PREDICTIVE MAINTENANCE OF AIRCRAFT USING DATA ANALYTICS AND MACHINE LEARNING

Project report submitted in partial fulfillment of the requirement for the

Degree of Bachelor of Technology

in

Computer Science and Engineering

By

Aisha Sajjad (191208)

Under the Supervision

Of

Dr. Amit Shrivastava and Dr. Vipul Sharma

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat,

Solan - 173234, Himachal Pradesh

Candidate's Declaration

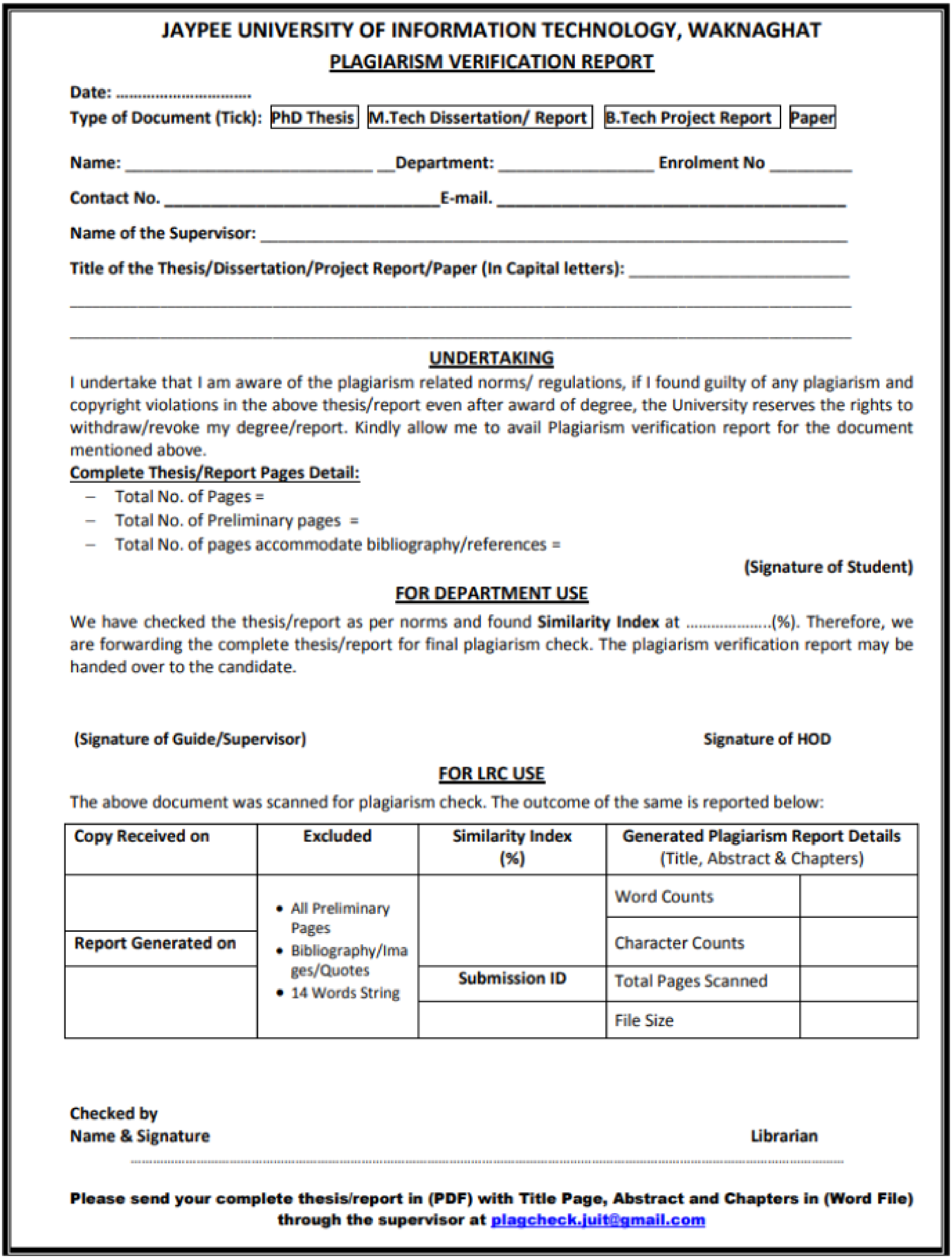
I hereby declare that the work presented in this report entitled "**Predictive Maintenance of Aircrafts using Data Analytics and Machine Learning**" in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted to the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from January 2023 to May 2023 under the supervision of Dr. Amit Srivastava (Associate Professor and Head, Department of Humanities and Social Sciences) and Dr. Vipul Sharma (Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology).

I also authenticate that I have carried out the above-mentioned project work under the proficiency stream Data Science. The matter embodied in this report has not been submitted for the award of any other degree or diploma.

Aisha Sajjad
191208

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Amit Srivastava	Dr. Vipul Sharma
Associate Professor and HoD	Assistant Professor
Department of HSS	Department of CS and IT
Dated:	Dated:



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ....

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: ______________________ __**Department:** ______________ **Enrolment No** _______

Contact No. _________________________**E-mail.** _______________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _______________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

Acknowledgment

Firstly, I express my heartiest thanks and gratefulness to almighty God for his divine blessing in making it possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to supervisors Dr. Amit Srivastava, Associate Professor, and Head, Department of Humanities and Social Sciences, and Dr. Vipul Sharma, Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat. The deep knowledge & keen interest of my supervisors in the field of "Data Analytics" and "Artificial Intelligence" to carry out this project has helped me out a lot. Their endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr. Amit Srivastava and Dr. Vipul Sharma for their kind help to finish my project. I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I would want to express my gratitude to all of the staff members, instructional and non-instructional, who have made my task easier by providing handy assistance.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Aisha Sajjad
191208

Table of Contents

List of Abbreviations

1. TTF - Time to Failure
2. RUL - Remaining Useful Life
3. MTOWs - Maximum Take-Off Weights
4. CM - Corrective Maintenance
5. PM - Preventive Maintenance
6. PdM - Predictive Maintenance
7. PHM - Prognosis & Health Management
8. IBM - International Business Machines
9. P&W - Pratt and Whitney
10. IVHM - Integrated Vehicle Health Management
11. OSA-CBM - Open System Architecture for Condition-Based Maintenance.
12. NASA - National Aeronautics and Space Administration
13. ACMS - Aircraft condition monitoring system
14. QAR - Quick access recorder
15. ACARS - Aircraft Communication Addressing and Reporting System
16. RFE - Recursive Feature Elimination
17. IPYNB - IPython Notebook
18. RBF - Radial Basis Function
19. TPR - True Positive Rate
20. FPR - False Positive Rate
21. TNR - True Negative Rate
22. FNR - False Negative Rate
23. SVC - Support Vector Machine
24. AUC - Area Under the ROC Curve
25. ROC - Receiver Operating Characteristic Curve
26. EDA - Exploratory Data Analysis
27. FE – Feature Engineering
28. AIRMAN – Airbus Industries Aircraft Maintenance Analysis
29. IVHM – Integrated Vehicle Health Management

List of Figures

List of Graphs

List of Tables

Abstract

The nature of international business is changing as a result of big data and AI/ML. Today, data is the most important asset for businesses across all sectors. Businesses are utilizing data-driven insights to gain a competitive edge. As a result, machine learning-based data analytics are quickly gaining traction in a variety of industries, creating autonomous systems that assist in human decision-making.

In the aviation industry, machine learning algorithms can handle massive amounts of diverse data, eliminating extraneous data points to produce a precise image of each individual aircraft component. Multiple condition-based monitoring and predictive maintenance processes are made more efficient by this feature. Data from several sources, including flight data recorders and logbooks, are used by machine learning for predictive maintenance in the aviation industry.

The visual inspection of airplanes can be labor-intensive, time-consuming, and error-prone when done manually. Because maintenance engineers must reach areas of an aircraft that are in hostile environments, it can also be a very dangerous task. However, one of the key benefits of machine learning-based solutions is their capacity to significantly increase the efficiency of human-oriented activities. Two extremely crucial capabilities are made possible by machine learning's deep learning capability: rapid diagnostics and component failure prediction.

In this research, we investigate the use of machine learning in component failure prediction. Regression and classification methods for supervised machine learning were used to analyze patterns in an existing dataset, resulting in predictive analytics for use in forecasting the performance of aircraft equipment/sensors. To be more precise, we have created machine learning-based analytics to forecast the time to failure (TTF) or the equipment's remaining usable life (RUL).

Chapter 1: INTRODUCTION

Over the several years in which the aviation industry has been in service, it has grown to become a significantly reliable and rapid mode of transportation for freight as well as for passengers despite the pressure being placed on the industry for maintaining operational affordability. As a trustworthy mode of transportation, air travel has developed over time. As of 2022, it carried more than 68.4 million metric tonnes of cargo and 3,781 million passengers, accounting for almost 35% of world trade [1] in quantitative terms.

The airline sector has been under pressure to cut costs due to predictions that by 2042, air freight will grow by 4.2% and passenger traffic will increase by 4.5%, respectively [2]. Additionally, the industry has seen fuel prices account for almost 40% of operational costs on average [3], surpassing labor expenses in a labor-dependent industry [4] in the current setting, which has necessitated the need for extremely strategic decision-making.

Herb Kelleher, the former CEO of Southwest Airlines once jokingly said, "If the Wright Brothers were living today, Wilbur would have to terminate Orville to save costs." This claim demonstrates how strenuous and irritating it can be to continually reduce the operating costs in the aviation industry. However, it is a behavior that cannot be overlooked. Failures of airlines are frequently attributable to one crucial factor, namely the "Cost of Operations."

Due to the essence of the varying and fixed expenditures, the expense of operating an airline is back-breaking. May it be for aircraft maintenance, renting out airport space, or IT systems, the fixed costs are very high. Amortizing this over a sizable base is the best way to handle this, but not all airlines have this as part of their strategic plan. Regarding variable costs, they are always risky for the airline due to their nature. When traveling between two points, the cost of the crew is constant, there are minimum fuel needs, the cost of parking and landing is based on the Maximum Take-Off Weights (MTOWs), and the airline is responsible for any disruption expenses. Thus, for airlines, there is a rush to attack each cost item no matter how minuscule it may be.

Without predictive maintenance, unplanned maintenance occurrences and component failure rates in the aviation sector would probably rise. Flight delays, greater aircraft downtime, and increased maintenance expenses could result from this. Since predictive maintenance has long been a common practise in the aviation industry, it is challenging to give a precise estimate in terms of statistics. To estimate the probable impact, we can look at historical data.

For instance, the US commercial aviation industry experienced an average of 9.4 engine failures per 100,000 flight hours between 2001 and 2010, according to a report by the Federal Aviation Administration (FAA). According to the FAA, unscheduled maintenance events can have costs that are up to four times higher than those associated with planned maintenance. Without predictive maintenance, we could anticipate an increase in these expenses, which would probably have a substantial effect on airline profitability and the price of air travel for customers.

The ability of machinery to function cannot be guaranteed; occasionally, it will fail due to outdated operation. Sensor-equipped machinery systems can only monitor a machine's condition; they cannot determine if it is in excellent or bad shape. A maintenance strategy must be applied to scheduled machinery systems in order to prevent the worst event (failure) and obtain information about a machine's status. Corrective (CM), Preventive (PM), and Predictive Maintenance (PdM) are the three best practices for a maintenance plan.

A sort of maintenance called "corrective maintenance" is carried out after a piece of equipment has already broken down or stopped working properly. In order to lessen the impact of the breakdown on the operations, the major objective of CM is to return the equipment to its normal operational condition as soon as feasible. Initiated only after the equipment has already failed, CM is frequently unplanned and reactive. As a result, there may be unanticipated downtime and productivity losses, as well as potential safety issues if the faulty equipment is essential to the operation's security. In conclusion, corrective maintenance is a type of maintenance that is carried out in response to equipment failure. The general consensus is that PM and PdM, which are more proactive and can help to prevent equipment failures before they happen, are more desirable than other types of maintenance, even though it can be beneficial in treating current issues.

In order to avoid equipment failure or malfunction, PM is a sort of maintenance that is carried out on equipment at regular intervals. PM is to maintain equipment in good operating order, lessen the likelihood of malfunctions, and increase the equipment's usable life. Preventive maintenance is typically scheduled in accordance with a predetermined maintenance schedule that considers the manufacturer's recommendations, best practices in the industry, and the particular working circumstances of the equipment. PM can involve a variety of maintenance procedures, including as inspection, cleaning, lubrication, part replacement, and instrument calibration. In general, PM is seen as a proactive method to maintenance because it is carried out on a regular basis whether or not the equipment is displaying failure symptoms. The chance of unanticipated downtime and lost productivity is decreased as a result of this method's ability to identify possible difficulties before they develop into serious concerns. Many industries, including manufacturing, transportation, healthcare, and utilities, use preventive maintenance frequently to keep equipment running smoothly and lower maintenance costs over time. In conclusion, preventative maintenance is a proactive approach to maintenance that is carried out on equipment on a regular basis to keep it in excellent operating order, avoid breakdowns, and prolong its usable life.

In order to forecast when equipment breakdown is likely to occur, PdM is a sort of maintenance plan that makes use of data, analytics, and machine learning algorithms. To minimise downtime, lower maintenance costs, and increase overall equipment reliability, predictive maintenance aims to carry out maintenance tasks proactively, before equipment failure occurs. Sensors and other monitoring tools are used in predictive maintenance to gather information on the equipment's operational parameters, such as temperature, pressure, vibration, and others. Then, machine learning algorithms are used to analyse this data in order to find patterns and anomalies that might point to potential equipment failure.

The most effective maintenance plan depends on the particular needs and requirements of the organisation. Each style of maintenance has benefits and drawbacks of its own. Because it is reactive and deals with equipment failures after they happen, CM is the least effective type of maintenance. Due to emergency repairs, this sort of maintenance may cause unanticipated downtime, lost production, and higher repair expenses. Because it is proactive and carried out on a regular basis, PM is a more effective type of maintenance than CM. This kind of maintenance lessens the possibility of unplanned downtime and lost productivity and helps to prevent equipment failures. PM, however, could lead to unneeded maintenance actions that raise maintenance expenses. Because it uses data and analysis to determine when equipment failure is likely to happen, PdM is typically regarded as the most efficient type of maintenance because it enables proactive scheduling of maintenance activities. This strategy lowers the likelihood of unplanned downtime, lowers repair expenses, and increases the equipment's usable life. However, the implementation of advanced data analytics and machine learning capabilities, which are necessary for PdM, can be costly. In conclusion, the form of maintenance that is most effective will rely on the particular needs and demands of the organisation. Although most organisations believe that PdM is the most effective strategy, it might not always be practical or cost-effective. While preventive maintenance is a good middle ground between corrective and predictive maintenance, it may also

result in a disproportionate amount of maintenance activities.
For example, annual maintenance of the machine, and post-maintenance, if the machine has spares, can be applied, otherwise, the whole operation will stop, which is a disadvantage [5]. Predictive maintenance is based on preventive maintenance, but it also involves constant monitoring of the machine's condition and doing maintenance only as necessary or in the best way possible. Based on historical data, integrity considerations, and statistical inference methodologies, PdM suggested what state the machine was in and when maintenance should be conducted [6]. The use of statistical methods and engineering techniques in the maintenance strategy is necessary for the prediction process, but as technology advances, ML has the potential to use PdM in any situation.

1.1 Performance Optimisation and Predictive Maintenance

Note that proper maintenance of an aircraft has a proportional impact on its performance, as explained below:

To maintain sustainability for airliners both in the short and long term, performance optimization in the aviation industry directly targets operational constraints resulting from fuel efficiency, labor costs, pollution, resource utilization, and aircraft expenses [1]. The demand on the aviation sector to increase sustainability and lessen environmental effect is growing. Reducing energy use, greenhouse gas emissions, and the use of natural resources like water and energy are all part of this. In order to be competitive in the market, airlines must also control their costs. Airlines put a strong emphasis on performance optimisation to achieve sustainability. By identifying and addressing operational restrictions that have an impact on fuel efficiency, labour costs, pollution, resource utilisation, and aircraft costs, performance optimisation seeks to increase the effectiveness of aircraft operations. Airlines can operate more sustainably and efficiently by resolving these limitations. The main goal of performance optimisation is fuel efficiency. Airlines optimise flight patterns, use more fuel-efficient aircraft, and enhance operating procedures in an effort to reduce fuel consumption. This lowers the cost of fuel for airlines while simultaneously reducing the environmental effect of air travel. Another area of focus for performance optimisation is labor costs. Airlines can lower labor costs while preserving safety and efficiency by streamlining procedures and decreasing the strain of ground staff and aircraft crew. Performance optimisation must also prioritise reducing pollution. To lessen the impact of air travel on the environment, airlines are putting money into new technology and alternative fuels. In order to cut emissions, this includes using biofuels, electric and hybrid aircraft, and optimising ground operations. Another crucial factor in performance optimisation is resource use. By using more effective procedures and technologies, airlines want to use less energy and water. This lowers the costs for the airlines and lessens their impact on the environment. Last but not least, performance optimisation strives to lower aircraft costs by enhancing maintenance processes, decreasing downtime, and lengthening the lifespan of aircraft. By attaining these objectives, airlines can become more sustainable over the short and long terms, lowering expenses and having a less impact on the environment while still competing in the market.

Aerospace performance optimization can be used to solve current or future issues that limit the operating effectiveness of an aircraft's equipment and systems, with a focus on the higher cost centers like the engines and auxiliary power units.

Fundamentally, there is a strong correlation between how well the component performs concerning these constraints and its "health," which shows that proper maintenance is needed. Health monitoring and anomaly tolerance levels would boost effectiveness.

Predictive maintenance is a technology that grasps the status of equipment in operation, predicts maintenance procedures before failures occur, and prevents failures by performing maintenance [7]. These techniques ideally reduce maintenance frequency and prevent unplanned reactive maintenance. As a result, this strategy eliminates the obligation to pay for frequent preventative maintenance.

1.2 Implication of Data-Centric Sciences in Aviation

The use of data-centric techniques for predictive maintenance and performance optimization currently available in the aviation industry is clearly visible compared to current approaches [8] such as:

- Decision-making is improved by recognizing trends in consumption and maintenance duties [9]. By isolating risks and flaws in components, inefficient situations can be identified early for maintenance purposes, allowing for process simplification and lowering the danger of cascading failures. An essential component of decision-making in maintenance management is identifying patterns in consumption and maintenance requirements. Proactive maintenance can be planned to minimise cascading failures and enhance safety by analysing data and spotting hazards and defects in components early. This strategy may also result in streamlined procedures and increased effectiveness in maintenance management.
- Manual diagnosis procedures may result in prolonged aircraft downtime and a reduction in the ability to handle passengers in real time. To reduce unscheduled maintenance, data science on aviation data could improve and aid in making quicker, educated judgments. It is possible to find patterns and trends that can be suggestive of prospective problems or maintenance needs by gathering and analysing data from multiple sources, including sensors, maintenance records, and flight data. Predictive models can be developed using this data to assist find possible problems before they arise. Early detection of these problems enables proactive maintenance scheduling, reducing aircraft downtime and increasing overall performance. Data science can decrease unscheduled maintenance while simultaneously increasing the precision of diagnosis processes. It is feasible to spot prospective difficulties and hone in on probable root causes of a problem by analysing data from a variety of sources.
- 5–10% less support staff is needed for maintenance since automated

problem isolation and detection are supported by data analytics.

1.3 Problem Statement

Many industries, including the airline sector, place a strong emphasis on failure prediction through predictive maintenance in order to improve operations and cut down on flight delays. By anticipating the engine's RUL or TTF, monitoring the health and condition of engines using sensors and telemetry data can facilitate this type of maintenance.

A maintenance schedule can be developed based on which engines are most likely to fail in the current period or cycle window by precisely projecting an engine's remaining usable life, which can assist forecast engine failures by period. By doing this, airlines are able to undertake maintenance procedures in a proactive manner, lowering the possibility of unplanned, expensive equipment breakdowns and raising the overall dependability of their operations.

By offering a proactive and preventative approach to maintenance, predictive maintenance can help businesses cut expenses, increase equipment reliability, and improve overall performance. Businesses can cut down on overall equipment downtime and maintenance expenses by using predictive maintenance. Businesses may monitor the state of their equipment in real-time, spot possible issues before they arise, and schedule maintenance tasks appropriately by putting into practice a predictive maintenance approach.

Predictive maintenance is particularly important in the aviation industry because any equipment breakdown can result in lengthy delays and cancellations, displeasing passengers and costing businesses money. Therefore, being able to anticipate equipment breakdowns and schedule maintenance in advance can help airlines make sure that their business operations are effective and efficient.

In conclusion, predictive maintenance is an essential tactic for companies trying to streamline their processes, cut expenses, and raise the dependability of their equipment. Businesses can utilise information about the health and condition of their equipment to predict and avoid equipment breakdowns by utilising data from sensors and telemetry systems. Predictive maintenance is turning into a more crucial tool to assure organisations' continuous success as they continue to rely on technology and equipment to power their operations.

1.4 Objectives

The target of this project is -

- To improve maintenance operations by utilizing data handling methodologies and ML algorithms to forecast maintenance requirements more accurately.
- To support time-based preventive maintenance planning

1.5 Methodology

ML algorithms can understand the association between sensor readings and changes in sensor readings from previous faults, and predict future faults by examining aircraft engine sensor readings over time.

- A regression model algorithm was used to forecast the number of remaining cycles before an engine fails.
- A binary classification algorithm was used to predict whether the engine will malfunction within a certain cycle window.
- A multi-class classification algorithm was employed to predict the cycle window for engine failure.

1.6 Organization

The rest of the report is organized as follows. Chapter 02 reviews the related work in aviation. Chapter 03 critically discusses data acquisition, and system development including data cleaning and exploratory data analysis while chapter 04 carefully examines the implication of utilizing data science methodologies and machine learning algorithms in the aviation industry followed by the limitations of the project and the conclusion in chapter 05.

Chapter 2: LITERATURE SURVEY

Since the 1980s, there has been a growing body of research on using machine learning, big data, and data analytics approach to the aviation industry. In particular, the technology was adopted for the goal of customer-focused marketing at the same time as the early adopters in related industries. Engine, auto, and mobile equipment manufacturers on a large scale have used big data technologies to improve production operations and reliability [10], [11], and [12], and have also proposed the terms "Industrial Internet" and Industrial IoT. In fact, particularly in the context of real-time analysis, these form the basis of aviation big data.

A real-time monitoring and problem diagnosis tool called the AIRMAN helps to detect anomalies early and improve effective solutions in a prompt way to minimize aircraft downtime. The tool shows signs of effectively managing huge amounts of real-time data, but the predictability of failures is not covered. However, the predictive prognostic model “Predictivity”, introduced in 2013, builds on the latest concepts in prognosis and health management (PHM) to establish a prognostic model of behavior using real-time data [10]. It made it possible to operate with less fuel consumption. The biggest winner was AirAsia, as it was able to save itself $10 million in gas expenditure. But the details of "predictability" are still a mystery.

To analyze aviation data from about 35,000 airplane engines and provide anomaly warning and fuel prediction models to customers, General Electrics inaugurated its first cloud platform called the "Predix"[10], [11].

In order to forecast engine maintenance, IBM and P&W tracked the performance of about 4,000 plane engines that were in use and employed big data analytics to analyze the results [11]. P&W measured 1000 parameters from a running engine in a modern environment and determined a 50% reduction in on-air shutdowns [10], believed to give a prediction accuracy of approximately 90%.

The general overview of such IVHM is indicated in Fig. 1.

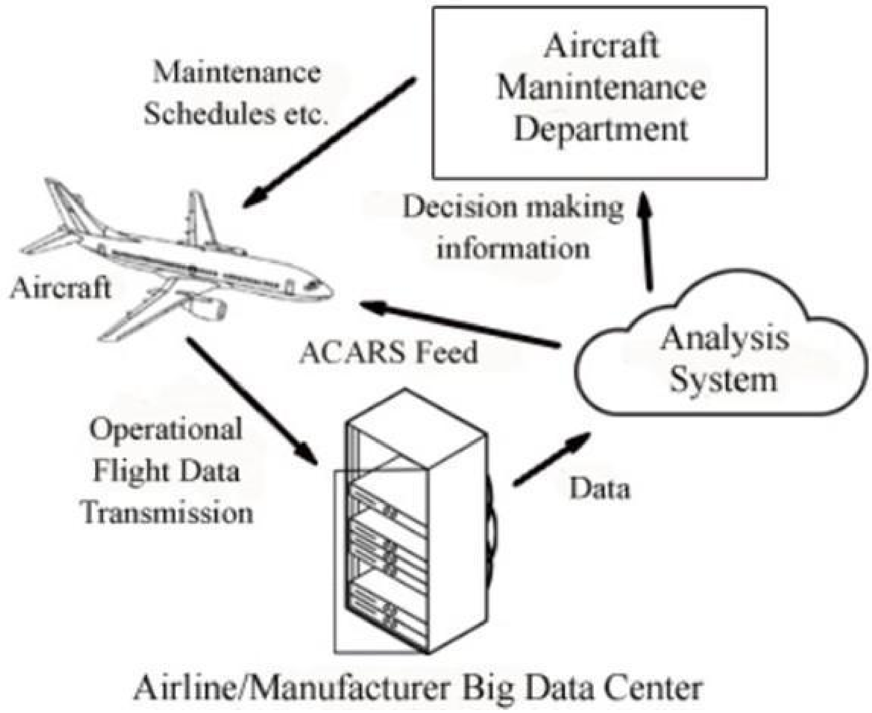


Fig. 1. IVHM framework for aircraft [13].

To enhance energy consumption levels and improve engine operational efficiency, Rolls Royce chose the Microsoft Azure cloud platform for the collection, integration, and analysis of vast amounts of airline data gathered by flight instruments [22], [13]. The system provided extensive opportunities for “predictive” optimization of aircraft performance. This includes service factors such as improved fuel burn and spare parts with consequent higher conversion efficiencies. However, these techniques used above had a significant impact on improving the performance of existing airlines due to the impossibility of acquiring and replacing fleets of aircraft frequently, thus IVHM I have isolated the area of PdM to the performance of engine only. In contrast to the 'integrated' aspects of airplane performance, which are the main focus of most of the studies discussed below, there are more issues. ISIR Labs' research [22] on fault-tolerant systems based on wireless transmission of real-time data from anti- icing systems deserves credit for establishing the

feasibility of using real-time parametric data analysis for flight and ground-based decisions.

Numerous research have looked into the use of machine learning and data analytics in PdM for aircraft.. One such study, conducted by Hsu et al. in [16], presented a statistical process control and machine learning-based approach for diagnosing faults in wind turbines and predicting maintenance needs.

Amruthnath and Gupta (2018) investigated the use of unsupervised machine learning algorithms for early issue detection in PdM in [17]. The authors applied different clustering algorithms on sensor data to detect anomalies and predict potential faults in the system.

Bruneo and Vita (2019) explored the use of Long Short-Term Memory (LSTM) networks for PdM in smart industries in [18]. The authors developed an LSTM-based model that can predict the RUL of equipment by analyzing the sensor data.

A hybrid machine learning strategy was put forth by Cho et al. (2019) for PdM in smart factories in [19]. The authors used a combination of support vector machines (SVM), artificial neural networks (ANN), and random forests to forecast equipment failures and maintenance needs.

Machine learning was used by Gohel et al. (2020) to create a PdM architecture for nuclear infrastructure in [21]. The authors used a combination of anomaly detection and classification algorithms to predict equipment failures and maintenance needs.

In [22], Spiru Haret and A. Mihai discussed the applications of business intelligence systems in the airline industry. The authors discussed the various ways in which the airline sector can benefit from the use of data analytics to enhance operations and maintenance.

Similarly, in [23], Campbell discussed the evolution of flight data analysis and the various techniques used to analyze aircraft data. The author discussed the benefits of using data analysis to improve aircraft maintenance and safety.

The research done in [26] reviews the trends and challenges of PdM for aircraft engines using machine learning techniques. The study highlights the potential benefits of predictive maintenance in reducing maintenance costs, minimizing downtime, and enhancing safety. The paper also discusses the challenges of data acquisition, data quality, and model accuracy in implementing predictive maintenance for aircraft engines. Our research explores the application of machine learning techniques, specifically regression and classification methods, for predicting component failure. The study utilized an existing dataset to identify patterns and develop predictive analytics for forecasting the performance of aircraft equipment/sensors. The focus was on creating machine learning-based models that could forecast the TTF or RUL of the equipment.

The research done in [27] explores the application of data analytics techniques in the PdM and performance optimization of aircraft. The study suggests that data-driven predictive maintenance can help in identifying potential faults and optimizing aircraft performance by analyzing various data sources such as aircraft telemetry data, maintenance records, and environmental data.

In addition to the papers mentioned earlier, several studies have investigated the implementation of data analytics and machine learning techniques in PdM for aircraft.

Overall, these studies demonstrate the potential of data analytics and machine learning techniques in predictive maintenance for various industries, including aircraft, wind turbines, and nuclear infrastructure. By utilizing real-time data, these techniques can help identify potential faults and predict maintenance needs, leading to reduced costs, minimized downtime, and enhanced safety.

2.1 Recent Trends of Predictive Maintenance

Any system can benefit from applying predictive maintenance, and using ML methods that include deep learning is the best approach to do it. Table 1 displays the summary articles that matched the set criteria. The average number of articles published each year from 2016 to 2021 was only 3.7. therefore, the year of publication in Figure 1 suggests that the ML applied to PdM is a relatively recent development in work. Additionally, it was covered in papers released by Indonesian universities in 2020 and 2021.

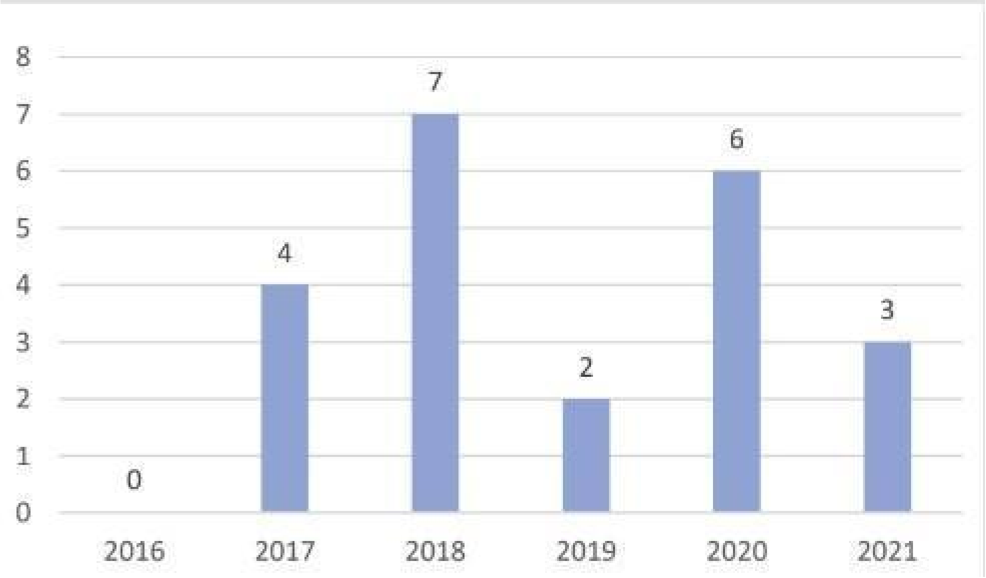


Fig. 2: No. of articles published on the predictive maintenance of aircraft from 2016 to 2021 [26]

Table 1. Recent trends in PdM

References	Year	Methods	Description of Applied Predictive Maintenance
Hsu, et al [16]	2020	K-means fold; Random Forests; Decision Trees	With an accuracy of more than 90%, statistical processes to identify faults and machine learning forecast maintenance requirements for wind turbines can identify failure states.
Amruthnath and Gupta [17]	2018	PCA; C-means; K- means; Analysis of Clusters; Gaussian Mixture Modeling	Observing the exhaust fan with a vibrator sensor, dividing the results into three categories—healthy warning, fault condition, and condition prediction using machine learning (ML), followed by the high accuracy T2 method.
Bampoula, et al. [15]	2021	LSTM; Autoencoder	The rolling milling machine uses LSTM-autoencoder to estimate RUL with great accuracy, however, it is limited in its ability to use different neural networks to determine RUL.
Bruneo and Vita [18]	2019	LSTM and Tunning of Hyper-parameters	We use this to tune hyperparameters for predictive maintenance of jet engines with larger RMSE than other ML techniques.
Cho et al. [19]	2018	Hybrid of semi-supervised and unsupervised algorithms	Using hybrid ML to calculate predictive maintenance, problems with a smart factory's multiple machine operations without sufficient maintenance data were resolved.

Demidova [20]	2020	GRU; LSTM; RNN Hybrid	Used LSTM and GRU neurons with an accuracy of over 90% to compute RUL for preventative maintenance on aircraft engines utilizing one and two layers of RNN.
Gohel, et al [21]	2020	Logistic Regression and SVM	Because employing SVM and LR has the comprehensive capability for nuclear powerplant infrastructure, ML is applied to make Predictive maintenance on nuclear infrastructure, which is a crucial site.

Chapter 3: SYSTEM DEVELOPMENT

3.1 Data Acquisition

Several data sources are fundamentally taken into consideration to attain real-time data analytical capabilities of signified dependability and precision.

1) Quick Access Recorder

Flight parameters are recorded by numerous sensors in the avionics and other parts of the aircraft (such as the engine). The ACMS collects and pre-processes the recorded data and makes it accessible to the aircraft and it is done with the help of a Quick Access Recorder (QAR) [10]. The usage of the IoT as an upcoming technology is primarily driven by recent advances in QAR that have developed to transmit via GPRS and wireless transmission over the Internet [23], [24], [25]. As shown in Figure 3, this approach provides direct access to the data warehouse for data storage.

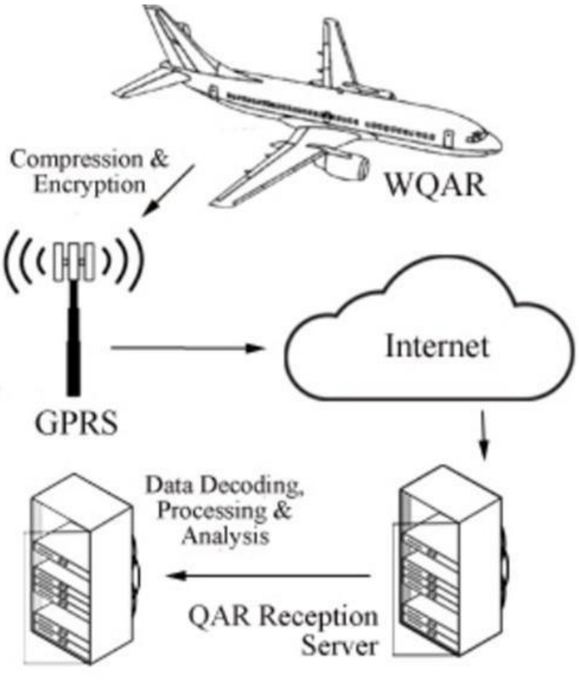


Fig. 3. Data transfer process of WQAR [10].

2) ACARS

As shown in Figure 4 [10], [11], the ACARS is a dual air-to-ground data link that relays real- time diagnostic and error messages to operators over specialized networks. In contrast, data is organized in ACARS and the same privacy considerations apply. However, real-time performance data can be full of meaningless data due to the high capacity and speed of analysis, as well as the sensitivity of the sensors, which can be affected by physical and non- physical interference, transmission issues, etc.

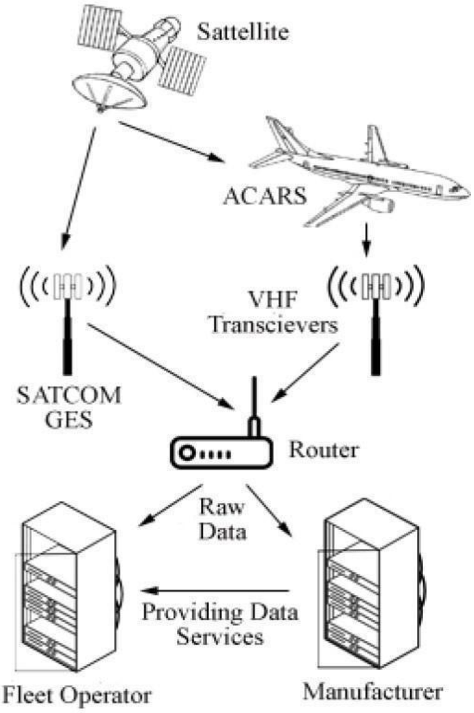


Fig. 4. ACARS [10].

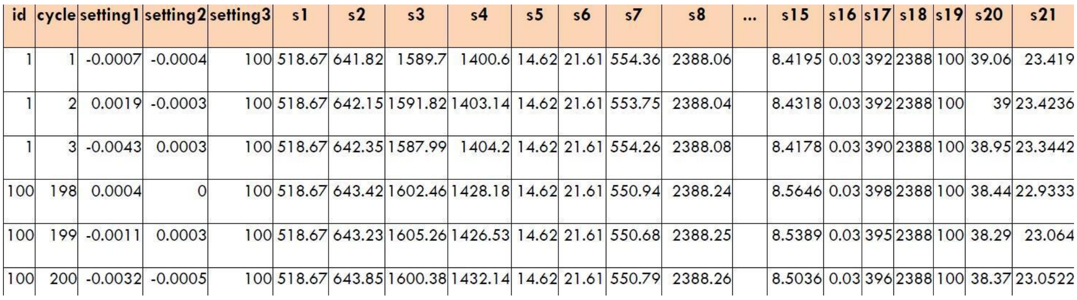
3) Repair and Maintenance Records

Because beneficial insights are obtained by comparing operational data with maintenance records, this data could be regarded as key data sources concerning predictive maintenance. Even though repair data are only produced in certain circumstances, they can be viewed as static for a long time [10], meaning that their validity may deteriorate over time

3.2 The Dataset

Text files provide operating settings, 21 sensor measurements provided by Microsoft, and simulated aircraft engine run-to-failure events. It is considered that the engine's sensor measurements represent the pattern of engine degradation as it progresses.

- Run-to-failure statistics for aviation engines are contained in training data files. records of 20,000+ cycles for 100 engines.
- Engine operating data from test aircraft are documented in test data files without failure events. A separate Ground truth data file is provided with the remaining cycles.



id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	s6	s7	s8	...	s15	s16	s17	s18	s19	s20	s21
1	1	-0.0007	-0.0004	100	518.67	641.82	1589.7	1400.6	14.62	21.61	554.36	2388.06		8.4195	0.03	392	2388	100	39.06	23.419
1	2	0.0019	-0.0003	100	518.67	642.15	1591.82	1403.14	14.62	21.61	553.75	2388.04		8.4318	0.03	392	2388	100	39	23.4236
1	3	-0.0043	0.0003	100	518.67	642.35	1587.99	1404.2	14.62	21.61	554.26	2388.08		8.4178	0.03	390	2388	100	38.95	23.3442
100	198	0.0004	0	100	518.67	643.42	1602.46	1428.18	14.62	21.61	550.94	2388.24		8.5646	0.03	398	2388	100	38.44	22.9333
100	199	-0.0011	0.0003	100	518.67	643.23	1605.26	1426.53	14.62	21.61	550.68	2388.25		8.5389	0.03	395	2388	100	38.29	23.064
100	200	-0.0032	-0.0005	100	518.67	643.85	1600.38	1432.14	14.62	21.61	550.79	2388.26		8.5036	0.03	396	2388	100	38.37	23.0522

Features contained in the dataset are –

ID: Engine ID in the range of 1 to 100.

Cycle: per engine sequence, starting from 1 to the cycle number where the error occurred.

Setting1 to Setting3: Engine operation settings s1 to s21: sensor readings

3.3 Approach

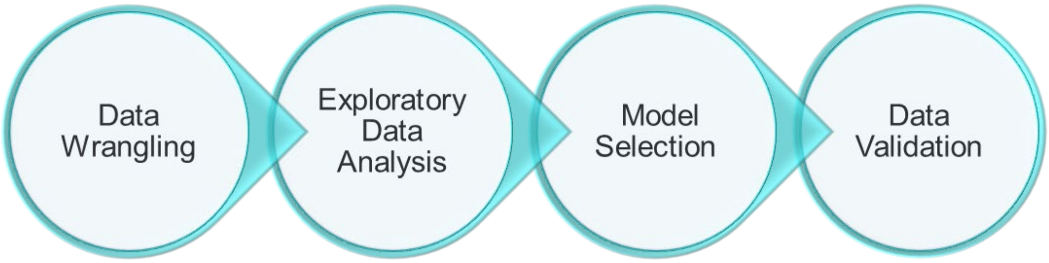


Fig 5: Flowchart of the approach used

3.3.1 Data Wrangling

Large amounts of data need to be stored and organized for analysis because the amount of data and data sources available today are expanding quickly. Data cleaning often referred to as data wrangling or data munging, is the act of organizing, manipulating, and cleaning raw data into the required format that analysts can utilize to make quick decisions.

For data science and data analysis, data wrangling is a vital subject. Python's Pandas Framework, an open-source library created primarily for data analysis and data science is used for data wrangling.

The following functionalities are dealt with via data wrangling in Python:

1. Data exploration: It involves displaying data representations in

order to study, analyze, and comprehend the data.

2. Dealing with missing values: The majority of datasets with a large quantity of data have missing values of NaN; they need to be dealt with by replacing them with the mean, mode, or the column's most frequent value, or simply by removing the row containing the NaN value.
3. Data reshaping: It is the process of manipulating data to meet specific needs. New data can be added or current data can be changed.
4. Data filtering: Datasets occasionally contain undesired rows or columns that need to be eliminated or filtered.
5. Other: By combining the aforementioned features with the raw dataset, we can produce an effective dataset that meets our needs. This dataset can then be utilized for the necessary tasks, such as data analysis, machine learning, data visualization, model training, etc.

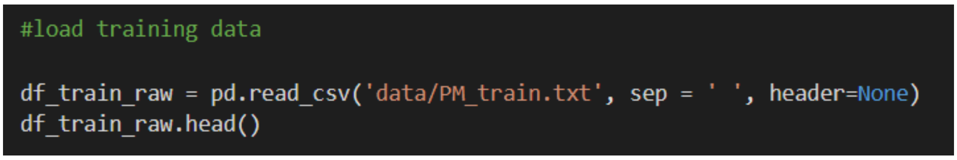
The primary significance of using data-cleaning tools is explained below:

1. Making raw data functional. Data that has been correctly wrangled ensures that high- quality data is used in the subsequent analysis.
2. Putting all information from many sources in one place so that it can be utilized.
3. Assembling raw data in the required format and comprehending the data's business context
4. Data visualization becomes easier. Starting with a clean dataset enables you to concentrate on developing an effective visualization rather than trying to identify and address problems as you go.

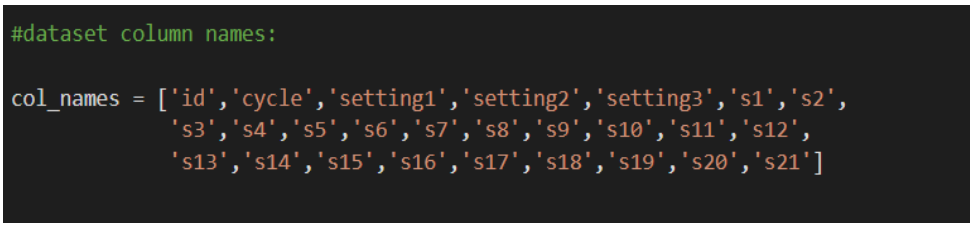
6 Steps to Perform Data Wrangling-

1) Data Discovery: Data discovery is the first stage of the Data Wrangling process. This is a general phrase for comprehending or becoming acquainted with your data. In order to make your data easier to

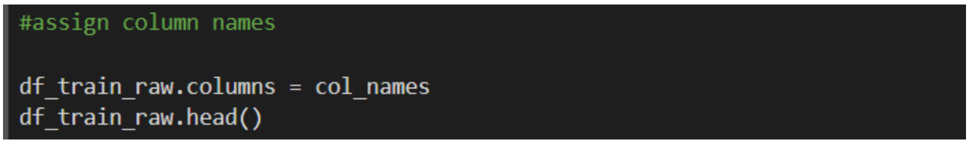
use and analyse, you must look at it and consider how you would like it to be arranged. As a result, you start with an unruly crowd of data that has been gathered from many sources and is in a variety of formats. The objective at this point is to gather the many, siloed data sources and set each one up such that it is possible to understand them and look for patterns and trends in the data.



```
#load training data

df_train_raw = pd.read_csv('data/PM_train.txt', sep = ' ', header=None)
df_train_raw.head()
```



```
#dataset column names:

col_names = ['id','cycle','setting1','setting2','setting3','s1','s2',
             's3','s4','s5','s6','s7','s8','s9','s10','s11','s12',
             's13','s14','s15','s16','s17','s18','s19','s20','s21']
```



```
#assign column names

df_train_raw.columns = col_names
df_train_raw.head()
```

We start with loading our training data into an IPYNB file and storing the imported data in a pandas' data frame. Since our data had no column names, we provide it manually. Moreover, our data frame contained some extra columns and they had to be removed. Next, we move on to exploring the data. Using the. describe() method we try to get some statistical information about our data.

From the training data, we can conclude that there are 100 engines and the average is 108 cycles per engine. The failure of the engine is represented by its last cycle.

Similarly, import test data and repeat the above process. Like the training data, we have 100 engines and each engine has an average of 76 cycles. However, in the test data file, no error data was provided. They were provided in a separate truth file. To get purposeful test data, we need to combine the last cycle and truth data (TTF) of each engine in the test data. This gives us a test set of 100 engines with TTF data. We accomplish this later when we create regression and classification labels for our data.

2) Data Structuring: When raw data is gathered, it comes in a variety of sizes and forms. It lacks a clear structure, which indicates that it lacks a model and is wholly disorganized. Giving it a framework enables better analysis and allows it to be reformed to fit in with the analytical model used by your company. Unstructured data frequently has a lot of text and contains elements like dates, numbers, ID codes, etc. The dataset has to be parsed at this point in the Data Wrangling procedure. This will produce a spreadsheet with more valuable data and more user-friendly columns, classes, headings, etc.

3) Data Cleaning: Raw data typically contains a number of inaccuracies that must be corrected before moving on to the next step. Data cleaning includes addressing outliers, making corrections, fully erasing bad data, etc. This is accomplished by sanitising and cleaning up the dataset using algorithms. The following happens when data is cleaned:

- It eliminates outliers from your dataset that can cause your data analysis results to be skewed.
- To enhance the accuracy and consistency of the data, it modifies any null values and harmonises the data format.
- It locates duplicate values, standardises measurement techniques, corrects grammatical and typographical errors, and validates the data to make it more manageable.

```
# check the data types

df_train_raw.dtypes
```

```
# check for NaN values

df_train_raw.isnull().sum()
```

Using the above two methods we found out that all of the data we were working with was numeric and hence, we could say that the provided dataset was a clean dataset.

4) Data Enriching: You can increase the precision of your analysis by combining your raw data with extra data from sources such as internal systems, third-party providers, etc. with your raw data. Alternatively, you could just want to fill in any informational gaps.

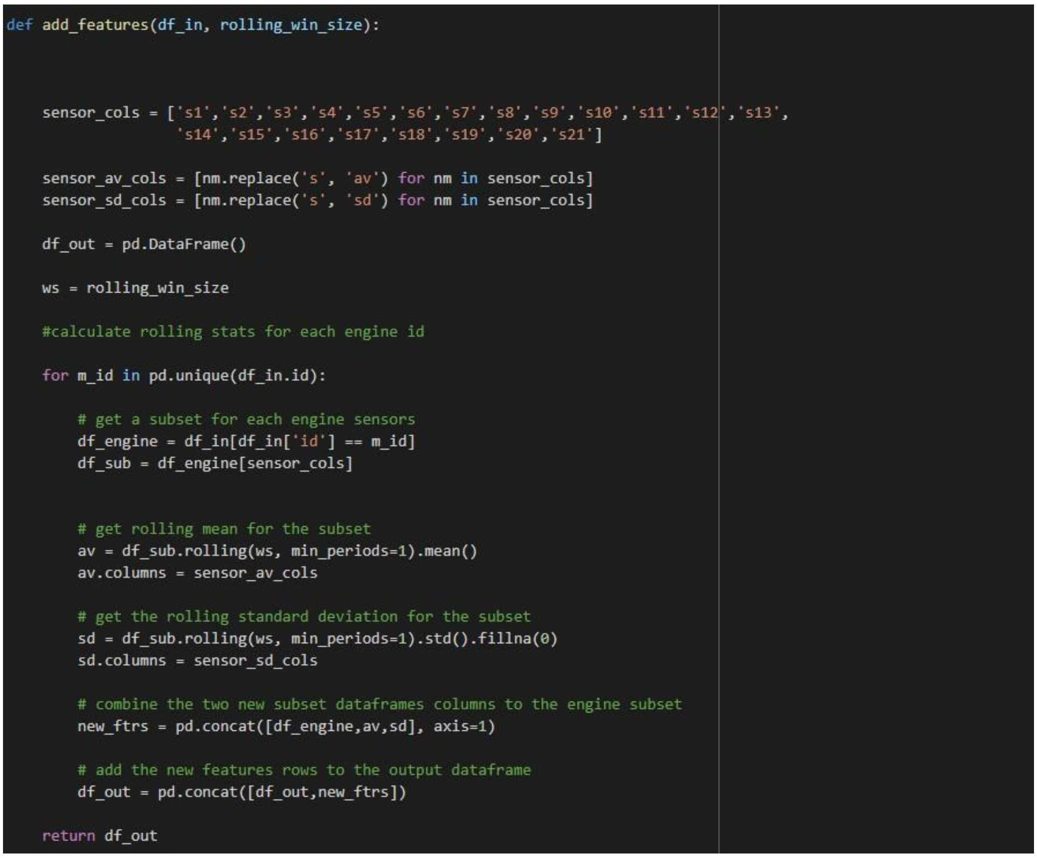
5) Data Publishing: All the steps have been finished by this point, and the data is prepared for analysis. The freshly wrangled data has to be published somewhere where you and other stakeholders can readily access it and use it. Once the data has been cleaned and is ready for the analyses and modeling phase, we'll save the data to a CSV file. Both training and test data are stored in a CSV file for later stages.

Feature Extraction: The massive datasets, that we work with, consist of a wide variety of features. Processing these features requires a lot of computational power. To effectively decrease the amount of data, FE helps in extracting the heavily weighted features from these huge amounts of data by selecting variables and combining them into features.

Ultimately, data reduction speeds up the learning and generalization phases of the machine learning process, allowing models to be built with less machine effort. Rolling and Moving Averages: A rolling average, often known as a moving average, is a measure that uses a set of data to determine patterns over brief periods of time. In particular, it makes it easier to calculate trends when it could otherwise be challenging to do so. For instance, you could be unable to determine whether your data collection exhibits upward or downward trends over time if it contains numerous occasions where the values sharply increase or decrease. Long-term patterns can be found using rolling averages when they are otherwise hidden by sporadic changes.

Rolling average = sum of data over time/time period

Moving Standard Deviation: A statistical concept known as standard deviation gives a decent measure of volatility. It quantifies how far values (like closing prices) deviate from the mean. Dispersion is the discrepancy between the closing price's actual value and its average value (mean closing price).



```
def add_features(df_in, rolling_win_size):



    sensor_cols = ['s1','s2','s3','s4','s5','s6','s7','s8','s9','s10','s11','s12','s13',
                   's14','s15','s16','s17','s18','s19','s20','s21']

    sensor_av_cols = [nm.replace('s', 'av') for nm in sensor_cols]
    sensor_sd_cols = [nm.replace('s', 'sd') for nm in sensor_cols]

    df_out = pd.DataFrame()

    ws = rolling_win_size

    #calculate rolling stats for each engine id

    for m_id in pd.unique(df_in.id):

        # get a subset for each engine sensors
        df_engine = df_in[df_in['id'] == m_id]
        df_sub = df_engine[sensor_cols]


        # get rolling mean for the subset
        av = df_sub.rolling(ws, min_periods=1).mean()
        av.columns = sensor_av_cols

        # get the rolling standard deviation for the subset
        sd = df_sub.rolling(ws, min_periods=1).std().fillna(0)
        sd.columns = sensor_sd_cols

        # combine the two new subset dataframes columns to the engine subset
        new_ftrs = pd.concat([df_engine,av,sd], axis=1)

        # add the new features rows to the output dataframe
        df_out = pd.concat([df_out,new_ftrs])

    return df_out
```

Adding Regression and classification labels to the training data and the test data:

- Regression: TTF for each cycle/engine TTF (cycles to failure) is the number of cycles between the last cycle of the same engine and that particular cycle.

- Binary classification: If the remaining cycles are less than a certain number of cycles (eg Duration = 15), the motor will fail in this duration. Else the motor is fine.

- Multi-class classification: By dividing the TTF into cycle ranges (eg, duration: 0-15, 16- 30, 30+), we were able to identify periods during which the engine would fail.

We prepare the training and test data by adding features and labels. Once we're done with that, we save our data frames for both the test data and training data for further processing.

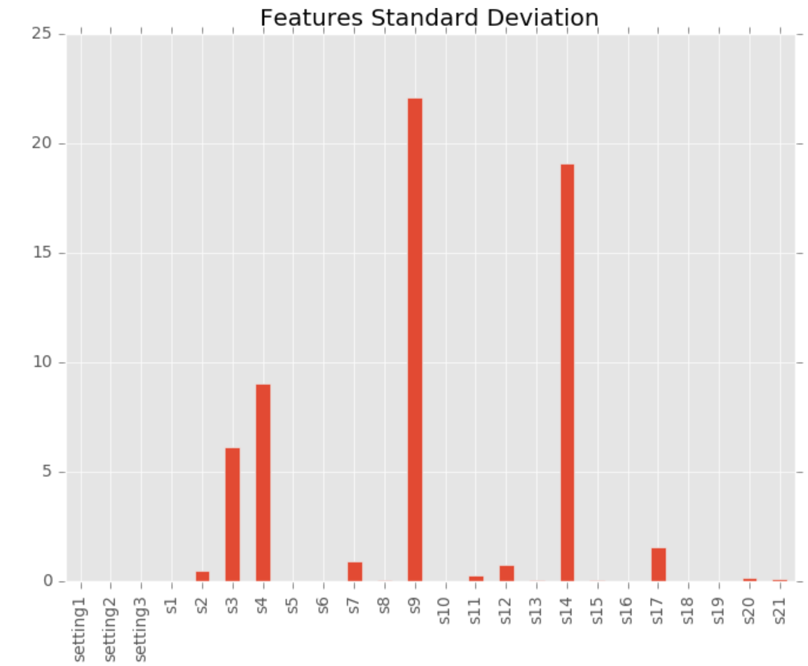
3.3.2 Exploratory Data Analysis

EDA is an important way to first look at data to find patterns, discover anomalies, test hypotheses, and validate assumptions using summary statistics and graphs.

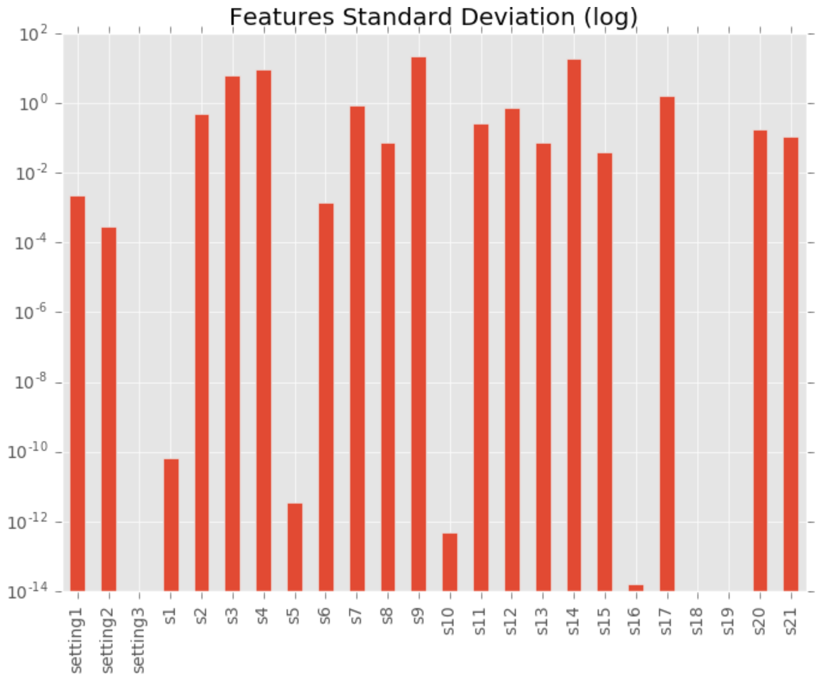
Benefits of conducting exploratory data analysis

- Organising a dataset
- Understanding variables
- Identifying relationships between variables
- Choosing the right model
- Finding patterns in a dataset

We start by comparing the standard deviation of selected input features. We also plot a graph for comparing the log standard deviation of selected input features.



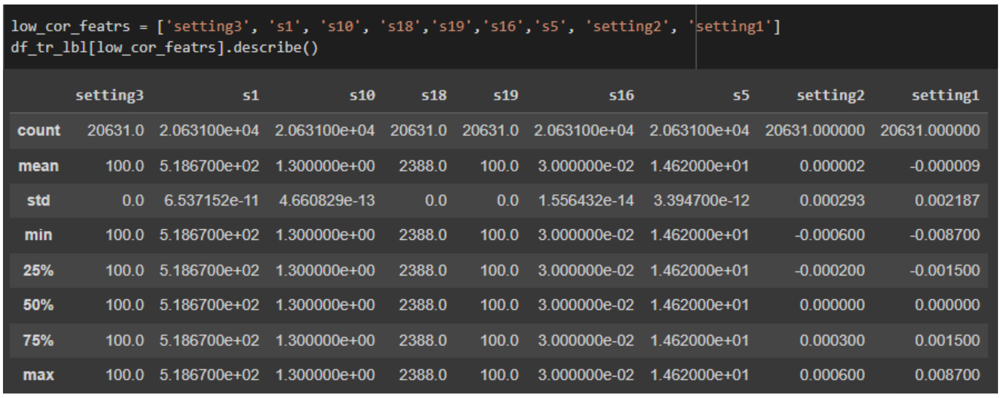
Graph 1: standard deviation of input features



Graph 2: Log standard deviation of input features

The statistical relationship between two different variables is measured via correlation analysis. The outcome will demonstrate the effects of changing one parameter on the other parameter. A crucial idea that is well-known in the field of predictive analytics is correlation analysis. Additionally, the correlations study must be finished before developing the model and drawing any conclusions regarding the links between the variables. While correlation analysis aids in our knowledge of the relationship between two variables in a dataset, it is unable to identify or quantify the underlying cause.

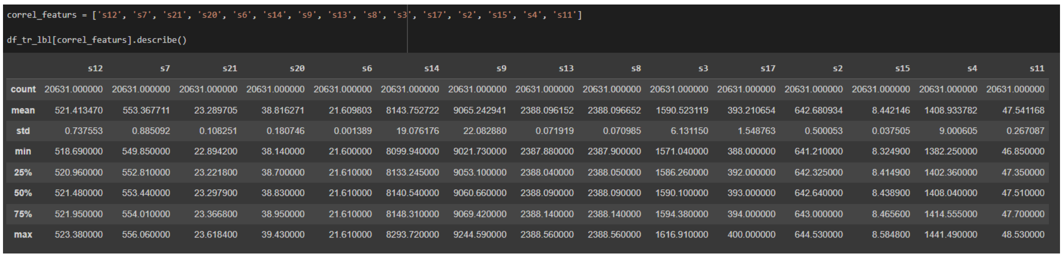
Using the graphs - Graph 1 and Graph 2, we can find out the top variance features. Once we're done with that, we can get the ordered list of features and their correlation with regression label TTF. The features having low or no correlation with regression label ttf and very low or no variance will be target for removal in feature selection.



```
low_cor_featrs = ['setting3', 's1', 's10', 's18','s19','s16','s5', 'setting2', 'setting1']
df_tr_lbl[low_cor_featrs].describe()
```

	setting3	s1	s10	s18	s19	s16	s5	setting2	setting1
count	20631.0	2.063100e+04	2.063100e+04	20631.0	20631.0	2.063100e+04	2.063100e+04	20631.000000	20631.000000
mean	100.0	5.186700e+02	1.300000e+00	2388.0	100.0	3.000000e-02	1.462000e+01	0.000002	-0.000009
std	0.0	6.537152e-11	4.660829e-13	0.0	0.0	1.556432e-14	3.394700e-12	0.000293	0.002187
min	100.0	5.186700e+02	1.300000e+00	2388.0	100.0	3.000000e-02	1.462000e+01	-0.000600	-0.008700
25%	100.0	5.186700e+02	1.300000e+00	2388.0	100.0	3.000000e-02	1.462000e+01	-0.000200	-0.001500
50%	100.0	5.186700e+02	1.300000e+00	2388.0	100.0	3.000000e-02	1.462000e+01	0.000000	0.000000
75%	100.0	5.186700e+02	1.300000e+00	2388.0	100.0	3.000000e-02	1.462000e+01	0.000300	0.001500
max	100.0	5.186700e+02	1.300000e+00	2388.0	100.0	3.000000e-02	1.462000e+01	0.000600	0.008700

We also find out the features having a high correlation with regression label TTF.

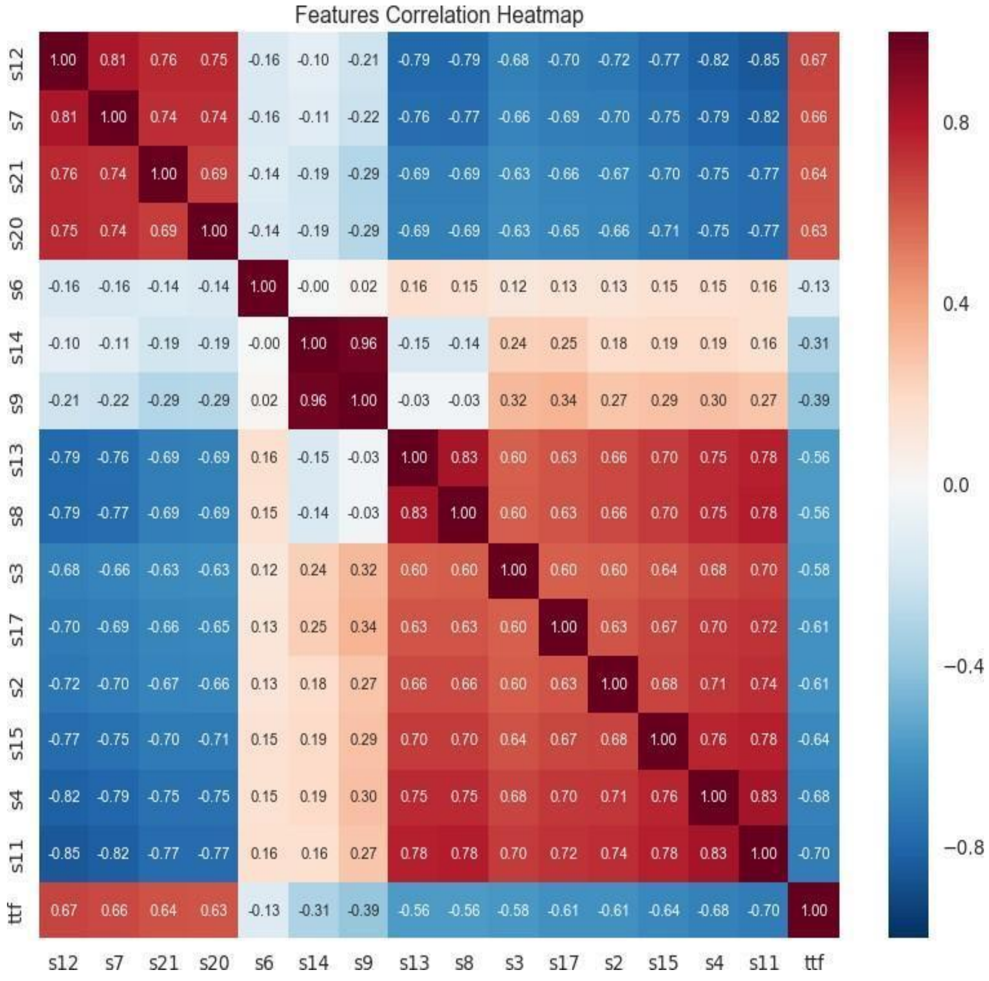


```
correl_featurs = ['s12', 's7', 's21', 's20', 's6', 's14', 's9', 's13', 's8', 's3', 's17', 's2', 's15', 's4', 's11']

df_tr_lbl[correl_featurs].describe()
```

	s12	s7	s21	s20	s6	s14	s9	s13	s8	s3	s17	s2	s15	s4	s11
count	20631.000000	20631.000000	20631.000000	20631.000000	20631.000000	20631.000000	20631.000000	20631.000000	20631.000000	20631.000000	20631.000000	20631.000000	20631.000000	20631.000000	20631.000000
mean	521.413470	553.367711	23.289705	38.816271	21.609803	8143.752722	9065.242941	2388.096152	2388.096652	1590.523119	393.210654	642.680934	8.442146	1408.933782	47.541168
std	0.737553	0.885092	0.108251	0.180746	0.001389	19.076176	22.082880	0.071919	0.070985	6.131150	1.548763	0.500053	0.037505	9.000605	0.267087
min	518.690000	549.850000	22.894200	38.140000	21.600000	8099.940000	9021.730000	2387.880000	2387.900000	1571.040000	388.000000	641.210000	8.324900	1382.250000	46.850000
25%	520.960000	552.810000	23.221800	38.700000	21.610000	8133.245000	9053.100000	2388.040000	2388.050000	1586.260000	392.000000	642.325000	8.414900	1402.360000	47.350000
50%	521.480000	553.440000	23.297900	38.830000	21.610000	8140.540000	9060.660000	2388.090000	2388.090000	1590.100000	393.000000	642.640000	8.438900	1408.040000	47.510000
75%	521.950000	554.010000	23.366800	38.950000	21.610000	8148.310000	9069.420000	2388.140000	2388.140000	1594.380000	394.000000	643.000000	8.465600	1414.555000	47.700000
max	523.380000	556.060000	23.618400	39.430000	21.610000	8293.720000	9244.590000	2388.560000	2388.560000	1616.910000	400.000000	644.530000	8.584800	1441.490000	48.530000

["s7", "s20", "s21", "s15", "s6", "s9", "s4", "s8", "s12", "s3", "s17", "s2", "s14", "s13", "s11"]

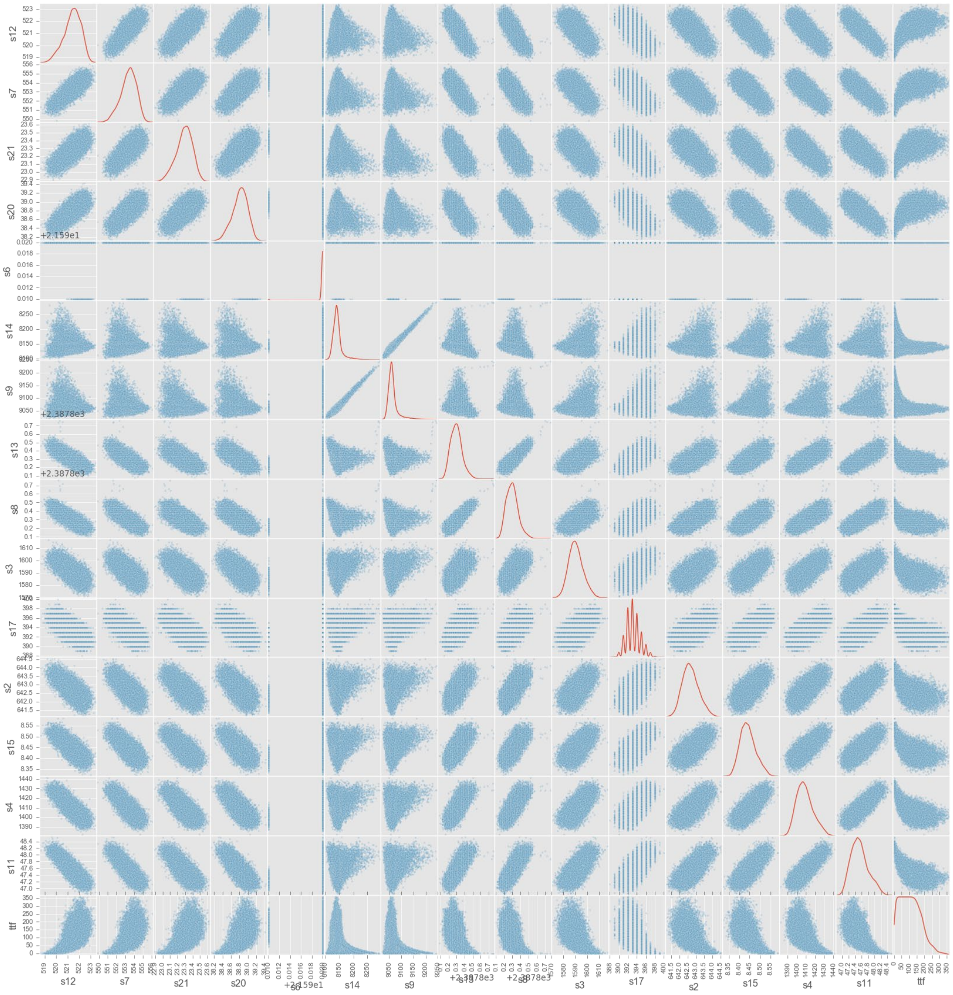
Considering that their correlation with TTF is larger than that of other features, they might be a focus for feature selection during modeling. We'll show this correlation using a heatmap.



	s12	s7	s21	s20	s6	s14	s9	s13	s8	s3	s17	s2	s15	s4	s11	ttf
s12	1.00	0.81	0.76	0.75	-0.16	-0.10	-0.21	-0.79	-0.79	-0.68	-0.70	-0.72	-0.77	-0.82	-0.85	0.67
s7	0.81	1.00	0.74	0.74	-0.16	-0.11	-0.22	-0.76	-0.77	-0.66	-0.69	-0.70	-0.75	-0.79	-0.82	0.66
s21	0.76	0.74	1.00	0.69	-0.14	-0.19	-0.29	-0.69	-0.69	-0.63	-0.66	-0.67	-0.70	-0.75	-0.77	0.64
s20	0.75	0.74	0.69	1.00	-0.14	-0.19	-0.29	-0.69	-0.69	-0.63	-0.65	-0.66	-0.71	-0.75	-0.77	0.63
s6	-0.16	-0.16	-0.14	-0.14	1.00	-0.00	0.02	0.16	0.15	0.12	0.13	0.13	0.15	0.15	0.16	-0.13
s14	-0.10	-0.11	-0.19	-0.19	-0.00	1.00	0.96	-0.15	-0.14	0.24	0.25	0.18	0.19	0.19	0.16	-0.31
s9	-0.21	-0.22	-0.29	-0.29	0.02	0.96	1.00	-0.03	-0.03	0.32	0.34	0.27	0.29	0.30	0.27	-0.39
s13	-0.79	-0.76	-0.69	-0.69	0.16	-0.15	-0.03	1.00	0.83	0.60	0.63	0.66	0.70	0.75	0.78	-0.56
s8	-0.79	-0.77	-0.69	-0.69	0.15	-0.14	-0.03	0.83	1.00	0.60	0.63	0.66	0.70	0.75	0.78	-0.56
s3	-0.68	-0.66	-0.63	-0.63	0.12	0.24	0.32	0.60	0.60	1.00	0.60	0.60	0.64	0.68	0.70	-0.58
s17	-0.70	-0.69	-0.66	-0.65	0.13	0.25	0.34	0.63	0.63	0.60	1.00	0.63	0.67	0.70	0.72	-0.61
s2	-0.72	-0.70	-0.67	-0.66	0.13	0.18	0.27	0.66	0.66	0.60	0.63	1.00	0.68	0.71	0.74	-0.61
s15	-0.77	-0.75	-0.70	-0.71	0.15	0.19	0.29	0.70	0.70	0.64	0.67	0.68	1.00	0.76	0.78	-0.64
s4	-0.82	-0.79	-0.75	-0.75	0.15	0.19	0.30	0.75	0.75	0.68	0.70	0.71	0.76	1.00	0.83	-0.68
s11	-0.85	-0.82	-0.77	-0.77	0.16	0.16	0.27	0.78	0.78	0.70	0.72	0.74	0.78	0.83	1.00	-0.70
ttf	0.67	0.66	0.64	0.63	-0.13	-0.31	-0.39	-0.56	-0.56	-0.58	-0.61	-0.61	-0.64	-0.68	-0.70	1.00

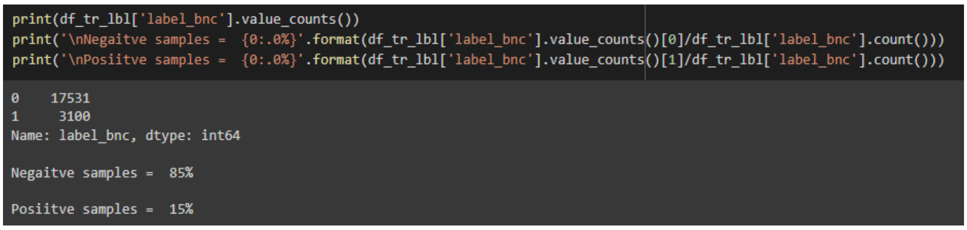
Graph 3: Features Correlation heatmap

The correlation between some features is very high (> 0.8). Some of these feature pairs are: (s4, s11), (s9, s14), (s7, s11), (s12, s4), (s12, s11), (s8, s13), (s12, s7). This can affect the performance of some ML algorithms. So, some of the features mentioned above are removed from the feature selection. Next, we plot a scatter matrix to display relationships and distribution among features and regression labels. A matrix of scatterplots is called a scatterplot matrix and is used to plot bivariate correlations between sets of variables. Scatterplots in the matrix show each association between pairs of variables, allowing you to examine many associations in one graph.



Graph 4: Scatter matrix to display relationships and distribution among features and regression labels.

Since most features have a gaussian distribution, ML algorithms benefit from this. Utilizing polynomial models may produce better results because maximum features have non-linear relationships with the label TTF. We also observe different input features and explore the time series plot each sensor selecting random sample engines. The regression label has non-linear relationships with maximum features, hence adding their polynomial transforms may enhance the performance of the model. At the end of EDA phase, we try to get some stats for the classification labels.



```
print(df_tr_lbl['label_bnc'].value_counts())
print('\nNegaitve samples =  {0:.0%}'.format(df_tr_lbl['label_bnc'].value_counts()[0]/df_tr_lbl['label_bnc'].count()))
print('\nPosiitve samples =  {0:.0%}'.format(df_tr_lbl['label_bnc'].value_counts()[1]/df_tr_lbl['label_bnc'].count()))
```

```
0    17531
1     3100
Name: label_bnc, dtype: int64

Negaitve samples =  85%

Posiitve samples =  15%
```

We should not rely on classification Accuracy as a model performance indicator because this dataset is obviously skewed. Instead, we can employ AUC ROC.

Feature Importance: Variable significance decides the contribution of each variable to the prediction capability of the model. Essentially, it determines how useful a specific variable is for a particular model and prediction. Scores are typically used to quantify feature importance. The larger the score, the higher the importance of the trait. Variable importance scores have many benefits. For example, you can determine the relationship between a feature variable (characteristic) and a target variable (goal).

By examining the significance values of the variables, irrelevant features can be identified and eliminated. Reducing the number of irrelevant variables can make your model run faster or even perform better. From the results, we can infer why the ML model predicts certain things and how changing its features can change that prediction.

There are many ways of calculating feature importance, but generally, we can divide them into two groups:

- Model agnostic
- Model dependent

3.3.3 Regression Analysis

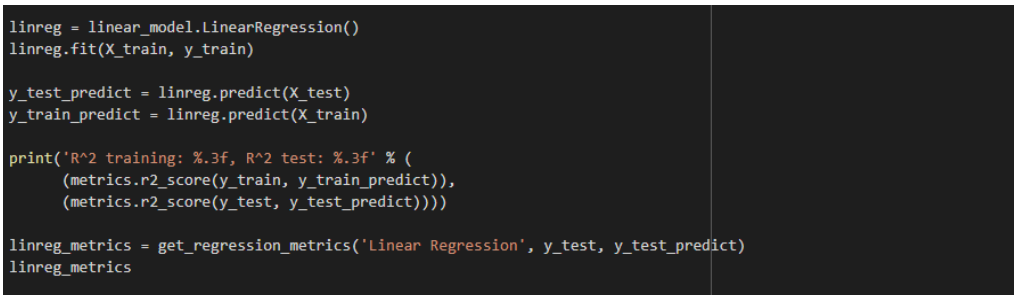
The technique for determining the relationship between target and independent variables in a dataset is regression analysis. It is commonly used when the target variable has a continuous range of values and the target and independent variables are in a linear or nonlinear relationship.

Regression analysis techniques are used to predict, model, and display causal associations among the variables across time series. Regression analysis is used to predict either the value of the target variable or the effect of an independent variable on the target variable when knowledge of the independent variables is available.

Various regression analysis prediction techniques are available. Other elements that affect the choice of the technique include the number of independent variables, the pattern of the regression line, and the type of target variable.

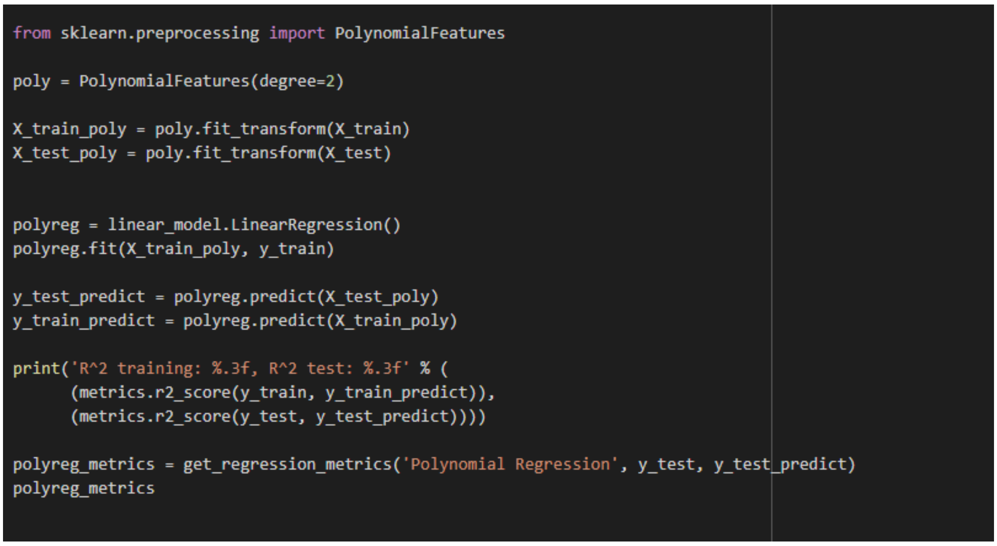
Linear Regression: This modeling technique assumes that an independent variable (V) and a dependent variable (Y) have a linear relationship (X). It generates a best-fit line. Y = c+m*X + e, where m denotes the slope, e denotes the error factor, and c denotes the intercept, is the equation for the linear regression.

The amount of dependent and independent variables vary between the simple and sophisticated versions of the linear regression model.



```
linreg = linear_model.LinearRegression()
linreg.fit(X_train, y_train)

y_test_predict = linreg.predict(X_test)
y_train_predict = linreg.predict(X_train)

print('R^2 training: %.3f, R^2 test: %.3f' % (
      (metrics.r2_score(y_train, y_train_predict)),
      (metrics.r2_score(y_test, y_test_predict))))

linreg_metrics = get_regression_metrics('Linear Regression', y_test, y_test_predict)
linreg_metrics
```

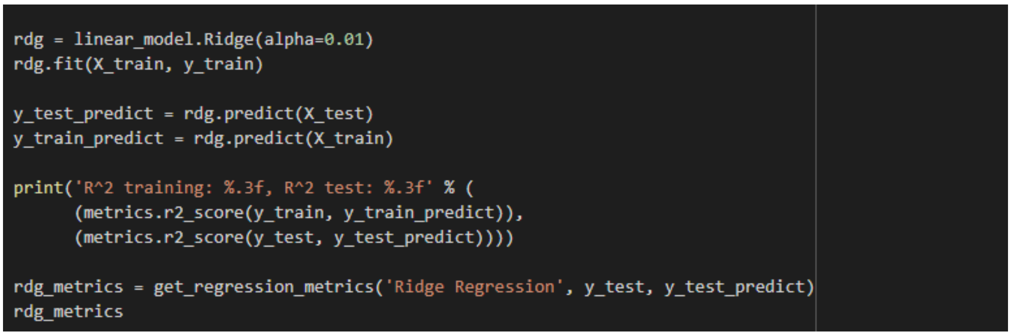
Polynomial Regression: Analysis using polynomial regression is performed to show the non- linear relationship between the dependent and independent variables. In this type of multiple linear regression model, the line of best fit is not straight but curved.



```
from sklearn.preprocessing import PolynomialFeatures

poly = PolynomialFeatures(degree=2)

X_train_poly = poly.fit_transform(X_train)
X_test_poly = poly.fit_transform(X_test)


polyreg = linear_model.LinearRegression()
polyreg.fit(X_train_poly, y_train)

y_test_predict = polyreg.predict(X_test_poly)
y_train_predict = polyreg.predict(X_train_poly)

print('R^2 training: %.3f, R^2 test: %.3f' % (
      (metrics.r2_score(y_train, y_train_predict)),
      (metrics.r2_score(y_test, y_test_predict))))

polyreg_metrics = get_regression_metrics('Polynomial Regression', y_test, y_test_predict)
polyreg_metrics
```

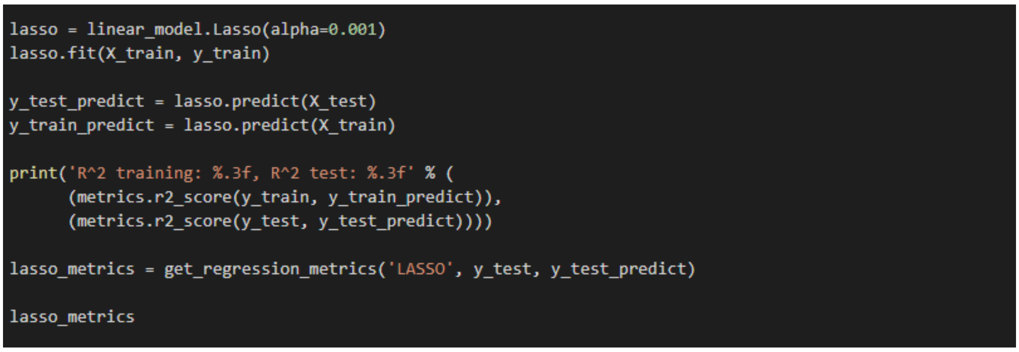
Ridge Regression: This technique is employed when the independent variables are highly correlated and the data we are working with exhibits multicollinearity. Even if the least- squares estimates are fair to multicollinearity, their variances are considerable enough to cause differences between the observed and true values. By adjusting the estimates of regression, ridge regression lowers the standard error. The issue of multicollinearity in the ridge regression equation is solved by the lambda variable.

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$



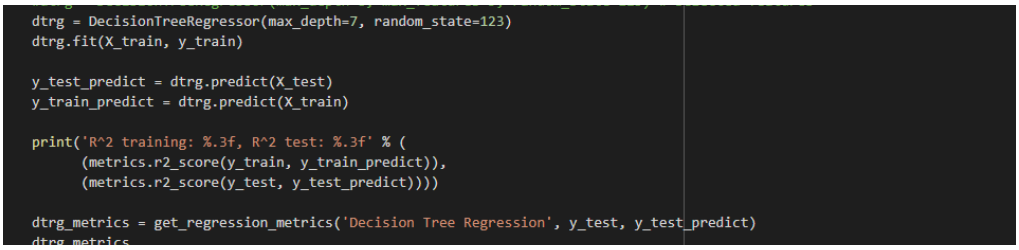
```
rdg = linear_model.Ridge(alpha=0.01)
rdg.fit(X_train, y_train)

y_test_predict = rdg.predict(X_test)
y_train_predict = rdg.predict(X_train)

print('R^2 training: %.3f, R^2 test: %.3f' % (
      (metrics.r2_score(y_train, y_train_predict)),
      (metrics.r2_score(y_test, y_test_predict))))

rdg_metrics = get_regression_metrics('Ridge Regression', y_test, y_test_predict)
rdg_metrics
```

Lasso Regression: The Lasso method penalizes the absolute size of the regression coefficients, similar to Ridge regression. The LASSO regression method also uses variable selection to drive the coefficient values to zero.



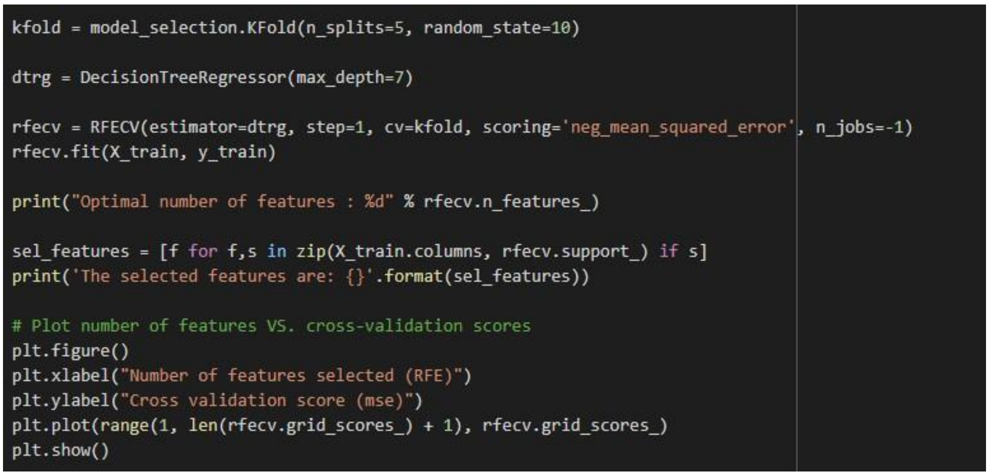
```
lasso = linear_model.Lasso(alpha=0.001)
lasso.fit(X_train, y_train)

y_test_predict = lasso.predict(X_test)
y_train_predict = lasso.predict(X_train)

print('R^2 training: %.3f, R^2 test: %.3f' % (
      (metrics.r2_score(y_train, y_train_predict)),
      (metrics.r2_score(y_test, y_test_predict))))

lasso_metrics = get_regression_metrics('LASSO', y_test, y_test_predict)

lasso_metrics
```

DT Regression: By looking into the element characteristics, DT regression trains a model in tree form to predict future dates and generate valuable ongoing results. The lack of discontinuous output, or being represented by a unique set of known numbers or values, is called continuous output.

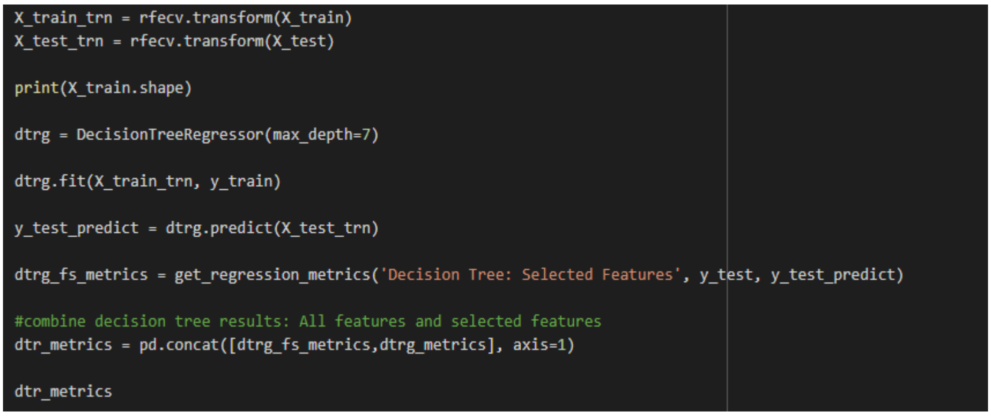


```
dtrg = DecisionTreeRegressor(max_depth=7, random_state=123)
dtrg.fit(X_train, y_train)

y_test_predict = dtrg.predict(X_test)
y_train_predict = dtrg.predict(X_train)

print('R^2 training: %.3f, R^2 test: %.3f' % (
      (metrics.r2_score(y_train, y_train_predict)),
      (metrics.r2_score(y_test, y_test_predict))))

dtrg_metrics = get_regression_metrics('Decision Tree Regression', y_test, y_test_predict)
dtrg_metrics
```

We try to optimize our decision tree regressor by using RFE. It is a feature selection technique that eliminates the least required feature (or features) from a model until the desired number of features is reached. RFE aims to eliminate any dependencies and collinearity that exist in the model by repeatedly deleting a limited number of features every cycle according to the model's coef_ or feature importances_ characteristics.

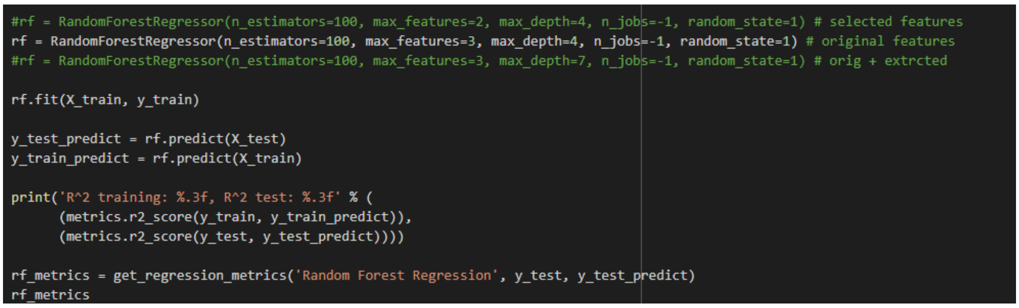


```
kfold = model_selection.KFold(n_splits=5, random_state=10)

dtrg = DecisionTreeRegressor(max_depth=7)

rfecv = RFECV(estimator=dtrg, step=1, cv=kfold, scoring='neg_mean_squared_error', n_jobs=-1)
rfecv.fit(X_train, y_train)

print("Optimal number of features : %d" % rfecv.n_features_)

sel_features = [f for f,s in zip(X_train.columns, rfecv.support_) if s]
print('The selected features are: {}'.format(sel_features))

# Plot number of features VS. cross-validation scores
plt.figure()
plt.xlabel("Number of features selected (RFE)")
plt.ylabel("Cross validation score (mse)")
plt.plot(range(1, len(rfecv.grid_scores_) + 1), rfecv.grid_scores_)
plt.show()
```



```
X_train_trn = rfecv.transform(X_train)
X_test_trn = rfecv.transform(X_test)

print(X_train.shape)

dtrg = DecisionTreeRegressor(max_depth=7)

dtrg.fit(X_train_trn, y_train)

y_test_predict = dtrg.predict(X_test_trn)

dtrg_fs_metrics = get_regression_metrics('Decision Tree: Selected Features', y_test, y_test_predict)

#combine decision tree results: All features and selected features
dtr_metrics = pd.concat([dtrg_fs_metrics,dtrg_metrics], axis=1)

dtr_metrics
```

Random Forest Regressor: Random forests, which use multiple decision trees and a method called bootstrapping and aggregation (also called bagging), are ensemble techniques that can handle both regression and classification tasks. The basic principle of this method is to integrate multiple DTs to get the final result instead of relying on only one DT. Some DTs act as basic learning models for random forests.



```
#rf = RandomForestRegressor(n_estimators=100, max_features=2, max_depth=4, n_jobs=-1, random_state=1) # selected features
rf = RandomForestRegressor(n_estimators=100, max_features=3, max_depth=4, n_jobs=-1, random_state=1) # original features
#rf = RandomForestRegressor(n_estimators=100, max_features=3, max_depth=7, n_jobs=-1, random_state=1) # orig + extrcted

rf.fit(X_train, y_train)

y_test_predict = rf.predict(X_test)
y_train_predict = rf.predict(X_train)

print('R^2 training: %.3f, R^2 test: %.3f' % (
      (metrics.r2_score(y_train, y_train_predict)),
      (metrics.r2_score(y_test, y_test_predict))))

rf_metrics = get_regression_metrics('Random Forest Regression', y_test, y_test_predict)
rf_metrics
```

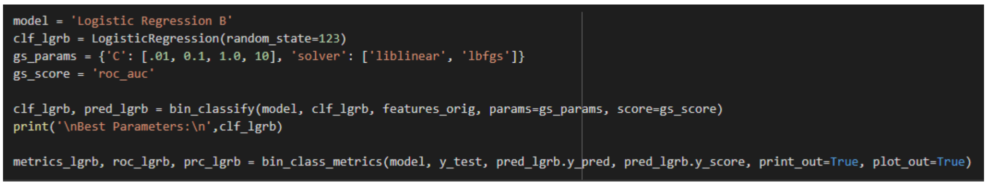
3.3.4 Classification Analysis

Machine learning programs utilize various algorithms to classify datasets to be used at later time into categories using pre-categorized training datasets. Classification is the process of recognizing, interpreting, and grouping data into predetermined groups. The two types of classification algorithms are:

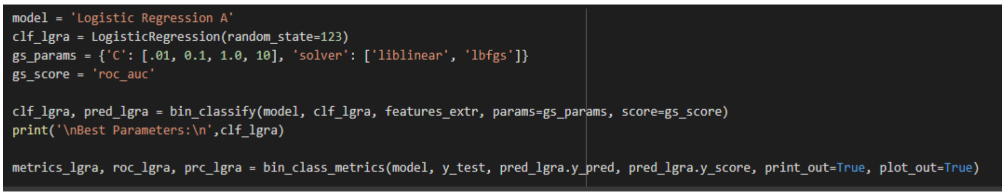
- Binary classification algorithms.
- Multi-class classification algorithms.

Binary classification classifies the data into two groups. Mostly, these two groups consist of '0' and '1'. In multiclass classification, on the other hand, there are more than two classes.

Logistic Regression: It is used to predict binary outcomes. It analyzes the independent variables to determine the binary outcome, the outcome he falls into is one of two categories. Independent variables can be categorical or numeric, but dependent variables are always categorical..

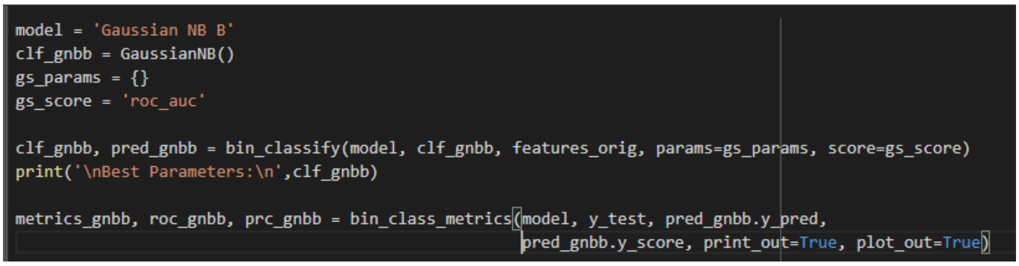


```
model = 'Logistic Regression B'
clf_lgrb = LogisticRegression(random_state=123)
gs_params = {'C': [.01, 0.1, 1.0, 10], 'solver': ['liblinear', 'lbfgs']}
gs_score = 'roc_auc'

clf_lgrb, pred_lgrb = bin_classify(model, clf_lgrb, features_orig, params=gs_params, score=gs_score)
print('\nBest Parameters:\n',clf_lgrb)

metrics_lgrb, roc_lgrb, prc_lgrb = bin_class_metrics(model, y_test, pred_lgrb.y_pred, pred_lgrb.y_score, print_out=True, plot_out=True)
```

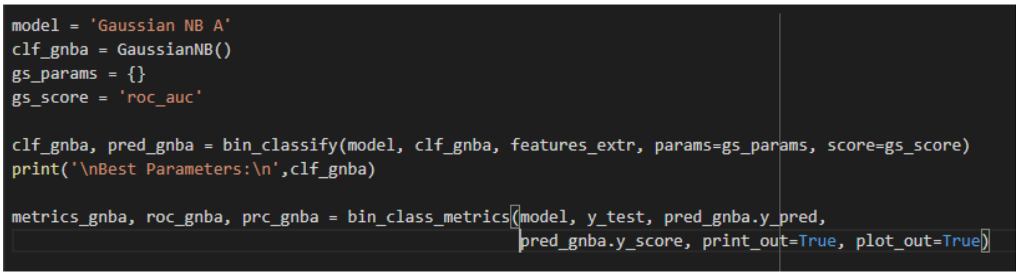


```
model = 'Logistic Regression A'
clf_lgra = LogisticRegression(random_state=123)
gs_params = {'C': [.01, 0.1, 1.0, 10], 'solver': ['liblinear', 'lbfgs']}
gs_score = 'roc_auc'

clf_lgra, pred_lgra = bin_classify(model, clf_lgra, features_extr, params=gs_params, score=gs_score)
print('\nBest Parameters:\n',clf_lgra)

metrics_lgra, roc_lgra, prc_lgra = bin_class_metrics(model, y_test, pred_lgra.y_pred, pred_lgra.y_score, print_out=True, plot_out=True)
```

Naive Bayes: Naive Bayes predicts the probability of a data point to fall into a specific category or not.

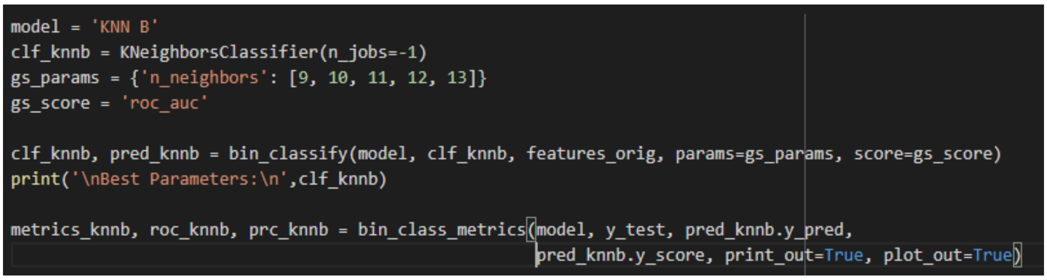


```
model = 'Gaussian NB B'
clf_gnbb = GaussianNB()
gs_params = {}
gs_score = 'roc_auc'

clf_gnbb, pred_gnbb = bin_classify(model, clf_gnbb, features_orig, params=gs_params, score=gs_score)
print('\nBest Parameters:\n',clf_gnbb)

metrics_gnbb, roc_gnbb, prc_gnbb = bin_class_metrics(model, y_test, pred_gnbb.y_pred,
                                                     pred_gnbb.y_score, print_out=True, plot_out=True)
```

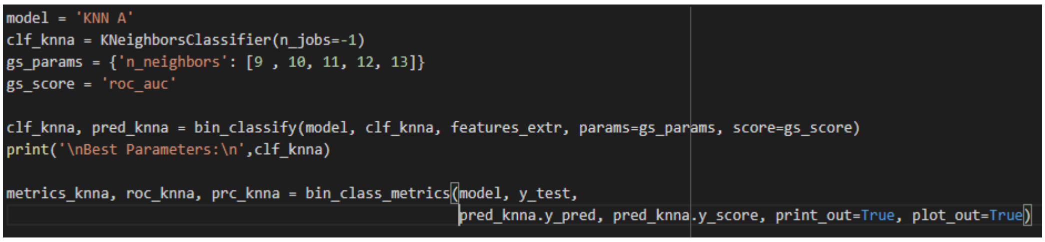


```
model = 'Gaussian NB A'
clf_gnba = GaussianNB()
gs_params = {}
gs_score = 'roc_auc'

clf_gnba, pred_gnba = bin_classify(model, clf_gnba, features_extr, params=gs_params, score=gs_score)
print('\nBest Parameters:\n',clf_gnba)

metrics_gnba, roc_gnba, prc_gnba = bin_class_metrics(model, y_test, pred_gnba.y_pred,
                                                     pred_gnba.y_score, print_out=True, plot_out=True)
```

K-nearest Neighbours: A pattern identification approach called K-nearest neighbors uses the training dataset to identify the k nearest neighbors in new cases. When using a k-NN for classification, find where to place the data relative to its nearest neighbors.

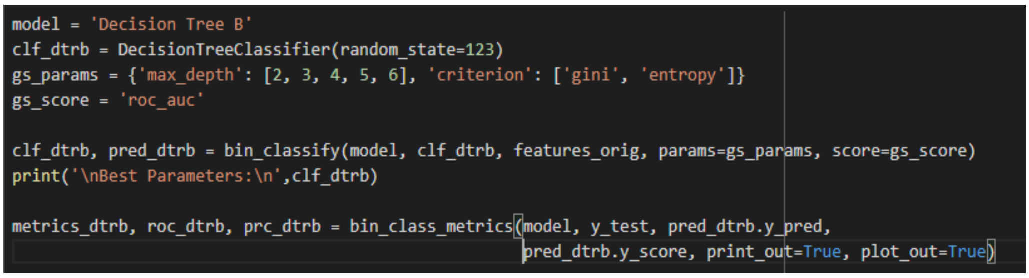


```
model = 'KNN B'
clf_knnb = KNeighborsClassifier(n_jobs=-1)
gs_params = {'n_neighbors': [9, 10, 11, 12, 13]}
gs_score = 'roc_auc'

clf_knnb, pred_knnb = bin_classify(model, clf_knnb, features_orig, params=gs_params, score=gs_score)
print('\nBest Parameters:\n',clf_knnb)

metrics_knnb, roc_knnb, prc_knnb = bin_class_metrics(model, y_test, pred_knnb.y_pred,
                                                     pred_knnb.y_score, print_out=True, plot_out=True)
```

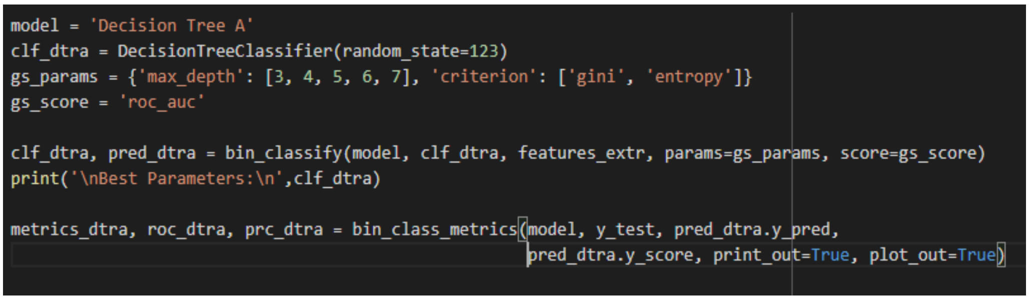


```
model = 'KNN A'
clf_knna = KNeighborsClassifier(n_jobs=-1)
gs_params = {'n_neighbors': [9 , 10, 11, 12, 13]}
gs_score = 'roc_auc'

clf_knna, pred_knna = bin_classify(model, clf_knna, features_extr, params=gs_params, score=gs_score)
print('\nBest Parameters:\n',clf_knna)

metrics_knna, roc_knna, prc_knna = bin_class_metrics(model, y_test,
                                                     pred_knna.y_pred, pred_knna.y_score, print_out=True, plot_out=True)
```

Decision trees are an ideal supervised learning technique for classification problems because they can classify classes accurately. Like a flowchart, it splits the data points into two similar categories simultaneously. Beginning with 'trunk', going through 'branches' and 'leaves', the categories become more closely related. The result is a category within a category, allowing organic taxonomy with the least manual oversight.

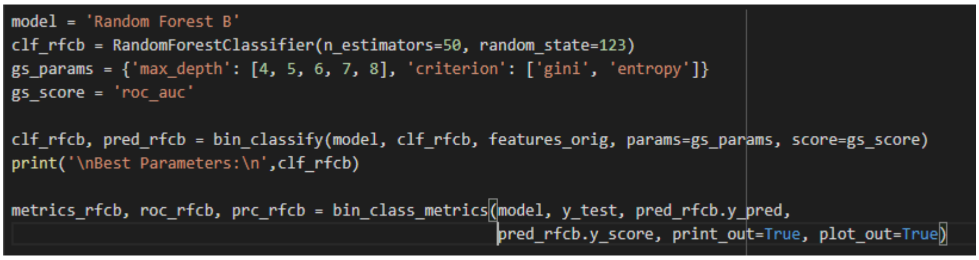


```
model = 'Decision Tree B'
clf_dtrb = DecisionTreeClassifier(random_state=123)
gs_params = {'max_depth': [2, 3, 4, 5, 6], 'criterion': ['gini', 'entropy']}
gs_score = 'roc_auc'

clf_dtrb, pred_dtrb = bin_classify(model, clf_dtrb, features_orig, params=gs_params, score=gs_score)
print('\nBest Parameters:\n',clf_dtrb)

metrics_dtrb, roc_dtrb, prc_dtrb = bin_class_metrics(model, y_test, pred_dtrb.y_pred,
                                                     pred_dtrb.y_score, print_out=True, plot_out=True)
```

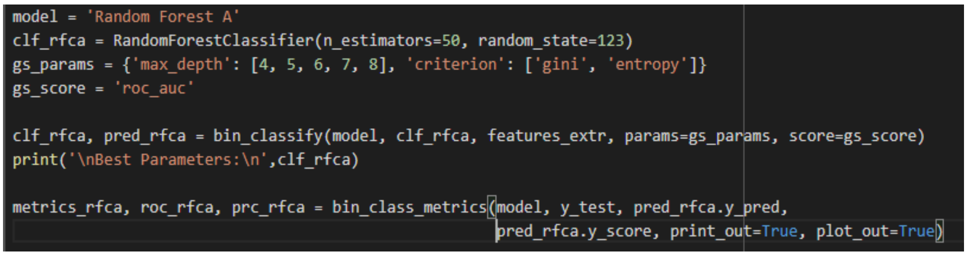


```
model = 'Decision Tree A'
clf_dtra = DecisionTreeClassifier(random_state=123)
gs_params = {'max_depth': [3, 4, 5, 6, 7], 'criterion': ['gini', 'entropy']}
gs_score = 'roc_auc'

clf_dtra, pred_dtra = bin_classify(model, clf_dtra, features_extr, params=gs_params, score=gs_score)
print('\nBest Parameters:\n',clf_dtra)

metrics_dtra, roc_dtra, prc_dtra = bin_class_metrics(model, y_test, pred_dtra.y_pred,
                                                     pred_dtra.y_score, print_out=True, plot_out=True)
```

Random Forest: This model is a development of a decision tree, where you first create a huge amount of DTs with training data, then fit your fresh data within one of the DTs as a "random forest." To connect your data to the nearest tree on the data scale, it simply averages your data. This model is useful because it addresses the DT’s issue of excessively "pushing" data points into a category.

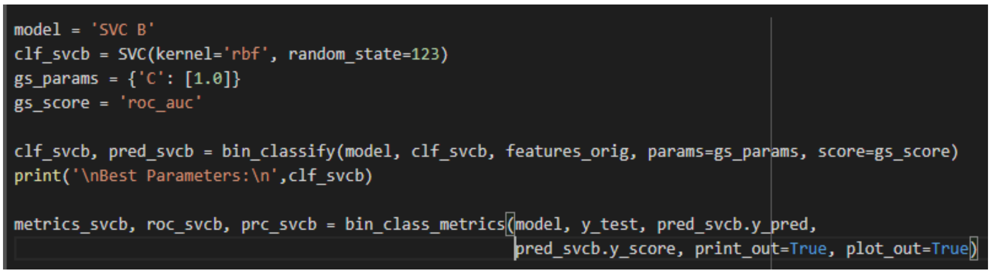


```
model = 'Random Forest B'
clf_rfcb = RandomForestClassifier(n_estimators=50, random_state=123)
gs_params = {'max_depth': [4, 5, 6, 7, 8], 'criterion': ['gini', 'entropy']}
gs_score = 'roc_auc'

clf_rfcb, pred_rfcb = bin_classify(model, clf_rfcb, features_orig, params=gs_params, score=gs_score)
print('\nBest Parameters:\n',clf_rfcb)

metrics_rfcb, roc_rfcb, prc_rfcb = bin_class_metrics(model, y_test, pred_rfcb.y_pred,
                                                     pred_rfcb.y_score, print_out=True, plot_out=True)
```

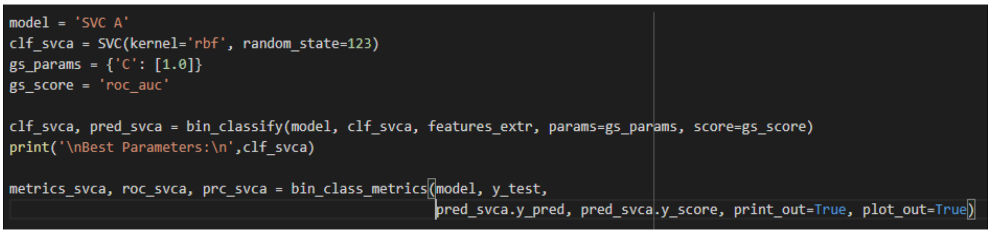


```
model = 'Random Forest A'
clf_rfca = RandomForestClassifier(n_estimators=50, random_state=123)
gs_params = {'max_depth': [4, 5, 6, 7, 8], 'criterion': ['gini', 'entropy']}
gs_score = 'roc_auc'

clf_rfca, pred_rfca = bin_classify(model, clf_rfca, features_extr, params=gs_params, score=gs_score)
print('\nBest Parameters:\n',clf_rfca)

metrics_rfca, roc_rfca, prc_rfca = bin_class_metrics(model, y_test, pred_rfca.y_pred,
                                                     pred_rfca.y_score, print_out=True, plot_out=True)
```

SVM : A Support Vector Machine exceeds A/B prediction by using techniques to train and classify data according to degrees of variation. A hyperplane that optimally splits up the tags is chosen by the SVM. It is only a line in two dimensions. Class 1 items fall on one side of the line, and class 2 items fall on the other. For example, this would be good and negative in sentiment analysis. The optimal hyperplane has the greatest distance between each tag in order to maximize machine learning. However, it might not be viable to classify the data with a single line as data sets grow more complicated. Using this algorithm, the more complex the data, the more precise the predictor will become. Since Support Vector Machine is multidimensional, it allows for more precise machine learning.



```
model = 'SVC B'
clf_svcb = SVC(kernel='rbf', random_state=123)
gs_params = {'C': [1.0]}
gs_score = 'roc_auc'

clf_svcb, pred_svcb = bin_classify(model, clf_svcb, features_orig, params=gs_params, score=gs_score)
print('\nBest Parameters:\n',clf_svcb)

metrics_svcb, roc_svcb, prc_svcb = bin_class_metrics(model, y_test, pred_svcb.y_pred,
                                                     pred_svcb.y_score, print_out=True, plot_out=True)
```



```
model = 'SVC A'
clf_svca = SVC(kernel='rbf', random_state=123)
gs_params = {'C': [1.0]}
gs_score = 'roc_auc'

clf_svca, pred_svca = bin_classify(model, clf_svca, features_extr, params=gs_params, score=gs_score)
print('\nBest Parameters:\n',clf_svca)

metrics_svca, roc_svca, prc_svca = bin_class_metrics(model, y_test,
                                                     pred_svca.y_pred, pred_svca.y_score, print_out=True, plot_out=True)
```

Multiclass classification: A machine learning classification problem called "multiclass classification" has more than two classes or outputs. Multiclass classification is perhaps the most frequent machine learning job, excluding regression. In classification, we design a machine learning model to determine which of the K different classes some previously unobserved data belongs to after being presented with a large number of training samples (ie. the animal types from the previous example). By analysing the training dataset, the model discovers patterns unique to each class, which it then employs to forecast the membership of upcoming data. In order to subset the original data frames into original features and original + extracted features, we first create the feature sets needed. Additionally, we build a series of training and test data labels and convert them to binary format so that the multiclass classification algorithms may use them. We design a helper function that will adjust a classifier's Grid Search hyperparameters in a manner similar to binary classification. We also Calculate main multiclass classification metrics, plot AUC ROC and Precision-Recall curves. For multiclass classification, we employ different classification algorithms - SVC, Decision tree classifier, Naive Bayes, KNN, as well as Neural Net MLP. We apply these algorithms on two different datasets - one in which we have the original features and the other in which we have only the extracted features.

Chapter 4: PERFORMANCE ANALYSIS

4.1 Regression Analysis

1. Root mean square error is the standard deviation of the prediction errors. We use residuals to compute the distance between the data points and the regression line and the variability of these residuals is calculated by RMSE. In other words, it describes the density of data around the best fit line. The equation is:

$$RMSE = \sqrt{(f - o)^2}$$

Where: f = prediction (unknown outcomes) and o = known outcome.

2. Absolute error is the amount of error in a measurement. The difference between the computed value and the actual value. The equation for absolute error (Δx): (Δx) = xi - x, where: xi is the measured value and x is the true value.

Mean Absolute Error is the mean of all absolute errors. The equation is:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} | x_i - x |$$

Where n = the number of errors, Σ = summation symbol (which means “add them all up”),

|xi – x| = the absolute errors.

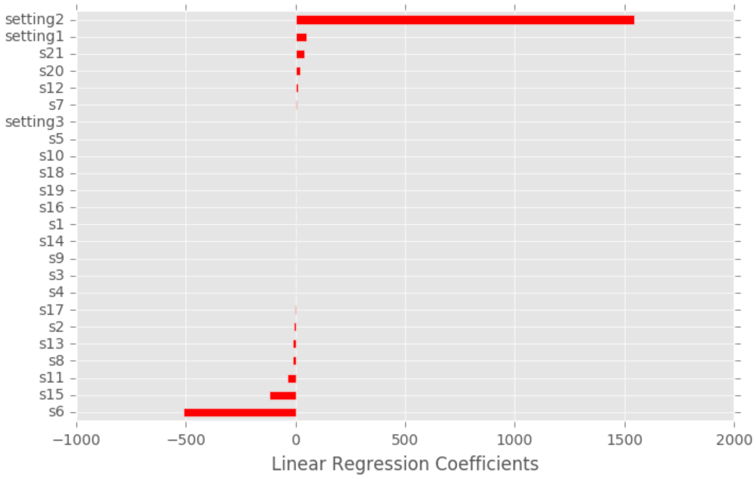
3. The "standardised version of MSE" is R-Squared. Instead of MSE to get the error, R- Squared shows the percentage of variance of the actual value of the response obtained by the regression model. For assessing the effectiveness of regression models, R-Squared or rather adjusted R-Squared is advised. This is mostly due to the fact that R-Squared captures the percentage of variation of real values captured by the regression model and has a tendency to provide a more accurate representation of the regression model's accuracy. Whether or not the response variable's values are scaled also affects the MSE values. The RMSE, which accounts for the issue of whether the values of the response variable are scaled or not, is a superior metric than the mean squared error (MSE).

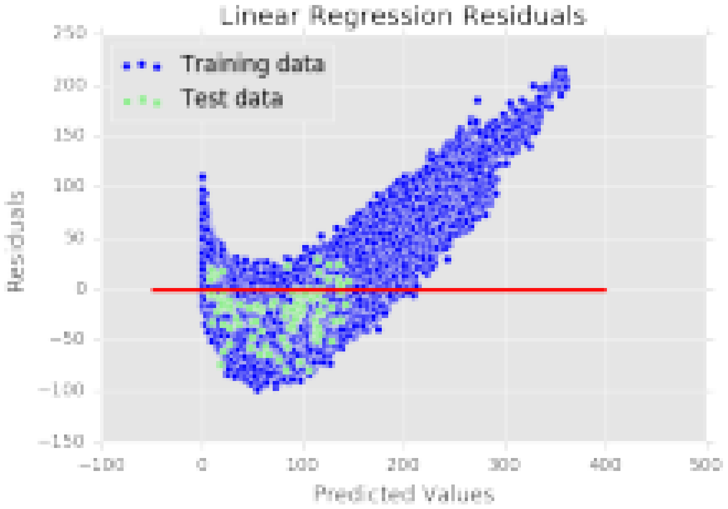
4. The difference between a model's predictions and the actual data is quantified using explained variance. It can be said that it is the portion of the model's total variation that can be accounted for by genuine components rather than error variance. A stronger strength of correlation is indicated by higher explained variance percentages. It also implies that your predictions are more accurate.

Table 2: Regression metrics value for different regression algorithms

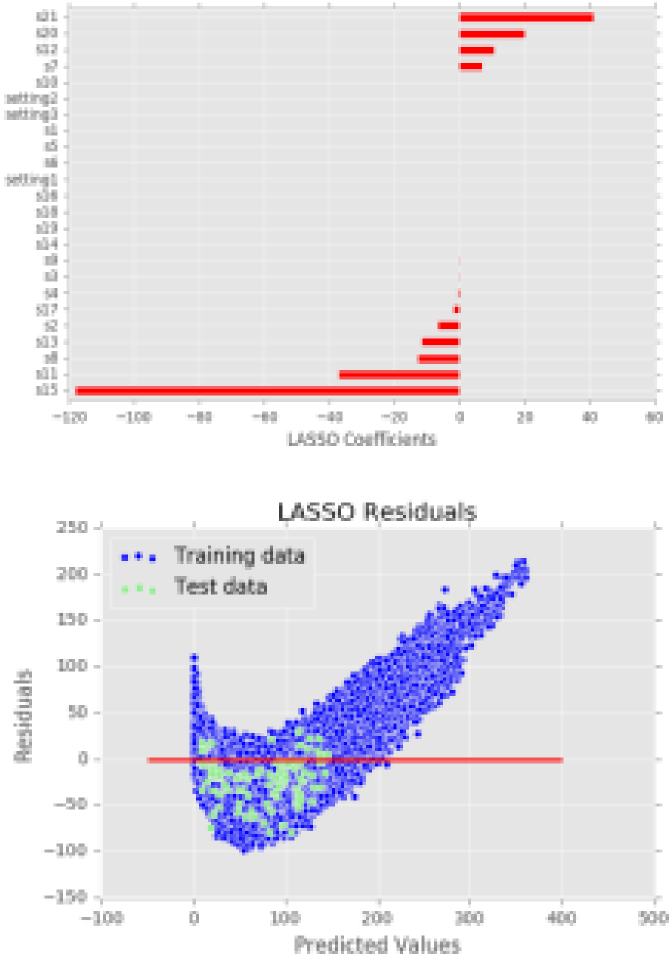
Regression Algorithm	Root Mean Squared Error	Mean Absolute Error	R^2 (training)	R^2 (test)	Explained Variance
Linear Regression	32.041095	25.591780	0.580	0.405495	0.665297

Lasso Regression	31.966099	25.551808	0.579	0.408	0.668206
Ridge Regression	31.965740	25.544620	0.580	0.408289	0.667607
Polynomial Regression	29.677417	22.377344	0.626	0.489974	0.645374
Decision Tree Regression	32.095349	24.319068	0.625	0.403480	0.632767
Decision Tree with selected features	34.212392	25.866117	0.580	0.322191	0.593892
Random Forest Regression	28.634253	23.167130	0.594	0.525198	0.767320

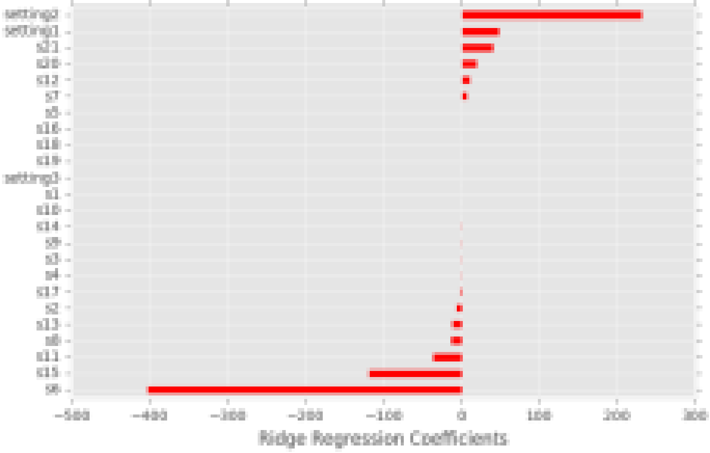


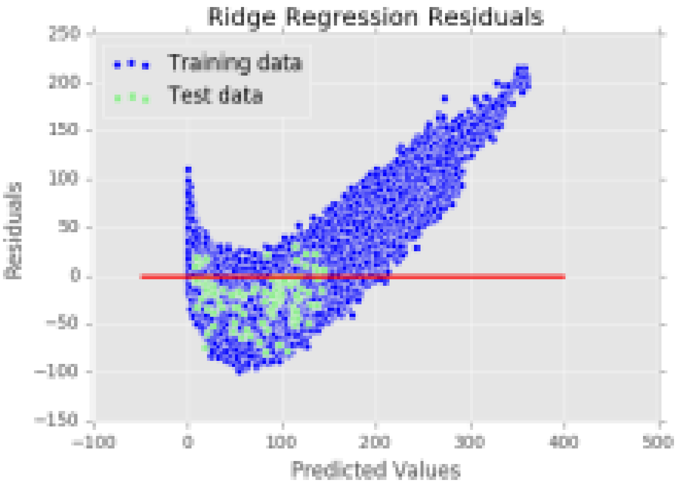


Graph 5 and 6: Coefficients and Residuals for linear regression model

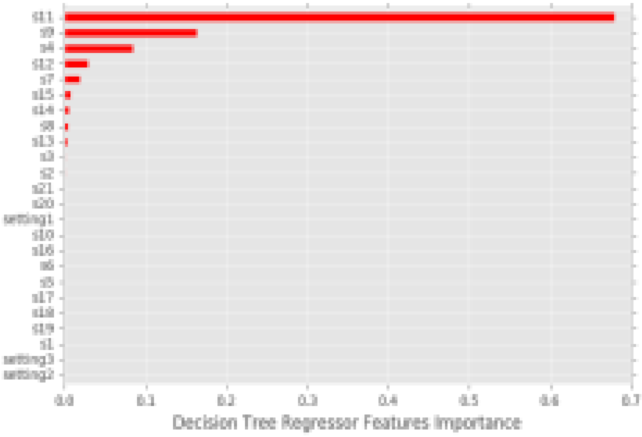


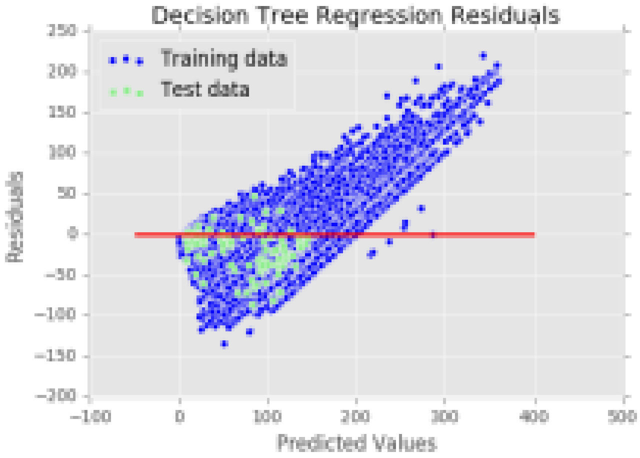
Graph 7 and 8: Coefficients and Residuals for lasso regression model



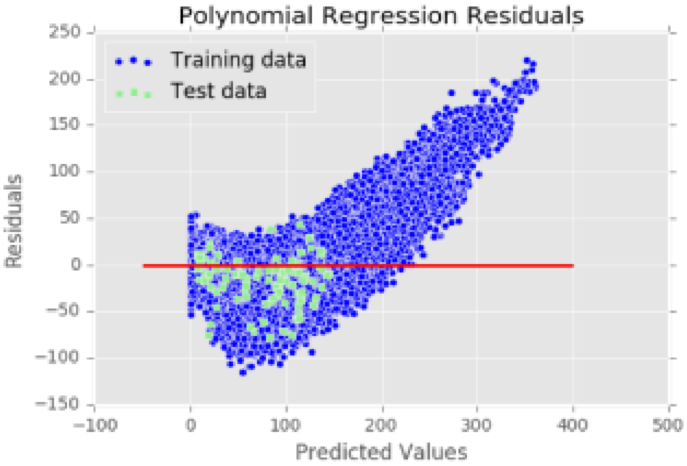


Graph 9 and 10: Coefficients and Residuals for ridge regression model



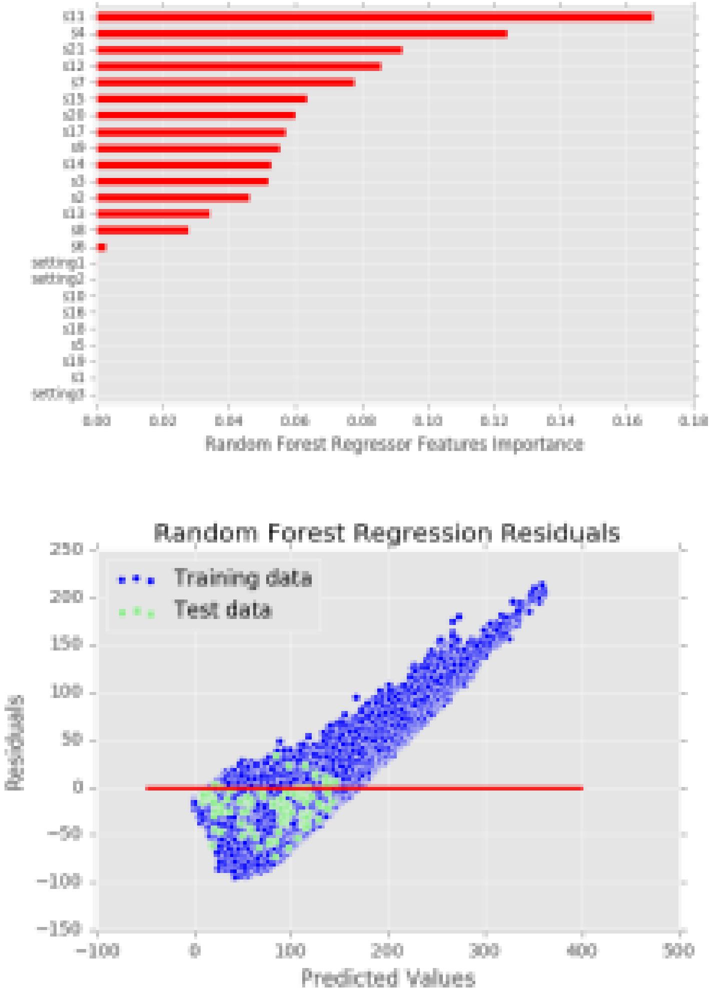


Graph 11 and 12: Coefficients and Residuals for decision tree regression model



Graph 13: Polynomial Regression residuals Polynomial

Regression has scored better than linear models



Graph 14 and 15: Coefficients and Residuals for random forest regressor

It was observed that the regression residuals were not randomly distributed around the mean of the residuals. We could have improved this by fixing the data (e.g. fixing or removing outliers, resampling) or tuning the model parameters. Our study during the data exploration phase showed that linear models, such as Linear, Ridge and Lasso regression were outperformed by non-linear regression models, such as Random Forest and Polynomial Regression. The model predicts TTF within an average error range of 28.63 cycles, clearly outperforming other models with root mean squared error of RF regressor being 28.63 cycles.

The manual hyper-parameter adjustment for the RF Regressor, Ridge and lasso models might have been processed more effectively with Grid Search or Random Search with Cross Validation.

4.2 Classification Analysis

In model names: B stands for applying the model on the original set of features, before feature extraction and A stands for applying the model on the original + extracted features set, After feature extraction

Classification Metrics

The percentage of precisely identified data points out of all the instances is called accuracy.

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

If the dataset is unbalanced, accuracy might not be an acceptable metric (both negative and positive classes have different number of data instances).

Precision is formulated as :

$$Precision = \frac{TP}{TP + FP}$$

The definition of TPR, also called sensitivity or recall, is:

$$Recall = \frac{TP}{TP + FN}$$

The ideal TPR and precision of a competent classifier are 1, which indicates that FN and FP are nil. Therefore, we must use statistics that consider both precision and recall. Definition of the F1 score, a statistic that considers both precision and recall:

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The harmonic mean of precision and sensitivity is referred to as the F1 score and is a better measure than precision. F1 scores are high only when the values for both precision and TPR are high.

AUC-ROC curves are used to evaluate the performance of multiclass classification problems. It is an essential criterion for evaluating the potency of classification models.

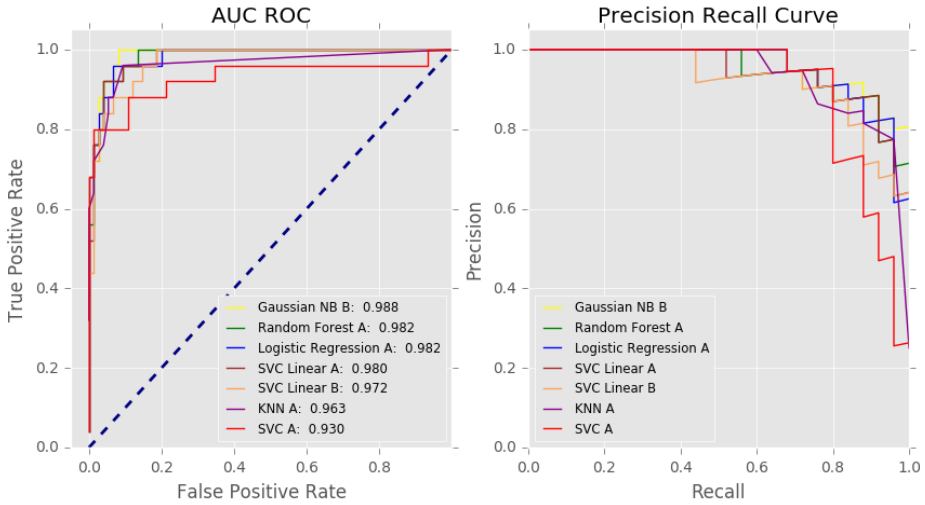
The area under the Curve stands for the degree of separability and the Receiver Operating Characteristic is the probability curve. This shows how carefully our model can distinguish classes. The higher the AUC, the more accurately the model classifies. Similarly, the higher the area under the curve, the better the model differentiation.

Table 3: Classification metrics value for different algorithms

Classification Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
Logistic Regression B	0.880000	0.933333	0.560000	0.700000	0.980267
Logistic Regression A	0.920000	1.000000	0.680000	0.809521	0.981867

Decision Tree B	0.880000	0.933333	0.560000	0.700000	0.945067
Decision Tree A	0.920000	0.947368	0.720000	0.818182	0.962933
Random Forest B	0.910000	0.944444	0.680000	0.790698	0.980267
Random Forest A	0.910000	0.944444	0.680000	0.790698	0.982400
SVC B	0.910000	0.944444	0.680000	0.790698	0.891733
SVC A	0.920000	0.947368	0.720000	0.818182	0.930133
SVC Linear B	0.850000	1.000000	0.400000	0.571429	0.971733
SVC Linear A	0.670000	0.431034	1.000000	0.602410	0.979733
KNN B	0.910000	0.944444	0.680000	0.790698	0.935200
KNN A	0.920000	0.947368	0.720000	0.818182	0.963467
Gaussian NB B	0.940000	0.827586	0.960000	0.888889	0.987733
Gaussian NB A	0.940000	0.827586	0.960000	0.888889	0.980533

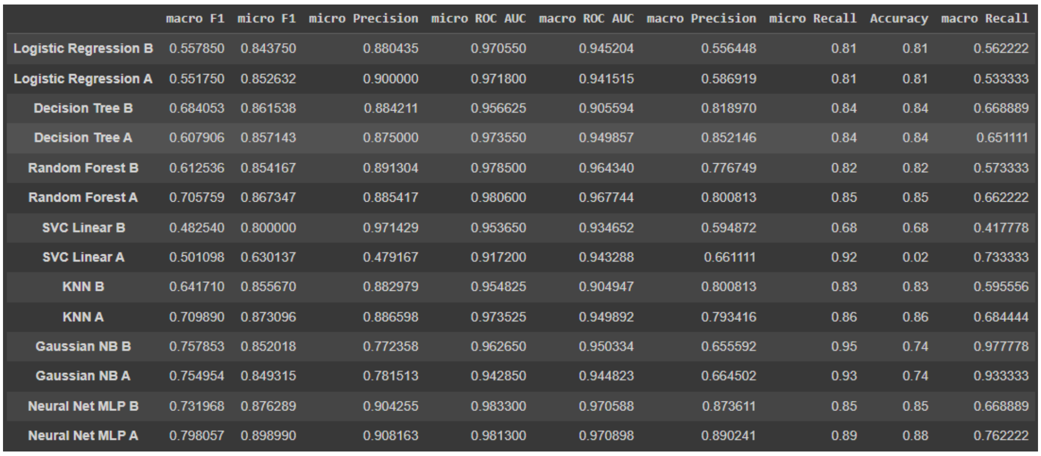
Random Forests and Naive Bayes scored the highest AUC ROC. We also found that feature extraction improved performance metrics for most models.



Graph 16: AUC-ROC and Precision-Recall curve for different classification algorithms

Binary classification summary:

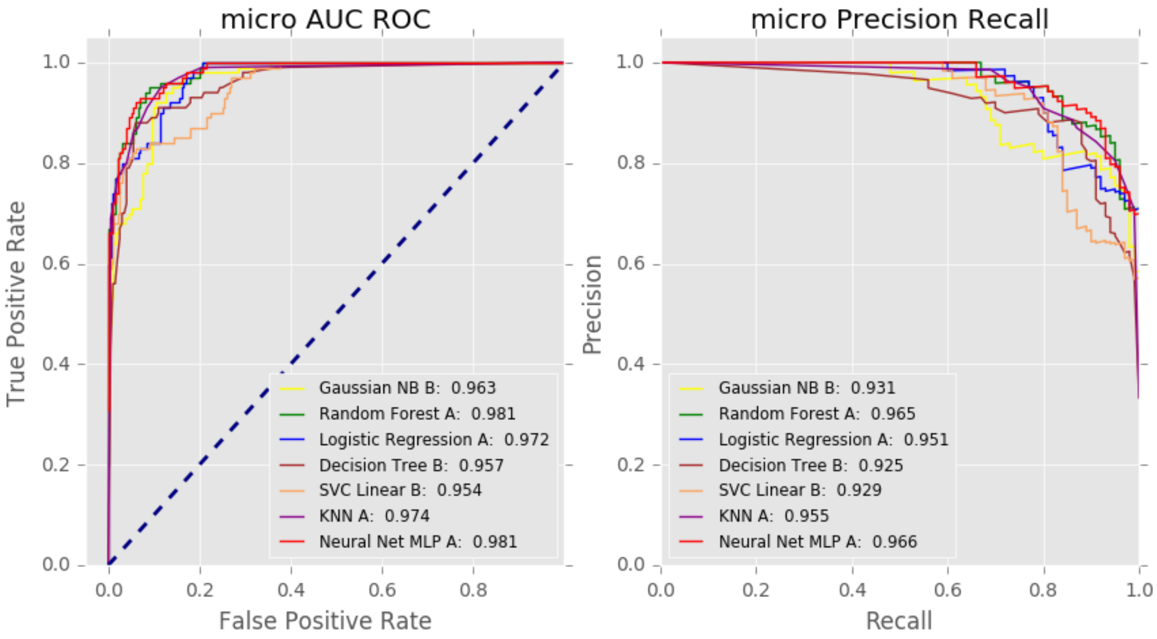
- With the addition of new features, it was observed that most binary classifiers showed better performance metrics.
- Random Forest and Naive Bayes performed similar pre-feature engineering and post- feature engineering.
- Naive Bayes and Linear SVC performed better on Recall (Sensitivity) than other classifiers, while the other algorithms performed better on Precision.
- Linear SVC has completely distinct evaluation indexes pre-FE and post-FE, and switches between Recall and Precision.
- SVC (RBF) has the least AUROC, but the highest precision sensitivity curve operating at a threshold of 0.17, giving nil precision and recall for the engine.
- Associate TPR, FPR, and Engine charts with (True positive, False positive, True negative, and False negative) cost matrices to compute the anticipated value at various operational thresholds to optimize strategic decision-making need to do it.



	macro F1	micro F1	micro Precision	micro ROC AUC	macro ROC AUC	macro Precision	micro Recall	Accuracy	macro Recall
Logistic Regression B	0.557850	0.843750	0.880435	0.970550	0.945204	0.556448	0.81	0.81	0.562222
Logistic Regression A	0.551750	0.852632	0.900000	0.971800	0.941515	0.586919	0.81	0.81	0.533333
Decision Tree B	0.684053	0.861538	0.884211	0.956625	0.905594	0.818970	0.84	0.84	0.668889
Decision Tree A	0.607906	0.857143	0.875000	0.973550	0.949857	0.852146	0.84	0.84	0.651111
Random Forest B	0.612536	0.854167	0.891304	0.978500	0.964340	0.776749	0.82	0.82	0.573333
Random Forest A	0.705759	0.867347	0.885417	0.980600	0.967744	0.800813	0.85	0.85	0.662222
SVC Linear B	0.482540	0.800000	0.971429	0.953650	0.934652	0.594872	0.68	0.68	0.417778
SVC Linear A	0.501098	0.630137	0.479167	0.917200	0.943288	0.661111	0.92	0.02	0.733333
KNN B	0.641710	0.855670	0.882979	0.954825	0.904947	0.800813	0.83	0.83	0.595556
KNN A	0.709890	0.873096	0.886598	0.973525	0.949892	0.793416	0.86	0.86	0.684444
Gaussian NB B	0.757853	0.852018	0.772358	0.962650	0.950334	0.655592	0.95	0.74	0.977778
Gaussian NB A	0.754954	0.849315	0.781513	0.942850	0.944823	0.664502	0.93	0.74	0.933333
Neural Net MLP B	0.731968	0.876289	0.904255	0.983300	0.970588	0.873611	0.85	0.85	0.668889
Neural Net MLP A	0.798057	0.898990	0.908163	0.981300	0.970898	0.890241	0.89	0.88	0.762222

Fig 6: Metric comparison of different classification algorithms

The Random Forests classifier came in second, with the Neural Net Multi-layer Perceptron classifier easily outperforming rival models across all measures.



Graph 16: Micro AUROC curves & micro precision-sensitivity curves

Neural Network MLP is the top model in both the AUROC and Precision-sensitivity curves, according to the graph.

Chapter 5: CONCLUSION AND LIMITATIONS

Three key questions in predictive maintenance were addressed by the project: When will an engine fail? Which engines will malfunction during this time? How could maintenance be planned more effectively?

The project was able to offer some solutions to the issue by applying machine learning regression, binary classification, and multiclass classification algorithms to historical data of engine sensors. Our study during the data exploration phase showed that linear models, such as Linear, Ridge and Lasso regression were outperformed by non-linear regression models, such as Random Forest and Polynomial Regression. The model predicts TTF within an average error range of 28.63 cycles, clearly outperforming other models with root mean squared error of RF regressor being 28.63 cycles. In classification analysis, the Random Forests classifier came in second, with the Neural Net Multi-layer Perceptron classifier easily outperforming rival models across all measures.

Regression performance needs to be improved because it is essential to all types of modelling used in this project. This may be done by adjusting model parameters, attempting different models, or fixing data (outliers, resampling, etc.). To improve model efficiency and speed, features selection and dimensionality reduction approaches should also be used. Significant computation and time were required for neural nets and SVMs with RBF kernels.

References

[1] IATA, Fact Sheet Industry Statistics. 2018.

[2] ICAO, “ICAO Long-Term Traffic Forecasts,” International Civil Aviation Organization, June. 2018.

[3] G. Li, “Machine learning in fuel consumption prediction of aircraft,” in Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on, 2010, pp. 358–363.

[4] M. Ward, N. McDonald, R. Morrison, D. Gaynor, and T.Nugent, “A performance improvement case study in aircraft maintenance and its implications for hazard identification,” Ergonomics, vol. 53, no. 2, pp. 247–267,Feb. 2010.

[5] Bloch, “Petrochemical Machinery Insights”, Butterworth-Heinemann , H. P. (2017), pp.191-222.

[6] Susto, G. A., Member, S., Beghi, A., & Luca, C. D, A predictive maintenance system for epitaxy processes based on filtering and prediction techniques. IEEE Transactions on Semiconductor Manufacturing, 25, 2012, pp.638–649.

[7] R. K. Mobley, An Introduction to Predictive Maintenance, 2nd Edition. United States of America:Butterworth-Heinemann, 2002.

[8] IBM Coperation, “Predictive Maintenance Benefits for the Airline Industry.” Sep-2014

[9] X. Li, H. Wang, Y. Shen, and H. Fu, “Integrated vehicle health management in the aviation field,” in Prognostics and System Health Management Conference (PHMChengdu),
2016, pp. 1–5.

[10] J. Chen et al., “A Big Data Analysis and Application Platform for Civil Aircraft Health Management,” 2016, pp. 404–409

[11] S. Li, Y. Yang, L. Yang, H. Su, G. Zhang, and J. Wang,“Civil Aircraft Big Data Platform,” 2017, pp. 328–333.

[12] A. M. Chandramohan, D. Mylaraswamy, B. Xu, and P.Dietrich, “Big Data Infrastructure for Aviation Data Analytics,” in Cloud Computing in

Emerging Markets (CCEM), 2014 IEEE International Conference on, 2014,pp. 1–6.

[13] G. Zhang et al., “A integrated vehicle health management framework for aircraft—A preliminary report,” in Prognostics and Health Management (PHM),2015 IEEE Conference on, 2015, pp. 1–8.

[14] S. Smith, “Predictive Maintenance for Aircraft Engines,”Revolutions, 25-May-2016

[15] Bampoula, X., et al. (2021). A Deep Learning Model for Predictive Maintenance in Cyber-Physical Production Systems Using LSTM Autoencoders. Sensors 21,2021, no. 3. pp. 972.

[16] Hsu, J-Y., et al. Wind turbine fault diagnosis and predictive maintenance through statistical process control and machine learning. IEEE Access 8, 2020, pp. 23427-23439.

[17] Amruthnath, N., and Gupta, T. A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), 2018, pp. 355-361.

[18] Bruneo, D., and Vita, F. D. On the use of LSTM networks for predictive maintenance in smart industries." In 2019 IEEE International Conference on Smart Computing (SMARTCOMP), 2019, pp. 241-248.

[19] Cho, S., et al. A hybrid machine learning approach for predictive maintenance in smart factories of the future. In IFIP International Conference on Advances in Production Management Systems, 2018, pp. 311-317.

[20] Demidova, L. A. Re-current neural networks’ configurations in the predictive maintenance problems. In IOP Conference Series: Materials Science and Engineering, vol. 714, 2020, no. 1. pp. 012005.

[21] Gohel, H. A., et al. (Predictive maintenance architecture development for nuclear infrastructure using machine learning. Nuclear Engineering and Technology 52, 2020, no. 7. Pp.1436-1442.

[22] Spiru Haret University Str. Ion Ghica 13, Bucharest 030045, Romania and A.Mihai, “Airline Applications of Business Intelligence Systems,” INCAS Bull, Sep.2015, vol. 7, no. 3, pp. 153–160.

[23] N. A. H. Campbell, “The Evolution of Flight Data Analysis,” p. 22.

[24] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. F.Tsang, and J. Rodriguez, “Massive Internet of Things for Industrial Applications: Addressing Wireless IIoT Connectivity Challenges and Ecosystem Fragmentation,” IEEE Ind. Electron. Mag.,Mar. 2017, vol. 11, no. 1, pp. 28–33.

[25] S. Sciancalepore, G. Piro, F. Bruni, E. Nasca, G. Boggia,and L. A. Grieco, “An IoT- based measurement system for aerial vehicles,” in Metrology for Aerospace

[26] Fitrayudha, Adryan & Sastra, K. (2021). Predictive Maintenance for Aircraft Engine Using Machine Learning: Trends and Challenges. AVIA. 3. 10.47355/avia.v3i1.45.

[27] S. Weerasinghe and S. Ahangama, "Predictive Maintenance and Performance Optimisation in Aircrafts using Data Analytics," 2018 3rd International Conference on Information Technology Research (ICITR), Moratuwa, Sri Lanka, 2018, pp. 1-8, doi: 10.1109/ICITR.2018.8736157.

Aisha_Updated

ORIGINALITY REPORT

8%	7%	7%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Shakthi Weerasinghe, Supunmali Ahangama. "Predictive Maintenance and Performance Optimisation in Aircrafts using Data Analytics", 2018 3rd International Conference on Information Technology Research (ICITR), 2018 Publication	3%
2	avia.ftmd.itb.ac.id Internet Source	2%
3	dspace.lib.cranfield.ac.uk Internet Source	1%
4	ieomsociety.org Internet Source	<1%
5	dl.lib.uom.lk Internet Source	<1%
6	F A Adryan, K W Sastra. "Predictive Maintenance for Aircraft Engine Using Machine Learning: Trends and Challenges", AVIA, 2021 Publication	<1%