


12-2023

EARLY-WARNING PREDICTION FOR MACHINE FAILURES IN AUTOMATED INDUSTRIES USING ADVANCED MACHINE LEARNING TECHNIQUES

Satnam Singh

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd>

 Part of the [Business Analytics Commons](#), [Business Intelligence Commons](#), [Other Computer Engineering Commons](#), and the [Technology and Innovation Commons](#)

Recommended Citation

Singh, Satnam, "EARLY-WARNING PREDICTION FOR MACHINE FAILURES IN AUTOMATED INDUSTRIES USING ADVANCED MACHINE LEARNING TECHNIQUES" (2023). *Electronic Theses, Projects, and Dissertations*. 1812.
<https://scholarworks.lib.csusb.edu/etd/1812>

This Project is brought to you for free and open access by the Office of Graduate Studies at CSUSB ScholarWorks. It has been accepted for inclusion in Electronic Theses, Projects, and Dissertations by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

EARLY-WARNING PREDICTION FOR MACHINE FAILURES IN AUTOMATED
INDUSTRIES USING ADVANCED MACHINE LEARNING TECHNIQUES

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Information Systems and Technology

by
Satnam Singh
December 2023

EARLY-WARNING PREDICTION FOR MACHINE FAILURES IN AUTOMATED
INDUSTRIES USING ADVANCED MACHINE LEARNING TECHNIQUES

A Project
Presented to the
Faculty of
California State University,
San Bernardino

by
Satnam Singh
December 2023
Approved by:

Dr. Conrad Shayo, Committee Member, Chair

Dr. William Butler, Committee Member, Reader

Dr. Conrad Shayo, Department Chair, Information and Decision Science

© 2023 Satnam Singh

ABSTRACT

This Culminating Experience Project explores the use of machine learning algorithms to detect machine failure. The research questions are: Q1) How does the quality of input data, including issues such as outliers, and noise, impact the accuracy and reliability of machine failure prediction models in industrial settings? Q2) How does the integration of SMOTE with feature engineering techniques influence the overall performance of machine learning models in detecting and preventing machine failures? Q3) What is the performance of different machine learning algorithms in predicting machine failures, and which algorithm is the most effective? The research findings are: Q1) Effective outlier handling is vital for predictive maintenance as the variables distribution initially showed a right-skewed pattern but after rectifying, it became more centralized, with correlations between specific sensors showing potential for further exploration. Q2) Data balancing through SMOTE and feature engineering is essential due to the rarity of actual failure instances. Substantial challenges are observed when predicting 'Failure' instances, with a lower true positive rate (73%), resulting in low precision (0.02) and recall (0.73) for 'Failure' predictions. This is further reflected in the low F1-Score (0.03) for 'Failure,' indicating a trade-off between precision and recall. Despite a commendable overall accuracy of 94%, the class imbalance within the dataset (92,200 'Running' instances vs. 126 'Failure' instances) remains a contributing factor to the model's limitations. Q3)

Machine learning algorithm performance varies, with Catboost excelling in accuracy and failure detection. The choice of algorithm and continuous model refinement are critical for enhanced predictive accuracy in industrial contexts. The main conclusions are: Q1) Addressing outliers in data preprocessing significantly enhances the accuracy of machine failure prediction models. Q2) focuses on addressing the issue of equipment failure parameter imbalance. It was found in the research findings that there was a significant imbalance in the failure data, with only 0.14% of the dataset representing actual failures and 99.86% of the dataset pertaining to non-failure data. This extreme class disparity can result in biased models that underperform on underrepresented classes, which is a common problem in machine learning. Q3) Catboost outperforms other algorithms in predicting machine failures with remarkable accuracy and failure detection rates of 92% accuracy and 99% times it is correct, and further exploration of diverse data and algorithms is needed for tailored industrial applications. Future research areas include advanced outlier handling, sensor relationships, and data balancing for improved model accuracy. Addressing rare failures, enhancing model performance, and exploring diverse machine learning algorithms are critical for advancing predictive maintenance.

DEDICATION

For my Family and Friends

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF FIGURES	vii
CHAPTER ONE: INTRODUCTION	1
Background.....	3
Problem Statement	5
Research Question	6
Summary	6
Organization of the Project	7
CHAPTER TWO: LITERATURE REVIEW.....	8
CHAPTER THREE: METHODOLOGY	21
Algorithms.....	27
Random Forest	27
XGBoost.....	29
Support Vector Machine (SVM).....	31
CatBoost	34
CHAPTER FOUR: DATA COLLECTION, ANALYSIS AND FINDINGS.....	38
Data Collection	38
Analysis and Findings.....	40
CHAPTER FIVE: CONCLUSION AND FUTURE WORK	58
Discussion	58
Conclusion	58
Areas for Further Study.....	60

APPENDIX A: DATASET	62
APPENDIX B: CODE.....	65
REFERENCES	75

LIST OF FIGURES

Figure 1. Random Forest (ML Random Forest Algorithm - JavatPoint, n.d.)	28
Figure 2. XGBoost ("Introduction to XGBoost in Python," 2023).....	30
Figure 3. Support Vector Classifier (SMV).....	32
Figure 4. Catboost (Yenigün, 2022).....	36
Figure 5. Qualitative Data	41
Figure 6. Quantitative Data Information.....	42
Figure 7. Data Distribution, Outliers After Rectifying Outliers S15.....	42
Figure 8. Data Distribution, Outliers After Rectifying Outliers S17.....	43
Figure 9. Data Distribution, Outliers After Rectifying Outliers S13.....	43
Figure 10. Data Distribution, Outliers After Rectifying Outliers S5.....	44
Figure 11. Data Distribution, Outliers After Rectifying Outliers S16.....	44
Figure 12. Data Distribution, Outliers After Rectifying Outliers S19.....	45
Figure 13. Data Distribution, Outliers After Rectifying Outliers S18.....	45
Figure 14. Data Distribution, Outliers After Rectifying Outliers S8.....	46
Figure 15. Equipment Failure Information	47
Figure 16. Confusion Matrix of Support Vector Machine	48
Figure 17. Classification Report.....	48
Figure 18. Importance of the Features Based on Correlation.....	50
Figure 19. Using SMOTE for Balancing Data	51
Figure 20. Classification Report of Random Forest	52

Figure 21. Confusion Matrix of Without and With Normalization of Random Forest	52
Figure 22. Classification Report of Support Vector Machine	53
Figure 23. Confusion Matrix of Without and With Normalization of SVM.....	54
Figure 24. Classification Report of XG Boost	55
Figure 25. Confusion Matrix of Without and With Normalization of XG Boost	55
Figure 26. Classification Report of CatBoost.....	56
Figure 27. Confusion Matrix of Without and With Normalization of CatBoost.....	57

CHAPTER ONE

INTRODUCTION

In recent years, automation has become an integral part in various industries, from manufacturing to healthcare. Automated systems have brought about significant improvements in productivity, efficiency, and cost savings. However, with increasing reliance on automation, the occurrence of machine failures has become a significant concern (Timothy, 2005). Machine failures can result in significant downtime, increased costs, and reduced customer satisfaction. As our dependency on automated systems grows, the occurrence of machine breakdowns has emerged as a major issue, resulting in significant operational downtime, increased expenses, and decreased consumer satisfaction (Abidi et al., 2022).

Predictive maintenance, a cornerstone in machinery and asset management, has traditionally relied on several methodologies to anticipate and prevent machine failures. Initially, the industry favored Time-Based Maintenance (TBM) which serviced machinery at fixed intervals based on their anticipated life cycles, though this often resulted in either unnecessary maintenance or overlooking impending failures (Cavalieri & Salafia, 2020). On the other hand, Reactive Maintenance, which waited for machines to fail before intervention, was simpler but had associated high costs from emergency repairs and unexpected downtimes (Rojek et al., 2023). This led to the emergence of Condition-Based

Maintenance (CBM) that used sensor data to prompt maintenance when deviations from predefined thresholds were noted, albeit the manual interpretation was fraught with errors and certain failure modes could be overlooked (Goyal & Pabla, 2015). To optimize strategies, the industry also employed the comprehensive yet time-consuming Reliability-Centered Maintenance (RCM), which despite its depth, struggled with real-time adaptability to changes in machinery behavior or operational environments (Tiddens et al., 2023).

The integration of AI and machine learning into predictive maintenance has revolutionized machinery maintenance, overcoming many of the limitations inherent in traditional methodologies such as TBM, Reactive Maintenance, CBM, and RCM. These advanced ML algorithms, adept at analyzing extensive datasets, provide critical data-driven insights, identifying patterns and anomalies potentially overlooked by human analysts (Cross, 2015). Their real-time analytical prowess facilitates prompt interventions, marking a departure from conventional periodic checks (Heron & Smyth, 2010). Additionally, the continuous adaptability of these models ensures updated and accurate predictions, minimizing false alarms and enhancing cost efficiency by focusing on genuine maintenance needs rather than relying on predetermined schedules or reactive measures. This technological evolution, combining AI's capabilities with data insights, signifies a transformative phase in achieving operational excellence in numerous industrial sectors (Brock & Von Wangenheim, 2019).

Background

Machine failures, which encompass the malfunction or breakdown of machines causing suboptimal performance, pose a critical challenge in various industries. Such failures often arise from wear and tear, misuse, inherent design flaws, or external environmental factors. These breakdowns can severely affect an organization's operations and profitability, leading to consequences such as reduced output, compromised efficiency, escalated maintenance costs, and even potential safety hazards (Ku, 2018). Given these repercussions, there's a compelling need for organizations to adopt a proactive stance towards the prevention of machine failures. Embracing such an approach not only safeguards operational integrity but also paves the way for enhanced productivity and reduced operational costs. Consequently, as we delve deeper into this topic, it becomes evident that exploring and instituting robust preventive mechanisms is crucial for the seamless functioning of industries.

In the realm of equipment management and optimization, certain methodologies stand out as pivotal in ensuring longevity and operational efficiency. Preventive Maintenance serves as the front line of defence against potential machine failures. It encapsulates a proactive regimen where regular checks and repairs play a quintessential role in ensuring machinery functions optimally. By staving off foreseeable issues, this approach aims to reduce

unforeseen downtimes and subsequent operational losses (Chanda & Banerjee, 2022b).

Parallel to this, there's an ever-evolving need for real-time insights into machine health, ushering in the significance of Condition Monitoring. As opposed to reactive measures, condition monitoring consistently oversees machinery performance, flagging inconsistencies or anomalies. Such vigilant oversight ensures timely interventions, circumventing failures that could otherwise have been detrimental (Natarajan & Srinivasan, 2010).

Yet, in the broader spectrum of system efficiency, a holistic approach becomes imperative. Enter Reliability Engineering, a discipline that transcends mere maintenance. With its roots in systematic methodologies, reliability engineering doesn't merely address, but strategically enhances both the performance and reliability of machinery, systems, and even end products. This holistic view ensures that every cog in the machinery, be it a physical component or a process, functions in harmony, thereby driving optimal outputs (Tsang, 2018).

In addition to these proactive techniques, it is critical to do root cause analysis (RCA) following a machine failure. RCA is the process of determining the root cause of an issue to avoid it from recurring in the future. This frequently entails a thorough investigation of the machine, its components, and the circumstances underlying the failure (Lokrantz, Gustavsson & Jirstrand, 2018).

Several important factors must be addressed when evaluating machine failure data in order to accurately evaluate the probability of machine failure and identify possible causes (Angelopoulos et al., 2019). Time of failure, Type of machine, Operating parameters, Service history, Component failures, and Location are some typical criteria that are frequently included in machine failure data (Angelopoulos et al., 2019).

Problem Statement

Machine failures can result in significant costs and downtime for industries that rely on automation. In order to minimize the impact of machine failures, it is important to detect them early and carry out preventative maintenance (Chen & Zhang, 2018). Traditional machine failure prediction methods are based on manual inspections and threshold-based approaches, which are often reactive and time-consuming (Salfner et al., 2010).

The operational integrity of automation systems hinges significantly on the accuracy and reliability of machine failure prediction models (Chanda & Banerjee, 2022). The repercussions of unpredictable machine failures can ripple across various domains, from operational downtimes to economic ramifications. While there's consensus on the pivotal role of real-time prediction models, there remains a void when it comes to their seamless integration within automated industries (Navarro et al., 2022). Advanced machine learning algorithms, with their potential to revolutionize this domain, appear as a beacon of promise.

However, a concerning lacuna has been identified in contemporary research: there's a limited exploration into the application of these avant-garde algorithms specifically for machine failure prediction within the context of automated industries. Dalzochio et al. (2020) have accentuated this gap, emphasizing that previous studies often tend to overlook this niche that is yet crucial application. Consequently, there's a pressing need to delve into this uncharted territory, building upon the foundation set by prior research while addressing the areas they've earmarked for further exploration.

Research Question

1. How does the quality of input data, including issues such as outliers, and noise, impact the accuracy and reliability of machine failure prediction models in industrial settings?
2. How does the integration of SMOTE with feature engineering techniques influence the overall performance of machine learning models in detecting and preventing machine failures?
3. What is the performance of different machine learning algorithms in predicting machine failures, and which algorithm is the most effective?

Summary

Automated systems have improved productivity, efficiency, and cost savings, machine failures can result in significant downtime, increased costs, and

reduced customer satisfaction. Traditional machine failure prediction methods are often reactive and time-consuming, leading to a growing need for more accurate and reliable machine failure prediction models that can provide real-time results and easily integrate into automation systems.

Advanced machine learning algorithms have shown great promise in predicting machine failures and allowing for early detection and preventative maintenance. However, there is a lack of research on the application of these algorithms to machine failure prediction in automated industries (Jung et al., 2021). The aim of this study is to address this gap in knowledge by exploring the use of machine learning algorithms to predict machine failures, the factors influencing machine failures in automated industries, and the performance of different machine learning algorithms in predicting machine failures.

Organization of the Project

This Culminating Experience Project is organized as follows:

Chapter 1 covers the Introduction and Problem Statement

Chapter 2 covers the Literature Review

Chapter 3 covers the Methodology

Chapter 4 cover Data Collection and Analysis

Chapter 5 covers Conclusion and Areas for the Further Studies

CHAPTER TWO

LITERATURE REVIEW

Question 1: How does the quality of input data, including issues such as outliers, and noise, impact the accuracy and reliability of machine failure prediction models in industrial settings?

The ability to harness advanced machine learning algorithms for predicting machine failures aligns with the industry's pursuit of minimizing downtime, optimizing maintenance operations, and maximizing overall operational efficiency. Numerous variables, spanning from mechanical wear and tear to environmental conditions, play pivotal roles in the degradation and eventual malfunction of machinery. Factors such as vibration levels, temperature variations, humidity, and operational loads contribute to the intricate web of influences on machine performance (Qing et al., 2022).

Understanding these multifaceted factors is imperative for developing effective predictive maintenance strategies that can mitigate downtime and optimize operational efficiency. Singh et al., (2016) reviewed the literature on the sensitivity of machines to different mechanical and electrical flaws that might result in motor failure and unplanned downtime. Since it is frequently not economically practical to physically inspect machines after they fail, computer models have been created to simulate motor failure and the resulting changes in

measured parameters. The authors provide a summary of mathematical models that have been applied to research motors in malfunction (Singh et al., 2016). Many linked circuit models, dq models, magnetic equivalent circuit models, and finite element models are the different types of models. The merits and cons of each type of model in simulating various fault types are examined in detail.

The study by Gaddam et al. (2020) says, in the era of Industry 4.0, the fusion of advanced machine learning techniques with smart sensors and robust communication technologies is fundamentally transforming our interactions with the physical world, permeating sectors such as work, learning, innovation, and entertainment. However, the operational challenges stemming from the deployment of these smart sensors in demanding industrial environments have been found to generate outliers—erroneous and unusual data readings. Gaddam et al. (2020) emphasizes in their research that while extensive efforts have been dedicated to devising sensor outlier detection models, the unique and intricate operational context of machine learning within industrial settings necessitates tailored approaches. Detecting sensor faults and anomalies takes center stage, ensuring not only data quality but also the reliability of machine learning models deployed for the processing of sensor data and the advancement of various industrial applications. In this comprehensive investigation, we delve into the multifaceted landscape of detecting sensor faults, anomalies, and outliers in industrial domains, unveiling the core findings and results that underscore the significance of selecting specialized outlier detection models for bolstering the

dependability of machine learning systems tasked with the management and interpretation of sensor data.

The study by De Jesus et al. (2021) says, in the domain of environmental monitoring, where sensor platforms often confront harsh conditions while monitoring complex phenomena, the task of designing dependable systems faces formidable challenges due to external disturbances affecting sensor measurements. Even the seemingly simple task of outlier detection in sensor data becomes intricate, as it requires distinguishing genuine data errors arising from sensor faults from deviations caused by natural phenomena (De Jesus et al., 2021). Existing solutions for runtime outlier detection rely on precise physical process modeling or the assumption that outliers exhibit conspicuous deviations, easily filtered by predefined thresholds. Alternatively, they depend on redundant data from multiple sensors for voting-based techniques. To address these complexities, this article introduces an innovative methodology that leverages machine learning to model individual sensor behavior, using correlated data from related sensors to accurately estimate environmental parameters and construct failure detectors. This approach not only distinguishes genuine abnormalities from natural deviations but also quantifies measurement quality, enhancing data reliability. Applied to real datasets from an aquatic monitoring system, the methodology showcases its effectiveness in identifying outliers, surpassing existing solutions in accuracy and promising improved reliability for environmental monitoring systems.

The field of predictive maintenance, a critical area in automated industries. Leveraging the proposed methodology, the researchers Garouani et al., (2022) have conducted rigorous testing, centered on the analysis of over 360 databases derived from the domain of predictive maintenance. Through these tests, the researchers demonstrate the efficacy of their approach in addressing the pertinent challenge of predicting machine failures. By harnessing automated machine learning, the study contributes to the overarching discourse on optimizing machine learning techniques for real-world industrial scenarios.

The study by Konstantinidis et al., (2022) says that since Industry 4.0 permits more flexibility and customization in production processes, it has a significant impact on traditional business models. Also, it has a big impact on small and medium-sized businesses since it enables them to compete with bigger businesses by implementing digital technologies. Furthermore, highlights how crucial decision-making procedures are in the context of Industry 4.0. It is essential to have efficient decision-making processes in place to make use of the vast amount of data provided by digital technology.

Study by Wang et al., (2017) says the rapid evolution of optical networks serving as the backbone of many essential services, there's an increasing demand for sophisticated performance monitoring and failure prediction solutions. Machine learning has emerged as a promising approach to address these challenges, specifically utilizing algorithms like the Support Vector Machine (SVM) and Double Exponential Smoothing (DES). While SVM's prowess in

handling high-dimensional data for classification makes it apt for predicting equipment failures, DES, known for forecasting network traffic patterns, captures the trend and seasonality of data, ensuring accurate modelling of optical network behaviours. Despite traditional risk-aware models providing insights, their static nature often overlooks the evolving dynamics of network behaviors. However, our pioneering effort to integrate DES and SVM has shown a remarkable 95% prediction accuracy in optical network equipment failure, signifying its potential in enhancing network resilience and filling the noticeable research gap in proactive failure prediction methodologies.

Question 2: How does the integration of SMOTE with feature engineering techniques influence the overall performance of machine learning models in detecting and preventing machine failures?

The accurate identification of failures and defects in industrial machines plays a pivotal role in assessing their warranty and overall performance (Sridhar & Sanagavarapu, 2021b). Industrial machines undergo depreciation owing to a variety of factors, with common culprits including tool wear, strain, heat, and power failure. This paper focuses on the development of machine learning algorithms aimed at predicting machine failures. To achieve this, a synthesized dataset mirroring real-time failures encountered in industrial settings was employed to construct a predictive maintenance model. However, the presence

of class data imbalance posed a challenge to the performance of machine learning algorithms. To address this issue, SMOTE-based oversampling techniques were evaluated. The results demonstrated a notable 7.83% increase in the Area Under Curve (AUC) score, signifying improved performance of the Random Forest classifier in effectively distinguishing between instances of non-failure and machine failures.

Modern condition monitoring systems for electrical machines have increasingly relied on data-driven methods, offering a straightforward approach to effective fault detection and diagnostics (Swana et al., 2022b). Despite their advantages, practical implementation encounters challenges such as data imbalance. The scarcity of reliable labelled fault data from real-world machines poses a significant obstacle to developing accurate supervised learning-based condition monitoring systems. This study explores the application of the Naïve Bayes classifier, support vector machine, and k-nearest neighbours in conjunction with synthetic minority oversampling technique, Tomek link, and their combined use for fault classification using both simulated and experimentally imbalanced data. A comprehensive comparative analysis across various imbalanced data scenarios assesses their suitability for condition monitoring in a wound-rotor induction generator. Performance evaluation utilizes precision, recall, and f1-score metrics, revealing that the combination of the synthetic minority oversampling technique with the Tomek link consistently delivers the best performance across all classifiers. Particularly, the k-nearest neighbours,

when coupled with this combined resampling technique, achieves the most accurate classification results. This research is valuable to both researchers and practitioners in the field of condition monitoring for electrical machines, aiding in the selection of appropriate techniques for handling imbalanced fault data, a critical consideration in the limited data landscape of condition monitoring for electrical rotating machines (Swana et al., 2022b).

A study by Kim et al., (2016) states, in semiconductor manufacturing, the prediction of faults in the FAB (wafer fabrication) process is instrumental in enhancing product quality and reliability through effective classification performance. However, the FAB process occasionally experiences faults, with most of these instances categorized as "pass." Consequently, this leads to data imbalance in the pass/fail class distribution. This data imbalance poses challenges for prediction models as it tends to introduce bias towards the majority class (pass class), making it difficult to accurately predict instances of the "failure" class (Kim et al., 2016).

In this Culminating Experience Project, we propose a solution to this problem by introducing the SMOTE (Synthetic Minority Oversampling Technique)-based oversampling method (Duan et al., 2022). This approach aims to address the imbalance between the "pass" and "fail" classes by oversampling the minority class, "fail." By applying this method, we effectively mitigate data imbalance and enhance the performance of fault detection prediction models in

the FAB process, thus ultimately contributing to improved product quality and reliability.

Industrial manufacturing processes often grapple with the disruptive impact of machine failures, which can lead to unplanned downtime and substantial revenue losses for manufacturers (Vuttipittayamongkol & Arreeras, 2022). To address this challenge, numerous machine learning-based approaches have been proposed to enable the instant detection of occurring failures or the prediction of potential breakdowns (Vuttipittayamongkol & Arreeras, 2022). However, several limitations and issues persist, demanding focused attention.

In their study, Vuttipittayamongkol & Arreeras (2022) address the challenges of acquiring real-world industrial data, particularly in the context of big data, the intricate task of feature selection, and the under-representation of machine failure events within the data. They present a novel approach that leverages a relatively small predictive maintenance dataset and basic supervised learning algorithms for industrial machine failure detection. Vuttipittayamongkol & Arreeras (2022) also highlight the importance of addressing the imbalanced class distribution inherent in such data to enhance detection accuracy. They employ a range of non-deep learning algorithms for classification, complemented by data resampling methods to improve model performance. Their findings indicate that decision tree algorithms are effective in achieving robust classification results, with the implementation of an undersampling method yielding a notable detection accuracy of 91%. This study by Vuttipittayamongkol & Arreeras (2022)

contributes significantly to the field, recommending further exploration in the area of machine failure detection using limited data resources.

Imbalanced datasets can be a formidable hurdle in classification problems, and this study delves into the intricacies of imbalanced classification utilizing support vector machines (SVM). The research by Illán et al., (2019) aims to comprehensively understand and quantify the challenges posed by imbalanced datasets when employing SVMs. Through a combination of theoretical analysis and experimental exploration, the study identifies the conditions under which SVM failures can manifest. Interestingly, the research reveals that SVM failures can be relevant even in scenarios characterized by very slight imbalances in the data distribution. To mitigate these challenges and avoid SVM failures, the study also provides guidelines for conducting exploratory data analysis. By following these guidelines, practitioners can navigate the complexities of imbalanced classification more effectively and improve the reliability of SVM-based classification models (Illán et al., 2019).

Question 3: What is the performance of different machine learning algorithms in predicting machine failures, and which algorithm is the most effective?

The most commonly used algorithms in predicting machine failures is Support Vector Machine, as evidenced in study (Sridhar & Sanagavarapu, 2021). This machine learning algorithms used in predictive maintenance, with a focus on determining the most effective choice. Support Vector Machine (SVM) has been

acknowledged as a valuable algorithm for predicting machine failures, as substantiated by several studies in the literature (Sridhar & Sanagavarapu, 2021). SVM's performance in predictive maintenance has consistently demonstrated its effectiveness in accurately identifying potential failures and contributing to improved maintenance strategies.

The evaluation of various machine learning algorithms in predicting machine failures has been the subject of extensive investigation within the realm of predictive maintenance (Rousopoulou et al., 2020, Nguyen et al., 2023). Researchers have explored a wide array of algorithms, each with its unique strengths and limitations, to determine their performance in anticipating impending machine failures. The selection of the most effective algorithm depends upon the specific characteristics of the dataset, the complexity of the problem, and the interpretability of the results. A significant study conducted by Costa et al., (2019) compared and evaluated the efficacy of various statistical and machine-learning methods for forecasting industrial robot breakdowns. This is a crucial job since failure predictions can enhance maintenance plans, reduce accidents, and save money for businesses that use these robots. According to the outcomes provided by Costa et al., (2019), the hybrid gradient boosting method is particularly good at failure prediction. This method mixes components from statistical and machine learning techniques, which might enable it to benefit from the best aspects of each. It is noteworthy to notice that while failure

categorization improves from information use, local joint information is particularly helpful for failure detection (Costa et al., 2019).

Balamurugan et al. (2019) describe the substantial transformation occurring in the global manufacturing sector, transitioning into what is often termed "Manufacturing 4.0." This period is marked by the deep integration of digital technologies, which is reshaping traditional manufacturing and management practices. They emphasize that the keystones of Industry 4.0, including Artificial Intelligence (AI), machine learning, the Internet of Things (IoT), and cyber-physical systems, are paving the way for the future of "smart factories." In these advanced environments, the interconnectedness of interfaces, machines, and modules allows for an unprecedented level of communication and data exchange, leading to a potential revolution in manufacturing efficiency and adaptability.

Balamurugan et al. highlight that the proliferation of extensive data, combined with AI and machine learning, is leading to a new era of industrial automation that optimizes manufacturing processes. This integration goes beyond mere automation; it facilitates intelligent, data-driven decision-making, revolutionizing traditional manufacturing approaches. The impact of merging AI and machine learning with Industry 4.0 extends to modernizing operational techniques and establishing an adaptive, innovative, and responsive manufacturing ecosystem. This integration positions the sector to effectively face future challenges and seize opportunities.

The industrial sector's efficiency often hinges on the uninterrupted operation of electric motors, with unexpected failures leading to significant economic and operational setbacks. Emphasizing the need for proactive solutions, the spotlight has shifted to condition monitoring and predictive maintenance. The paper presented delves into an innovative Machine Learning (ML) architecture for predictive maintenance, harnessing the power of the Random Forest algorithm. This ensemble learning method, acclaimed for its applications in classification and regression, is employed to monitor and predict equipment health. With data sourced from diverse sensors, machine PLCs, and varied communication protocols, the system capitalizes on Azure Cloud architecture, ensuring real-time processing and storage of information. The preliminary findings by Paolanti et al., (2018) from the implementation are promising. The Random Forest-based ML model not only captures a comprehensive perspective of equipment performance but also forecasts various machine states with marked accuracy, surpassing traditional simulation tools in some regards. This convergence of machine learning with cloud capabilities, exemplified by the Azure platform, signals a paradigm shift in predictive maintenance. Such innovations stand to offer industries a dynamic, scalable, and data-driven approach, aiming to enhance equipment reliability and drastically reduce unforeseen downtimes.

In the realm of wireless sensor networks (WSNs), data collected is susceptible to various forms of faults attributed to both internal and external

influences, including issues like calibration discrepancies, low battery levels, environmental interference, and sensor aging. This paper specifically delves into the identification and classification of faults arising from low battery levels and calibration errors within WSNs (Warraich et al., 2017). Leveraging the efficacy of machine learning algorithms in fault detection and classification, this study evaluates and compares the performance of three prominent algorithms: k-nearest neighbor (kNN), support vector machine (SVM), and Naive Bayes. Utilizing real-world datasets, the paper conducts a comprehensive comparative analysis of these approaches. The methodology is validated using empirical data obtained from motes deployed in an actual living lab over an extended period. The results highlight that the k-nearest neighbor (kNN) algorithm outperforms other methods, demonstrating superior fault detection rates according to the specified performance metrics.

CHAPTER THREE

METHODOLOGY

As noted in Chapter 1, this project will seek to answer the following questions:

1. How does the quality of input data, including issues such as outliers, and noise, impact the accuracy and reliability of machine failure prediction models in industrial settings?
2. How does the integration of SMOTE with feature engineering techniques influence the overall performance of machine learning models in detecting and preventing machine failures?
3. What is the performance of different machine learning algorithms in predicting machine failures, and which algorithm is the most effective?

Question 1: How does the quality of input data, including issues such as outliers, and noise, impact the accuracy and reliability of machine failure prediction models in industrial settings?

In my project, I will address the limitations of traditional prediction methodologies in automated industries, which historically have relied on static analysis and predefined logics. As noted by Lee et al. (2014), these traditional methods, while being foundational, often lack adaptability to the dynamic variables and complex behaviors typical in automated industries. To overcome

these limitations, I will apply machine learning, a field that excels in processing vast, complex datasets and iteratively learning from them. Machine learning's ability to identify patterns and anomalies that extend beyond the capabilities of traditional methods, as discussed by Erhan et al. (2021), positions it as an ideal solution for the challenges faced by automated industries. My project will harness this potential of machine learning to bring a more dynamic, data-driven approach to predicting and managing the intricate dynamics of automated industrial systems.

I will initiate an extensive data extraction process, gathering data from a wide range of automated industries. The focus will be on key parameters crucial to machine operations and identifying potential malfunctions. This includes data on the manufacturer, the last maintenance date, regional clustering, type of machine, machine age, sensor readings, machine runtimes, and a historical record of machine failures. Utilizing this diverse dataset, I plan to employ advanced machine learning algorithms, specifically Random Forest and Support Vector Machines, tailored to the unique challenges of this project.

A selected subset of this data will be used to train these algorithms. After the training phase, the algorithms will be rigorously tested on previously unseen data segments to evaluate their effectiveness in real-world scenarios. I will then assess their predictive accuracies through a detailed comparative analysis, employing a classification evaluation matrix. This matrix will include various performance metrics such as the confusion matrix, True Positives, True

Negatives, False Positives, False Negatives, overall Accuracy, Sensitivity, and the F-beta score. This approach, inspired by the work of Subashini et al. (2009), is designed to provide a comprehensive understanding of the effectiveness of these machine learning models in predicting and managing potential machine malfunctions in automated industrial settings.

I will apply advanced machine learning algorithms, drawing from a systematic approach and initial analyses to align closely with the complex dynamics of automated industries. Based on insights from Anumbe et al. (2022), I anticipate that these algorithms will surpass the performance benchmarks set by traditional models. My project will leverage the adaptability and continuous learning capabilities of machine learning algorithms, aiming to enhance their potential in predicting machine failures in automated settings. I intend to model predictions with a focus on accuracy and reliability, pertinent to the automated industry sector. This approach is expected to demonstrate the transformative capacity of these algorithms, highlighting the significant value they can add in preempting and addressing machine failures, thereby contributing to more efficient and effective maintenance strategies in automated industries.

Question 2: How does the integration of SMOTE with feature engineering techniques influence the overall performance of machine learning models in detecting and preventing machine failures?

In answer, I'll describe how combining feature engineering methods with SMOTE (Synthetic Minority Over-sampling Technique) can greatly improve machine learning models' overall efficacy in detecting and preventing machine failures. In automated industries, this method is essential for handling the complexity of machine breakdowns.

Feature Identification and Extraction: A wide range of features are found and extracted in the context of predicting machine failure. These characteristics cover a wide range of information, such as machine kinds, age, sensor values, runtimes, region clusters, manufacturer data, and last maintenance dates. Each of these characteristics offers insightful data that may affect forecasts of machine failure (Zhuhadar & Lytras, 2023).

SMOTE is incorporated into the procedure to address class imbalance, where machine failures are uncommon in comparison to typical machine operations. To balance the dataset, SMOTE creates synthetic samples of the minority class, in this case, machine failures (Akbulut et al., 2023). By doing this, we can make sure the machine learning model has enough data to train and can predict machine failures more accurately.

Another essential component of improving machine learning models is feature engineering. To give the data a deeper understanding, this entails adding new features or altering current ones. Richer and more informative data for the model can be obtained, for example, by computing the mean or variance of

sensor values over specified timeframes and using other feature engineering techniques (Hoseinitabatabaei et al., 2013).

The data must be pre-processed in order for machine learning to be possible. To make sure the dataset is clean and prepared for modeling, one-hot encoding, data normalization, and handling missing values are used (Bouramtane et al., 2023). When preprocessing, SMOTE is especially helpful in making sure the minority class is fairly represented.

After preprocessing, the improved features—including those made better by SMOTE and feature engineering—are added to machine learning models. The effectiveness of machine learning models in identifying and averting machine failures is significantly impacted by the combination of a balanced dataset and informative features.

When SMOTE is combined with feature engineering, preliminary analyses and research (Akbulut et al., 2023) show a marked increase in the capacity of machine learning models to identify and stop machine failures. This method lessens the problems caused by class imbalance and gives the models access to more pertinent and instructive data. It therefore results in more precise forecasts and improves the dependability and effectiveness of intervention and maintenance plans in automated industries.

In conclusion, when working with imbalanced datasets, in particular, the overall performance of machine learning models in identifying and averting machine failures will be significantly improved by the integration of SMOTE with

feature engineering techniques. In automated industries, this combined approach improves prediction accuracy and increases the efficacy of maintenance and intervention strategies.

Question 3: What is the performance of different machine learning algorithms in predicting machine failures, and which algorithm is the most effective?

To answer this question, I will find accuracy with four algorithms—CatBoost, SVM, Random Forest, and XGBoost—and compare them to answer the question.

Machine learning offers a rich assortment of algorithms, each equipped with distinct strengths, to predict machine failures (Ma & Sun, 2020). Predicting machine failures is a complex task that requires a balance of accuracy, interpretability, and computational efficiency. While numerous algorithms exist, Random Forest, Decision Tree, and XGBoost are often championed for their efficacy in handling intricate datasets characteristic of automated industries (Chen et al., 2020).

The best algorithm to predict machine failures in this situation is a question that needs to be answered. We will evaluate the effectiveness of these four algorithms by gauging how well they predict machine failures using a particular dataset to respond to this question. We will be able to compare these

algorithms intelligently and decide which one works best for the application because of this empirical evaluation.

To summary, the effectiveness of these four algorithms—CatBoost, SVM, Random Forest, and XGBoost—will be evaluated based on how well they predict machine failures. A thorough analysis will assist in determining which algorithm performs best for this specific prediction task.

Algorithms

Random Forest

Random Forest is a bagging-based ensemble method that constructs multiple decision trees to produce more accurate and stable predictions. (Li et al., 2018)

Working:

The algorithm begins by bootstrapping, or sampling with replacement, multiple subsets from the dataset. A decision tree is grown on each subset. When predicting, each tree votes, and the majority class is chosen. For machine failure features like "Manufacturer" or "last date maintenance," RF can account for a variety of scenarios and nuances that a single tree might overlook.

Advantages:

- Reduces overfitting which is commonly found in decision trees.
- Can handle large datasets with higher dimensionality.

- Can estimate missing values.
- Maintains accuracy even when a large proportion of the data is missing.

Process in the Context of Machine Failure:

Given machine failure features like "machine age" and "seven sensor values machine runtimes," RF will consider diverse scenarios across its multitude of trees. For instance, while one tree might focus on the importance of machine age, another could prioritize sensor values.

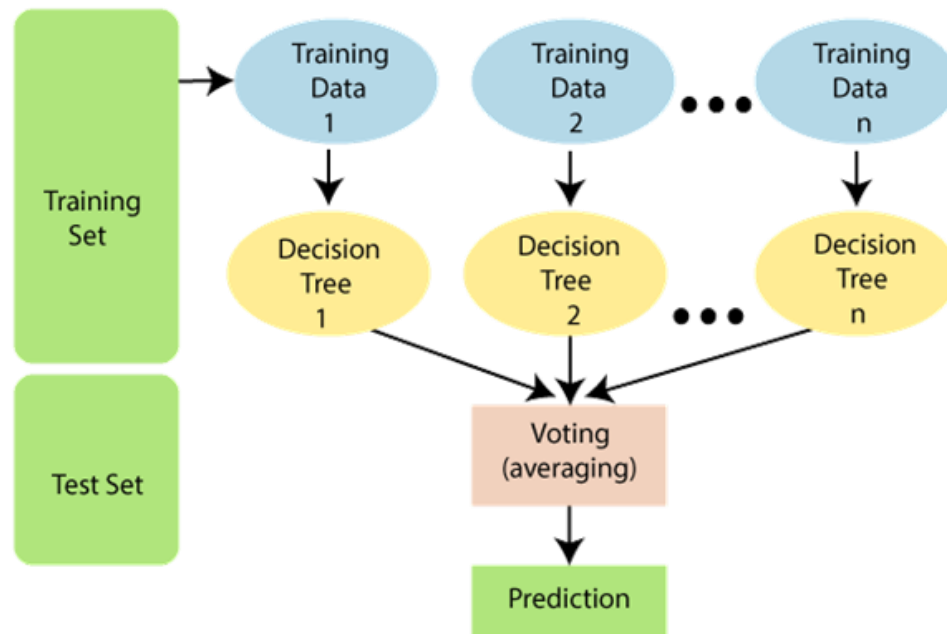


Figure 1. Random Forest (ML Random Forest Algorithm - JavatPoint, n.d.)

Key Parameters Includes:

- `n_estimators`: The number of trees in the forest.
- `max_features`: The number of features to consider for the best split.
- `min_samples_split`: The minimum number of samples required to split an internal node

XGBoost

XGBoost is an optimized gradient boosting algorithm known for its speed and performance (Keck, 2016).

Working:

It works by iteratively adding weak learners (typically decision trees) in a manner that each subsequent tree corrects the errors of its predecessor. XGBoost places a significant emphasis on regularization to prevent overfitting. In the context of machine failures, if "machine runtimes" indicate excessive usage leading to faster wear and tear, XGBoost can weigh this feature heavily during its boosting rounds.

Advantages:

- Regularization prevents overfitting, making XGBoost robust.
- Capability to handle missing values internally.
- Parallelizable, making it faster. (Qin et al., 2021).

Process in the Context of Machine Failure:

Given machine failure features, XGBoost will rank the importance of each feature after its boosting rounds. Features like "last date maintenance" or "temperature fluctuations" can be assessed in terms of their influence on machine health, allowing industries to prioritize maintenance schedules or environment controls accordingly.

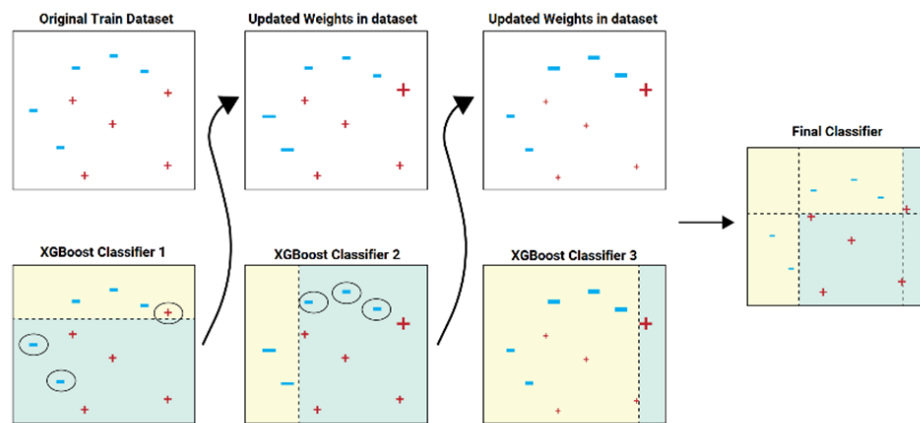


Figure 2. XGBoost (*"Introduction to XGBoost in Python,"* 2023)

Key parameters include:

- `learning_rate`: Step size shrinkage used to prevent overfitting.
- `max_depth`: Maximum depth of the decision trees.
- `n_estimators`: Number of boosting rounds or trees to be run (Huang et al., 2023).

Support Vector Machine (SVM)

Support Vector Machine, often abbreviated as SVM, is a supervised machine learning algorithm mainly used for classification and, to a lesser extent, regression tasks. It primarily works by finding a hyperplane that best divides a dataset into classes. (Ullah et al., 2021)

Working:

The primary objective of the SVM is to segregate the given dataset in the best possible way. When the segregation is done, the distance between the nearest data point (either class) or the hyperplane should be maximized. These nearest data points are termed as "support vectors." Essentially, SVM finds the optimal hyperplane that minimizes the margin between support vectors of two classes.

In cases where data isn't linearly separable, SVM employs a technique known as the "kernel trick." This maps the input data into higher-dimensional space where a separating hyperplane can be found. Common kernels include polynomial, radial basis function (RBF), and sigmoid.

Advantages:

- Effective in high dimensional spaces or when the number of dimensions exceeds the number of samples.

- Memory efficient, as it uses a subset of training points (the support vectors).
- Flexible, due to the decision function being defined by the support vectors and the kernel trick enabling the algorithm to tackle non-linear relationships. (Xu et al., 2006)

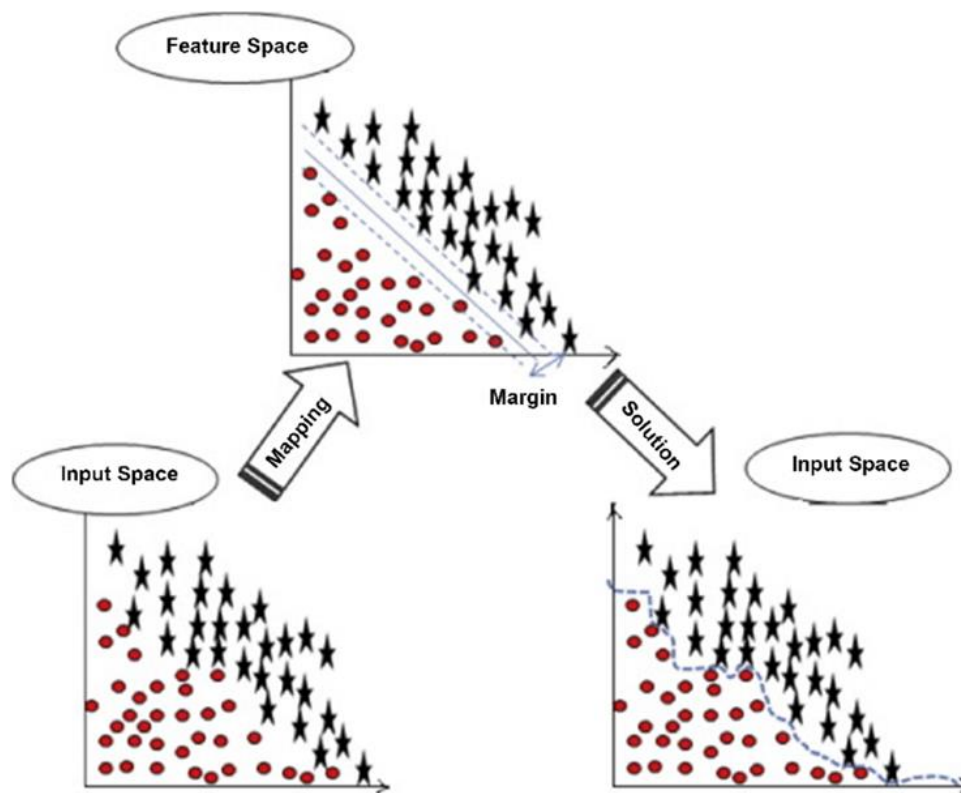


Figure 3. Support Vector Classifier (SMV)

Applications in Machine Failure Prediction:

In the context of machine failure prediction, SVM can be crucial when considering features that are not linearly correlated with failure patterns. For example, the relationship between machine age and failure likelihood may not be strictly linear. In such scenarios, SVM can map these features into a space where their relationship becomes clearer. Features like sensor readings which may come in high dimensions, SVM stands out because of its efficacy in handling high-dimensional data. If, for instance, a machine's health is determined by readings from various sensors (Ken J. Kubota BS, SEP,2016) (temperature, pressure, vibration, etc.), SVM can effectively analyze this multivariate data to classify the health state of the machine.

Crucial Parameters Support Vector Machine:

- **C (Cost parameter):** It controls the trade-off between maximizing the margin (distance between the decision boundary and the nearest data point of any class) and minimizing classification error. A small C value creates a wider margin, which may result in more misclassifications. In contrast, a large C value seeks a smaller margin and can lead to overfitting if it tries too hard to classify all points correctly.
- **Kernel:** The SVM algorithm uses a variety of kernel functions to transform the input data into higher-dimensional spaces. Common kernel functions include:
 - **Linear:** No transformation.

- Polynomial: Non-linear transformation with the degree of polynomial as an additional parameter.
- Radial Basis Function (RBF): A popular choice; it has another parameter ' γ ' which needs tuning.
- Sigmoid: Sigmoid function as a kernel.
- γ (Gamma): It's used in the RBF kernel and determines the shape of the decision boundary. A low value will produce a more flexible curve, while a high value will result in a more defined or tight curve.

CatBoost

CatBoost is a state-of-the-art, open-source gradient boosting library developed by Yandex, primarily known for its prowess in handling categorical features directly. It has gained popularity in various machine learning tasks due to its efficient and advanced implementations, which lead to superior results with less extensive hyperparameter tuning.

Working:

Like other gradient boosting methods, CatBoost works by building an ensemble of decision trees, where each successive tree aims to correct the errors of its predecessor. One of the unique features of CatBoost is its treatment of categorical variables:

- **Categorical Features Handling:** Traditionally, categorical features need to be transformed into a numerical format using techniques like one-hot encoding. However, this can dramatically increase the data's dimensionality. CatBoost eliminates this need by using a technique called "ordered boosting," where it calculates statistics on the target variable for each category and uses them for training, but in a way that avoids target leakage.
- **Oblivious Trees:** Unlike traditional decision trees, where each node has its own splitting criteria, oblivious trees use a single feature split for each level of the tree. This results in more balanced trees and reduces overfitting.
- **Regularization:** CatBoost incorporates the L2 regularization in its loss function, reducing the chance of overfitting.

Advantages:

- Direct handling of categorical variables reduces preprocessing steps and potential information loss.
- Provides built-in support for text data.
- It is less sensitive to hyperparameter configurations compared to other gradient boosting algorithms, making it user-friendly for beginners.
- Efficient implementations and faster training times.
- Offers built-in support for visualization which helps in understanding feature importance, model performance, and other diagnostics.

Applications in Machine Failure Prediction

Given that machinery data can often contain a mix of numerical sensors' readings and categorical data (e.g., machine type, manufacturer, region cluster), CatBoost's innate ability to handle categorical data is a boon. Instead of relying on transformations that could dilute the interpretability of such features, CatBoost preserves the essence of categorical variables, thereby potentially uncovering nuanced relationships that lead to machine failures.

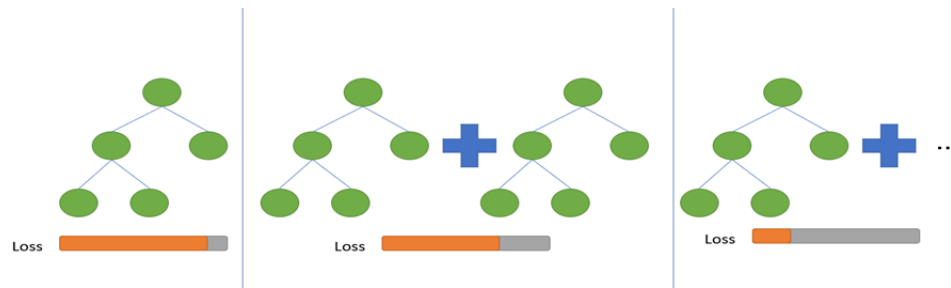


Figure 4. Catboost (Yenigün, 2022)

Crucial Parameters CatBoost:

- **Learning Rate (eta):** It controls the contribution of each tree in the ensemble. Smaller values make the optimization more robust but require more trees.
- **Depth:** The maximum depth of the tree. Deep trees can lead to overfitting, while shallow trees might not capture intricate patterns.

- `L2_leaf_reg`: L2 regularization term on weights. It's used for preventing overfitting.
- `Iterations`: The maximum number of trees that can be built. More trees can make the model more complex and potentially overfit.
- `Border_count`: The number of splits for numerical features. It's equivalent to the number of bins used to bucket continuous features.

Methods Applied:

To evaluate the performance of these algorithms:

- **Data Splitting**: The dataset was divided into training and test sets, ensuring a fair assessment of algorithmic prowess.
- **Algorithm Implementation**: Each algorithm was trained on the training set, incorporating the features related to machine failures. Algorithm-specific parameters were tuned using cross-validation to optimize performance.

Performance Evaluation: Post training, predictions were made on the test set. The classification report provided metrics like precision, recall, and F1-score. The confusion matrix further delineated true positives, false positives, true negatives, and false negatives, offering a consolidated view of the model's accuracy and misclassifications (Rahmati et al., 2019).

CHAPTER FOUR

DATA COLLECTION, ANALYSIS AND FINDINGS

This chapter used analytical approaches to address the questions from the earlier chapters of our research. This study seeks to understand the underlying causes and trends of machine breakdowns and have employed machine learning to forecast these failures.

Data Collection

The dataset offers insights into machine breakdowns and is designed for crafting machine learning solutions for predicting such failures. The columns in the dataset include information like the failure's date and time, the machine's ID, and various sensor outputs. Sourced from GitHub, this dataset comprises 307,751 entries across 16 columns (Shadgriffin, n.d.) & (Griffin, 2022).

The dataset's description is given in the table below.

s.no	Attribute name	Description	Range of value
1	ID	ID field that represents a specific machine	100001 to 100617
2	DATE	The date of the observation.	731

3	REGION_CLUSTER	A field that represents the region in which the machine resides.	8
4	MAINTENANCE_VENDOR	A field that represents the company that provides maintenance and service to the machine	8
5	MANUFACTURER	The company that manufactured the equipment in question.	10
6	WELL, _GROUP	A field representing the type of machine	1 to 8
7	EQUIPMENT_AGE	Age of the machine, in days.	0 to 15170
8	S15	A Sensor Value	0 to 59.04
9	S17	A Sensor Value	0 to 2555.52
10	S13	A Sensor Value.	0 to 592.89
11	S16	A Sensor Value.	0 to 24.6
12	S19	A Sensor Value.	0 to 511
13	S18	A Sensor Value	0 to 4151.7
14	S8	A Sensor Value.	-16.49 to 2068.11
15	S5	A Sensor Value	0 to 52767
16	EQUIPMENT_FAILURE	A '1' means that the equipment failed. A '0' means the equipment did not fail.	0 to 1

Analysis and Findings

Question 1. How does the quality of input data, including issues such as outliers, and noise, impact the accuracy and reliability of machine failure prediction models in industrial settings?

The dataset on machine failures, sourced from GitHub, offers a comprehensive view by amalgamating details about a machine's initial operating conditions and sensor readings. Factors like the machine's age, regional cluster, and its manufacturer can be pivotal in forecasting machine malfunctions. Concurrently, sensor data sheds light on the machine's operational state, highlighting patterns indicative of impending breakdowns. The fusion of multiple datasets is essential for crafting precise machine learning models that predict equipment malfunctions. Integrating diverse information, such as initial conditions and sensor readings, equips maintenance crews with the ability to detect potential problems early, aiding in averting expensive repairs.

Analyzing qualitative data, statistical tools can assist in comprehending data distribution and discerning patterns or trends (Leech & Onwuegbuzie, 2007). The techniques used in data analysis differ for qualitative and quantitative data.

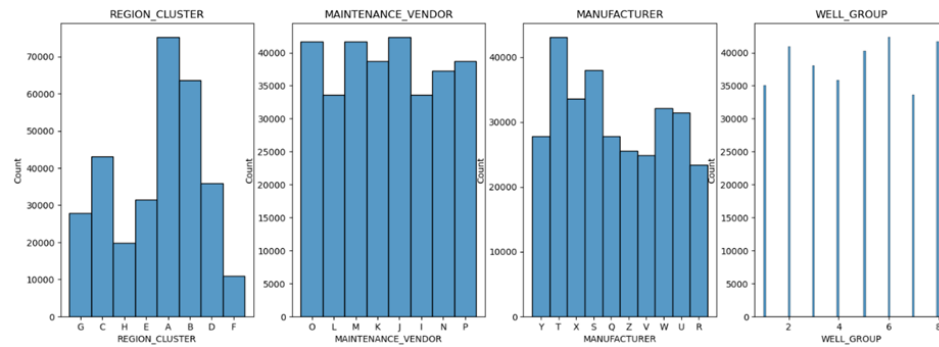


Figure 5. Qualitative Data

The prevalence of specific types of machine malfunctions or detect frequently occurring issues, providing a clearer picture of recurrent patterns or potential anomalies. When we shift our focus to quantitative data, such as metrics detailing machine uptime, durations of downtime, or specific sensor measurements, we occasionally encounter outliers. These anomalous values can skew the analysis, potentially leading to imprecise conclusions about machine failures.

	count	mean	std	min	25%	50%	75%	max
ID	307751.0	100310.826603	177.574390	100001.00	100161.000000	100311.000000	100467.000000	100617.00
WELL_GROUP	307751.0	4.543943	2.284121	1.00	3.000000	5.000000	6.000000	8.00
S15	307751.0	14.585192	8.817056	0.00	7.694100	11.661600	22.560000	59.04
S17	307751.0	80.265541	85.804273	0.00	0.000000	31.680000	160.080000	2555.52
S13	307751.0	35.018249	14.446585	0.00	28.200000	34.940000	41.610000	592.89
S5	307751.0	4675.848252	2521.074632	0.00	3209.000000	4237.047619	5743.000000	52767.00
S16	307751.0	7.972097	2.321949	0.00	6.621500	8.004000	9.460000	24.60
S19	307751.0	9.069123	16.898887	0.00	0.900000	4.200000	10.600000	511.00
S18	307751.0	137.963064	238.890128	0.00	11.798276	38.200000	150.900000	4151.70
EQUIPMENT_FAILURE	307751.0	0.001368	0.036961	0.00	0.000000	0.000000	0.000000	1.00
S8	307751.0	144.665715	240.773926	-16.49	9.250000	53.080000	165.092608	2068.11
AGE_OF_EQUIPMENT	307751.0	2524.192399	3158.930976	0.00	721.000000	1113.000000	2784.000000	15170.00

Figure 6. Quantitative Data Information

Outliers were identified and addressed by replacing them with the mean of the dataset to maintain a representative average value. This method helps prevent the replaced value from being affected by other potential outliers. In cases with frequent outliers or where removing them would lead to significant data loss, data transformation techniques were applied to the dataset.

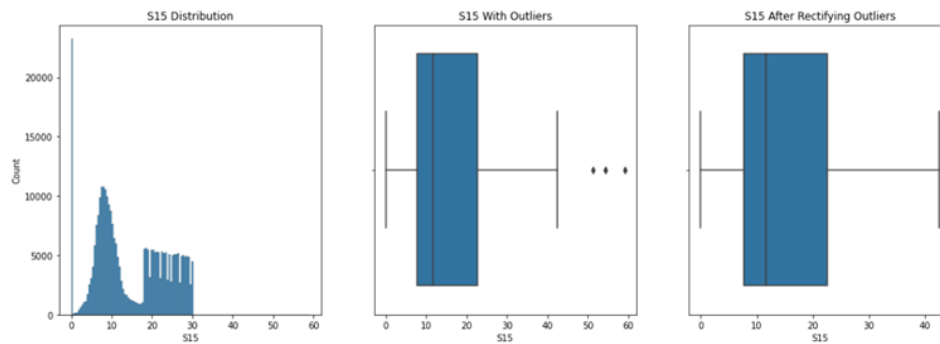


Figure 7. Data Distribution, Outliers After Rectifying Outliers S15

The S15 variable's distribution initially shows a right-skewed pattern. After rectifying outliers, the distribution becomes more centralized and less spread, indicating a successful adjustment.

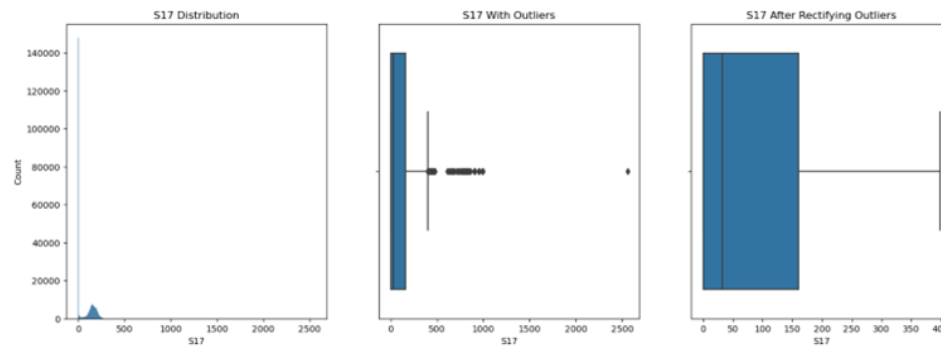


Figure 8. Data Distribution, Outliers After Rectifying Outliers S17

Outlier adjustment for the S17 variable results in a narrower and more consistent data distribution, indicating improved data quality for analysis.

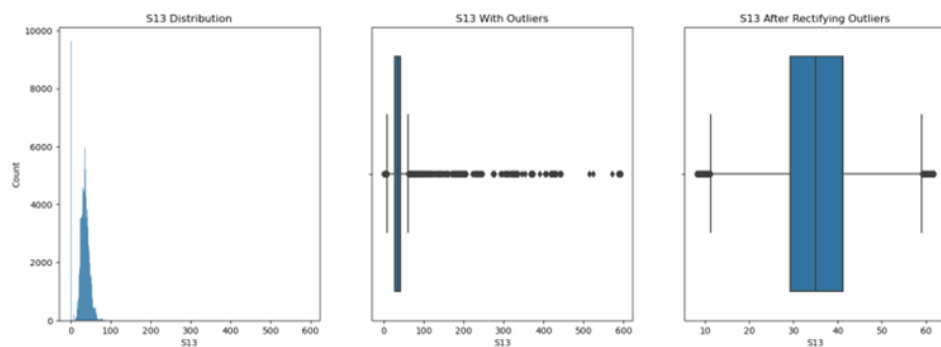


Figure 9. Data Distribution, Outliers After Rectifying Outliers S13

The distribution of the S13 variable tightens significantly after outliers are rectified, indicating a more uniform dataset for subsequent analysis.

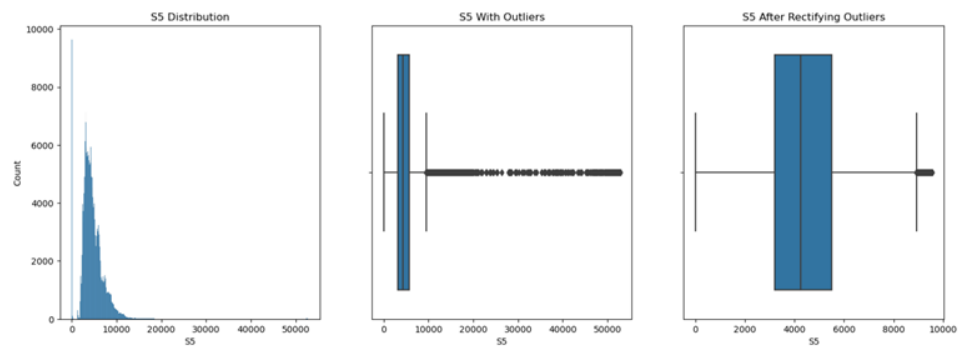


Figure 10. Data Distribution, Outliers After Rectifying Outliers S5

Post-outlier rectification, the S5 variable shows a more centralized distribution, indicating a reduction of extreme values.

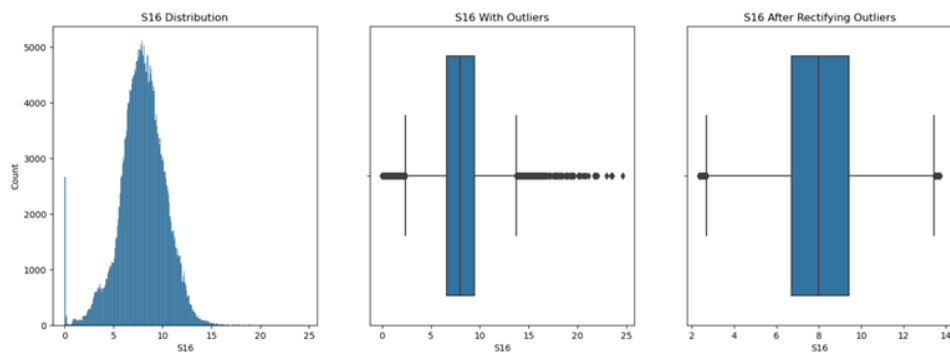


Figure 11. Data Distribution, Outliers After Rectifying Outliers S16

The S16 variable's distribution is refined, and outliers are reduced, resulting in a more normalized data representation.

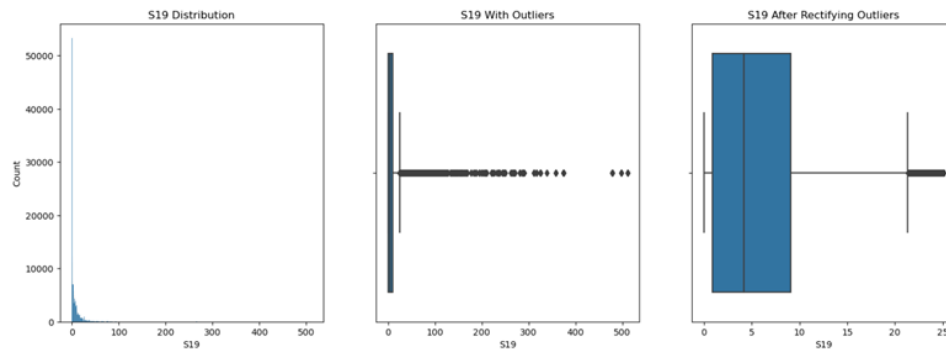


Figure 12. Data Distribution, Outliers After Rectifying Outliers S19

The S19 variable's distribution is significantly concentrated after outlier.

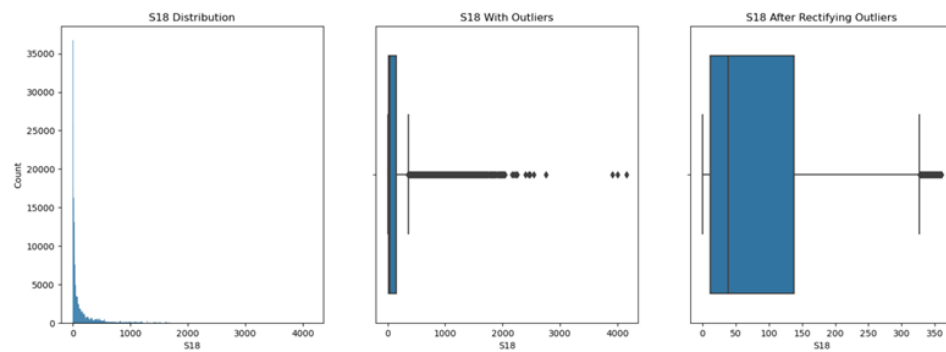


Figure 13. Data Distribution, Outliers After Rectifying Outliers S18

After outliers are addressed, the S18 variable displays a more uniform and tighter distribution, suggesting enhanced data stability.

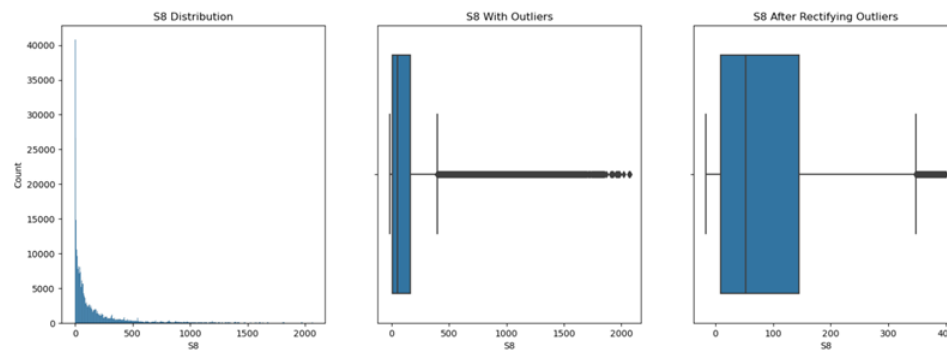


Figure 14. Data Distribution, Outliers After Rectifying Outliers S8

The S8 variable's distribution becomes noticeably more centralized after outlier correction, showing a reduction in data variance.

Question 2. How does the integration of SMOTE with feature engineering techniques influence the overall performance of machine learning models in detecting and preventing machine failures?

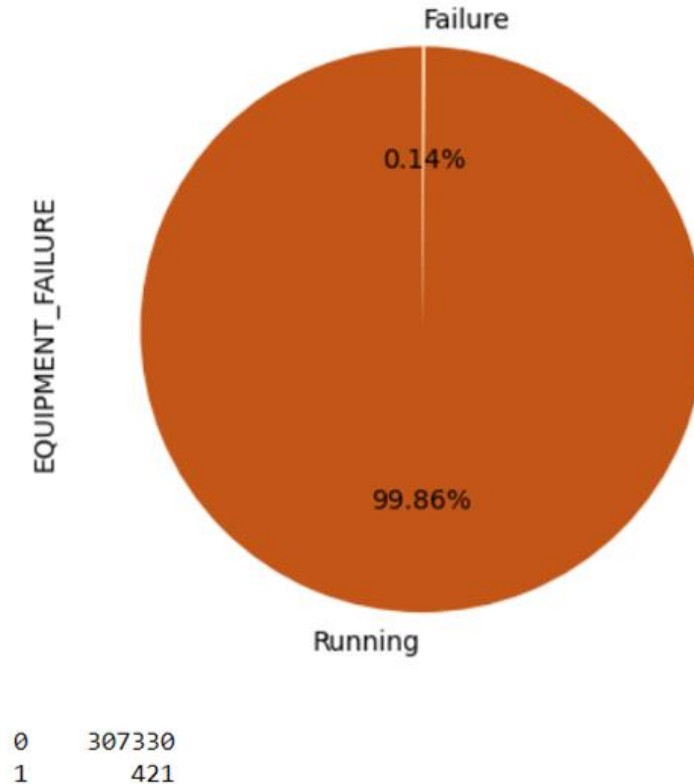


Figure 15. Equipment Failure Information

Oversample the training data using SMOTE and trains with different machine learning algorithms. The classification report provides important metrics such as precision, recall, and f1-score for each class in the target variable. It also provides an overall accuracy score for the model. This report can be used to evaluate the performance of the classification model and identify areas for improvement.

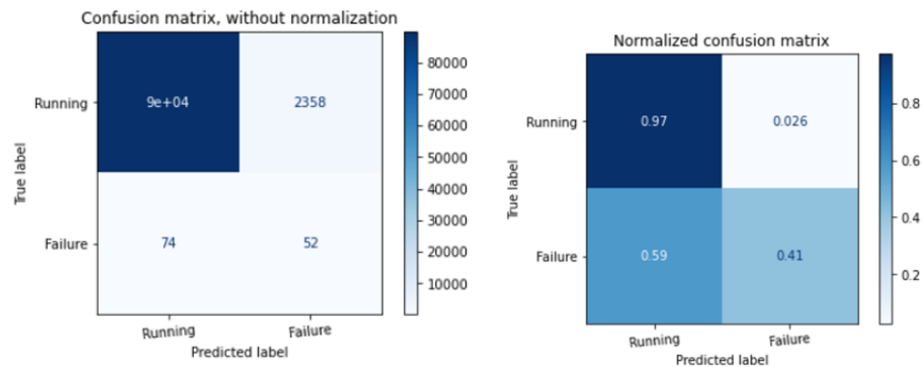


Figure 16. Confusion Matrix of Support Vector Machine

	precision	recall	f1-score	support
0	1.00	0.97	0.99	92200
1	0.02	0.41	0.04	126
accuracy			0.97	92326
macro avg	0.51	0.69	0.51	92326
weighted avg	1.00	0.97	0.99	92326

Figure 17. Classification Report

The confusion matrix and classification report indicate that the predictive model has a high true positive rate for 'Running' equipment, correctly identifying 90,000+ instances. However, it struggles with 'Failure' predictions, with a true positive rate of only 41%. This is further reflected in the precision and recall scores for failure prediction, which are 0.02 and 0.41, respectively, leading to a low F1-score of 0.04 for failures. While the overall accuracy of the model is high

(0.97), the low F1-score for failures suggests that the model is not effective at identifying actual failures, which could be due to the imbalanced nature of the dataset (92,200 running vs. 126 failure instances). The weighted averages indicate that while the model is reliable for the majority class ('Running'), it lacks precision and recall for the critical minority class ('Failure'), emphasizing the need for improved balance handling in the dataset.

Encoding techniques like one-hot and label encoding, facilitating the conversion of categorical data into a numerical format suitable for machine learning (Pau Rodríguez, Miguel A. Bautista, 2018). The selection of target and feature variables and partitioning data into training and test sets are also emphasized as vital steps in constructing and assessing machine learning models.

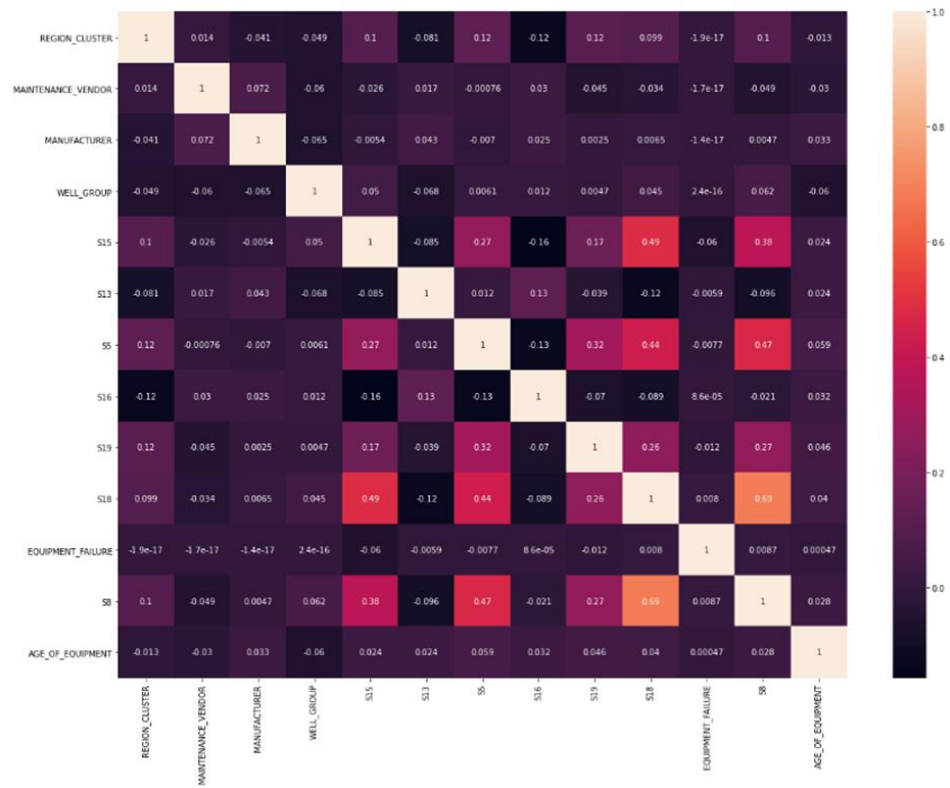


Figure 18. Importance of the Features Based on Correlation.

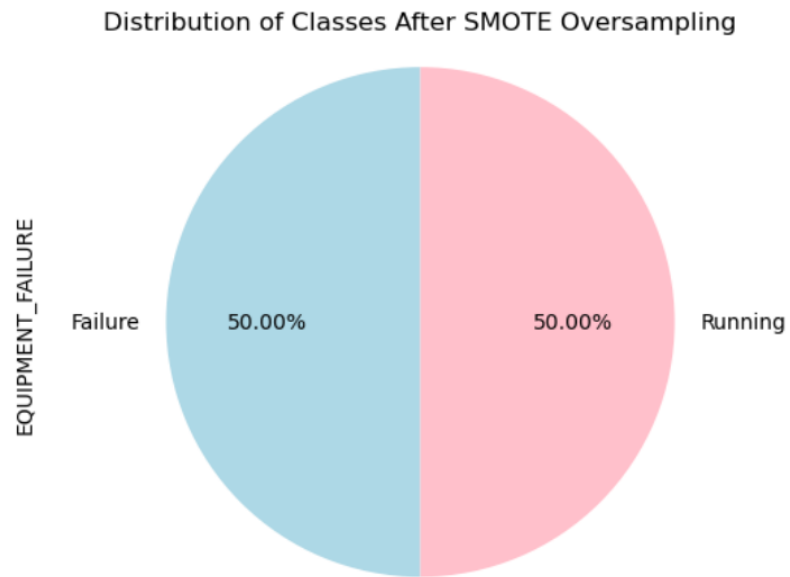


Figure 19. Using SMOTE for Balancing Data

Question 3. What is the performance of different machine learning algorithms in predicting machine failures, and which algorithm is the most effective?

The different machine learning algorithms and classification report provides important metrics such as precision, recall, and f1-score for each class in the target variable. It also provides an overall accuracy score for the model. This report can be used to evaluate the performance of the classification model and identify areas for improvement.

Random Forest:

	precision	recall	f1-score	support
0	1.00	0.94	0.97	92200
1	0.02	0.73	0.03	126
accuracy			0.94	92326
macro avg	0.51	0.84	0.50	92326
weighted avg	1.00	0.94	0.97	92326

Figure 20. Classification Report of Random Forest

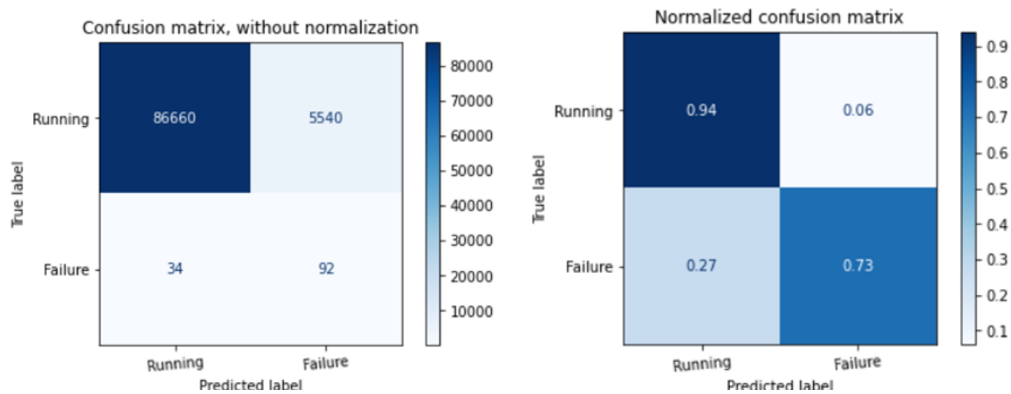


Figure 21. Confusion Matrix of Without and With Normalization of Random Forest

An examination of the confusion matrix and classification report provides valuable insights into the predictive model's performance. Notably, the model

demonstrates a high true positive rate (73%) for 'Running' equipment, correctly identifying a substantial number of instances. However, the model faces substantial challenges when predicting 'Failure' instances, with a lower true positive rate (73%), resulting in low precision (0.02) and recall (0.73) for 'Failure' predictions. This is further reflected in the low F1-Score (0.03) for 'Failure,' indicating a trade-off between precision and recall. Despite a commendable overall accuracy of 94%, the class imbalance within the dataset (92,200 'Running' instances vs. 126 'Failure' instances) remains a contributing factor to the model's limitations.

Support Vector Classifier:

	precision	recall	f1-score	support
0	1.00	0.97	0.99	92200
1	0.02	0.41	0.04	126
accuracy			0.97	92326
macro avg	0.51	0.69	0.51	92326
weighted avg	1.00	0.97	0.99	92326

Figure 22. Classification Report of Support Vector Machine

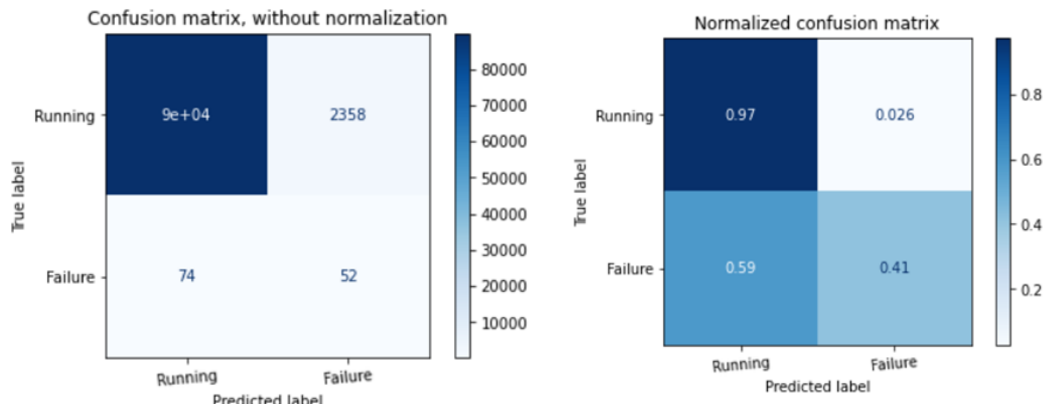


Figure 23. Confusion Matrix of Without and With Normalization of SVM

The analysis of the confusion matrix and classification report reveals crucial insights into the predictive model's performance. The model exhibits a robust true positive rate for 'Running' equipment, correctly identifying more than 90,000 instances, highlighting its effectiveness in classifying 'Running' instances. However, it faces significant challenges when predicting 'Failure' instances, with a true positive rate of only 41%, leading to low precision (0.02) and recall (0.41) for 'Failure' predictions. The model's low F1-Score of 0.04 for 'Failure' underscores the trade-off between precision and recall and the limitations in 'Failure' prediction. Despite an impressive overall accuracy of 97%, the dataset's class imbalance (92,200 'Running' instances vs. 126 'Failure' instances) contributes to these challenges.

XG Boost:

	precision	recall	f1-score	support
0	1.00	0.79	0.88	92200
1	0.01	1.00	0.01	126
accuracy			0.79	92326
macro avg	0.50	0.89	0.45	92326
weighted avg	1.00	0.79	0.88	92326

Figure 24. Classification Report of XG Boost

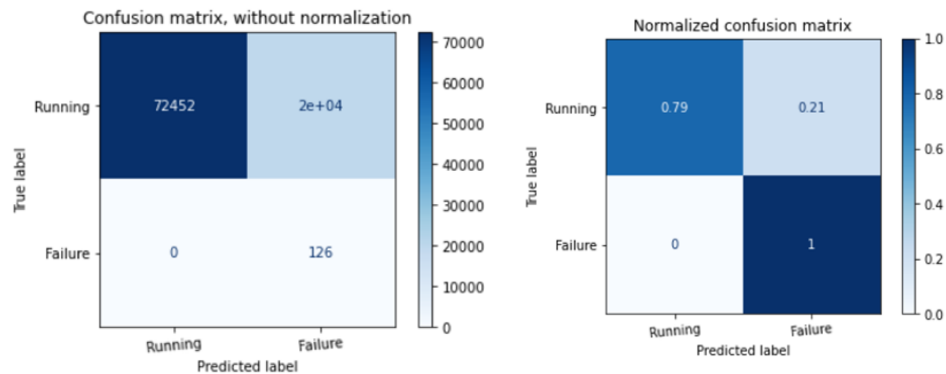


Figure 25. Confusion Matrix of Without and With Normalization of XG Boost

An in-depth analysis of the confusion matrix and classification report provides critical insights into the model's performance. The model demonstrates a 100%

true positive rate for 'Running' equipment, correctly identifying a substantial number of instances in this class. However, it faces significant challenges when predicting 'Failure' instances, as reflected in the 0% true positive rate for 'Failure,' leading to low precision (0.01) and recall (1.00) for 'Failure' predictions. This is further evidenced by the low F1-Score (0.01) for 'Failure,' underlining the trade-off between precision and recall. Despite an overall accuracy of 79%, the severe class imbalance within the dataset (92,200 'Running' instances vs. 126 'Failure' instances) significantly affects the model's effectiveness.

Cat Boost:

	precision	recall	f1-score	support
0	1.00	0.92	0.96	92200
1	0.02	0.99	0.03	126
accuracy			0.92	92326
macro avg	0.51	0.96	0.50	92326
weighted avg	1.00	0.92	0.96	92326

Figure 26. Classification Report of CatBoost

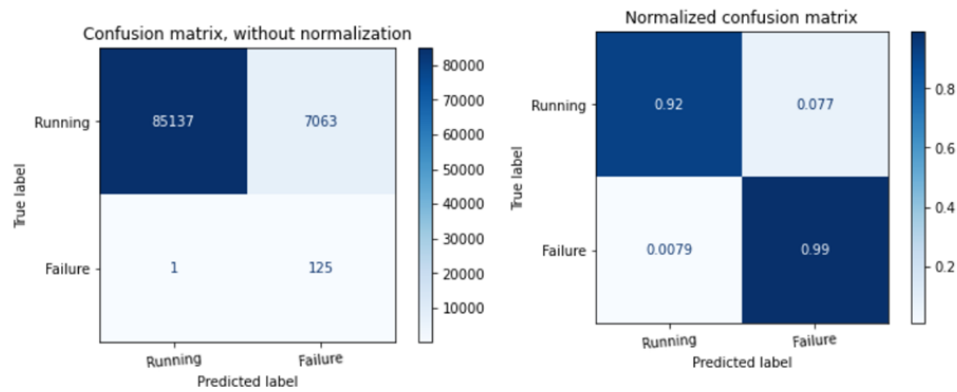


Figure 27. Confusion Matrix of Without and With Normalization of CatBoost

An in-depth examination of the confusion matrix and classification report sheds light on the model's performance. The model demonstrates an impressive true positive rate of 99% for 'Running' equipment, correctly identifying the majority of instances in this class. However, it faces challenges when predicting 'Failure' instances, as indicated by the 0.02 precision and 0.99 recall for 'Failure' predictions. The model achieves a low F1-Score of 0.03 for 'Failure,' illustrating a trade-off between precision and recall. Despite an overall accuracy of 92%, the substantial class imbalance within the dataset (92,200 'Running' instances vs. 126 'Failure' instances) continues to impact the model's performance.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

Discussion

In this project, we are addressing the substantial issue of machine failures in automated industries, which can lead to significant costs and operational downtime. We focus on enhancing the accuracy and reliability of machine failure prediction models through advanced machine learning algorithms, with the aim of seamlessly integrating them into the realm of automation. Our research questions delve into data quality, the impact of integrating SMOTE with feature engineering, and the performance of different machine learning algorithms in predicting machine failures, ultimately aiming to optimize predictive maintenance and reduce the disruptions caused by unexpected machinery breakdowns in industrial settings.

Conclusion

In conclusion, our investigation into predictive maintenance and machine failure prediction has unearthed essential insights. Question 1's examination of sensor parameters underscored the critical role of addressing outliers during data preprocessing, which consistently improved model accuracy (Figure 6-14). The fascinating correlation between Sensor S18 and S8 offers avenues for further

exploration, while negative correlations among other sensors reveal the dataset's complexity (Figure 18). To optimize predictive capabilities, exploring alternative outlier rectification techniques and assessing their impact on model accuracy is essential. The pursuit of perfecting machine failure prediction is ongoing, and these methodologies will play a pivotal role in enhancing predictive outcomes. Question 2 focuses on addressing the issue of equipment failure parameter imbalance. It was found in the research findings that there was a significant imbalance in the failure data, with only 0.14% of the dataset representing actual failures and 99.86% of the dataset pertaining to non-failure data (Figure 15). This extreme class disparity can result in biased models that underperform on underrepresented classes, which is a common problem in machine learning.

We used the Synthetic Minority Over-sampling Technique (SMOTE) to rectify the imbalance. In the training data, SMOTE was utilized to create artificial instances of the minority class—in this case, equipment failure instances. Through this process, the dataset was effectively rebalanced, giving the model access to a more equitable representation of both failure and non-failure scenarios for learning purposes. SMOTE successfully tackled the problem of class imbalance in machine learning and increased the model's capacity for accurate prediction (Figure 19).

In Question 3, we examined various Machine Learning algorithms and observed significant variations in model performance. The Catboost model stood out with remarkable accuracy and failure detection rates with 92% accuracy and

failure detection 99% correct (Figure 26-27). In contrast, the Random Forest model delivered a 94% accuracy rate and a 73% failure recall rate (Figure 20-21). The Support Vector Machine model excelled with a 97% accuracy rate but had a lower failure recall rate of 41% (Figure 22-23). Meanwhile, the XGBoost model achieved a remarkable 100% failure recall rate but the accuracy rate of only 79% (Figure 24-25). Finally, the CatBoost model, known for its accuracy, duplicated the 92% accuracy rate and 99% failure recall rate, showcasing its consistency and effectiveness in the study. As we move forward, experimenting with diverse data types and refining data balancing techniques is crucial for further fine-tuning our models and achieving even more reliable predictions in the future.

Areas for Further Study

Our research lays the groundwork for future studies in predictive maintenance. Building upon our findings, future research can explore advanced outlier rectification techniques, including median substitution and normalization, to further optimize predictive models. Investigating the intricate relationships among sensors and data types can provide additional insights to enhance predictive accuracy. Furthermore, there is potential for future studies to delve deeper into data scaling and balancing methods, striving to achieve the most effective techniques that can be tailored to specific industrial contexts. Additional research can focus on addressing the challenge of rare failure events and

improving the overall performance of predictive models in such scenarios. The exploration of machine learning algorithms should continue, encompassing a wider array of algorithms and data types to identify the best-performing models for diverse industrial applications. As predictive maintenance evolves, it is imperative to stay at the forefront of technology and data science advancements, ensuring that the predictive models become even more reliable and efficient in preventing machine failures.

APPENDIX A

DATASET

DATASET

s.no	Attribute name	Description	Range of value
1	ID	ID field that represents a specific machine	100001 to 100617
2	DATE	The date of the observation.	731
3	REGION_CLUSTER	A field that represents the region in which the machine resides.	8
4	MAINTENANCE_VENDOR	A field that represents the company that provides maintenance and service to the machine	8
5	MANUFACTURER	The company that manufactured the equipment in question.	10
6	WELL, _GROUP	A field representing the type of machine	1 to 8
7	EQUIPMENT_AGE	Age of the machine, in days.	0 to 15170
8	S15	A Sensor Value	0 to 59.04
9	S17	A Sensor Value	0 to 2555.52
10	S13	A Sensor Value.	0 to 592.89
11	S16	A Sensor Value.	0 to 24.6
12	S19	A Sensor Value.	0 to 511
13	S18	A Sensor Value	0 to 4151.7

14	S8	A Sensor Value.	-16.49 to 2068.11
15	S5	A Sensor Value	0 to 52767
16	EQUIPMENT_FAILURE	A '1' means that the equipment failed. A '0' means the equipment did not fail.	0 to 1

APPENDIX B

CODE

Importing Necessary Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn import preprocessing
import sklearn
```

Loading First Half DataSet

```
df1=pd.read_csv('first_half.txt')
df1.head(3)
```

Shape of the First Half DataSet

```
df1.shape
```

Loading Second Half DataSet

```
df2=pd.read_csv('second_half.txt')
df2.head(2)
```

Shape of the Second Half DataSet

```
df1.shape
```

Concatinating df1 and df2 datasets

```
df=pd.concat([df1,df2],ignore_index=True)
df.head(3)
```

Shape of the Concatenate Dataset of Df1 and Df2

```
df.shape
```

Finding Whole Data Set information

```
df.info()
```

Finding the Null values in the Data Sets

```
df.isnull().sum()
```

Displaying the Column Names

```
df.columns
```

Statistics Analysis of the qualitative Data

```
df.describe(include='object')
```

```
plt.figure(figsize=(12,12))

plt.subplot(2, 2, 1)
sns.histplot(df, x="REGION_CLUSTER")
plt.title("REGION_CLUSTER")

plt.subplot(2, 2, 2)
sns.histplot(df, x="REGION_CLUSTER")
plt.title("REGION_CLUSTER")

plt.subplot(2, 2, 3)
sns.histplot(df, x="MANUFACTURER")
plt.title("MANUFACTURER")

plt.subplot(2, 2, 4)
sns.histplot(df, x="WELL_GROUP")
plt.title("WELL_GROUP")

plt.suptitle("Categorical Data Distribution")
plt.show()
```

Statistic Analysis of the Quantitative Data

```
df.describe()
```

Rectifying the Outliers

```
def mea(df,col):    #to replace null values with median

    q1= df[col].quantile(0.25)
    q3= df[col].quantile(0.75)
    IQR= q3-q1
    upper= q3+(1.5*IQR)
    lower= q1-(1.5*IQR)
    mean= df[col].mean()
    df[col]= np.where(df[col]>upper,mean,np.where(df[col]<lower,mean,df[col]))
    sns.boxplot(data=df,x=df[col])
```

```

plt.figure(figsize=(18,6))

plt.subplot(1, 3, 1)
sns.histplot(data=df, x="S15")
plt.title("S15 Distribution")

plt.subplot(1, 3, 2)
sns.boxplot(data=df, x="S15")
plt.title("S15 With Outliers")

plt.subplot(1, 3, 3)
mea(df, 'S15')
plt.title("S15 After Rectifying Outliers")

plt.suptitle("Numerical Data Distribution")
plt.show()

```

```

plt.figure(figsize=(18,6))

plt.subplot(1, 3, 1)
sns.histplot(data=df, x="S17")
plt.title("S17 Distribution")

plt.subplot(1, 3, 2)
sns.boxplot(data=df, x="S17")
plt.title("S17 With Outliers")

plt.subplot(1, 3, 3)
mea(df, 'S17')
plt.title("S17 After Rectifying Outliers")

plt.suptitle("Numerical Data Distribution")
plt.show()

```

```

plt.figure(figsize=(18,6))

plt.subplot(1, 3, 1)
sns.histplot(data=df, x="S13")
plt.title("S13 Distribution")

plt.subplot(1, 3, 2)
sns.boxplot(data=df, x="S13")
plt.title("S13 With Outliers")

plt.subplot(1, 3, 3)
mea(df, 'S13')
plt.title("S13 After Rectifying Outliers")

plt.suptitle("Numerical Data Distribution")
plt.show()

```

```

plt.figure(figsize=(18,6))

plt.subplot(1, 3, 1)
sns.histplot(data=df, x="S5")
plt.title("S5 Distribution")

plt.subplot(1, 3, 2)
sns.boxplot(data=df, x="S5")
plt.title("S5 With Outliers")

plt.subplot(1, 3, 3)
mea(df, 'S5')
plt.title("S5 After Rectifying Outliers")

plt.suptitle("Numerical Data Distribution")
plt.show()

```

```

plt.figure(figsize=(18,6))

plt.subplot(1, 3, 1)
sns.histplot(data=df, x="S16")
plt.title("S16 Distribution")

plt.subplot(1, 3, 2)
sns.boxplot(data=df, x="S16")
plt.title("S16 With Outliers")

plt.subplot(1, 3, 3)
mea(df, 'S16')
plt.title("S16 After Rectifying Outliers")

plt.suptitle("Numerical Data Distribution")
plt.show()

```


Target Value Analysis

```
➤ sns.countplot(data=df, x="EQUIPMENT_FAILURE")  
plt.show()
```

```
➤ df['EQUIPMENT_FAILURE'].value_counts()
```

Dropping Unnecessary Columns

```
➤ # we dropped 'S17' because value 0 is more than 305 i.e. 45%.  
df.drop(['ID', 'DATE', 'S17'], axis = 1, inplace = True)  
df.head(2)
```

Encoding Method to convert character to number

```
➤ label_encoder=preprocessing.LabelEncoder()  
df['REGION_CLUSTER']=label_encoder.fit_transform(df['REGION_CLUSTER'])  
df['MAINTENANCE_VENDOR']=label_encoder.fit_transform(df['MAINTENANCE_VENDOR'])  
df['MANUFACTURER']=label_encoder.fit_transform(df['MANUFACTURER'])  
df.head(2)
```

checking correlation.

```
➤ plt.figure(figsize=(20,15))  
a=sns.heatmap(df.corr(),annot=True)
```

Selecting Target and Feature

```
➤ x=df.drop(['EQUIPMENT_FAILURE'], axis=1)  
y=df['EQUIPMENT_FAILURE']
```

Splitting Data into Train and Test Based on 70 and 30%

```
➤ #doing Test Train.  
from sklearn.model_selection import train_test_split  
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0,stratify=y)
```

Applying Standard Scaler on Features

```
➤ # feature scaling  
from sklearn.preprocessing import StandardScaler  
sc=StandardScaler()  
x_train=sc.fit_transform(x_train)  
x_test=sc.fit_transform(x_test)
```

Applying SMOTE to convert imbalance to balance data

```
# from imblearn.over_sampling import SMOTE
oversampled = SMOTE()
x_train, y_train = oversampled.fit_resample(x_train, y_train)

# Importing SMOTE
from imblearn.over_sampling import SMOTE
# Instantiate SMOTE
oversampled = SMOTE(random_state=27)
# Fitting SMOTE to the train set
X_train_smote, y_train_smote = oversampled.fit_resample(x_train, y_train)

# print('Before SMOTE oversampling X_train shape=', x_train.shape)
print('After SMOTE oversampling X_train shape=', X_train_smote.shape)

print("\nSMOTE Data of two classes")
print(y_train_smote.value_counts())

y_train_smote.value_counts().plot(kind='pie', autopct="%.2f%%", labels=['Failure', 'Running'],
                                  startangle = 90, colors = ['#ADD8E6', '#FFC0CB'])

plt.title('Distribution of Classes After SMOTE Oversampling')
plt.axis('equal')
plt.show();
```

Model

1. Random Forest Regression

```
# from sklearn.ensemble import RandomForestClassifier
rf_classifier = RandomForestClassifier(n_estimators = 10, random_state = 24)
rf_classifier.fit(x_train, y_train)
```

Calculating Evaluation Metrics on Model

```
# y_pred = rf_classifier.predict(x_test)
# Classification Model
from sklearn.metrics import classification_report
print("\nClassification Report: \n")
target_names = ['Running', 'Failure']
print(classification_report(y_test, y_pred))

# Evaluation Matrix
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay
# Confusion Matrix
class_names = ['Running', 'Failure']
print("\n Confusion matrix: \n")
title_options = [
    ("Confusion matrix, without normalization", None),
    ("Normalized confusion matrix", "true")]
for title, normalize in title_options:
    display = ConfusionMatrixDisplay.from_estimator(
        rf_classifier,
        x_test,
        y_test,
        display_labels=class_names,
        xticks_rotation=5,
        cmap=plt.cm.Blues,
        normalize=normalize,
    )
    display.ax_.set_title(title)
plt.show()

df_pred = pd.DataFrame({'Actual': y_test, 'Predict': y_pred})
df_pred.head()
```

2. Support Vector Machines

```
from sklearn.svm import SVC
rf_Classifier = SVC()
rf_Classifier.fit(x_train,y_train)
```

Calculating Evaluation Metrics on Model

```
y_pred = rf_Classifier.predict(x_test)
# Classification Model
from sklearn.metrics import classification_report
print("\033[1m Classification Report: \033[0m \n")
target_names = ['Running', 'Failure']
print(classification_report(y_test, y_pred))

# Evaluation Matrix
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay
# Confusion Matrix
class_names = ['Running', 'Failure']
print('\n \033[1m Confusion matrix: \033[0m \n \n')
title_options = [
    ("Confusion matrix, without normalization", None),
    ("Normalized confusion matrix", "true")]
for title, normalize in title_options:
    display = ConfusionMatrixDisplay.from_estimator(
        rf_Classifier,
        x_test,
        y_test,
        display_labels=class_names,
        xticks_rotation=5,
        cmap=plt.cm.Blues,
        normalize=normalize,
    )
    display.ax_.set_title(title)
plt.show()

df_pred = pd.DataFrame({'Actual':y_test,'Predict':y_pred})
df_pred.head()
```

3. XGBoost

```
from xgboost import XGBClassifier
xg_Classifier = XGBClassifier()
xg_Classifier.fit(x_train,y_train)
```

Calculating Evaluation Metrics on Model

```
y_pred = xg_Classifier.predict(x_test)
# Classification Model
from sklearn.metrics import classification_report
print("\033[1m Classification Report: \033[0m \n")
target_names = ['Running', 'Failure']
print(classification_report(y_test, y_pred))

# Evaluation Matrix
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay
# Confusion Matrix
class_names = ['Running', 'Failure']
print('\n \033[1m Confusion matrix: \033[0m \n \n')
title_options = [
    ("Confusion matrix, without normalization", None),
    ("Normalized confusion matrix", "true")]
for title, normalize in title_options:
    display = ConfusionMatrixDisplay.from_estimator(
        xg_Classifier,
        x_test,
        y_test,
        display_labels=class_names,
        xticks_rotation=5,
        cmap=plt.cm.Blues,
        normalize=normalize,
    )
    display.ax_.set_title(title)
plt.show()

df_pred = pd.DataFrame({'Actual':y_test,'Predict':y_pred})
df_pred.head()
```

4. CatBoost

```
M cat_features = list(range(0, x_train.shape[1]))
print(cat_features)

M from catboost import CatBoostClassifier

cat_Classifier = CatBoostClassifier(
    iterations=5,
    learning_rate=0.1,
    #Loss_function='CrossEntropy'
)

cat_Classifier.fit(
    x_train, y_train,
    eval_set=(x_test, y_test),
)
```

Calculating Evaluation Metrics on Model

```
M y_pred = cat_Classifier.predict(x_test)
# Classification Model
from sklearn.metrics import classification_report
print("\033[1m Classification Report: \033[0m \n")
target_names = ['Running', 'Failure']
print(classification_report(y_test, y_pred))

# Evaluation Matrix
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay
# Confusion Matrix
class_names = ['Running', 'Failure']
print('\n \033[1m Confusion matrix: \033[0m \n \n')
title_options = [
    ("Confusion matrix, without normalization", None),
    ("Normalized confusion matrix", "true")]
for title, normalize in title_options:
    display = ConfusionMatrixDisplay.from_estimator(
        cat_Classifier,
        x_test,
        y_test,
        display_labels=class_names,
        xticks_rotation=5,
        cmap=plt.cm.Blues,
        normalize=normalize,
    )
    display.ax_.set_title(title)
plt.show()

df_pred = pd.DataFrame({'Actual':y_test, 'Predict':y_pred})
df_pred.head()
```

REFERENCES

- Abidi, M. H., Mohammed, M. K., & Alkhalefah, H. (2022). Predictive Maintenance Planning for Industry 4.0 using machine learning for sustainable manufacturing. *Sustainability*, 14(6), 3387.
<https://doi.org/10.3390/su14063387>
- Akbulut, U., Çifçi, M. A., & Aslan, Z. (2023). Hybrid modeling for stream flow estimation: integrating machine learning and federated learning. *Applied Sciences*, 13(18), 10203. <https://doi.org/10.3390/app131810203>
- Angelopoulos, A., Michailidis, E. T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S., & Zahariadis, T. (2019b). Tackling Faults in the Industry 4.0 Era—A Survey of Machine-Learning Solutions and Key Aspects. *Sensors*, 20(1), 109.
<https://doi.org/10.3390/s20010109>
- Anumbe, N., Saidy, C., & Harik, R. (2022). A primer on the factories of the future. *Sensors*, 22(15), 5834. <https://doi.org/10.3390/s22155834>
- Balamurugan, E., Flaih, L. R., Yuvaraj, D., Sangeetha, K., Jayanthiladevi, A., & Kumar, T. S. (2019). Use Case of Artificial Intelligence in Machine Learning Manufacturing 4.0. *IEEE*.
<https://doi.org/10.1109/iccike47802.2019.9004327>
- Bouramtane, T., Leblanc, M., Kacimi, I., Ouatiki, H., & Boudhar, A. (2023). The contribution of remote sensing and input feature selection for groundwater

- level prediction using LSTM neural networks in the Oum Er-Rbia Basin, Morocco. *Frontiers in Water*, 5. <https://doi.org/10.3389/frwa.2023.1241451>
- Brock, J. K., & Von Wangenheim, F. (2019). Demystifying AI: What Digital Transformation Leaders Can Teach You about Realistic Artificial Intelligence. *California Management Review*, 61(4), 110–134. <https://doi.org/10.1177/1536504219865226>.
- Cavalieri, S., & Salafia, M. G. (2020). A model for predictive maintenance based on asset administration Shell. *Sensors*, 20(21), 6028. <https://doi.org/10.3390/s20216028>.
- Chanda, S. S., & Banerjee, D. N. (2022). Omission and commission errors underlying AI failures. *AI & Society*. <https://doi.org/10.1007/s00146-022-01585-x>.
- Chen, J., Kudjo, P. K., Mensah, S., Brown, S. A., & Akorfu, G. (2020). An automatic software vulnerability classification framework using term frequency-inverse gravity moment and feature selection. *Journal of Systems and Software*, 167, 110616. <https://doi.org/10.1016/j.jss.2020.110616>
- Chen, Y., & Zhang, Y. (2018). Predictive maintenance using machine learning algorithms: A review. *Journal of Manufacturing Systems*, 47, 120-130. <https://doi.org/10.1016/j.jmsy.2017.12.005>

Costa, M. A., Wulft, B., Norrlöf, M., & Gunnarsson, S. (2019). Failure detection in robotic arms using statistical modeling, Machine Learning and hybrid gradient boosting. *Measurement*, 146, 425–436.

<https://doi.org/10.1016/j.measurement.2019.06.039>

Cross, J. A. (2015, December 1). An adaptive ARX model to estimate an asset remaining useful life. <http://hdl.handle.net/2346/66152>.

Dalzochio, J., Kunst, R., De Freitas, E. P., Binotto, A. P. D., Sanyal, S., Favilla, J. R., & Barbosa, J. L. V. (2020);” Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges” *Computers in Industry*, 123, 103298.

<https://doi.org/10.1016/j.compind.2020.103298>

De Jesus, G., Casimiro, A., & Oliveira, A. (2021). Using machine learning for dependable outlier detection in environmental monitoring systems. *ACM Transactions on Cyber-Physical Systems*, 5(3), 1–30.

<https://doi.org/10.1145/3445812>

Duan, F., Zhang, S., Yan, Y., & Cai, Z. (2022). An oversampling method of unbalanced data for mechanical fault diagnosis based on MeanRadius-SMOTE. *Sensors*, 22(14), 5166. <https://doi.org/10.3390/s22145166>

Erhan, L., Ndubuaku, M. U., Di Mauro, M., Song, W., Chen, M., Fortino, G., Bagdasar, O., & Liotta, A. (2021). Smart anomaly detection in sensor

systems: A multi-perspective review. *Information Fusion*, 67, 64–79.

<https://doi.org/10.1016/j.inffus.2020.10.001>

Figure 3. Support vector machine SVM [3]. (n.d.). ResearchGate.

https://www.researchgate.net/figure/Support-vector-machine-SVM-3_fig2_334365773

Florio, A. M., Martins, P. S., Schiffer, M., Serra, T., & Vidal, T. (2023). Optimal decision diagrams for classification. *Proceedings of the . . . AAAI Conference on Artificial Intelligence*, 37(6), 7577–7585.

<https://doi.org/10.1609/aaai.v37i6.25920>

Gaddam, A., Wilkin, T., Angelova, M., & Gaddam, J. (2020). Detecting sensor faults, anomalies and outliers in the internet of Things: a survey on the challenges and solutions. *Electronics*, 9(3), 511.

<https://doi.org/10.3390/electronics9030511>

Garouani, M., Ahmad, A., Bouneffa, M. et al. Towards big industrial data mining through explainable automated machine learning. *Int J Adv Manuf Technol* 120, 1169–1188 (2022). <https://doi.org/10.1007/s00170-022-08761-9>

Goyal, D., & Pabla, B. S. (2015). Condition based maintenance of machine tools—A review. *Cirp Journal of Manufacturing Science and Technology*, 10, 24–35. <https://doi.org/10.1016/j.cirpj.2015.05.004>.

Griffin, S. (2022b, December 28). Machine learning for Equipment Failure Prediction and Predictive Maintenance (PM). *Medium*.

<https://medium.com/swlh/machine-learning-for-equipment-failure-prediction-and-predictive-maintenance-pm-e72b1ce42da1>

Hoseinitabatabaei, S. A., Gluhak, A., & Tafazolli, R. (2013). A survey on smartphone-based systems for opportunistic user context recognition. *ACM Computing Surveys*, 45(3), 1–51.

<https://doi.org/10.1145/2480741.2480744>

Huang, G., Wu, G., Yang, Z., Chen, X., & Wei, W. (2023). Development of surrogate models for evaluating energy transfer quality of high-speed railway pantograph-catenary system using physics-based model and machine learning. *Applied Energy*, 333, 120608.

<https://doi.org/10.1016/j.apenergy.2022.120608>

Illán, I. A., Górriz, J. M., Ramírez, J., Martínez-Murcia, F. J., Castillo-Barnes, D., Segovia, F., & Salas-González, D. (2019). Support vector machine failure in imbalanced datasets. In *Lecture Notes in Computer Science* (pp. 412–419). https://doi.org/10.1007/978-3-030-19591-5_42

Introduction to XGBoost in Python. (2023). Quantitative Finance & Algo Trading Blog by QuantInsti. <https://blog.quantinsti.com/xgboost-python/>

Jung, H., Jeon, J., Choi, D., & Park, J. (2021). Application of Machine learning techniques in injection molding Quality Prediction: Implications on

- Sustainable Manufacturing industry. *Sustainability*, 13(8), 4120.
<https://doi.org/10.3390/su13084120>.
- Keck, T. (2016, September 20). FastBDT: A speed-optimized and cache-friendly implementation of stochastic gradient-boosted decision trees for multivariate classification. arXiv.org. <https://arxiv.org/abs/1609.06119>
- Kim, J., Han, Y., & Lee, J. (2016). Data Imbalance Problem solving for SMOTE Based Oversampling: Study on Fault Detection Prediction Model in Semiconductor Manufacturing Process. *Advanced Science and Technology Letters*. <https://doi.org/10.14257/astl.2016.133.15>
- Konstantinidis, F. K., Myrillas, N., Mouroutsos, S. G., Koulouriotis, D., & Gasteratos, A. (2022). Assessment of industry 4.0 for Modern Manufacturing Ecosystem: A systematic survey of surveys. *Machines*, 10(9), 746.
<https://doi.org/10.3390/machines10090746>
- Ku, J. (2018). “A Study on Prediction Model of Equipment Failure through Analysis of Big Data Based on RHadoop” *Wireless Personal Communications*, 98(4), 3163–3176.
<https://doi.org/10.1007/s11277-017-4151-1>
- Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., & Siegel, D. (2014). Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications. *Mechanical Systems and Signal*

Processing, 42(1–2), 314–334.

<https://doi.org/10.1016/j.ymssp.2013.06.004>

Li, Y., Chun, Y., Liu, W., & Li, M. (2018). A principle component analysis-based random forest with the potential nearest neighbor method for automobile insurance fraud identification. *Applied Soft Computing*, 70, 1000–1009.

<https://doi.org/10.1016/j.asoc.2017.07.027>

Lokrantz, A., Gustavsson, E. K., & Jirstrand, M. (2018);" Root cause analysis of failures and quality deviations in manufacturing using machine learning" *Procedia CIRP*, 72, 1057–1062.

<https://doi.org/10.1016/j.procir.2018.03.229>

Ma, L., & Sun, B. (2020). Machine learning and AI in marketing – Connecting computing power to human insights. *International Journal of Research in Marketing*, 37(3), 481–504. <https://doi.org/10.1016/j.ijresmar.2020.04.005>

Machine Learning Random Forest Algorithm - JavatPoint. (n.d.).

www.javatpoint.com. <https://www.javatpoint.com/machine-learning-random-forest-algorithm>

Natarajan, S., & Srinivasan, R. (2010). Multi-model based process condition monitoring of offshore oil and gas production process. *Chemical Engineering Research & Design*, 88(5–6), 572–591.

<https://doi.org/10.1016/j.cherd.2009.10.013>.

Navarro, D. F., Kocaballi, A. B., Dras, M., & Berkovsky, S. (2022). Collaboration, not Confrontation: Understanding General Practitioners' Attitudes Towards Natural Language and Text Automation in Clinical Practice. *ACM Transactions on Computer-Human Interaction*.

<https://doi.org/10.1145/3569893>

Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., & Loncarski, J. (2018). Machine Learning approach for Predictive Maintenance in Industry 4.0. *IEEE*. <https://doi.org/10.1109/mesa.2018.8449150>

Qin, C., Zhang, Y., Bao, F., Zhang, C., Liu, P., & Liu, P. (2021). XGBOOST optimized by adaptive particle swarm optimization for credit scoring. *Mathematical Problems in Engineering*, 2021, 1–18.

<https://doi.org/10.1155/2021/6655510>

Rahmati, O., Kornejady, A., Samadi, M., Deo, R. C., Conoscenti, C., Lombardo, L., Dayal, K., Taghizadeh-Mehrjardi, R., Pourghasemi, H. R., Kumar, S., & Bui, D. T. (2019). PMT: New analytical framework for automated evaluation of geo-environmental modelling approaches. *Science of the Total Environment*, 664, 296–311.

<https://doi.org/10.1016/j.scitotenv.2019.02.017>

Rojek, I., Jasiulewicz-Kaczmarek, M., Piechowski, M., & Mikołajewski, D. (2023). An artificial intelligence approach for improving maintenance to supervise

- machine failures and support their repair. *Applied Sciences*, 13(8), 4971.
<https://doi.org/10.3390/app13084971>.
- Salfner, F., Lenk, M., & Malek, M. (2010b);" A survey of online failure prediction methods" *ACM Computing Surveys*, 42(3), 1–42.
<https://doi.org/10.1145/1670679.1670680>
- Shadgriffin. (n.d.-b). *GitHub - shadgriffin/machine_failure*. GitHub.
https://github.com/shadgriffin/machine_failure/tree/master
- Singh, A., Grant, B., DeFour, R., Sharma, C., & Bahadoorsingh, S. (2016). A review of Induction Motor Fault Modeling. *Electric Power Systems Research*, 133, 191–197.
<https://doi.org/10.1016/j.epsr.2015.12.017>
- Sridhar, S., & Sanagavarapu, S. (2021b). Handling Data Imbalance in Predictive Maintenance for Machines using SMOTE-based Oversampling. *IEEE*.
<https://doi.org/10.1109/cicn51697.2021.9574668>
- Subashini, T. S., Ramalingam, V., & Palanivel, S. (2009). Breast mass classification based on cytological patterns using RBFNN and SVM. *Expert Systems With Applications*, 36(3), 5284–5290.
<https://doi.org/10.1016/j.eswa.2008.06.127>

- Swana, E. F., Doorsamy, W., & Bokoro, P. N. (2022). Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset. *Sensors*, 22(9), 3246. <https://doi.org/10.3390/s22093246>
- Tagawa, Y., Maskeliūnas, R., & Damaševičius, R. (2021). Acoustic anomaly detection of mechanical failures in noisy Real-Life factory environments. *Electronics*, 10(19), 2329. <https://doi.org/10.3390/electronics10192329>
- Tiddens, W. W., Braaksma, A. J. J., & Tinga, T. (2023). Decision Framework for Predictive Maintenance Method selection. *Applied Sciences*, 13(3), 2021. <https://doi.org/10.3390/app13032021>.
- Timothy I. Salsbury," A SURVEY OF CONTROL TECHNOLOGIES IN THE BUILDING AUTOMATION INDUSTRY "[IFAC Proceedings Volumes Volume 38, Issue 1](#), 2005, Pages 90-100
<https://doi.org/10.3182/20050703-6-CZ-1902.01397>
- Ullah, H., Ahmad, B., Sana, I., Sattar, A., Khan, A., Akbar, S., & Asghar, M. Z. (2021). Comparative study for machine learning classifier recommendation to predict political affiliation based on online reviews. *CAAI Transactions on Intelligence Technology*, 6(3), 251–264. <https://doi.org/10.1049/cit2.12046>
- Vuttipittayamongkol, P., & Arreeras, T. (2022b). Data-driven industrial machine failure detection in imbalanced environments. *2022 IEEE International*

Conference on Industrial Engineering and Engineering Management (IEEM). <https://doi.org/10.1109/ieem55944.2022.9989673>

Wang, Z., Zhang, M., Wang, D., Song, C., Liu, M., Li, J., Lou, L., & Liu, Z. (2017). Failure prediction using machine learning and time series in optical network. *Optics Express*, 25(16), 18553.

<https://doi.org/10.1364/oe.25.018553>

Warriach, E. U., & Tei, K. (2017). A comparative analysis of machine learning algorithms for faults detection in wireless sensor networks. *International Journal of Sensor Networks*, 24(1), 1.

<https://doi.org/10.1504/ijsnnet.2017.084209>

Xu, Y., Zomer, S., & Brereton, R. G. (2006). Support vector Machines: a recent method for classification in chemometrics. *Critical Reviews in Analytical Chemistry*, 36(3–4), 177–188.

<https://doi.org/10.1080/10408340600969486>

Yenigün, O. (2022, September 7). Smart Aspects of CatBoost Algorithm - Python in Plain English. Medium. <https://python.plainenglish.io/smart-aspects-of-catboost-algorithm-2720a6de4da6>

Zuhadar, L. P., & Lytras, M. D. (2023). The Application of AutoML Techniques in Diabetes diagnosis: Current approaches, performance, and future directions. *Sustainability*, 15(18), 13484.

<https://doi.org/10.3390/su151813484>