[18] P. Dangeti, *Statistics for machine learning*. Packt Publishing Ltd, 2017.

[19] I. Daniyan, K. Mpofu, R. Muvunzi, and I. D. Uchegbu, "Implementation of artificial intelligence for maintenance operation in the rail industry," *Procedia CIRP*, vol. 109, pp. 449–453, Jan. 2022, ISSN: 2212-8271. DOI: 10.1016/J.PROCIR.2022.05.277.

[20] D. Dhital, *Artificial intelligence*, Jun. 2020. [Online]. Available: https://aviationoutlook.com/artificial-intelligence/.

[21] T. Dietterich, “Overfitting and undercomputing in machine learning,” *ACM computing surveys (CSUR)*, vol. 27, no. 3, pp. 326–327, 1995. [Online]. Available: `https://pastel.hal.science/tel-04307172`.

[22] A. Edell, *Understand these 4 advanced concepts to sound like a machine learning master*, Sep. 2020. [Online]. Available: `https://towardsdatascience.com/understand-these-4-advanced-concepts-to-sound-like-a-machine-learning-master-d32843840b52`.

[23] B. J. Erickson and F. Kitamura, *Magician's corner: 9. performance metrics for machine learning models*, 2021. [Online]. Available: `https://doi.org/10.1148/ryai.2021200126`.

[24] T. Fushiki, “Estimation of prediction error by using k-fold cross-validation,” *Statistics and Computing*, vol. 21, pp. 137–146, 2011. [Online]. Available: `https://doi.org/10.1007/s11222-009-9153-8`.

[25] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2019.

[26] A. Gilchrist, *Industry 4.0: the industrial internet of things*. Springer, 2016.

[27] W. Goeking, “Da máquina a vapor aos softwares de automação,” *O setor elétrico, São Paulo*, vol. 52, pp. 70–77, 2010.

[28] G. F. GREGÓRIO and A. M. SILVEIRA, “Manutenção industrial,” *Porto Alegre: SAGAH*, 2018.

[29] Z. Kang, C. Catal, and B. Tekinerdogan, “Remaining useful life (rul) prediction of equipment in production lines using artificial neural networks,” *Sensors 2021, Vol. 21, Page 932*, vol. 21, p. 932, 3 Jan. 2021, ISSN: 1424-8220. DOI: `10.3390/S21030932`. [Online]. Available: `https://www.mdpi.com/1424-8220/21/3/932/htm%20https://www.mdpi.com/1424-8220/21/3/932`.

[30] A. Kardec and J. Nascif, *Manutenção-função estratégica*. Qualitymark Editora Ltda, 2012, ISBN: 8541403629.

[31] T. Khan, K. Ahmad, J. Khan, I. Khan, and N. Ahmad, "An explainable regression framework for predicting remaining useful life of machines," *2022 27th International Conference on Automation and Computing: Smart Systems and Manufacturing, ICAC 2022*, 2022. DOI: 10.1109/ICAC55051.2022.9911162.

[32] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004, ISSN: 1353-7776.

[33] E. Levrat, E. Thomas, and B. Iung, "Odds-based decision-making tool for opportunistic production-maintenance synchronization," *International Journal of Production Research*, vol. 46, no. 19, pp. 5263–5287, 2008. DOI: 10.1080/00207540802273793. eprint: https://doi.org/10.1080/00207540802273793. [Online]. Available: https://doi.org/10.1080/00207540802273793.

[34] X. Lin, F. Yang, L. Zhou, *et al.*, "A support vector machine-recursive feature elimination feature selection method based on artificial contrast variables and mutual information," *Journal of chromatography B*, vol. 910, pp. 149–155, 2012.

[35] Lito, *Montando 367.000 peças*, Apr. 2009. [Online]. Available: https://avioesemusicas.com/montando-367000-pecas.html.

[36] G. F. Luger, *Artificial intelligence: structures and strategies for complex problem solving*. Pearson education, 2005.

[37] S. Madakam, V. Lake, V. Lake, V. Lake, *et al.*, "Internet of things (iot): A literature review," *Journal of Computer and Communications*, vol. 3, no. 05, p. 164, 2015. [Online]. Available: http://dx.doi.org/10.4236/jcc.2015.35021.

[38] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).*, vol. 9, no. 1, pp. 381–386, 2020. [Online]. Available: http://dx.doi.org/10.21275/ART20203995.

[39] D. Martinez, *Aviation revolution nears with artificial intelligence and deep learning*, Mar. 2018. [Online]. Available: https://datascience.aero/aviation-revolution-ai-deep-learning/.

[40] V. Mathew, T. Toby, V. Singh, B. M. Rao, and M. G. Kumar, "Prediction of remaining useful lifetime (rul) of turbofan engine using machine learning," *IEEE International Conference on Circuits and Systems, ICCS 2017*, vol. 2018-January, pp. 306–311, Mar. 2018. DOI: 10.1109/ICCS1.2017.8326010.

[41] A. Mora Gutiérrez, "Mantenimiento estratégico para empresas industriales o de servicios. enfoque sistemático kantiano," *AMG. Medellín, Colombia*, 2009.

[42] J. Moubray, *Reliability-centered maintenance*. Industrial Press Inc., 2001.

[43] A. C. Müller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.", 2016.

[44] NASA, "Nasa reliability-centered maintenance guide," *National Aeronautics and Space Administration*, p. 472, 2008, ISSN: 0831131462. [Online]. Available: https://www.wbdg.org/ffc/nasa/guides-handbooks/rcm-guide.

[45] N. Nikolakis, A. Papavasileiou, K. Dimoulas, K. Bourmpouchakis, and S. Makris, "On a versatile scheduling concept of maintenance activities for increased availability of production resources," *Procedia CIRP*, vol. 78, pp. 172–177, Jan. 2018. DOI: 10.1016/j.procir.2018.09.065.

[46] Norma Potuguesa, *NP EN 13306:2007, Terminologia da manutenção*, Norma Portuguesa, Lisboa, 2007.

[47] M. J. Page, J. E. McKenzie, P. M. Bossuyt, *et al.*, "The prisma 2020 statement: An updated guideline for reporting systematic reviews," *BMJ*, vol. 372, Mar. 2021, ISSN: 1756-1833. DOI: 10.1136/BMJ.N71. [Online]. Available: https://www.bmj.com/content/372/bmj.n71%20https://www.bmj.com/content/372/bmj.n71.abstract.

[48] PricewaterhouseCoopers, *Transportation amp; logistics*, 2016. [Online]. Available: https://www.pwc.com/us/en/industries/consumer-markets/transportation-logistics/airlines-airports.html.

[49] W. Qian, Y. Xiong, W. Ding, J. Huang, and C.-M. Vong, "Label correlations-based multi-label feature selection with label enhancement," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107 310, 2024, ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2023.107310. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095219762301494X.

[50] G. Rigotti, *Conheça as tecnologias habilitadoras da indústria 4.0 e algumas aplicações, associação brasileira de internet industrial*, Associação Brasileira de Internet Industrial, Nov. 2021. [Online]. Available: https://abii.com.br/com-abii-single-post-conheca-as-tecnologias-habilitadoras-da-industria-4-0-e-algumas-aplicacoes/.

[51] D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin, "Backpropagation: The basic theory," in *Backpropagation*, Psychology Press, 2013, pp. 1–34.

[52] A. R. C. Sanches, "Metodologia lean aplicada à manutenção aeronáutica," 2018.

[53] E. F. G. d. Santos, "Manutenção aeronáutica preditiva: Procedimentos, técnicas e business models," 2019.

[54] E. Saranya and P. B. Sivakumar, "Data-driven prognostics for run-to-failure data employing machine learning models," *Proceedings of the 5th International Conference on Inventive Computation Technologies, ICICT 2020*, pp. 528–533, Feb. 2020. DOI: 10.1109/ICICT48043.2020.9112411.

[55] A. Saxena and K. Goebel, "Turbofan engine degradation simulation data set," *NASA Ames Prognostics Data Repository*, pp. 1551–3203, 2008.

[56] S. Schwendemann, Z. Amjad, and A. Sikora, "A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding

machines," *Computers in Industry*, vol. 125, p. 103 380, Feb. 2021, ISSN: 0166-3615. DOI: 10.1016/J.COMPIND.2020.103380.

[57] J. Shaw and B. Wu, "Prediction of remaining useful life of wind turbine shaft bearings using machine learning," *Journal of Marine Science and Technology*, vol. 29, pp. 631–637, 5 Nov. 2021, ISSN: 2709-6998. DOI: 10.51400/2709-6998.2465. [Online]. Available: https://jmstt.ntou.edu.tw/journal/vol29/iss5/4.

[58] K. Siau, Y. Xi, and C. Zou, "Industry 4.0: Challenges and opportunities in different countries," *Cut. IT J*, vol. 2, pp. 23–34, 2019.

[59] D. G. d. Silva, *Aspectos positivos do uso da inteligncia artificial/machine learning na gestão e planejamento da manutenção aeronáutica.* Especializacao em Engenharia de Manutencao Aeronautica, Pontifica Universidade Catolica de Minas Gerais, Minas Gerais, BRA, Jan. 2021, p. 9.

[60] E. L. da Silva and M. V. Bahia, *Compostos orgânicos: Fórmulas estruturais e principais classes*, Site UOL Educação, Universidade Federal da Bahia, Feb. 2009. [Online]. Available: http://educacao.uol.com.br/disciplinas/quimica/compostos-organicos-formulas-estruturais-e-principais-classes.htm.

[61] N. Slack, S. Chambers, R. Johnston, *et al.*, *Administração da produção.* Atlas São Paulo, 2009, vol. 2.

[62] R. D. Sree, S. Jayanthy, and E. E. Vigneshwaran, "Estimation of remaining useful life(rul) of bldc motor using machine learning approaches," *7th International Conference on Communication and Electronics Systems, ICCES 2022 - Proceedings*, pp. 286–291, 2022. DOI: 10.1109/ICCES54183.2022.9835744.

[63] M. Tripathi, *Underfitting and overfitting in machine learning*, Jun. 2020. [Online]. Available: https://datascience.foundation/sciencewhitepaper/underfitting-and-overfitting-in-machine-learning.

[64] C. Ucler and O. Gok, "Innovating general aviation MRO's through IT: The sky aircraft management system - SAMS," *Procedia - Social and Behavioral Sciences*, vol. 195, pp. 1503–1513, Jul. 2015. DOI: 10.1016/j.sbspro.2015.06.452. [Online]. Available: https://doi.org/10.1016/j.sbspro.2015.06.452.

[65] D. A. Wiegmann and S. A. Shappell, "Applying the human factors analysis and classification system (hfacs) to the analysis of commercial aviation accident data," 2001.

[66] C. Wohlin, E. Mendes, K. R. Felizardo, and M. Kalinowski, "Guidelines for the search strategy to update systematic literature reviews in software engineering," *Information and Software Technology*, vol. 127, p. 106 366, 2020, ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2020.106366. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095058491930223X.

[67] H. Yu and X. Li, "On the chaos analysis and prediction of aircraft accidents based on multi-timescales," *Physica A: Statistical Mechanics and its Applications*, vol. 534, p. 120 828, Nov. 2019. DOI: 10.1016/j.physa.2019.04.064. [Online]. Available: https://doi.org/10.1016/j.physa.2019.04.064.

[68] O. E. Yurek and D. Birant, "Remaining useful life estimation for predictive maintenance using feature engineering," *Proceedings - 2019 Innovations in Intelligent Systems and Applications Conference, ASYU 2019*, Oct. 2019. DOI: 10.1109/ASYU48272.2019.8946397.

[69] X. Zhang and C.-A. Liu, "Model averaging prediction by k-fold cross-validation," *Journal of Econometrics*, vol. 235, no. 1, pp. 280–301, 2023, ISSN: 0304-4076. DOI: https://doi.org/10.1016/j.jeconom.2022.04.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304407622000975.

[70] T. Zonta, C. A. da Costa, F. A. Zeiser, G. de Oliveira Ramos, R. Kunst, and R. da Rosa Righi, "A predictive maintenance model for optimizing production schedule using deep neural networks," *Journal of Manufacturing Systems*, vol. 62, pp. 450–

462, Jan. 2022. DOI: 10.1016/j.jmsy.2021.12.013. [Online]. Available: https://doi.org/10.1016/j.jmsy.2021.12.013.