CHAPTER 19

1.INTRODUCTION

This chapter gives a picture about the background of the project and discusses the aim, problem description, research questions, the scope and limitations of this project work.

1.1 Background

A phenomenal boom during the industrial revolution gifted notable innovations in fields such as automation, rapid manufacturing and big data analytics. The fourth industrial revolution which is also termed as Industry 4.0 aims at digital transformation of manufacturing and production industries . This digital transformation creates value to companies as well as to its customers by improving productivity, quality and sustainability. Highly complex, dynamic, and integrated information systems, and very high computational powers are the key enablers of Industry 4.0

When it comes to adapting to changes, the manufacturing industry has always been the pioneer. This was mainly due to the enormous amount of competition and incriminating challenges to meet the customer expectations. The scenario was not different when it comes to application of data science for improving the standard and quality of production. This adaptability of the industry is visible since the first industrial revolution to the state-of-the-art Industry

4.0 techniques, and this has enabled the industry to improve their production sustainably .

Out of the key enablers of Industry 4.0, smart maintenance plays a crucial role. The integration of Information and Communication Technology (ICT) to the manufacturing process gave a breakthrough for gigabytes of data into the manufacturing industries. This big data and advanced computing algorithms enables us to use relevant data to generate precise operational decisions.

When the level of automation increases it becomes hard for the maintenance team to schedule and execute the maintenance activities with least possible impact on the production process. Conducting maintenance at the right time is very crucial for the life of the machine. Unpredictable breakdown leads to an economical wastage is also crucial, because these breakdowns at critical moments lead to disruption of production in an entire production line. This leads to huge financial issues which might result in losses such as eviction from the market and reputation loss. When it comes to big production lines, the loss would be very huge. Even a marginal downtime can cost significantly, when it comes to mega factories.

In order to avoid unwanted breakdowns there are different types of maintenance strategies that are being used in industries. This strategy involves conducting preventive maintenance at fixed time intervals, irrespective of the condition of the machines. Even though this strategy mostly solves the issue of unpredictable breakdowns, this approach has two major drawbacks.

By conducting scheduled maintenance a phenomenal part of the useful life of a machine will not be utilized for a beneficial production and this increases the overall maintenance cost.
The disruptions in the production lines for maintenance activities will also consume some of the useful production hours, which reduces the productivity of the plant.

1.2 Aim

The current maintenance practices at many industries are based on time-based preventive maintenance strategy. One of the major problems with this strategy is that a good amount of useful life of the machine is left unutilized.

The main objective of this project work is to apply machine learning (ML) methods to predict failure of the critical machines which are being utilized in production facilities. The final outcome of the project work would facilitate with data driven decision making for maintenance work, which would prevent unpredicted failures and also the remaining useful life estimation (RUL) estimation.

1.3 Research Questions

The following research questions are concluded based on our objectives of the project as follows:

RQ1 - Is the quantity and quality of available data sufficient to build the ML model for predictions?

RQ2 - Which is the best ML model in predicting RUL of the machine based on performance metrics?

RQ3 - How can we improve the predictive maintenance system considering data acquisition and create a future road map?

1.4 Problem Description

For most of the manufacturing industries, productivity in manufacturing is very important for meeting the very high customer demand . In order to achieve this efficiency, a company should have a well defined strategy for the maintenance activities to ensure maximum up-time for all of its machines.

Most manufacturing industries currently make use of a value driven maintenance strategy for maintaining the machines in good working conditions in their body shop factory. Value driven maintenance is always improving based on the learning from the previous feedback.

Currently the scheduling and prioritization is done only from the experience of the workers and the company wants to make maintenance decisions based on data driven approaches.

Industries are currently outlining a smart maintenance strategy, specifically describing the future development of data driven decision making. The aim is to move from calendar based, or sometimes even reactive maintenance to being able to predict equipment failure and planning maintenance on demand.

There are several machines in the plant for manufacturing activities. In our project we are working with a critical machine that was chosen for the study as it was acting like a bottle neck in the system. There were sudden, unpredictable break-downs in these machines which led to stoppage of the production line several times. This was causing many problems for the maintenance team as they often needed to do reactive maintenance.

On interviewing the maintenance experts from the company it was clear that the company's strategy was also to first change into predictive maintenance based on the cost of each process.

The aim of this project is to make use of the log data from the sensors attached to different parts of the station and analyze whether the data available is sufficient for a ML approach to predict the next failure .

The aim is to use ML models and analyze how good they are performing with the current data available from the machine, also calculate the RUL of the whole plant

From experience learned from modeling the next step will be to lay down a road map for the future data driven plan which would enable them to achieve their goal of attaining smart maintenance capability.

1.5 Scope and limitation

As mentioned in problem description the scope of the project is to conduct an exploratory data analysis to see the relevance of data available and to make use of State-of-the-art ML algorithms for future prediction of failure by calculating the remaining useful life. The predictions would help the maintenance department to plan the maintenance activities in advance and also make use of the valuable RUL information of the critical machine.

The project concentrates on critical machines from a shop floor consisting of many machines. In this project we are analyzing data from different sources, such as vibration data, temperature and RPM. Historical data was provided to us based on which we were asked to build the model. After the model is built the data that would be fed to the machine for prediction and also learn and store those data.

The aim of the project was to make a Machine learning model which would predict the RUL of the critical machine. In order to conduct supervised ML it is necessary to label the data with relevant data.

Limitation

Data acquisition process was very iterative and time consuming. There were many sources of data which were not relevant, understanding and acquiring the relevant data also consumed the majority of the time of the project. Also the historic data was not readily available, so at some points we had to wait for the data to be generated.

Other major limitations and challenges of the project were,

- Interpretability: Some AI models, such as deep neural networks, are considered black boxes, making it difficult to interpret their predictions and explanations. Explainable AI techniques are being explored to address this limitation.

- Model Deployment and Integration: Deploying AI models in real-world manufacturing environments and integrating them with existing systems and workflows can be complex and require careful planning and collaboration between data scientists and domain experts.

- Organizational Adoption and Skills Gap: Organizations may face challenges in terms of cultural acceptance, change management, and the availability of skilled personnel with expertise in AI and predictive maintenance.

2.LITERATURE SURVEY

2.1 INTRODUCTION

For this project work, an extensive literature study was carried out to gain knowledge and also to validate the results obtained. Literature study was carried out on various topics like maintenance strategies, smart maintenance, related works, ML models used for prediction and finally the evaluation techniques for assessing the models.

2.2 MAINTENANCE STRATEGIES

In the past years, we could see a considerable increase in automation in almost all fields. The story of the manufacturing industry is not so different. In order to meet the high customer demand, manufacturing industries are trying to automate most of their operations. As a byproduct to this, the importance of

maintenance activities are also increasing in order to keep these machines up and running. Different maintenance strategies have been evolved and are being used by manufacturing plants all across the globe. From the literature study, it was clear that the suitability of each method depends on the severity of the repercussions caused by the failure of the machine and depends on the type of production activity. For example if a product is continuously manufactured then, a breakdown in the production should be avoided. So for such a production strategy corrective or breakdown maintenance is not the correct fit.

Maintenance strategies are classified differently with different standards and depending on their application. shown in Figure 2.1

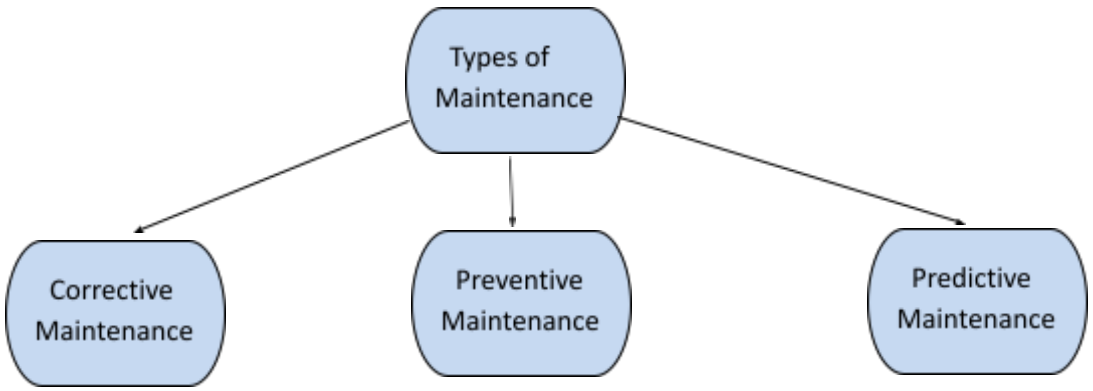


Figure 2.1: Types of maintenance

.2.2.1 Corrective Maintenance

Corrective maintenance is typically performed in response to unexpected events or failures that disrupt the normal operation of a system. These failures can result from various factors, such as equipment wear and tear, component failure, environmental conditions, operator error, or external factors.

The key steps involved in corrective maintenance include:

1. Failure identification: Recognizing and identifying the failure or malfunction in the system. This may involve investigating reported issues, conducting diagnostic tests, or analyzing system performance data.

2. Reporting and notification: Documenting the failure, its impact, and notifying the relevant personnel responsible for maintenance and repair.

3. Troubleshooting and diagnosis: Investigating the root cause of the failure to determine the underlying problem. This may involve inspecting components, examining system logs, or conducting tests and measurements.

4. Repair or replacement: Once the problem has been identified, appropriate repair actions are taken to fix the faulty equipment or components. In some cases, replacement of damaged parts may be necessary.

5. Testing and verification: After the repair, the system or equipment is tested to ensure that it is functioning correctly and

that the issue has been resolved. This may involve running diagnostic tests, conducting performance checks, or conducting operational tests.

6. Documentation and record-keeping: Proper documentation of the maintenance activities, including the details of the failure, repairs performed, replacement parts used, and any other relevant information. This helps in tracking maintenance history, analyzing trends, and improving maintenance processes.

Corrective maintenance is often seen as a reactive approach to maintenance, as it focuses on addressing failures as they occur. It is typically contrasted with preventive maintenance, which involves scheduled inspections, servicing, and replacement of components to prevent failures before they happen.

While corrective maintenance is necessary for dealing with unexpected failures, it is generally considered less efficient and more costly compared to preventive maintenance. Therefore, organizations often aim to strike a balance between both approaches to minimize unplanned downtime, optimize asset performance, and reduce maintenance costs.

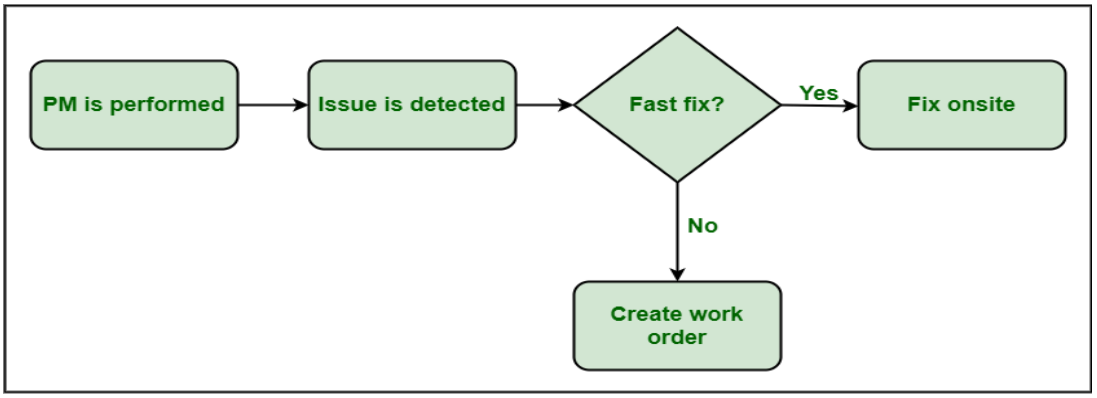


Fig No 2.2 Corrective maintenance workflow

2.2.2 Preventive Maintenance

Preventive maintenance (PM) is a proactive approach to maintenance that aims to prevent equipment failure, reduce the likelihood of breakdowns, and prolong the lifespan of assets. It involves regularly scheduled inspections, servicing, and maintenance activities to identify and address potential issues before they lead to equipment failure or performance degradation.

The primary goals of preventive maintenance are to:

1. Increase equipment reliability: By regularly inspecting and maintaining equipment, preventive maintenance helps identify and address potential problems before they cause a breakdown. This reduces unexpected downtime and improves overall equipment reliability.

2. Reduce maintenance costs: Proactively addressing potential issues through preventive maintenance can help minimize the need for costly repairs or emergency maintenance. It can also extend the lifespan of equipment, reducing the frequency of equipment replacements.
3. Optimize asset performance: Properly maintained equipment tends to operate more efficiently and effectively. Preventive maintenance helps ensure that equipment is operating at its peak performance, resulting in improved productivity and quality.
4. Enhance safety: Regular inspections and maintenance activities as part of preventive maintenance can help identify safety hazards or potential risks associated with equipment. By addressing these issues in advance, the risk of accidents or injuries can be minimized.

The key steps involved in implementing a preventive maintenance program include:

1. Asset identification: Identify the equipment, systems, or assets that require preventive maintenance based on their criticality, usage, or manufacturer's recommendations.
2. Establish maintenance schedules: Determine the appropriate frequency and intervals for inspections, servicing, and maintenance activities based on equipment type, operational requirements, and industry best practices.
3. Develop maintenance procedures: Create detailed procedures and checklists for each preventive maintenance task. These

procedures should outline the steps to be followed, necessary tools or resources, and any specific safety considerations.

4. Training and resources: Ensure that maintenance personnel are properly trained to perform preventive maintenance tasks. Provide them with the necessary tools, resources, and documentation to carry out their responsibilities effectively.
5. Documentation and record-keeping: Maintain a comprehensive record of all preventive maintenance activities, including dates, tasks performed, parts replaced, and any observations or recommendations. This information helps track maintenance history, identify trends, and make informed decisions regarding future maintenance activities.
6. Continuous improvement: Regularly evaluate the effectiveness of the preventive maintenance program and make adjustments as needed. Analyze maintenance data, identify recurring issues, and implement corrective actions to further optimize equipment performance and reliability.

By implementing a well-designed preventive maintenance program, organizations can proactively manage their assets, reduce unplanned downtime, extend equipment lifespan, and achieve overall cost savings.

While preventive maintenance offers numerous benefits, there are also some potential disadvantages to consider:

1. Increased upfront costs: Implementing a preventive maintenance program often requires an initial investment in

resources, such as training, tools, and software systems. These upfront costs can be significant, particularly for organizations with a large number of assets or complex equipment.

2. Time and labor-intensive: Preventive maintenance requires regular inspections, servicing, and maintenance activities. This can consume a significant amount of time and labor, especially for organizations with extensive equipment and maintenance schedules. It may require dedicated maintenance personnel or additional outsourcing of maintenance services.
3. Over-maintenance or unnecessary repairs: In some cases, preventive maintenance activities may lead to over-maintenance or unnecessary repairs. If maintenance tasks are performed too frequently or without proper justification, it can result in unnecessary downtime, increased costs, and potential disruptions to operations.
4. Disruption to operations: Performing preventive maintenance activities may require temporarily shutting down equipment or taking it out of service. This can cause disruptions to production or operations, particularly in industries where continuous operation is critical. Careful planning and coordination are necessary to minimize the impact on operations.
5. False sense of security: While preventive maintenance aims to reduce the likelihood of failures, it does not guarantee that all potential issues will be detected or prevented. Some failures can still occur despite regular maintenance efforts. Relying solely on preventive maintenance may create a false sense of

security, leading to complacency in other areas such as monitoring, condition-based maintenance, or proactive troubleshooting.

6. Difficulty in predicting failures: Predicting the exact timing of equipment failures can be challenging. While preventive maintenance schedules are based on historical data and manufacturer recommendations, unexpected failures can still occur. This means that some failures may happen before the scheduled maintenance, resulting in unplanned downtime and reactive maintenance efforts.
7. Costly maintenance for low-criticality assets: Applying the same level of preventive maintenance to all assets, regardless of their criticality, can lead to inefficient resource allocation. Some lower-criticality assets may not require as frequent or extensive maintenance, resulting in unnecessary costs.

To mitigate these disadvantages, organizations can employ a balanced maintenance approach that combines preventive maintenance with other strategies such as condition-based maintenance, predictive maintenance, and reliability-centered maintenance. This helps optimize maintenance efforts, reduce costs, and maximize asset performance. .

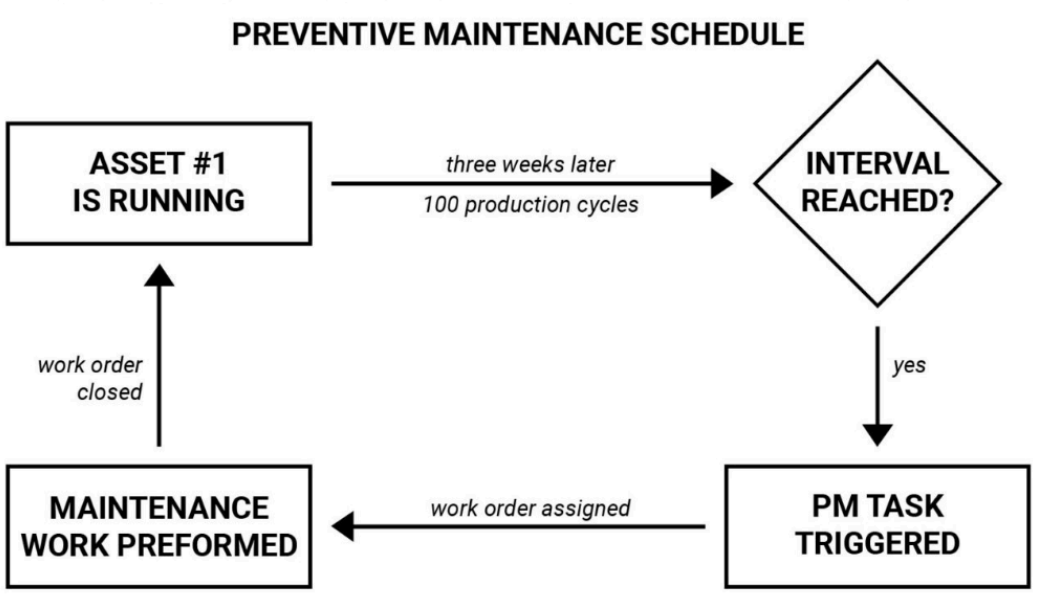


Fig no 2.3 Preventive maintenance

2.2.3 Predictive Maintenance

Predictive maintenance (PdM) is an advanced maintenance strategy that utilizes data analysis and condition monitoring techniques to predict when equipment failures are likely to occur. It aims to identify potential issues before they lead to breakdowns or performance degradation, allowing for proactive maintenance actions to be taken.

The key principle of predictive maintenance is to monitor the actual condition and performance of equipment in real-time or near real-time, analyze the data collected, and use that information to make informed predictions about when maintenance should be performed. By detecting early signs of deterioration or anomalies,

organizations can schedule maintenance activities at the most opportune times, minimizing downtime and reducing unnecessary maintenance.

Here are the main components and benefits of predictive maintenance:

1. Condition monitoring: Various sensors and monitoring devices are used to collect data on equipment performance, including vibration, temperature, pressure, lubricant analysis, acoustic emissions, and more. These sensors continuously measure and record data, providing insights into the condition of the equipment.
2. Data analysis and machine learning: Advanced data analysis techniques, including machine learning algorithms, are applied to the collected sensor data. The algorithms learn patterns and correlations between the data and equipment failures, allowing them to predict potential failures or performance degradation.
3. Anomaly detection: Predictive maintenance systems can identify anomalies or deviations from normal equipment behavior. These anomalies may indicate early signs of equipment deterioration or impending failure. By detecting such deviations, maintenance personnel can take proactive actions to address the issue before it worsens.
4. Condition-based maintenance: Predictive maintenance enables maintenance activities to be scheduled based on actual equipment condition rather than fixed time intervals.

This allows organizations to optimize maintenance efforts, reducing unnecessary maintenance tasks and minimizing equipment downtime.

5. Increased equipment uptime: By detecting potential failures in advance, predictive maintenance helps organizations minimize unplanned downtime. Planned maintenance actions can be scheduled during planned equipment downtime or low-demand periods, ensuring that maintenance activities have minimal impact on operations.
6. Cost savings: Predictive maintenance can help organizations reduce maintenance costs by avoiding unnecessary or premature maintenance. By focusing on equipment that genuinely requires attention, resources can be allocated more effectively, reducing labor, parts, and overall maintenance expenses.
7. Extended equipment lifespan: Timely maintenance based on equipment condition can help extend the lifespan of equipment. By addressing issues before they lead to more severe failures, equipment reliability and longevity can be improved.
8. Improved safety: Predictive maintenance can contribute to improving safety by detecting potential safety hazards or risks associated with equipment malfunctions. Proactive maintenance actions can prevent accidents, injuries, or other safety-related incidents.

Implementing predictive maintenance requires appropriate sensor deployment, data collection infrastructure, data analysis

capabilities, and the integration of predictive maintenance systems with existing maintenance workflows. It also relies on accurate historical data and continuous refinement of predictive models to improve accuracy over time.

Overall, predictive maintenance enables organizations to transition from reactive or time-based maintenance approaches to a more proactive and efficient maintenance strategy, optimizing resources, reducing costs, and enhancing equipment reliability.

While predictive maintenance offers several advantages, there are some potential disadvantages to consider:

1. Complex implementation: Implementing a predictive maintenance program requires specialized knowledge, expertise, and advanced technologies. It involves deploying sensors, setting up data collection infrastructure, implementing data analysis algorithms, and integrating predictive maintenance systems with existing maintenance processes. This complexity can present challenges in terms of initial investment, training, and technical integration.
2. Data requirements: Predictive maintenance relies heavily on data, including historical equipment performance data, sensor readings, and maintenance records. Organizations need to have access to reliable and comprehensive data to train predictive models and make accurate predictions. Insufficient

or poor-quality data can undermine the effectiveness of predictive maintenance efforts.

3. Cost of sensor deployment: Predictive maintenance often requires the installation of various sensors and monitoring devices to collect real-time equipment data. The cost of acquiring and installing these sensors can be significant, especially for organizations with a large number of assets or complex equipment.
4. Complexity of data analysis: Analyzing and interpreting the vast amounts of data collected from sensors can be challenging. It requires advanced data analysis techniques, including machine learning and data mining, as well as expertise in interpreting the results. Organizations may need to invest in specialized software tools or seek assistance from data scientists or domain experts.
5. False alarms and accuracy issues: Predictive maintenance models may generate false alarms or inaccurate predictions. Factors such as variations in operating conditions, data quality issues, or unexpected events can impact the accuracy of predictions. False alarms can lead to unnecessary maintenance activities or disruptions to operations, while inaccurate predictions can result in unexpected equipment failures.
6. Equipment complexity and variability: Some equipment may have complex failure patterns or exhibit high variability in performance. Predictive maintenance models may struggle to accurately predict failures in such cases. Additionally,

equipment with unique failure modes or limited historical data may pose challenges for developing accurate predictive models.

7. Maintenance scheduling constraints: Predictive maintenance relies on predicting when failures are likely to occur. However, scheduling maintenance activities based on these predictions can be challenging, as it requires coordination with operational schedules, availability of resources, and consideration of maintenance priorities. Balancing maintenance requirements with production demands can be a complex task.
8. Initial investment and ROI considerations: Implementing a predictive maintenance program involves upfront costs, including equipment, sensors, software, and training. Organizations need to carefully evaluate the potential return on investment (ROI) to ensure that the benefits derived from predictive maintenance outweigh the initial investment and ongoing maintenance costs.

Despite these challenges, predictive maintenance has the potential to optimize maintenance efforts, reduce costs, and improve equipment reliability. Organizations should carefully assess their specific needs, available resources, and the complexity of their equipment before deciding to implement predictive maintenance. It may be beneficial to start with pilot projects or focus on critical equipment with high potential for failure to maximize the advantages of predictive maintenance while minimizing the disadvantages.

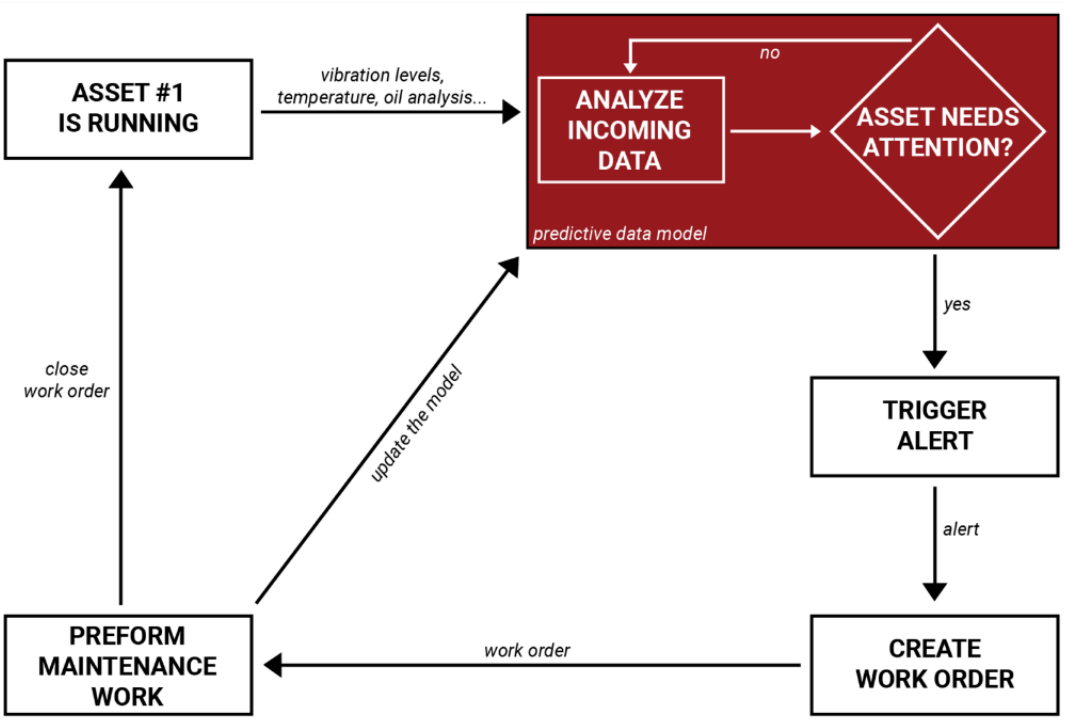


Fig no 2.4 Predictive maintenance

2.2.4 Maintenance Comparisons

Below charts explain different maintenance approaches that we have explained above. If you note carefully, when compared to reactive and preventive maintenance , predictive maintenance is able to tell us the failure of the machine or the systems as a whole

well in advance to the maintenance team for them to perform the maintenance.

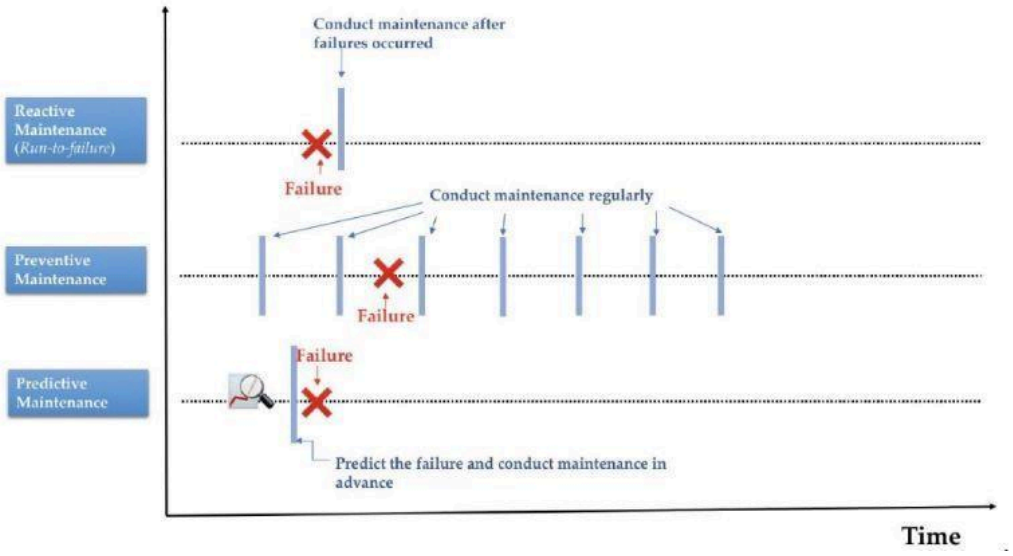


Fig no 2.5 Maintenance Comparison

2.3 Data-driven decision making

Data driven decision making is what differentiates traditional maintenance and predictive maintenance. Data driven decision making is deciding scheduling maintenance activities on the basis of analysis of historical data rather than from the intuitions and experience of humans. The advancements in high computation and the extensive research in the field of ML has enabled the capability for prediction,classification and anomalies detection which would support decision making.

Researchers have proved that data driven decision making is capable of providing better accuracy in decision making compared to traditional methods Some of the key enablers of data driven decision making are briefly discussed below.

2.3.1 Artificial Intelligence

AI is a broad terminology that can be used for any entity which can analyze data and recognize the patterns. It is a broad team and has many subdivisions like ML, Deep learning etc. describes the correlation between AI, ML and Deep learning and it is as shown in Figure below , points out that the development in parallel computing and massively parallel programs that has the capability to learn things has enabled human level intelligence in modern day machines.

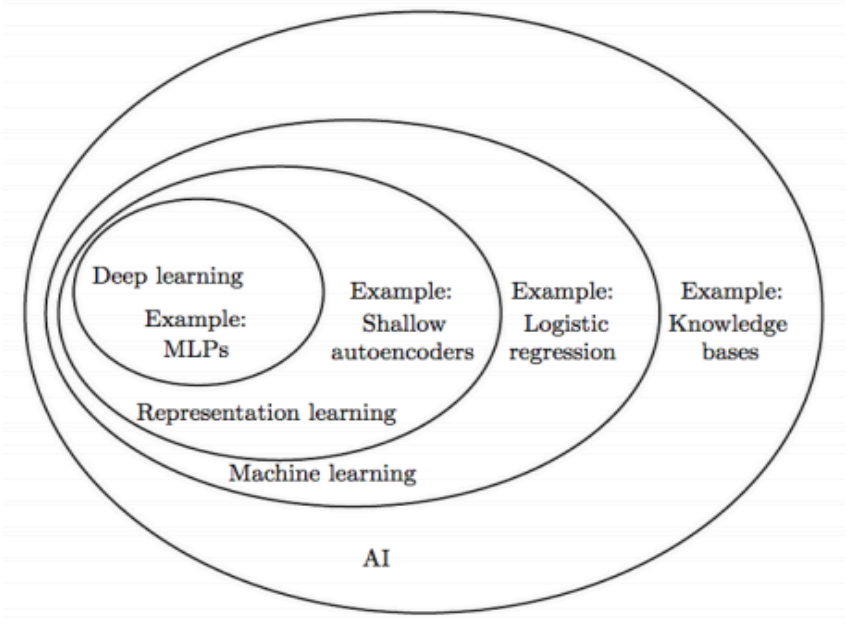


Fig. no 2.6: Venn diagram showing the correlation between AI, ML and Deep learning.

2.3.2 Machine learning

According to Burkov et al ML is a tool for finding the hidden patterns and trends from the available data. ML techniques make use of mathematical algorithms for optimization, prediction, classification, anomaly detection etc . The major role of a data scientist is to prepare the data set in the best possible way which is suitable. Literature Study for the algorithms or statistical model chosen based on the business requirement and application

2.4 Literature Review

As discussed in the above sections predictive maintenance is a key enabler for smart maintenance is a key enabler for smart maintenance. High computational power and highly accurate ML algorithms makes it possible to perform advanced predictions for implementing different Predictive Maintenance (PdM) applications such as health indicators construction, anomaly detection, and RUL estimations. The related part section concentrates on the research works conducted on RUL estimations and its applications in different fields.

Cline et al. [1] implies that the maintenance actions has to be taken at the right time otherwise it will not only cause waste of usable resources but also unnecessary downtime. The research team has collected all the related historical data from a very long period of time. It is proven that application of ML improves the estimation/ prediction the failure of risky assets. Predictive maintenance using AI / ML algorithms could aid in taking up maintenance activities just

in time, when they are really crucial. Scientific studies have proved that predictive maintenance reduces the breakdown time at a rate of 70 - 75%, reduces the maintenance cost by 25 - 35% and increases production by 25 to 35% . These statistics motivates industry to move forward in implementing predictive maintenance using AI/ ML algorithms.

Baptista et al. [2] compares the different AI and statistical approaches for PdM, and concludes that AI approach produces a better result than statistical approach and also discusses the ability of ML algorithms to handle high dimensional multivariate data for predictive maintenance applications in industries.

Kaparthi et al. [3], refers to performing predictive maintenance based on machine parts manufacturing industry. However this system can be applied to any industry because of its scalability. And it provides an introduction about decision tree-based ML strategies as well as an experiment based on real case study. Decision tree-based learning method follows the relationship identifified between the input variables. Literature Study focus in this research is conditional inference tree statistical methodology. A confusion matrix is used to evaluate the model performance and it is done by comparing the prediction data with real data.

Bekar et al. [4] introduces an intelligent approach for data pre-processing and analysis in PdM based on an industrial case study. The authors used a real-world industrial data collection

method and presented the results after preliminary analysis. This work was based on feature space dimension reduction and clustering of data-points with the idea of understanding the outliers in anomaly clusters. The first step of PdM implementation was the use of unsupervised ML and this formulated approach really helps to collect, analyze, describe, visualize, prepare and understand high-dimensional industrial big data. Apart from that, this study guided the transformation of the domain-expert knowledge to the ML work-flflow in the data preparation phase.

Fink et al. [5] discusses the five different levels of condition based and predictive maintenance and also arranges them in their level of complexity. Fink etal. [5] also states that availability of labeled data sets is one of the biggest challenges faced in conducting supervised learning projects. It also puts forward two solutions for this issue. The first one is obtaining labels directly or indirectly from the health indicator and the second one is to simulate the machine in a virtual environment and get the required data set from the digital twin. This is a great solution to start with initially and the data analysis can gradually be done with the real time data which would increase the accuracy of the predictions.

From the extensive literature study it was clear that there were not many research works on data quality issues in real time industrial scenario and the ways to solve them and overcoming these data challenges for performing RUL estimation. Different ML model give different accuracy values depending on the fitness of data which is

input in to the ML model. From the extensive literature survey it was clear that recurrent neural networks with a memory gave the best accuracy when it come to time series prediction.

2.5 Machine learning models for predictive maintenance

A machine learning model is defined as a mathematical representation of the output of the training process. Machine learning is the study of different algorithms that can improve automatically through experience & old data and build the model. A machine learning model is similar to computer software designed to recognize patterns or behaviors based on previous experience or data. The learning algorithm discovers patterns within the training data, and it outputs an ML model which captures these patterns and makes predictions on new data.

Machine Learning models can be understood as a program that has been trained to find patterns within new data and make predictions. These models are represented as a mathematical function that takes requests in the form of input data, makes predictions on input data, and then provides an output in response. First, these models are trained over a set of data, and then they are provided an algorithm to reason over data, extract the pattern from feed data and learn

from those data. Once these models get trained, they can be used to predict the unseen data set.

2.5.1 Classification of Machine Learning Models:

Based on different business goals and data sets, there are three learning models for algorithms. Each machine learning algorithm settles into one of the two models:In this project we are using the Supervised model.

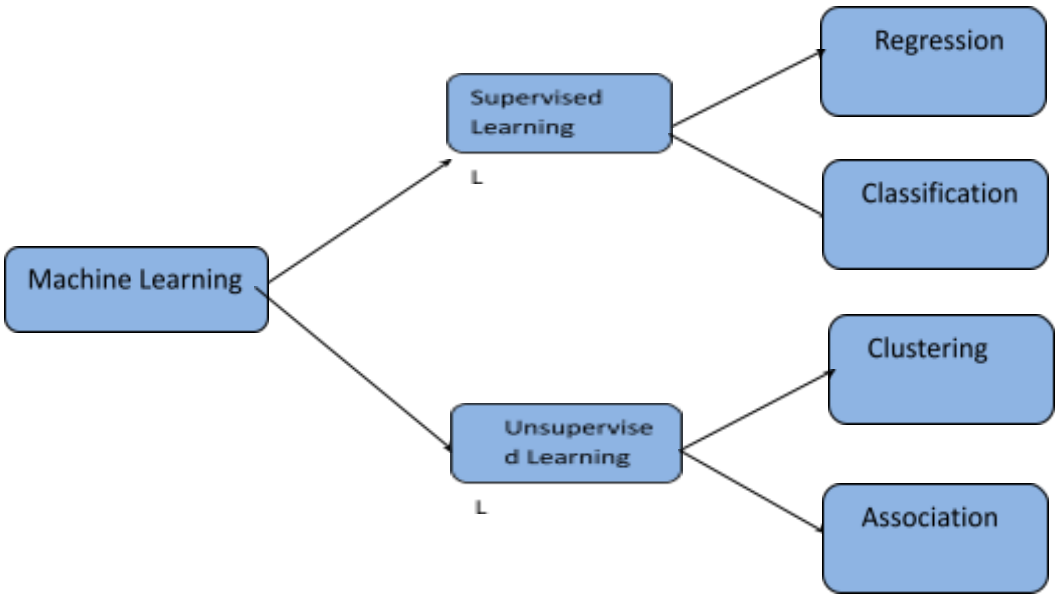


Fig no. 2.7 ML Models

2.5.2 Supervised Machine Learning Models

Supervised machine learning encompasses both regression and classification tasks. Let's explore each of them:

Regression: Regression is a type of supervised learning used to predict continuous numerical values. In predictive maintenance, regression models can be employed to predict the remaining useful life (RUL) of equipment, estimate the time until failure, or forecast numerical metrics related to equipment performance. Regression models take input features, such as historical sensor data, and output a continuous numerical value as the prediction. Examples of regression algorithms commonly used in predictive maintenance include linear regression, support vector regression (SVR), random forest regression, and neural networks.

Classification: Classification is another type of supervised learning that aims to assign input data into predefined classes or categories.

In the context of predictive maintenance, classification models can be used to categorize equipment into failure or non-failure states or to classify the severity of failures. Classification models take input features and assign the data to specific classes based on patterns and relationships learned from the training data. Commonly used classification algorithms in predictive maintenance include logistic regression, decision trees, random forests, support vector machines (SVM), and neural networks.

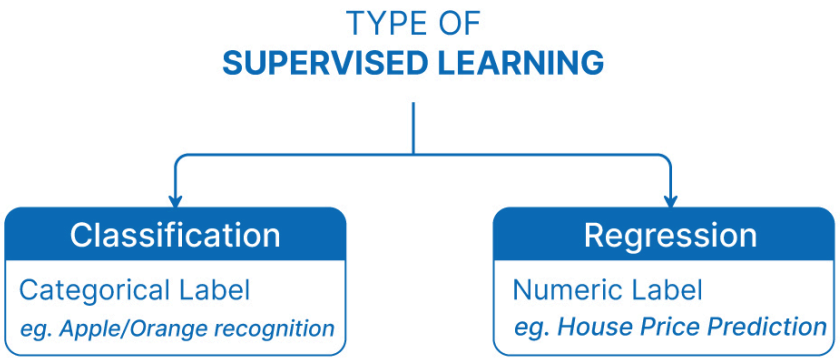


Fig no. 2.8 Supervised Learning

In both regression and classification tasks, the machine learning models are trained using labeled training data. Labeled data consists of input features along with corresponding target values or class labels. During the training phase, the models learn from the provided data to build a representation of the relationships between the input features and the target values or classes. Once the models are trained, they can be used to make predictions or classify new, unseen data.

It's important to note that the choice between regression and classification depends on the nature of the prediction task and the type of output required. If the objective is to predict a continuous numerical value, regression is appropriate. On the other hand, if the goal is to assign data to specific classes or categories, classification is the suitable approach.

Furthermore, it's worth mentioning that some predictive maintenance scenarios may involve a combination of regression and classification. For instance, a system could use regression to predict the remaining useful life of equipment and then apply classification to categorize the predicted RUL into specific maintenance actions, such as "replace," "repair," or "monitor." This hybrid approach allows for a more detailed and actionable prediction.

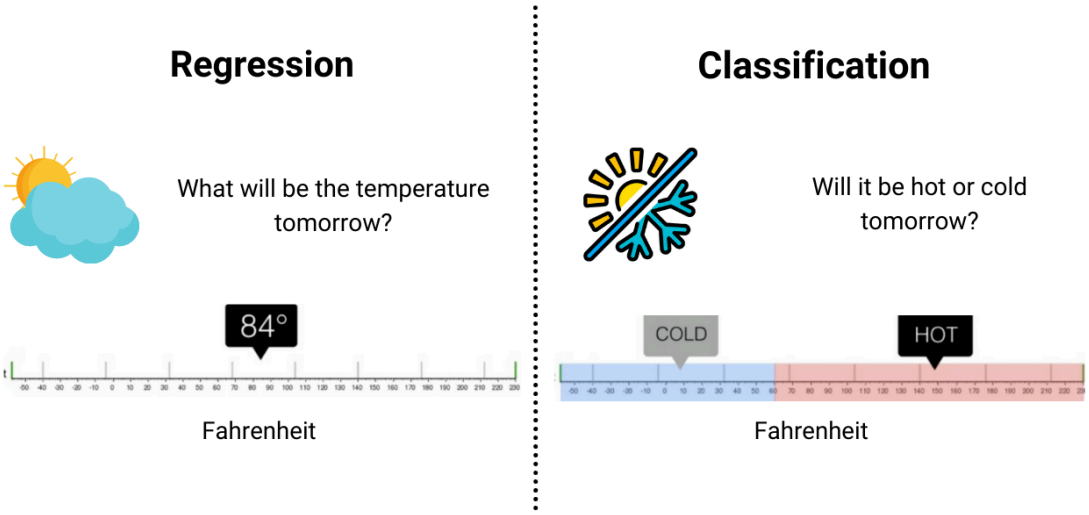


Fig.no 2.9 Regression Vs Classification

Machine learning models play a crucial role in predictive maintenance by analyzing historical and real-time data to make predictions about equipment failures. Here are some commonly used machine learning models for predictive maintenance:

1. Regression models: Regression models, such as linear regression or logistic regression, can be used to predict the remaining useful life (RUL) of equipment based on historical sensor data. These models estimate the time until failure or the probability of failure within a given time frame.

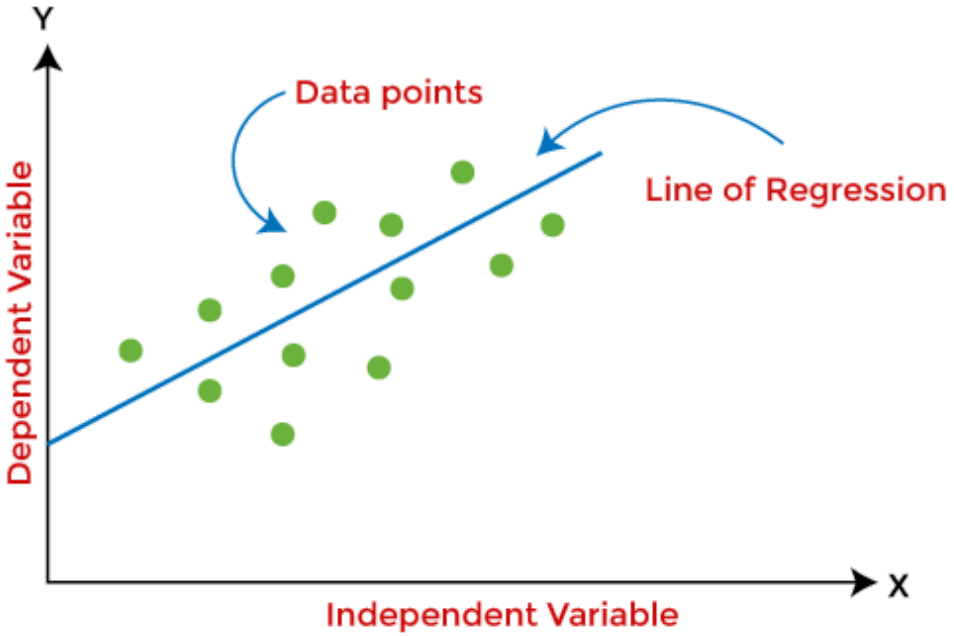


Fig no. 2.10 Regression model

2. Decision trees: Decision tree models, such as Random Forest or Gradient Boosting, are used to analyze sensor data and identify patterns or rules that indicate potential equipment failures. Decision trees are capable of handling non-linear relationships and can handle both numerical and categorical input features.

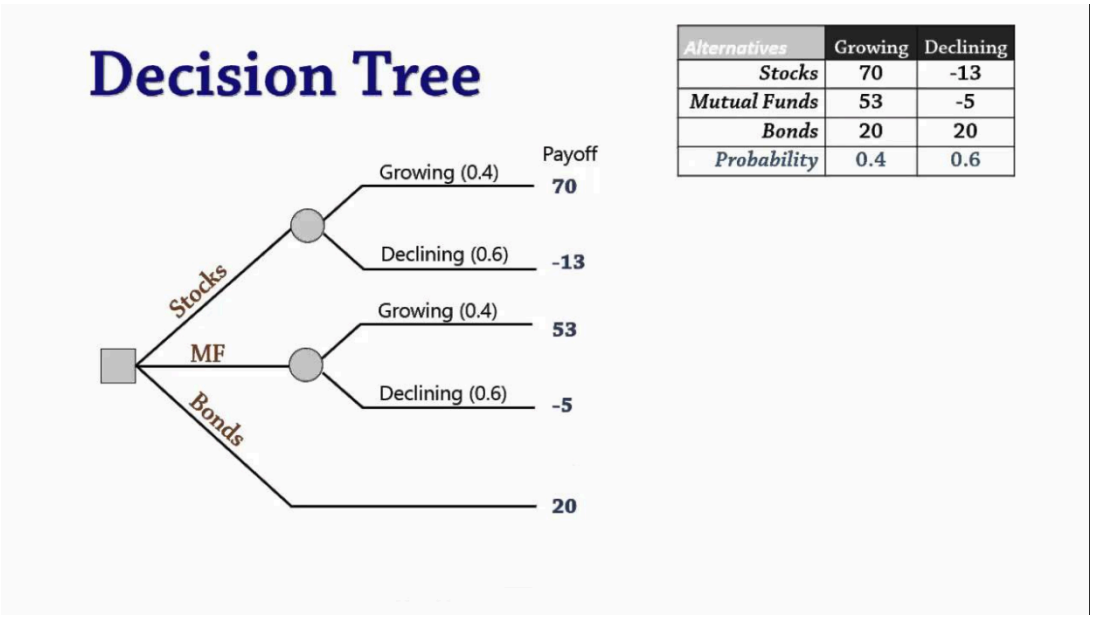


Fig no. 2.11 Decision Tree

3. Support Vector Machines (SVM): SVM models are effective for classification tasks in predictive maintenance. They can be used to classify equipment into different failure categories based on sensor data patterns. SVMs work well with small to medium-sized datasets and are known for their ability to handle high-dimensional data.

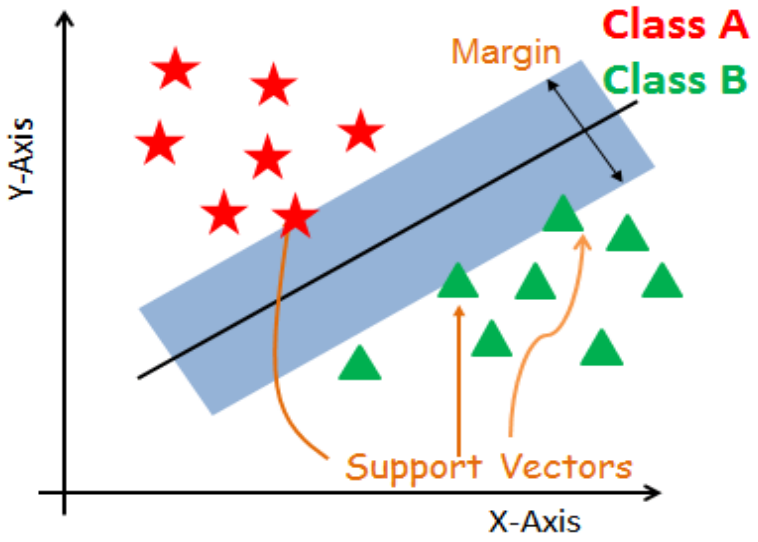


Fig. no. 2.12 SVM

4. Neural networks: Neural network models, including deep learning architectures like Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN), are increasingly used in predictive maintenance. These models can capture complex relationships in the sensor data and identify subtle patterns that may be indicative of impending failures.

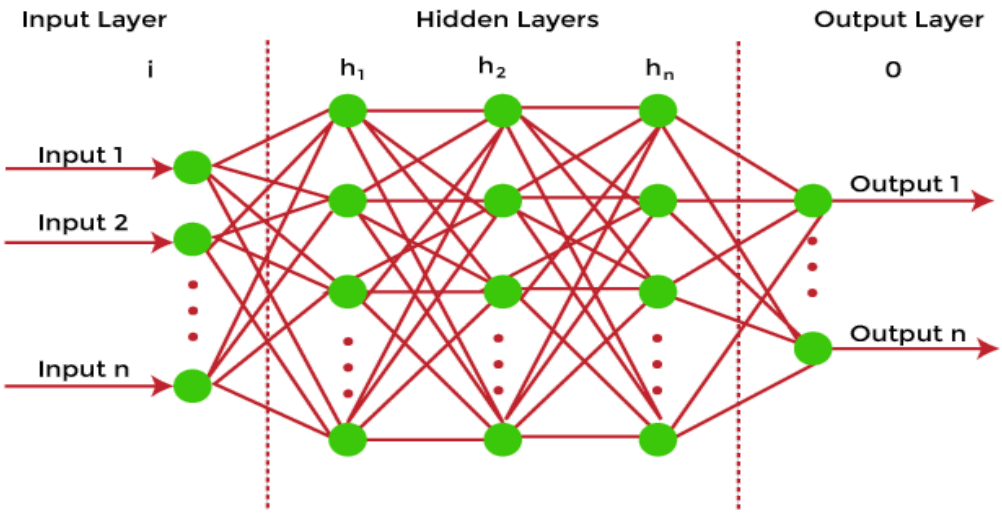


Fig. no. 2.13 Neural Networks

5. Ensemble methods: Ensemble methods, such as bagging or stacking, combine multiple machine learning models to improve predictive performance. By aggregating predictions from multiple models, ensemble methods can provide more robust and accurate predictions for equipment failures .

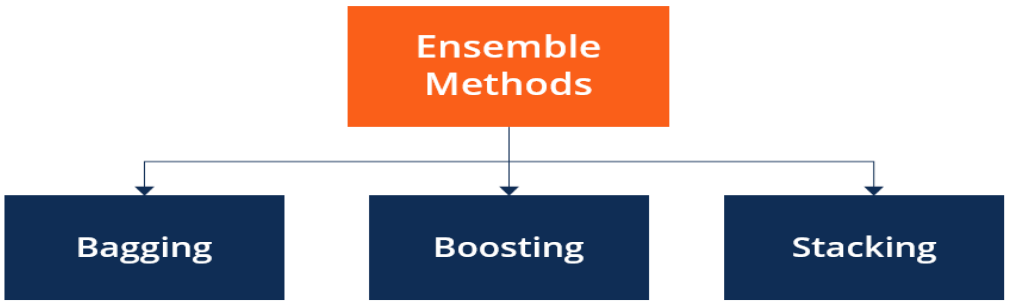


Fig. no. 2.14 Ensemble Methods

It's important to note that the selection of a specific machine learning model depends on various factors, including the type of data available, the complexity of the equipment, the desired prediction task, and the available computational resources. Additionally, data pre-processing, feature engineering, and model hyper parameter tuning are critical steps to optimize the performance of machine learning models in predictive maintenance applications.

Our project uses ExtraTreesRegressor and LGBM regressor,

ExtraTreesRegressor

Extra trees (short for extremely randomized trees) is an ensemble supervised machine learning method that uses decision trees and is used by the Train Using AutoML tool. This method is similar to random forests but can be faster.

The extra trees algorithm, like the random forests algorithm, creates many decision trees, but the sampling for each tree is random, without replacement. This creates a dataset for each tree with unique samples. A specific number of features, from the total set of features, are also selected randomly for each tree. The most important and unique characteristic of extra trees is the random selection of a splitting value for a feature. Instead of calculating a locally optimal value it split the data, the algorithm randomly selects a split value. This makes the trees diversified and uncorrelated.

LGBM Regressor

LightGBM is a gradient boosting ensemble method that is used by the Train Using AutoML tool and is based on decision trees. As with other decision tree-based methods, LightGBM can be used for both classification and regression. LightGBM is optimized for high performance with distributed systems.

LightGBM creates decision trees that grow leaf wise, which means that given a condition, only a single leaf is split, depending on the gain. Leaf-wise trees can sometimes overfit especially with smaller datasets. Limiting the tree depth can help to avoid overfitting.

LightGBM uses a histogram-based method in which data is bucketed into bins using a histogram of the distribution. The bins, instead of

each data point, are used to iterate, calculate the gain, and split the data. This method can be optimized for a sparse dataset as well. Another characteristic of LightGBM is exclusive feature bundling in which the algorithm combines exclusive features to reduce dimensionality, making it faster and more efficient

3. METHODOLOGY

This section focuses on the scientific concepts, selected approach and the data exploratory analysis that has been used. The initial planning were started to follow the standard CRISP-DM methodology which is having Business understanding,Data understanding, Data preparation, Modeling, Evaluation and Deployment as the main phases. However for practical reasons, this work represents the modified CRISP-DM methodology by putting more attention in data acquisition phase.

3.1 Enhanced CRISP-DM methodology

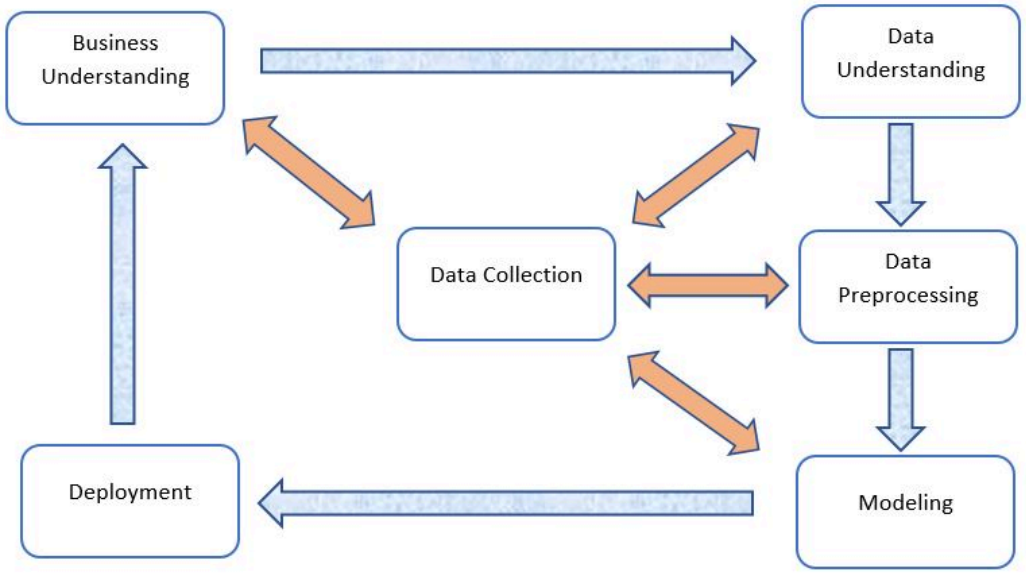


Fig.no. 3.1: Enhanced Methodology - adopted from the original CRISP-DM

The project starts with business understanding phase. It is essential for any project to understand customer needs. This is kind phenomenal because all the company objectives and goals have to be formulated. The data understanding part to identify the relevant data sources, collect data and evaluation of data comes secondly. The third main step would be data preparation and it takes the most of the time out of all the phases in the methodology. Once the sufficient amount of quality data is ready, the next phase is modeling and it is the shortest phase. The trial and error method should be performed to select the best possible model. In the training step, the model performance with the business requirements will be assessed. In the final step, the deployment is performed to conclude the work. The following content discuss about the main phases of

CRISP-DM structure and how it is modified in the present investigation.

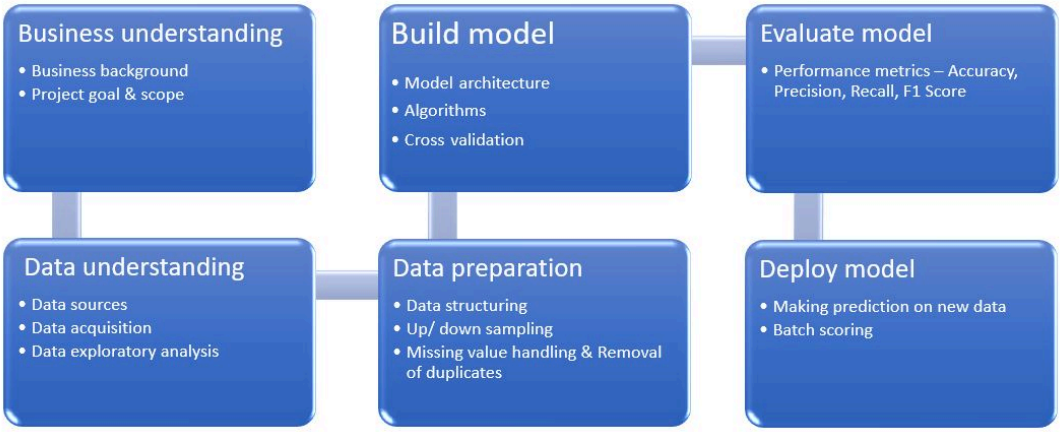


Fig. no. 3.2: Overall methodology of the project

As portrayed in the figure, all the phases from business understanding to modelling is connected to data collection phase since historical data was not available and it has to be collected during the project duration.

3.2 Business Understanding

The first phase of the methodology concentrates on the company requirements and objectives from the business point of view. It is necessary to learn the company goals, objectives and what are their expectations clearly before we set up the project scope and plan. The initial plan and the scope has to be decided based on the available resources and time. The time limitation is about six

months and few discussion sessions were carried out to understand the data availability. Hence, understanding the business objectives, assessment of available resources, and finalizing the project plan is done in this phase. The software tools incorporated in the project are identified to handle the work for data analysis, visualization and data modeling with the help from supervisor and team. Eventually this defined phase, represents the fundamental work in succeeding any data science project.

3.3 Data Understanding

Data understanding is a crucial step in any data-driven project, including predictive maintenance. It involves gaining knowledge and insights about the available data, understanding its structure, quality, and relevance to the predictive maintenance objectives. Here are the key aspects of data understanding in the context of predictive maintenance:

1. Data sources: Identify the sources of data that will be used for predictive maintenance. This may include historical maintenance records, sensor data collected from equipment, operational data, environmental data, and any other relevant sources. Understand where and how the data is generated and stored.
2. Data collection: Determine how the data is collected and stored. Consider the data collection methods, frequency, and granularity. Understand the sampling rate, resolution, and

potential gaps in data collection. Identify any data preprocessing steps that have been performed on the data.

3. Data quality: Assess the quality of the data. Look for missing values, outliers, noise, or inconsistencies in the data. Understand the data quality issues that may impact the reliability and accuracy of the predictive models. Address any data quality issues through data cleaning, imputation, or other necessary techniques.
4. Data attributes: Explore the different attributes or variables present in the data. Understand the meaning and significance of each attribute. Identify the input features that can be used for predictive modeling, such as sensor readings, maintenance records, or other relevant parameters. Also, identify the target variable or outcome that will be predicted, such as equipment failure or remaining useful life (RUL).
5. Data relationships: Analyze the relationships between different data attributes. Look for correlations or dependencies among the variables. Identify any contextual factors that may influence the equipment's health or failure. Consider the time dependencies and sequential patterns in the data, as they can be valuable for predictive maintenance.
6. Data volume and scalability: Evaluate the volume of data available for analysis. Consider the scalability of the data processing and storage infrastructure to handle the increasing volume of data over time. Assess the feasibility of handling big data challenges if large-scale data is involved.

7. Data integration: Explore the possibility of integrating data from different sources to enrich the predictive maintenance analysis. Consider combining maintenance data with other relevant datasets, such as weather data or production data, to gain more comprehensive insights.
8. Data privacy and security: Ensure compliance with data privacy regulations and security requirements. Identify any sensitive or confidential information present in the data and establish protocols to protect it. Implement appropriate data anonymization or encryption techniques if necessary.

Data understanding provides the foundation for subsequent steps in the predictive maintenance process, such as

- Data preprocessing,
- Feature engineering
- Model development

It helps in selecting the appropriate data for analysis, understanding its limitations, and making informed decisions regarding data handling and model development.

3.3.1 Data Collection

The relevant data sources and corrected data has to be identified to proceed with the project. It takes a considerable amount of time to scrutinize the data before the main phase. It is required to discuss

with the maintenance team and also suppliers in data collection to verify the data sources. The maintenance team has the relevant experience and knowledge on machine parts of the system. Therefore it is necessary to contact the data company before accessing the suitable data sources. Collected data comes in the form of csv or excel files and all of them are referenced with timestamps. The issues can occur when the timestamps from different data sources differ, and it has to be resolved by a sampling procedure.

3.3.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an essential step in understanding the characteristics and patterns within the data. It involves examining the data visually and statistically to gain insights, detect patterns, identify outliers, and determine the appropriate preprocessing steps. Following are the steps that were followed in our EDA process.

1. Data loading: Load the data into the analysis environment, whether it's a programming language like Python or R, or a data analysis tool such as Colab Notebook or Excel.
2. Data structure overview: Get a high-level understanding of the data by examining its structure. Check the dimensions (number of rows and columns), data types of variables, and any missing values.
3. Descriptive statistics: Calculate and analyze summary statistics of the data. This includes measures such as mean, median, standard deviation, minimum, maximum, quartiles,

and other relevant statistics. This provides insights into the central tendencies, spread, and distribution of the data.

4. Data visualization: Create visual representations of the data to understand its distribution and relationships. Common visualizations include histograms, box plots, scatter plots, line plots, bar plots, and correlation matrices. Visualizations help in identifying patterns, outliers, and relationships between variables.
5. Data cleaning: Identify and handle missing values, outliers, or inconsistent data points. Determine appropriate strategies for imputing missing values or deciding whether to remove or correct outliers. This step ensures the data is ready for further analysis and modeling.
6. Feature exploration: Analyze individual features or variables to understand their distributions, relationships with the target variable (if applicable), and any notable patterns. Identify variables that may be relevant for predictive modeling and further feature engineering.
7. Correlation analysis: Examine the correlations between variables to identify potential relationships or dependencies. Use correlation matrices, scatter plots, or other visualizations to understand the strength and direction of the relationships. This can help identify highly correlated variables or multicollinearity issues.

8. Data transformations: Explore the need for data transformations to address skewness, heteroscedasticity, or

non-linearity in the data. Consider techniques like log transformations, power transformations, or normalization to improve the data suitability for modeling.

9. Hypothesis testing: Perform statistical tests or hypothesis tests to validate assumptions, test relationships, or compare groups. This step helps in making objective decisions based on statistical evidence and supports further data analysis.
10. Iterative analysis: EDA is an iterative process, and the steps mentioned above may need to be repeated as new insights are gained or data preprocessing steps are applied. It involves exploring different angles of the data and refining the analysis based on the insights obtained.

3.4 Data Preparation

Data preparation is the basic step in data analysis and it requires most of the time in data engineering process. Preparing data to feed into the model will be the major task in this phase. After collecting all the relevant data a new data set must be prepared by using strategies like synthetic data generation and transformation.

Data integration from different sources, dimensionality reduction and transformation of data are part of the preparation process.

The reasons for data preparation is to alter the issues due to missing data and for formatting the data into a suitable ML format [78]. The data set obtained after preparation will be clear, quality ensured, error-free, complete and obviously smaller in size compared with the original data set.

3.5 Predictive Modeling

The Modeling phase is about trying out different modeling techniques and comparing the outcomes for selecting the best model with optimum hyper parameters. There is a connection between modeling and data preparation such that data issues can be identified during the modeling stage. A correct model type can be chosen depending on the available data in the preparation phase.

The expected outcome of the model is also considered during the model selection. Once the data set is finalized, the selection of model design and correct algorithm is chosen based on the data available and also the output requirement of work.

ExtraTreesRegressor model is suitable for this work as it can handle the data efficiently. An introduction and theory about these models are included in the literature review. The software tools used in this phase are colab Notebook with Python programming.

3.6 Evaluation

Building the model is not the final phase of the project. The generated model is improvised until the model is best suited. The Evaluation phase includes comparing the model results with the actual data and suggest evaluation metrics such as F1, score, Accuracy. Also, the outcomes of the model have to be evaluated with the business objectives .

3.7 Deployment

The best generated model and optimal results obtained should be presented to the organization in order to ensure the benefits of the results to the organization. Deploying a machine learning (ML) model involves making the model available for others to use in a production environment. The process can vary depending on the specific requirements and technologies involved.

4.RESULTS AND DISCUSSIONS

This chapter discusses the insights and results gained during the different phases of the thesis work, following CRISP-DM methodology. Business understanding section portrays the business objectives of the project work. The data collection process is described in data collection section. The different sources of data, data quality analysis and EDA is described in data understanding section. Data preparation section describes all the data pre-processing performed in order to make the data suitable for the ML model. Model design section discusses the test designing and all the models which were used for prediction. The evaluation of the ML models and the results are presented in the evaluation section.

4.1 Business Understanding

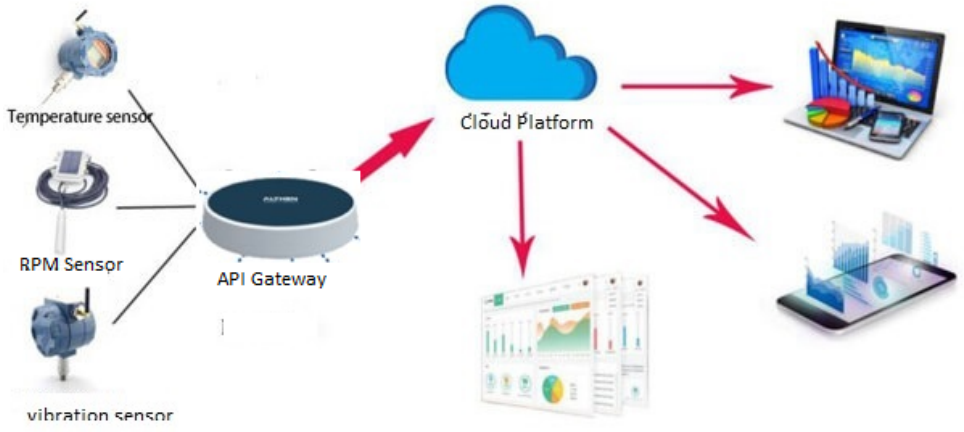


Fig 4.1 Business Model

Business Understanding — this phase consists of a very precise specification of the problem together with methods of evaluating the achievement of the goal.

Problem ⟹ Model ⟹ Solution

Our goal is to create a model of the problem, which we use in turn to find RUL of the plant as whole. The model should be precise enough to make the solution meaningful, otherwise, we will make too many assumptions and approximations which will make the solution far from real and meaningless.

Our project uses 51 sensors out of which 24 sensors are vibration sensor,12 Temperature sensor and 12 RPM sensors in their shop floor. Our goal is to create an model which can predict the RUL of the shop floor as whole .

The presented problem can be formulated as one or more machine learning models: classification, regression or others [**2**]. Finally, it is necessary to decide which metrics will be used to evaluate the model. This metric will also allow you to compare the models and determine whether the model is underfitting or overfitting

4.1 Data Collection

Data was provided by the senior Engineers which comprised of historical data and current data.

Client has provided the morphed data with 166441 rows to analyse

The data had total 53 columns,which had the corresponding values from the sensors and RUL values.Two columns cannot be used for predicting 1. Unnamed: 0 and 2.timestamp

Two more columns without correlation is removed.We were left with 48 Columns used for predicting and 1 column is used as target column.

4.2 Data loading and understanding

The raw data was loaded into python data frame as shown below.

```
from google.colab import drive
drive.mount('/content/drive')
location = "/content/drive/My Drive/Colab Notebooks/data/rul_hrs DataSet.csv"
df = pd.read_csv(location)
```

Next step is to understand the data. There are 166442 rows and 53 columns. The details of the columns are displayed below. For all the sensors there were numeric values and we know the data types of all the sensors and the total count of not null values in the columns. As we have found that there are no null values, The next process is to eliminate outliers if any. Data description is shown below.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 166441 entries, 0 to 166440
Data columns (total 53 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Unnamed: 0   166441 non-null  int64
 1   timestamp    166441 non-null  object
 2   sensor_00    166441 non-null  float64
 3   sensor_01    166441 non-null  float64
 4   sensor_02    166441 non-null  float64
 5   sensor_03    166441 non-null  float64
 6   sensor_04    166441 non-null  float64
 7   sensor_05    166441 non-null  float64
 8   sensor_06    166441 non-null  float64
 9   sensor_07    166441 non-null  float64
 10  sensor_08    166441 non-null  float64
 11  sensor_09    166441 non-null  float64
 12  sensor_10    166441 non-null  float64
 13  sensor_11    166441 non-null  float64
 14  sensor_12    166441 non-null  float64
 15  sensor_13    166441 non-null  float64
 16  sensor_14    166441 non-null  float64
 17  sensor_16    166441 non-null  float64
 18  sensor_17    166441 non-null  float64
 19  sensor_18    166441 non-null  float64
 20  sensor_19    166441 non-null  float64
 21  sensor_20    166441 non-null  float64
 22  sensor_21    166441 non-null  float64
 23  sensor_22    166441 non-null  float64
 24  sensor_23    166441 non-null  float64
 25  sensor_24    166441 non-null  float64
 26  sensor_25    166441 non-null  float64
 27  sensor_26    166441 non-null  float64
 28  sensor_27    166441 non-null  float64
 29  sensor_28    166441 non-null  float64
 30  sensor_29    166441 non-null  float64
 31  sensor_30    166441 non-null  float64
 32  sensor_31    166441 non-null  float64
 33  sensor_32    166441 non-null  float64
 34  sensor_33    166441 non-null  float64
 35  sensor_34    166441 non-null  float64
 36  sensor_35    166441 non-null  float64
 37  sensor_36    166441 non-null  float64
 38  sensor_37    166441 non-null  float64
 39  sensor_38    166441 non-null  float64
 40  sensor_39    166441 non-null  float64
 41  sensor_40    166441 non-null  float64
 42  sensor_41    166441 non-null  float64
 43  sensor_42    166441 non-null  float64
 44  sensor_43    166441 non-null  float64
 45  sensor_44    166441 non-null  float64
 46  sensor_45    166441 non-null  float64
 47  sensor_46    166441 non-null  float64
 48  sensor_47    166441 non-null  float64
 49  sensor_48    166441 non-null  float64
 50  sensor_49    166441 non-null  float64
 51  sensor_51    166441 non-null  float64
 52  rul          166441 non-null  float64
dtypes: float64(51), int64(1), object(1)
memory usage: 67.3+ MB
```

The spread of Remaining useful life, From the graph we can say it is distributed but it is right skewed. Descriptive analysis on the right also confirms it

4.3 Data quality analysis

Below we describe the data spread or inter quartile range., which gives the spread of data

index	count	mean	std	min	25%	50%	75%	max
sensor_00	166441	2	0	0	2	2	2	3
sensor_01	166441	47	3	22	46	48	49	56
sensor_02	166441	51	4	33	50	52	53	56
sensor_03	166441	43	3	32	42	44	45	48
sensor_04	166441	578	162	3	625	632	637	800
sensor_05	166441	74	19	0	72	77	82	100
sensor_06	166441	13	2	0	13	14	14	22
sensor_07	166441	16	2	0	16	16	16	24
sensor_08	166441	15	2	0	15	15	16	24
sensor_09	166441	15	2	0	15	15	15	25
sensor_10	166441	40	13	0	40	44	47	76
sensor_11	166441	39	14	0	37	43	48	60
sensor_12	166441	28	11	0	27	32	34	45
sensor_13	166441	5	6	0	1	2	5	31
sensor_14	166441	364	127	32	409	420	421	500
sensor_16	166441	402	141	0	451	463	464	740
sensor_17	166441	408	146	0	450	461	467	600
sensor_18	166441	2	1	0	2	3	3	5
sensor_19	166441	568	223	0	651	665	667	879
sensor_20	166441	349	114	0	390	399	400	449
sensor_21	166441	771	253	96	861	879	882	1108
sensor_22	166441	437	170	0	460	504	533	594
sensor_23	166441	870	316	0	947	979	1001	1156
sensor_24	166441	536	204	0	599	624	628	1000
sensor_25	166441	625	247	0	654	741	751	840
sensor_26	166441	760	272	43	752	867	903	1214
sensor_27	166441	477	154	0	442	477	525	2000
sensor_28	166441	878	340	4	833	1000	1058	1841
sensor_29	166441	589	257	1	524	699	768	1466
sensor_30	166441	586	215	0	617	652	685	1600
sensor_31	166441	854	320	24	831	923	993	1800
sensor_32	166441	782	283	0	750	859	923	1839
sensor_33	166441	479	171	6	488	517	569	1579
sensor_34	166441	225	94	55	166	190	315	426
sensor_35	166441	391	145	0	333	405	501	694
sensor_36	166441	532	304	2	209	610	838	984
sensor_37	166441	75	31	0	54	76	96	175
sensor_38	166441	49	11	24	45	49	54	409
sensor_39	166441	37	16	19	33	36	40	548
sensor_40	166441	65	21	23	56	64	73	513
sensor_41	166441	36	8	21	33	35	38	219
sensor_42	166441	36	11	22	33	35	38	374
sensor_43	166441	43	12	24	39	42	46	409
sensor_44	166441	40	9	26	36	39	43	291
sensor_45	166441	41	10	26	36	39	43	231
sensor_46	166441	45	14	26	39	43	48	321
sensor_47	166441	43	9	27	39	42	46	220
sensor_48	166441	122	64	26	73	110	163	502
sensor_49	166441	53	14	27	47	51	56	459
sensor_51	166441	202	120	28	178	198	211	1000
rul	166441	289	226	0	99	226	445	837

Fig No: 4.2 Statistical Data

The spread of Remaining useful life, From the graph we can say it is distributed but it is right skewed. Descriptive analysis on the right also confirms it

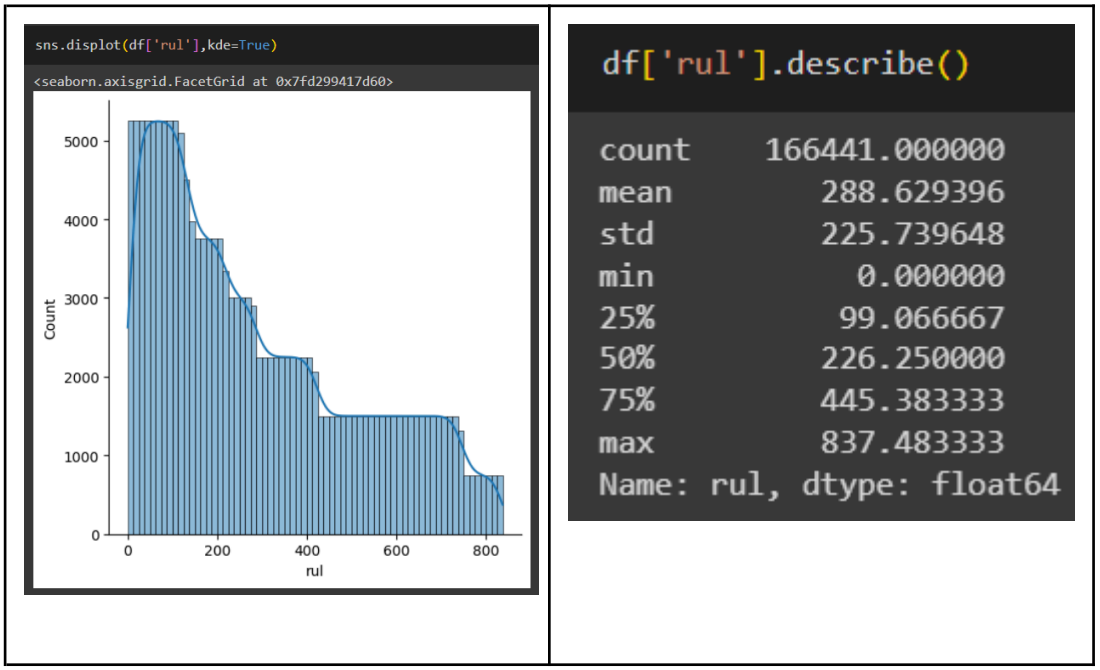
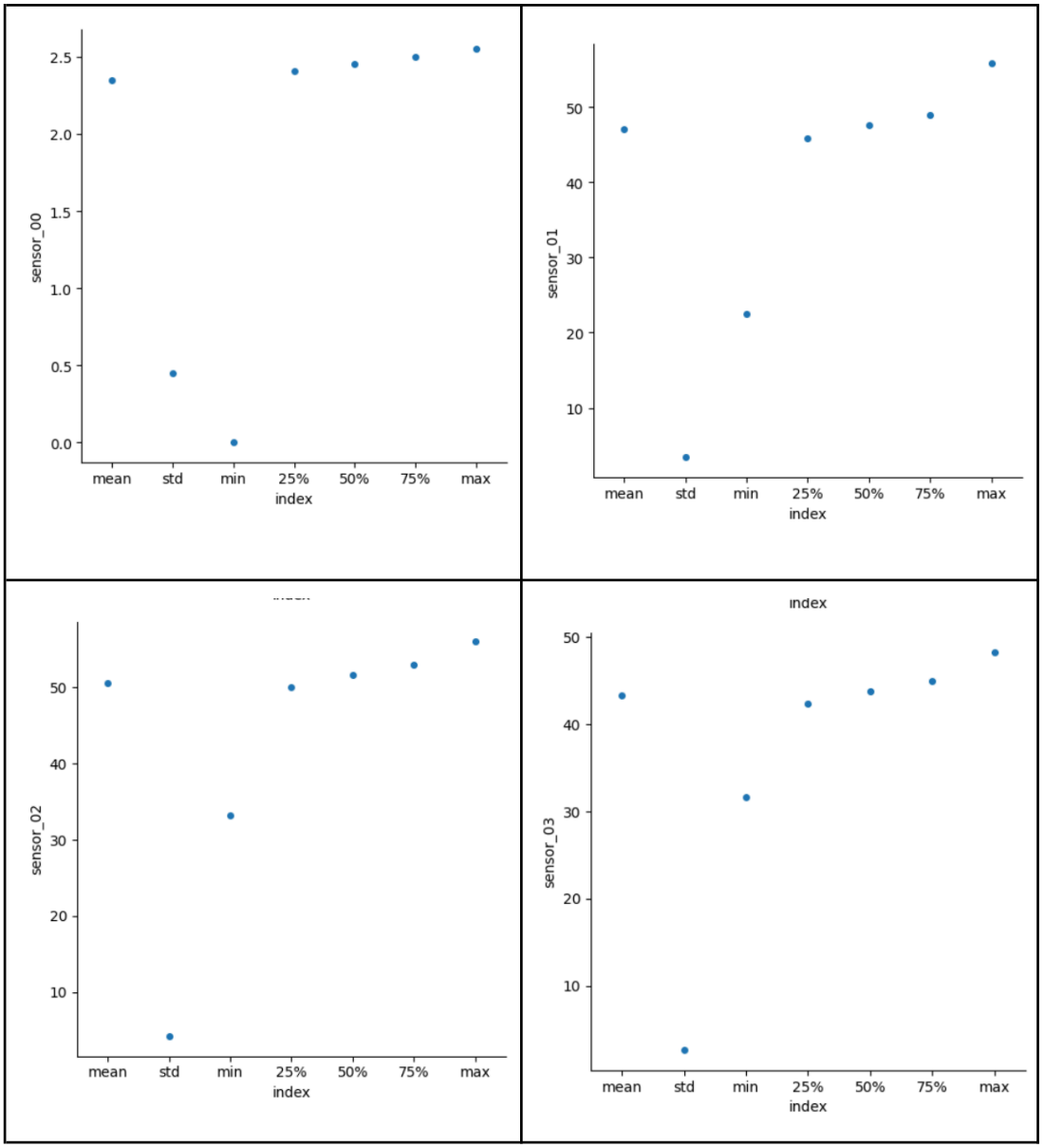
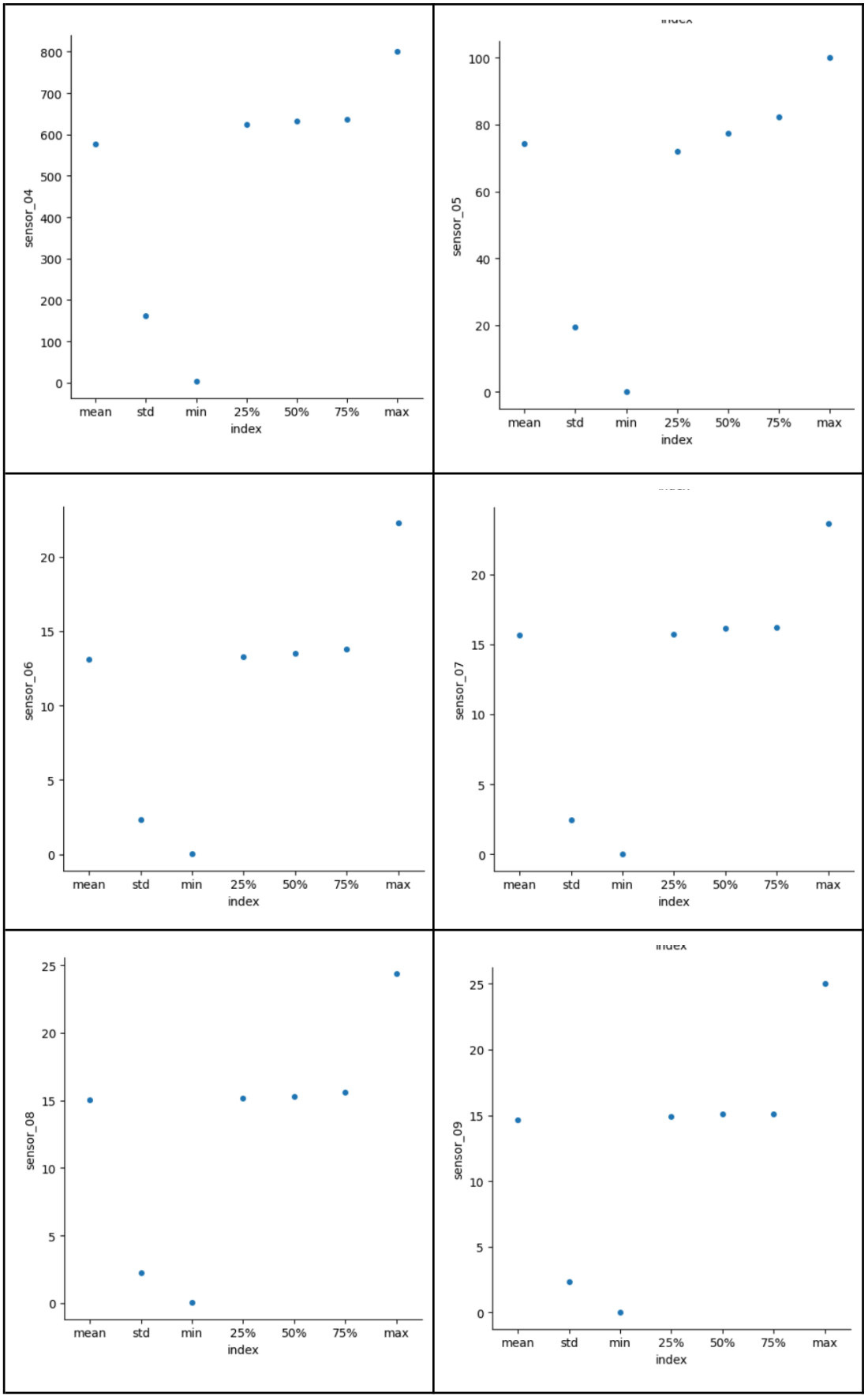
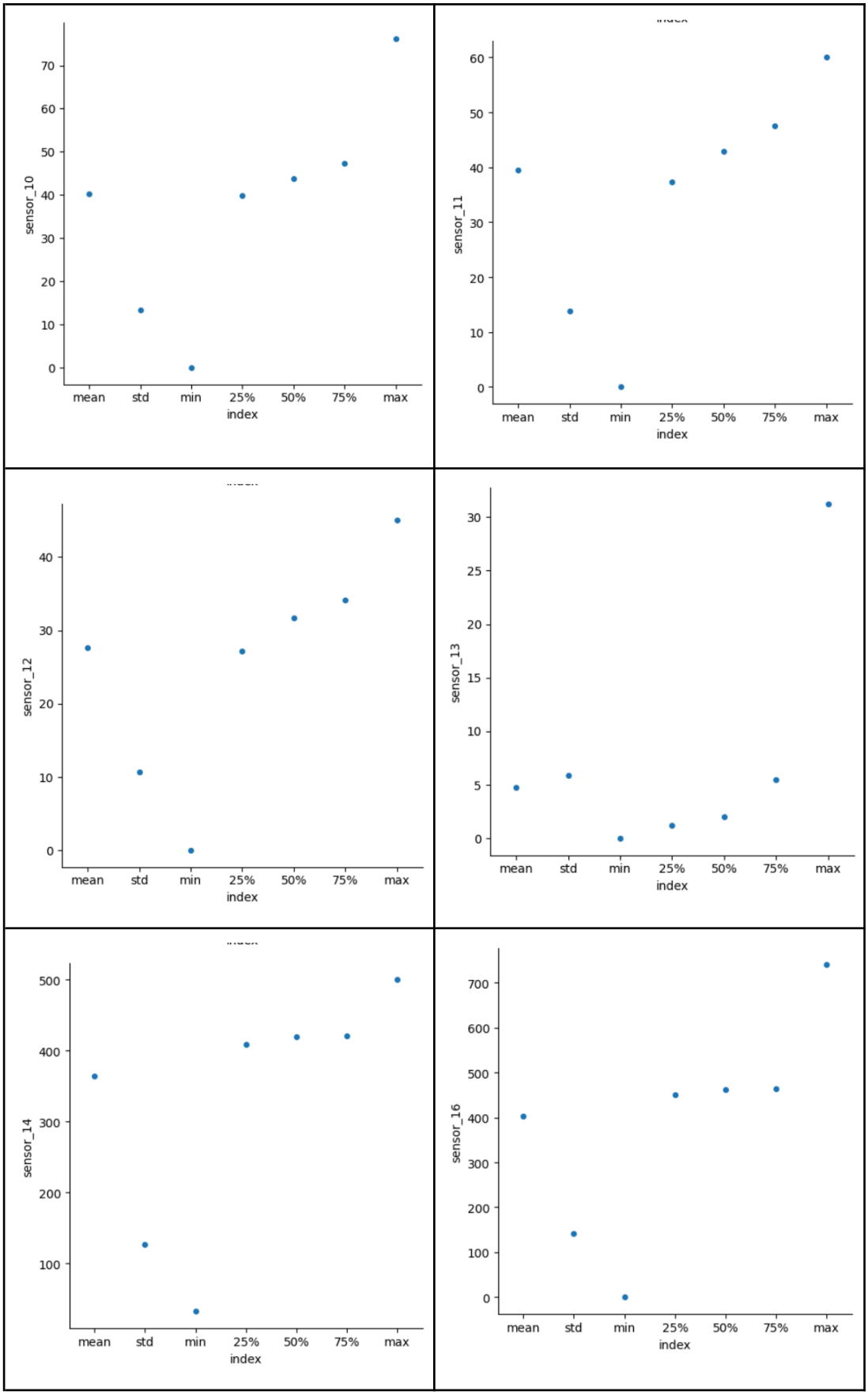


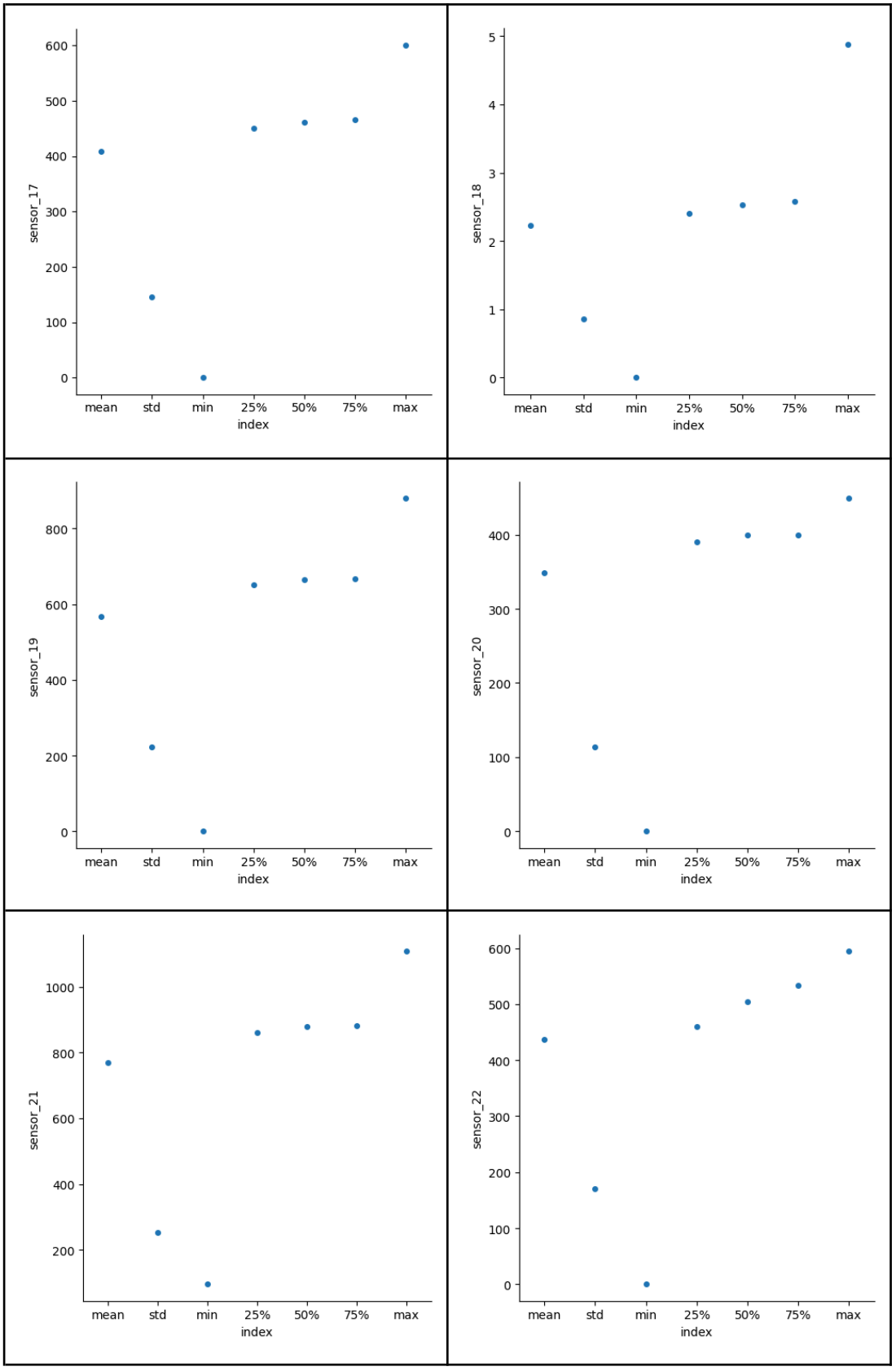
Fig No 4.3 RUL Distribution

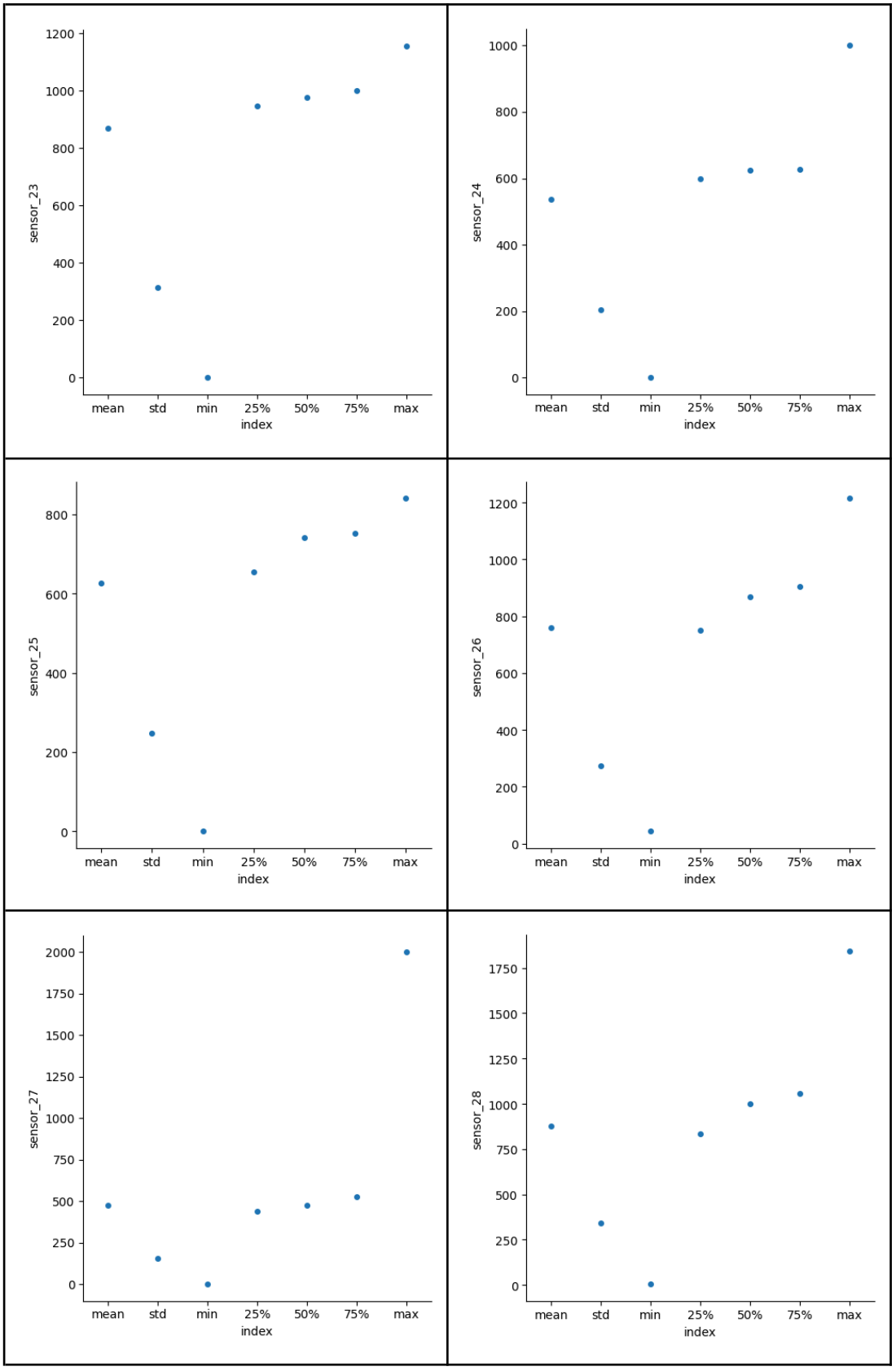
Plotting the descriptive statistics to find any outliers are there for each sensor. plot was done against stats vs count (Feature exploration)

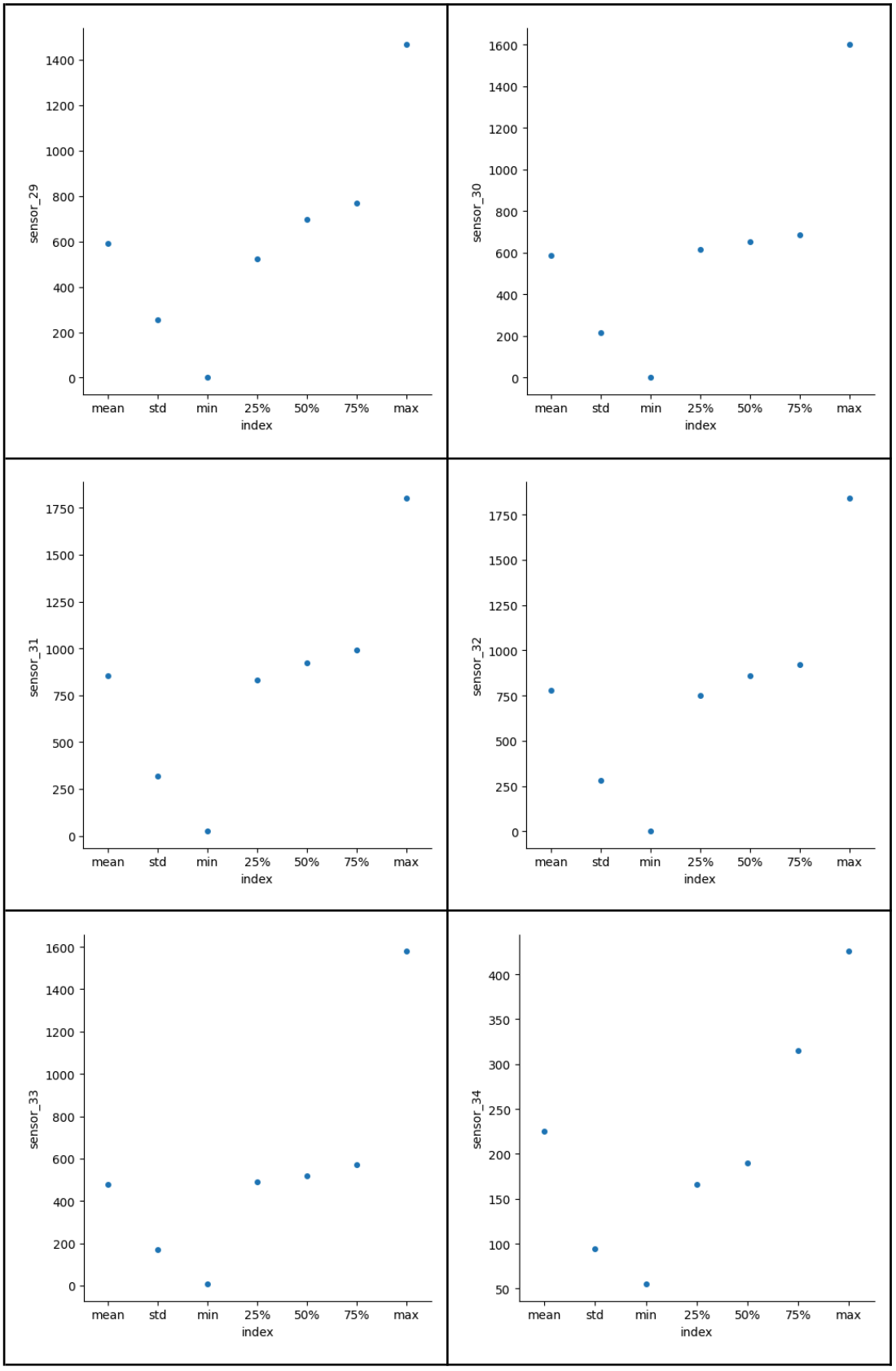

sensor_00
2.5
2.0
1.5
1.0
0.5
0.0
mean
std
min
25%
50%
75%
max
index
sensor_01
50
40
30
20
10
mean
std
min
25%
50%
75%
max
index
sensor_02
50
40
30
20
10
mean
std
min
25%
50%
75%
max
index
index
sensor_03
50
40
30
20
10
mean
std
min
25%
50%
75%
max
index

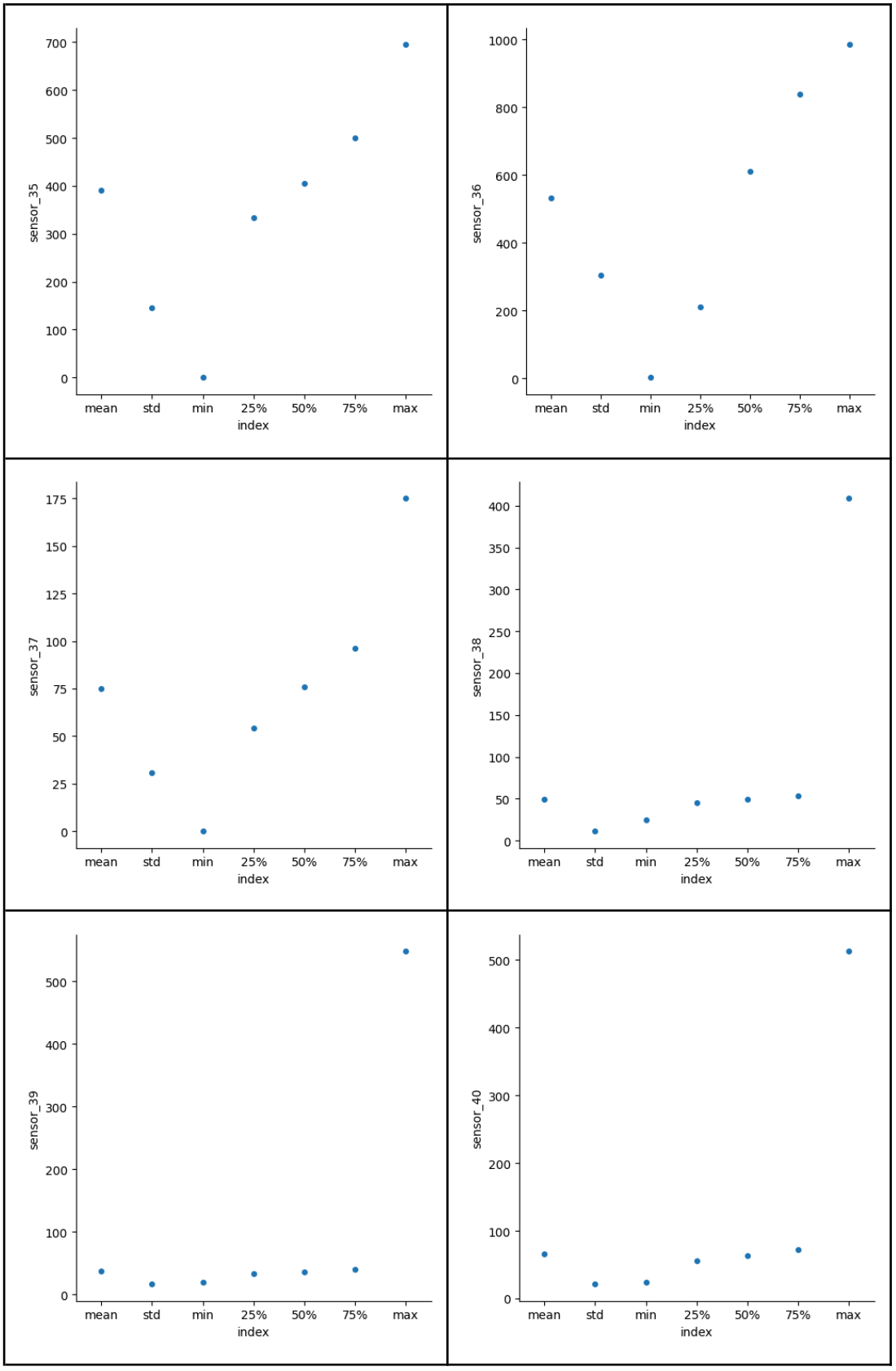

sensor_04
800
700
600
500
400
300
200
100
0
mean
std
min
25%
50%
75%
max
index
sensor_05
100
80
60
40
20
0
mean
std
min
25%
50%
75%
max
index
sensor_06
20
15
10
5
0
mean
std
min
25%
50%
75%
max
index
sensor_07
20
15
10
5
0
mean
std
min
25%
50%
75%
max
index
sensor_08
25
20
15
10
5
0
mean
std
min
25%
50%
75%
max
index
sensor_09
25
20
15
10
5
0
mean
std
min
25%
50%
75%
max
index

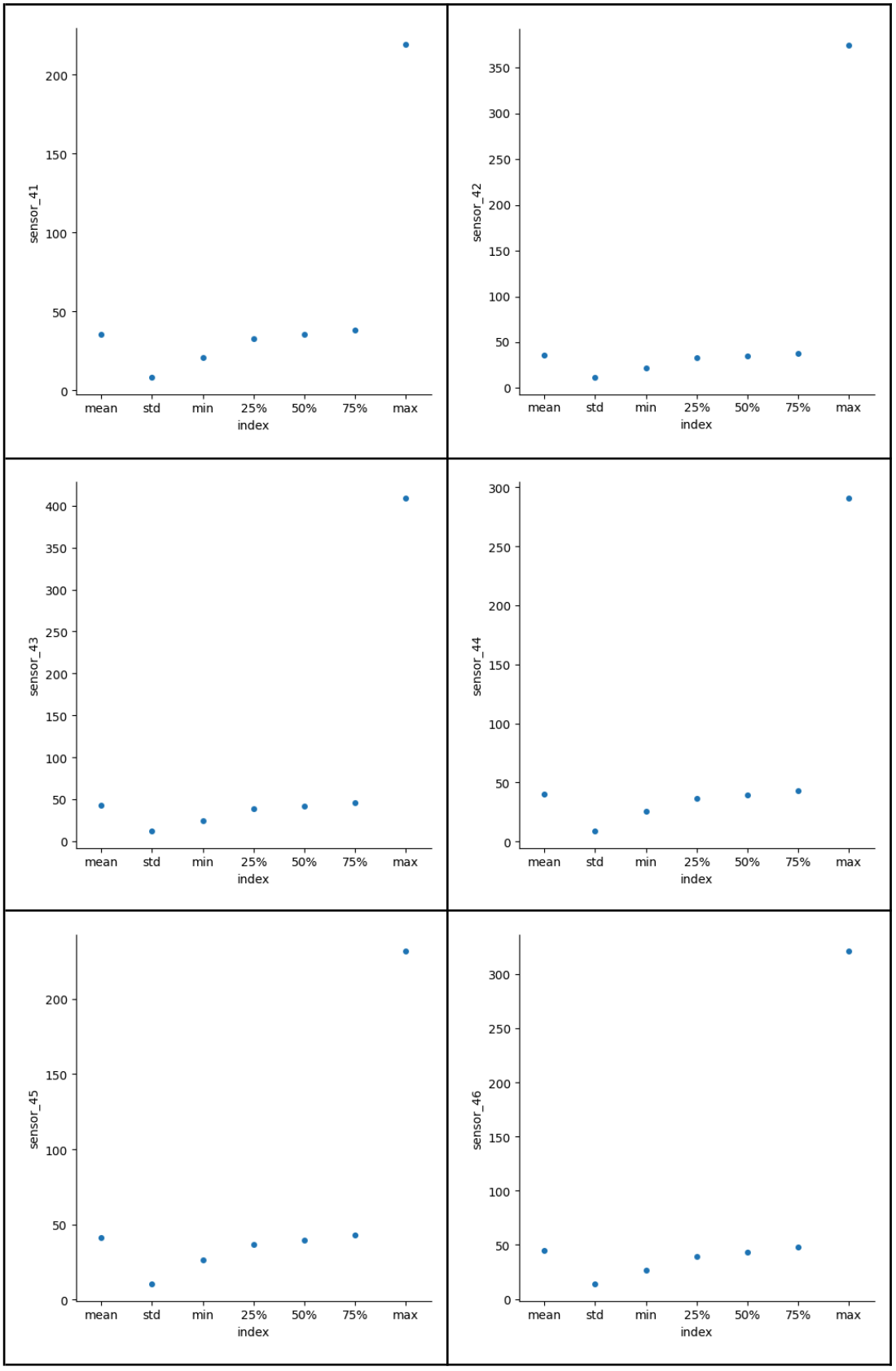

sensor_10
sensor_11
sensor_12
sensor_13
sensor_14
sensor_16
mean
std
min
25%
50%
75%
max
index


sensor_17
sensor_18
sensor_19
sensor_20
sensor_21
sensor_22
mean
std
min
25%
50%
75%
max
index


sensor_23
1200
1000
800
600
400
200
0
mean
std
min
25%
50%
75%
max
index
sensor_24
1000
800
600
400
200
0
mean
std
min
25%
50%
75%
max
index
sensor_25
800
600
400
200
0
mean
std
min
25%
50%
75%
max
index
sensor_26
1200
1000
800
600
400
200
0
mean
std
min
25%
50%
75%
max
index
sensor_27
2000
1750
1500
1250
1000
750
500
250
0
mean
std
min
25%
50%
75%
max
index
sensor_28
1750
1500
1250
1000
750
500
250
0
mean
std
min
25%
50%
75%
max
index


sensor_29
sensor_30
sensor_31
sensor_32
sensor_33
sensor_34
mean
std
min
25%
50%
75%
max
index


sensor_35
sensor_36
sensor_37
sensor_38
sensor_39
sensor_40
mean
std
min
25%
50%
75%
max
index


sensor_41
sensor_42
sensor_43
sensor_44
sensor_45
sensor_46
mean
std
min
25%
50%
75%
max
index

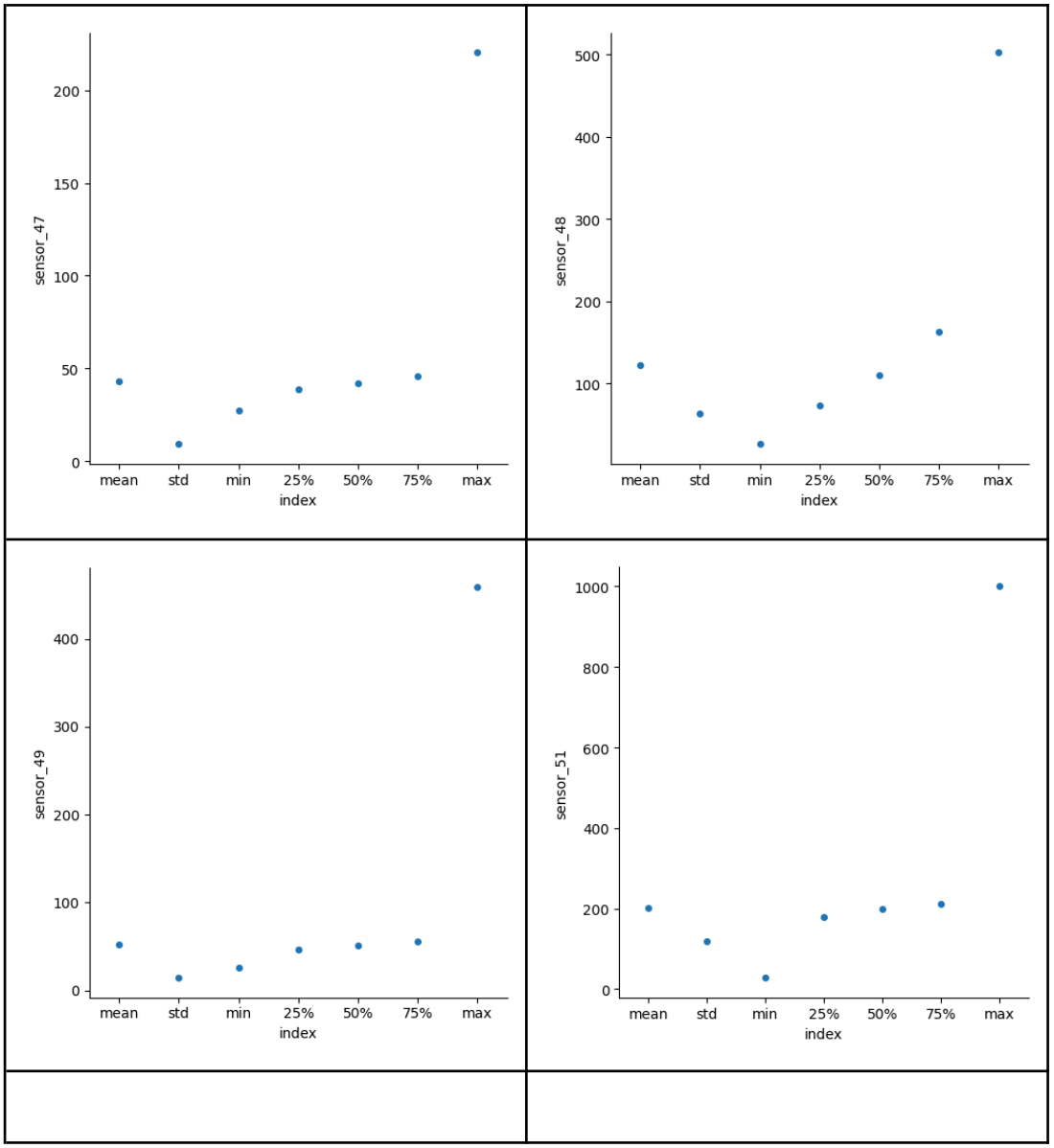


Fig No 4.4: Outliers

Histogram plot of data to see if the values are distributed and the skewness of the data with data spread.

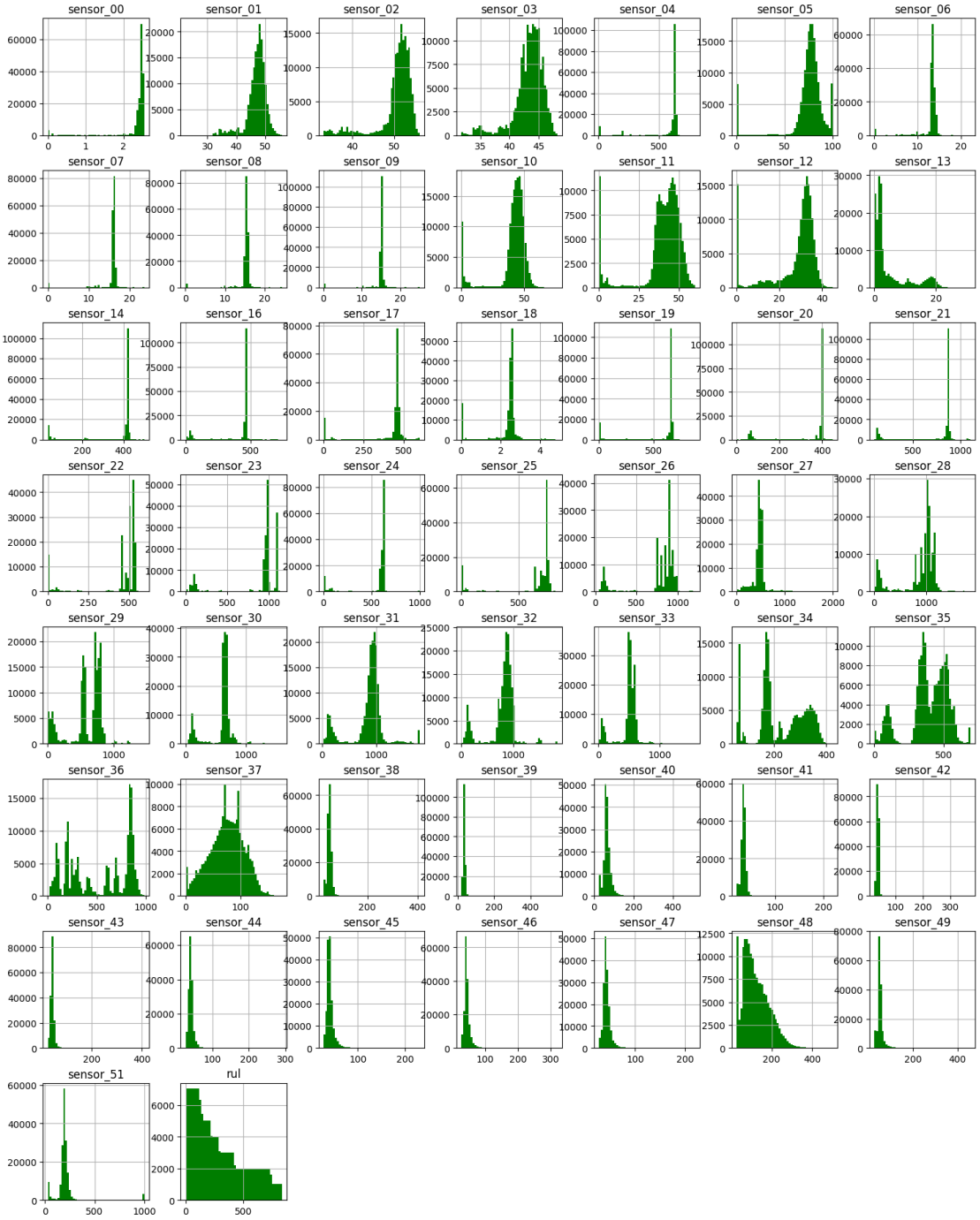
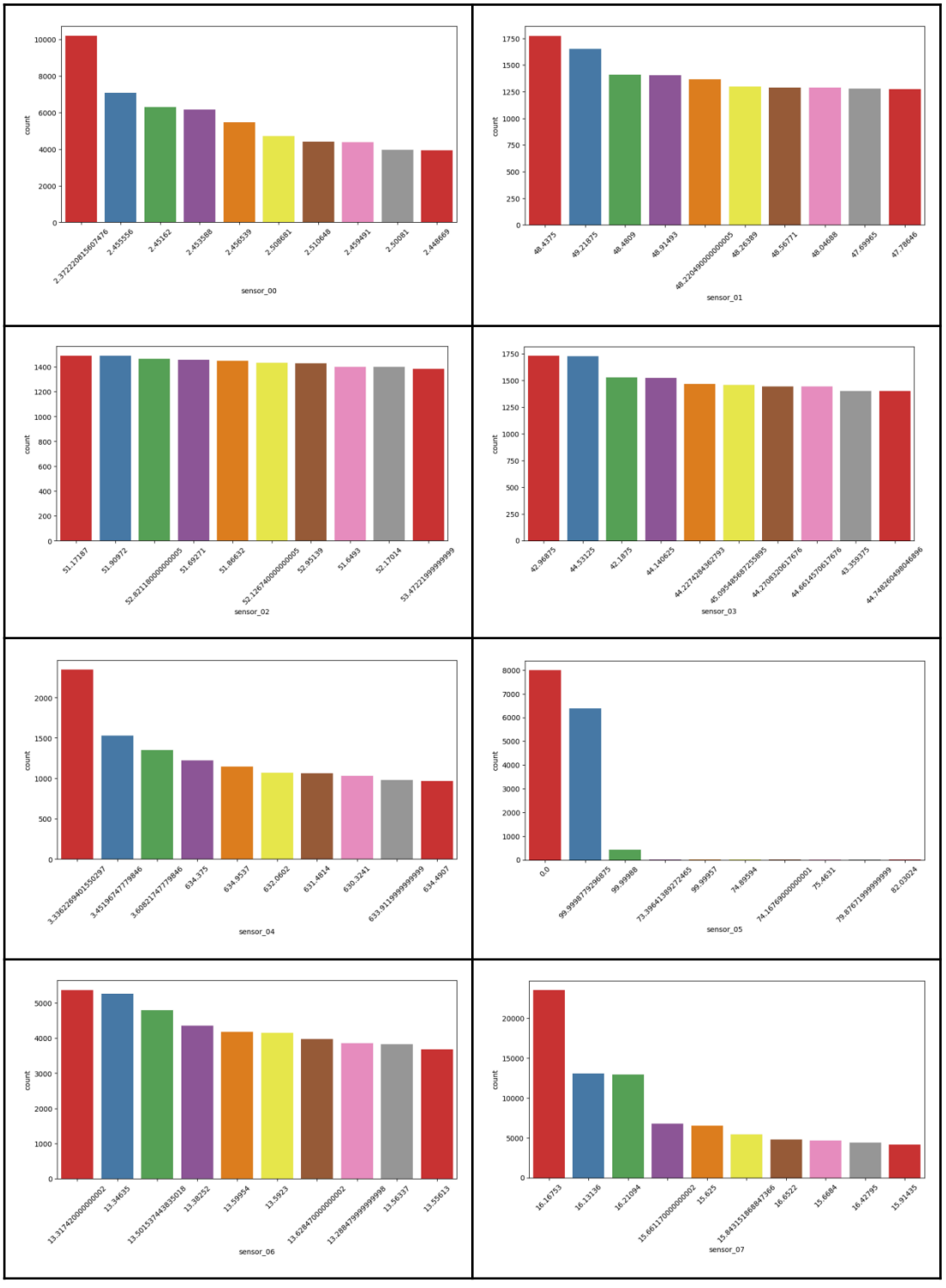
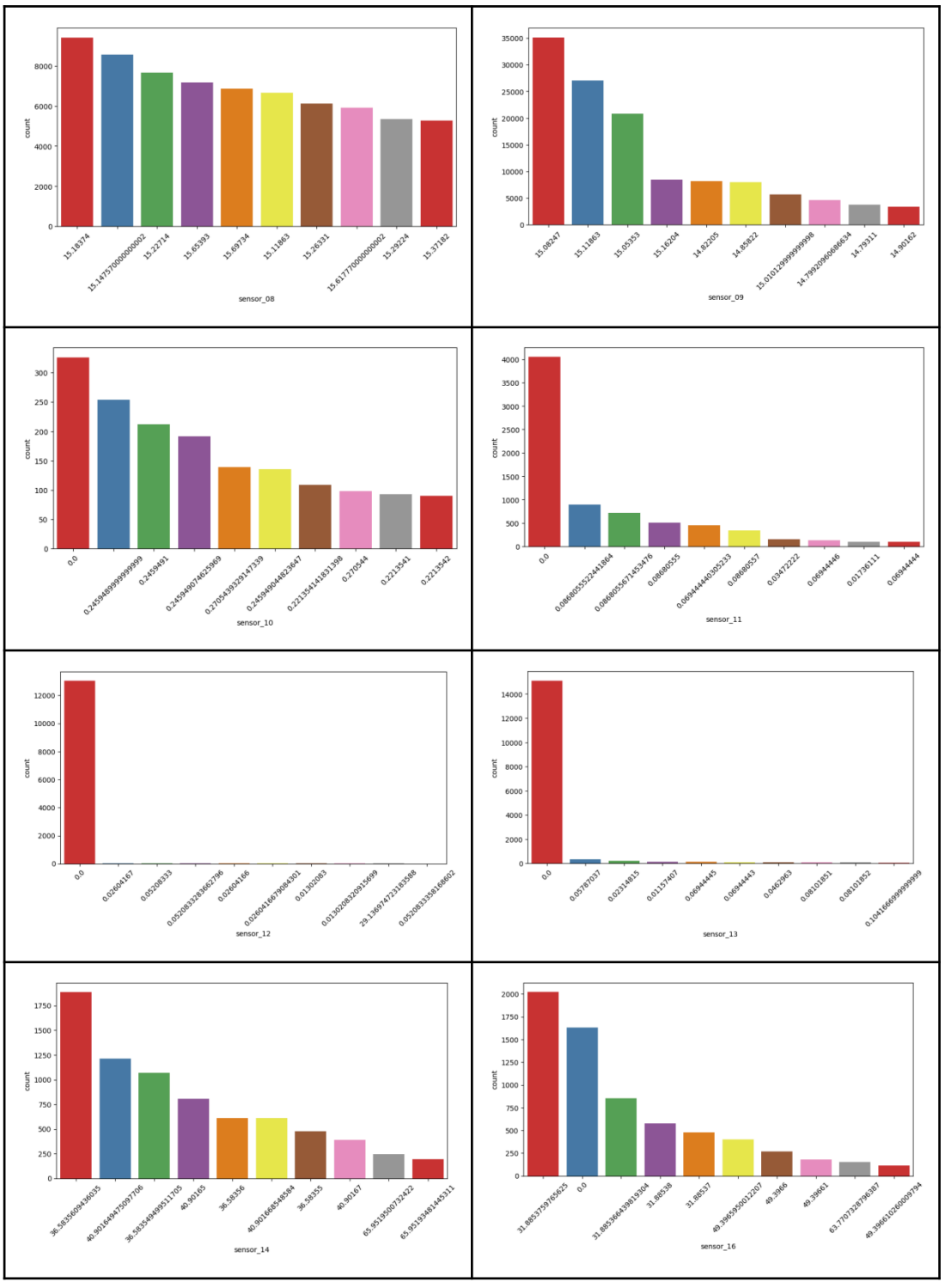
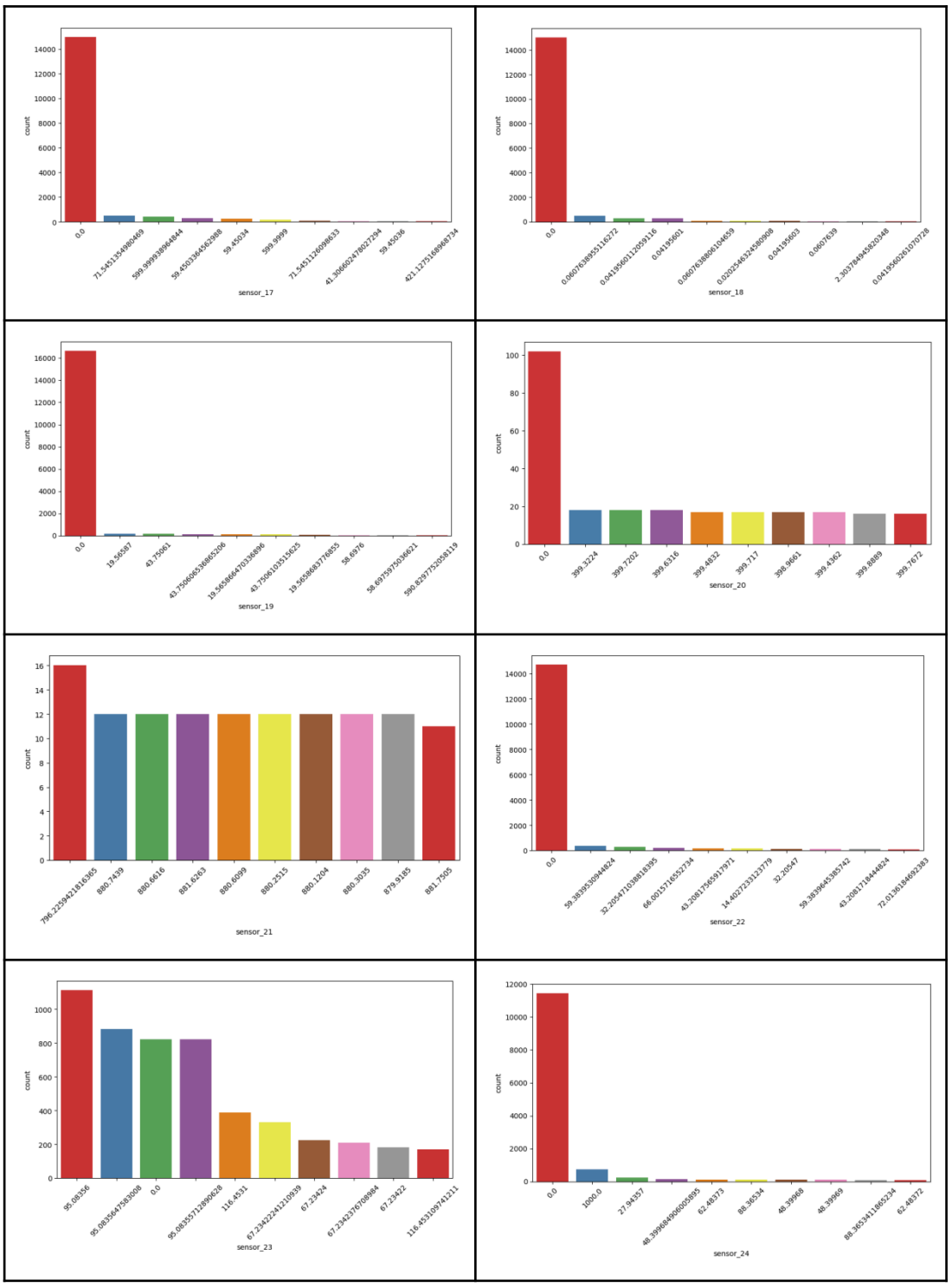


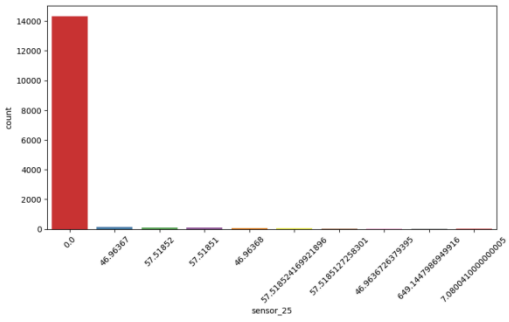
Fig No 4.5 Histogram plots for all sensors

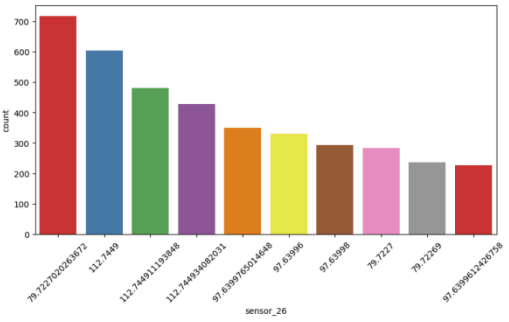
Below graphs are drawn to check the distribution of values against the count. This is to verify spread of data.

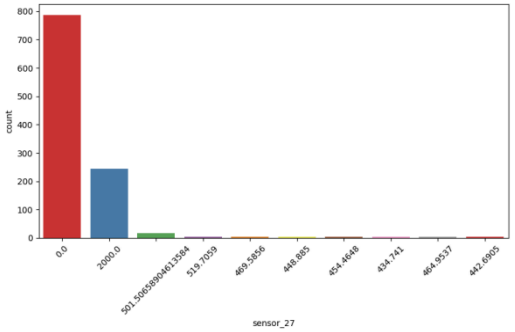

count
sensor_00
2.372220815607476
2.455556
2.45162
2.453588
2.456539
2.508681
2.510648
2.459491
2.50081
2.448669
sensor_01
48.4375
49.21875
48.4809
48.91493
48.22049000000005
48.26389
48.56771
48.04688
47.69965
47.78646
sensor_02
51.17187
51.90972
52.82218000000005
51.69271
51.86632
52.13674000000005
52.95139
51.6493
52.17014
53.47221999999999
sensor_03
42.96875
44.53125
42.1875
44.140625
44.227428436279
45.095485687255895
44.27082060176 76
44.661457061767 6
43.359375
44.74626049804 6896
sensor_04
3.3362269401550297
3.4519674778846
3.6082174778846
634.375
634.9537
632.0602
631.4814
630.3241
633.9119999999999
634.4907
sensor_05
0.0
99.99987792968 75
99.99988
73.396413892724 65
99.99957
74.89594
74.16769000000001
75.4631
79.67671999999999
82.03024
sensor_06
13.317420000000002
13.34635
13.501374438350 18
13.38252
13.59954
13.5923
13.628470000000002
13.388479999999998
13.56337
13.55613
sensor_07
16.16753
16.13136
16.21094
15.661170000000002
15.625
15.843151868847366
16.6922
15.6684
16.42795
15.91435

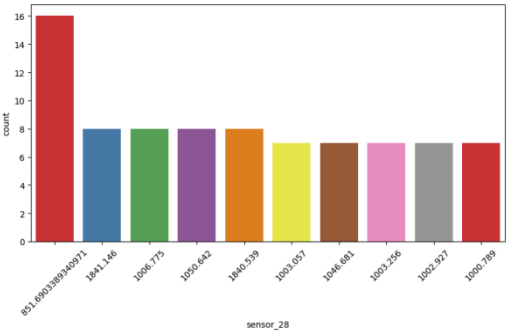

count
sensor_08
15.18374
15.14757000000002
15.22714
15.65393
15.69734
15.11863
15.26331
15.61777000000002
15.29224
15.37182
sensor_09
15.08247
15.11863
15.05353
15.16204
14.82205
14.85822
15.01012999999998
14.79929060686634
14.79311
14.90162
sensor_10
0.0
0.2459489999999999
0.2459491
0.2459490742025969
0.2705439329147339
0.2459490448236647
0.2213541418831398
0.270544
0.2213541
0.2213542
sensor_11
0.0
0.08680552241864
0.08680556714534476
0.0868055
0.06944444030523333
0.08680557
0.03472222
0.06944446
0.01736111
0.06944444
sensor_12
0.0
0.02604167
0.05208333
0.052083331836627996
0.026041666
0.026041667908430l
0.01302083
0.013020832091569
29.13697472318356
0.052083335816860
sensor_13
0.0
0.05787037
0.02314815
0.01157407
0.06944445
0.06944443
0.0462963
0.08101851
0.08101852
0.10416669999999999
sensor_14
36.583560943603
40.901649475097706
36.583549499511705
40.90165
36.58356
40.901668548584
36.58355
40.90167
65.95195007324220
65.95193481445311
sensor_16
31.885375976562
0.0
31.885366439819304
31.88538
31.88537
49.39695001220703
49.3966
49.39661
63.770732879638
49.396610260009766

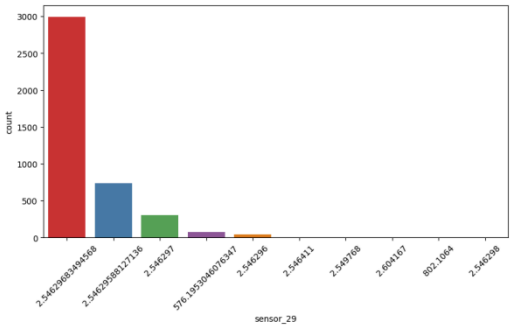

count
sensor_17
0.0
71.5451354980469
599.999389648444
59.4503364562988
59.45034
599.9999
71.5451126098633
41.3066024780273
59.45036
421.127516896734
sensor_18
0.0
0.0607638955116272
0.0419560112059116
0.041956
0.0607638806104659
0.0202546324580908
0.04195603
0.0607639
2.30378494582032
0.0419560261070728
sensor_19
0.0
19.56587
43.75061
43.7506653686206
19.5658664703368
43.7506103515625
19.5658683776855
58.6976
58.6975975036621
590.829775205811
sensor_20
0.0
399.3224
399.7202
399.6316
399.4852
399.717
398.9661
399.4362
399.8889
399.7672
sensor_21
796.225942816365
880.7439
880.6616
881.6363
880.6099
880.2515
880.1204
880.3035
879.9185
881.7505
sensor_22
0.0
59.3839530944824
32.2054710388816
66.0051716552734
43.2081756917971
14.4027231123779
32.20547
59.3839645385742
43.2081718444824
72.0136184692383
sensor_23
95.08356
95.0835647583008
0.0
95.0835712890628
116.4531
67.2342224121094
67.23424
67.2342376708984
67.23422
116.453109741211
sensor_24
0.0
1000.0
27.94357
48.3996849060059
62.48373
88.36534
48.39968
48.39969
88.3653411865234
62.48372

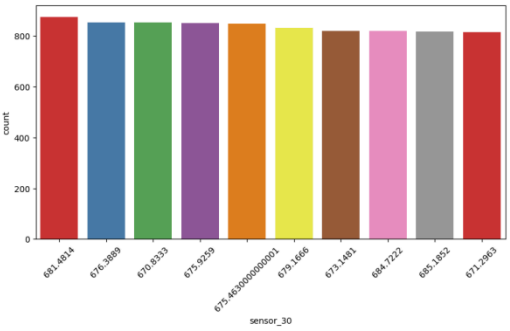

count
sensor_25
0.0
46.96367
57.51852
57.51851
46.96368
57.518524169921896
57.518512725830
46.963726379395
649.1447888849916
7.080041000000005

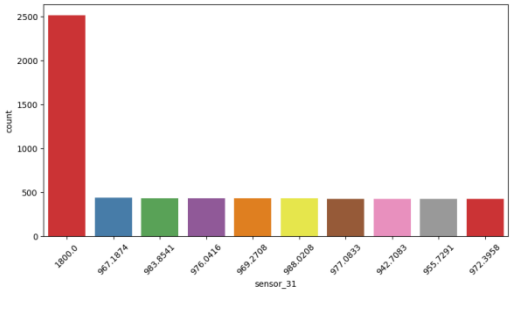

count
sensor_26
79.72270202636672
112.7449
112.74491119384
112.74493408203
97.63899765014648
97.63996
97.63998
79.7227
79.72269
97.63996124267578

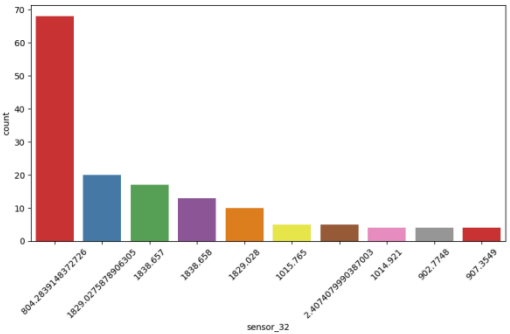

count
sensor_27
0.0
2000.0
501.5065890461384
519.7059
469.5856
448.885
454.4648
434.741
464.9537
442.6905

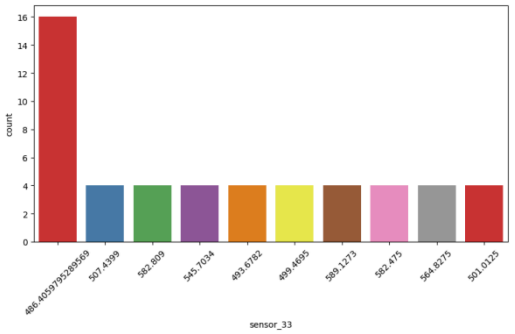

count
sensor_28
851.6903389340971
1841.146
1006.775
1050.642
1840.539
1003.057
1046.681
1003.256
1002.927
1000.789

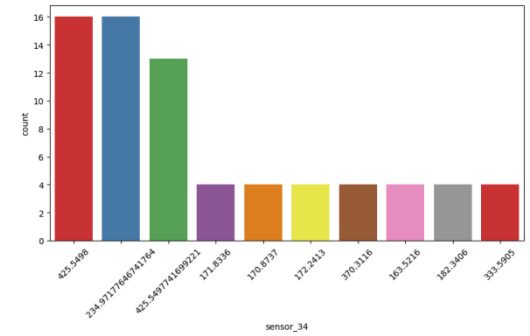

count
sensor_29
2.5462968349568
2.5462958812713
2.546297
576.1953046076347
2.546296
2.546411
2.549768
2.604167
802.1064
2.546298

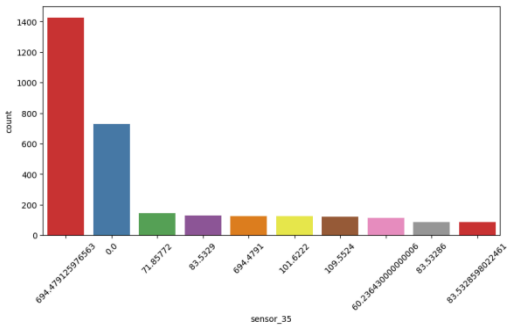

count
sensor_30
681.4814
676.3889
670.8333
675.9259
675.4630000000001
679.1666
673.1481
684.7222
685.1852
671.2963

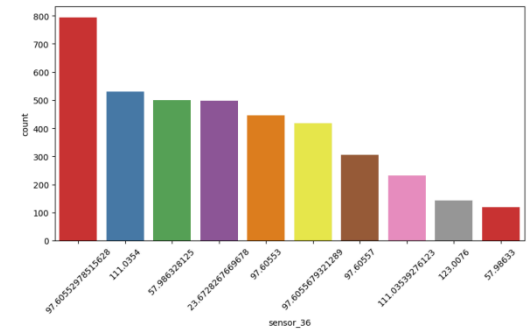

count
sensor_31
1800.0
967.1874
983.8541
976.0416
969.2708
988.0208
977.0833
942.7083
955.7291
972.3958

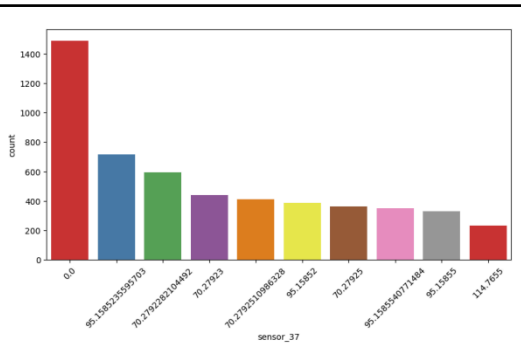

count
sensor_32
804.2839148372726
1839.0275878906305
1838.657
1838.658
1829.028
1015.765
2.4074079900387003
1014.921
902.7748
907.3549

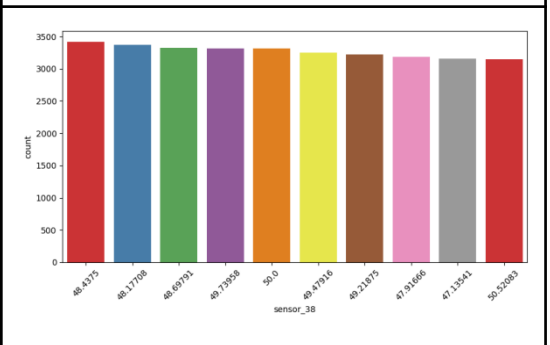

count
sensor_33
486.40597952895369
507.4399
582.809
545.7034
483.6782
499.4695
589.1273
582.475
564.8275
501.0125

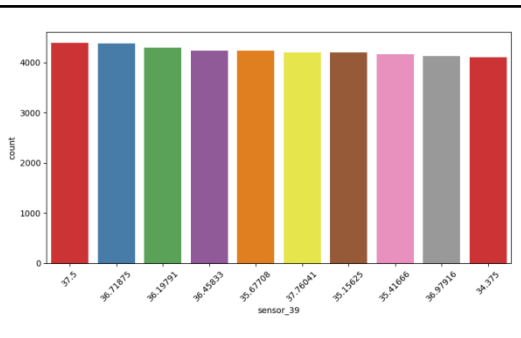

count
sensor_34
425.5498
234.97177664741764
425.5497741699221
171.8336
170.8737
172.2413
370.3116
163.5216
182.3406
333.5905

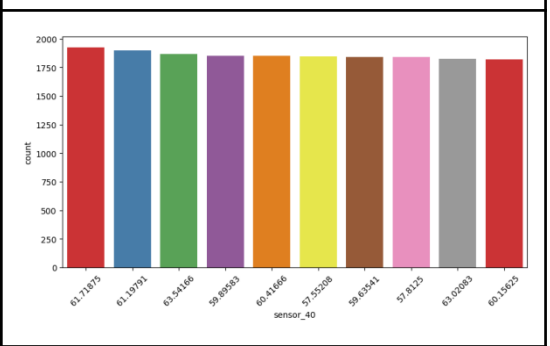

count
sensor_35
694.479125976563
0.0
71.85772
83.5329
694.4791
101.6222
109.5524
60.236430000000006
83.53286
83.53285980224161

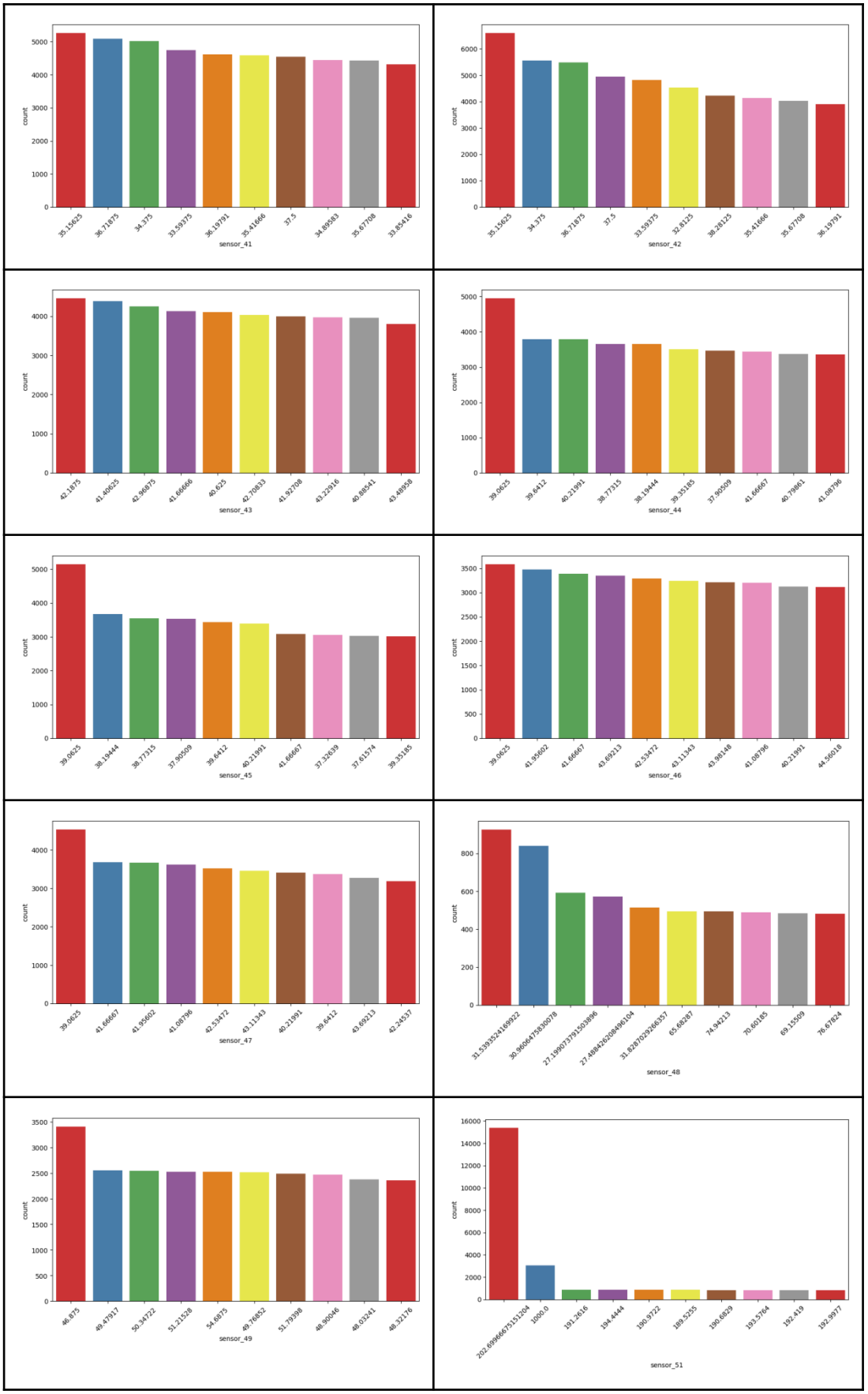

count
sensor_36
97.60552978515628
111.0354
57.986328125
23.67282676696678
97.60553
97.60556792321289
97.60557
111.03539276123
123.0076
57.98633


count
sensor_37
0.0
95.15852359570313
70.27923621044922
70.27923
70.27925109863328
95.15852
70.27925
95.15855407714844
95.15855
114.7655


count
sensor_38
48.4375
48.17708
48.69791
49.73958
50.0
49.47916
49.21875
47.91666
47.13541
50.52083


count
sensor_39
37.5
36.71875
36.19791
36.45833
35.67708
37.76041
35.15625
35.41666
36.97916
34.375


count
sensor_40
61.71875
61.19791
63.54166
59.89583
60.41666
57.55208
59.63541
57.8125
63.02083
60.15625


count
sensor_41
35.15625
36.71875
34.375
33.59375
36.19791
35.41666
37.5
34.69583
35.67708
33.85416
sensor_42
35.15625
34.375
36.71875
37.5
33.59375
32.8125
38.28125
35.41666
35.67708
36.19791
sensor_43
42.1875
41.40625
42.96875
41.66666
40.625
42.70833
41.97708
43.22916
40.88541
43.48958
sensor_44
39.0625
39.6412
40.21991
38.77315
38.19444
39.35185
37.90509
41.66667
40.79861
41.08796
sensor_45
39.0625
38.19444
38.77315
37.90509
39.6412
40.21991
41.66667
37.32639
37.61574
39.35185
sensor_46
39.0625
41.95602
41.66667
43.69213
42.53472
43.11343
43.98148
41.08796
40.21991
44.56018
sensor_47
39.0625
41.66667
41.95602
41.08796
42.53472
43.11343
40.21991
39.6412
43.69213
42.24537
sensor_48
31.539352416992
30.960647583008
27.199073791503896
27.488426208496104
31.828702926635742
65.68287
74.94213
70.60185
69.15509
76.67824
sensor_49
46.875
49.47917
50.34722
51.21526
54.6875
49.76852
51.79398
48.90046
48.03241
48.32176
sensor_51
202.69366671511204
1000.0
191.2616
194.4444
190.9722
189.5255
190.6629
193.5764
192.419
192.9977

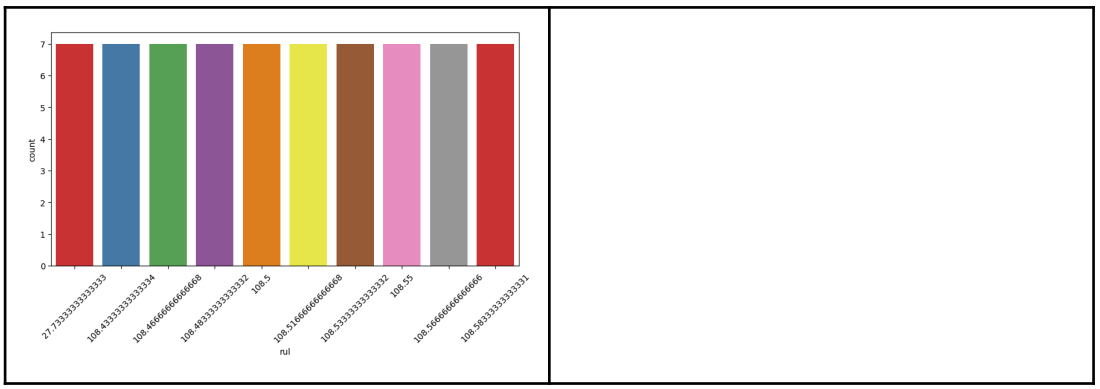


Fig.no.4.6 Count Vs Distribution

4.4 Correlation Matrices

Correlation Coefficient or Heat Map: Almost all the columns are related to the target variable.

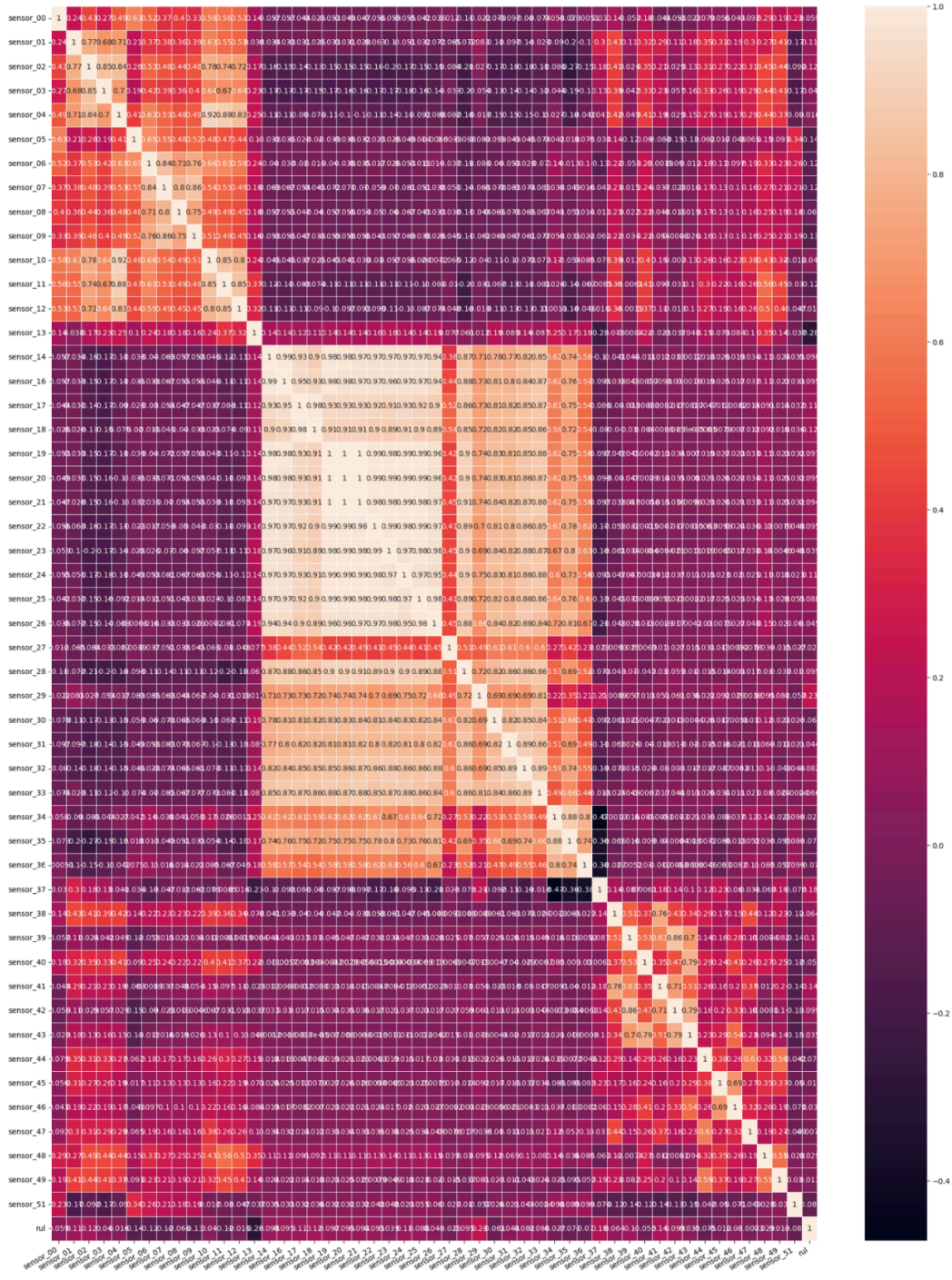


Fig No 4.7 : Correltion Matrix

Sum of all Null values in data

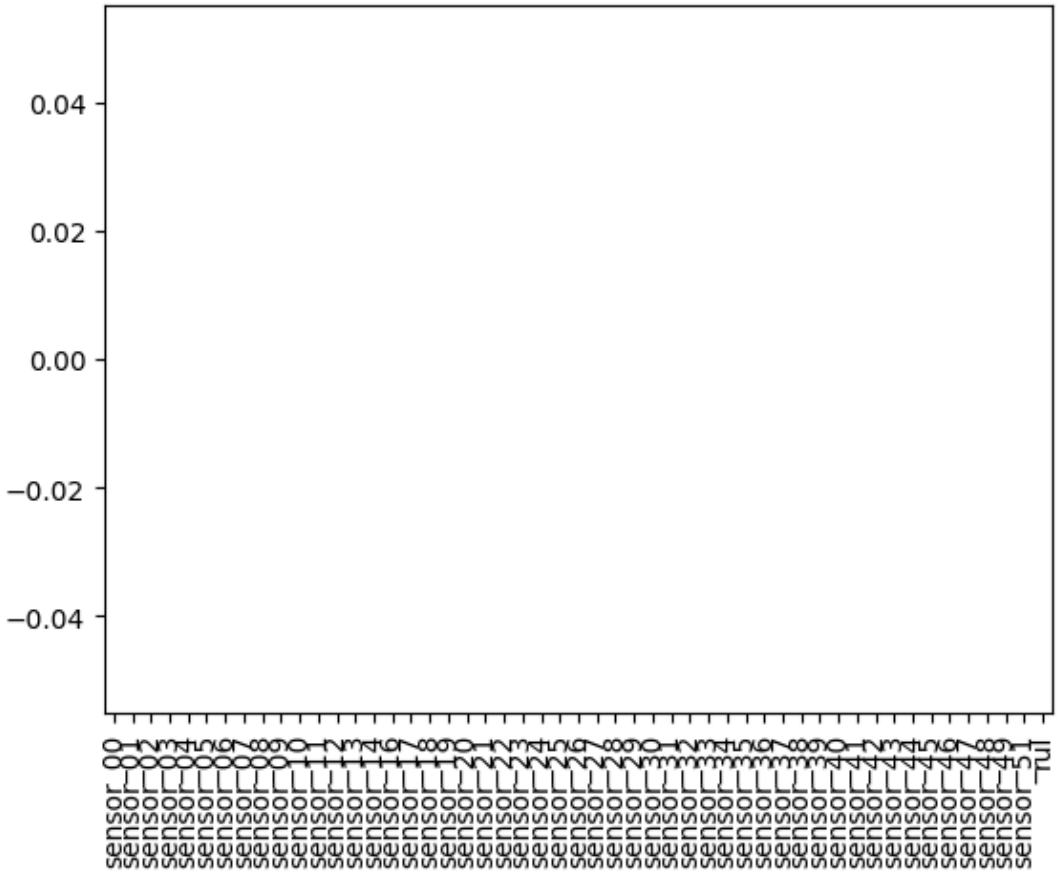


Fig.no 4.8 Sensors Vs Nullcount

In this data set we found zero null values but there are two columns that are of no use for predicting RUL, They are “Unnamed: 0” and “timestamp” columns. We are eliminating these values columns.

4.5 Spiting, Training and Testing

Next step is to prepare the data and split the available data into Train and test sets. Universally 30 % of data is used to test the model with the trained data set. The following code splits that data. In the below code we are randomly picking only 20,000 rows of data as we don't have capacity to run all the data in the data set.

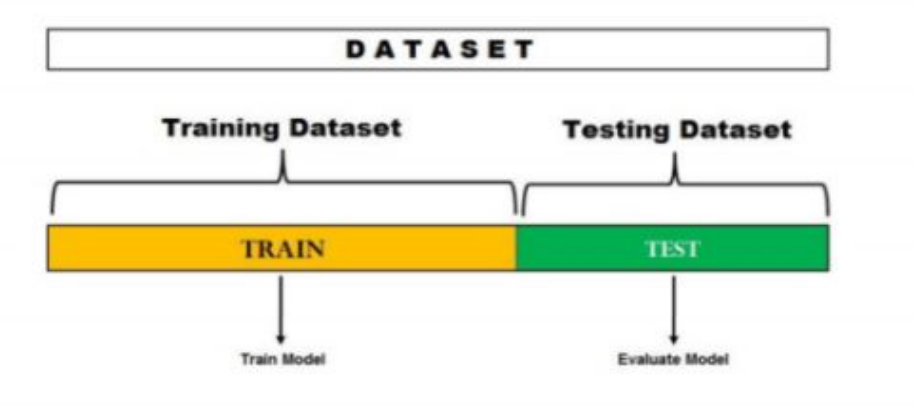


Fig.no 4.9: Data Splitting

```
dfSample= df.sample(n=20000)
X = dfSample.drop(['rul'], axis=1)
y = dfSample[['rul']]


from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

4.6 Building the Model

The data was run through 10 different algorithm and the accuracy and time taken run the algorithm are mentioned.

Model	Accuracy	Time Taken
ExtraTreesRegressor	0.96	2.5
LGBMRegressor	0.94	0.53
HistGradientBoostingRegressor	0.94	3.99
XGBRegressor	0.92	1.28
RandomForestRegressor	0.92	5.39
BaggingRegressor	0.89	0.6
GradientBoostingRegressor	0.86	2.53
DecisionTreeRegressor	0.77	0.15
KNeighborsRegressor	0.76	0.08
ExtraTreeRegressor	0.75	0.11

Fig.no.4.10 Algorithm and their Accuracy

From the above data it is clear the accuracy were more in ExtraTreesRegressor and the LGBMRegressor

ExtraTreesRegressor

Below is the code that runs the model and predicts the correctness of the model.

```
from sklearn.ensemble import ExtraTreesRegressor
reg = ExtraTreesRegressor(n_estimators=100, random_state=0).fit(
   X_train, y_train)
reg.score(X_test, y_test)
```

```
0.9961978959139908
```

The ExtraTreesRegressor is a machine learning algorithm that belongs to the ensemble methods family, specifically the tree-based ensemble methods. It is a variant of the Random Forest algorithm and shares some similarities with it. The term "Extra Trees" stands

for "Extremely Randomized Trees." The ExtraTreesRegressor algorithm is primarily used for regression tasks, where the target variable is continuous. It can be applied to various types of data, including numerical and categorical features. It is known for its fast training speed and can be effective in handling high-dimensional data.

4.7 Feature Importance

Feature importance refers to the significance or contribution of individual features or variables in a predictive model or statistical analysis. It helps to understand which features have the most impact on the target variable or outcome.

Feature importance can be assessed using various techniques, depending on the type of model and the specific problem at hand. In our model 19 features contribute to the prediction. Usually any value above .25 is considered to be good but this value changes based on the algorithm and the data set used

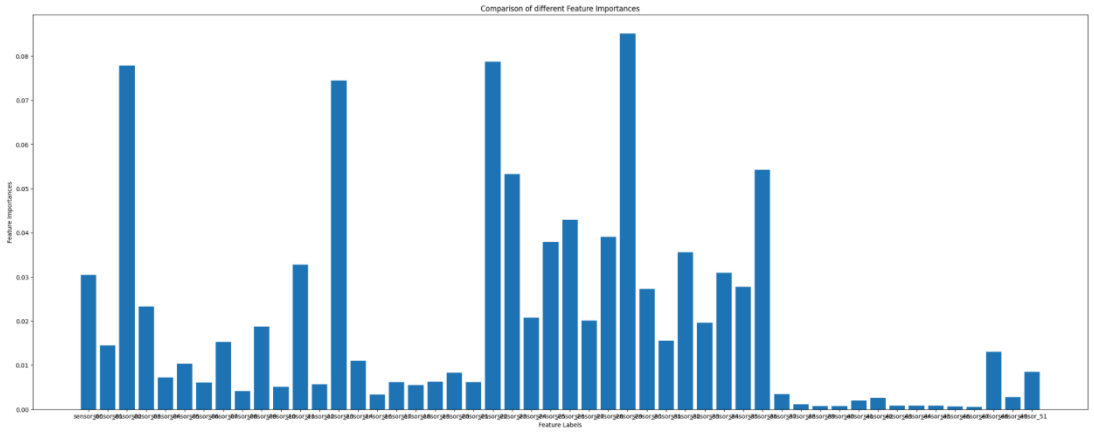


Fig.no 4.11: Feature Importance

4.7 Evaluatioans of ML models

The success of the model performance can be defined by different evaluation metrics such that comparing the predicted values with the actual values. The actual values are chosen from the test data which we have from the split data set as train and test. The commonly used accuracy metrics in evaluation towards regression modelling are;

- Min-Max Error (minmax)
- Mean Error (ME)
- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- Lag 1 Autocorrelation of Error (ACF1)
- Mean Absolute Percentage Error (MAPE)
- Mean Percentage Error (MPE)
- Correlation between the Actual and the Forecast (corr)
- R-squared
- Adjusted R -squared

There are other evaluation criteria in classification modeling are;

- Confusion matrix
- Classification accuracy
- Precision
- Recall
- AUC

However, we concentrate on two evaluation metrics in our thesis, which are:

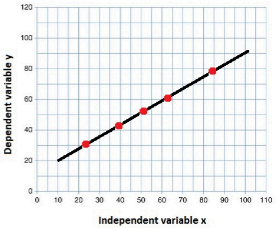
1. R-squared
2. Adjusted R -squared

R-squared

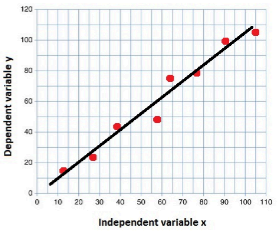
R-squared is a goodness-of-fit measure for linear regression models. This statistic indicates the percentage of the variance in the dependent variable that the independent variables explain collectively. R-squared measures the strength of the relationship between your model and the dependent variable on a scale of 0 to 1

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}},$$
$$= 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}.$$

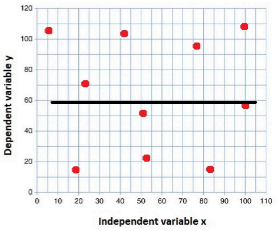
$R^2 = 1$ All the variation in the y values is accounted for by the x values



$R^2 = 0.83$ 83% of the variation in the y values is accounted for by the x values



$R^2 = 0$ None of the variation in the y values is accounted for by the x values



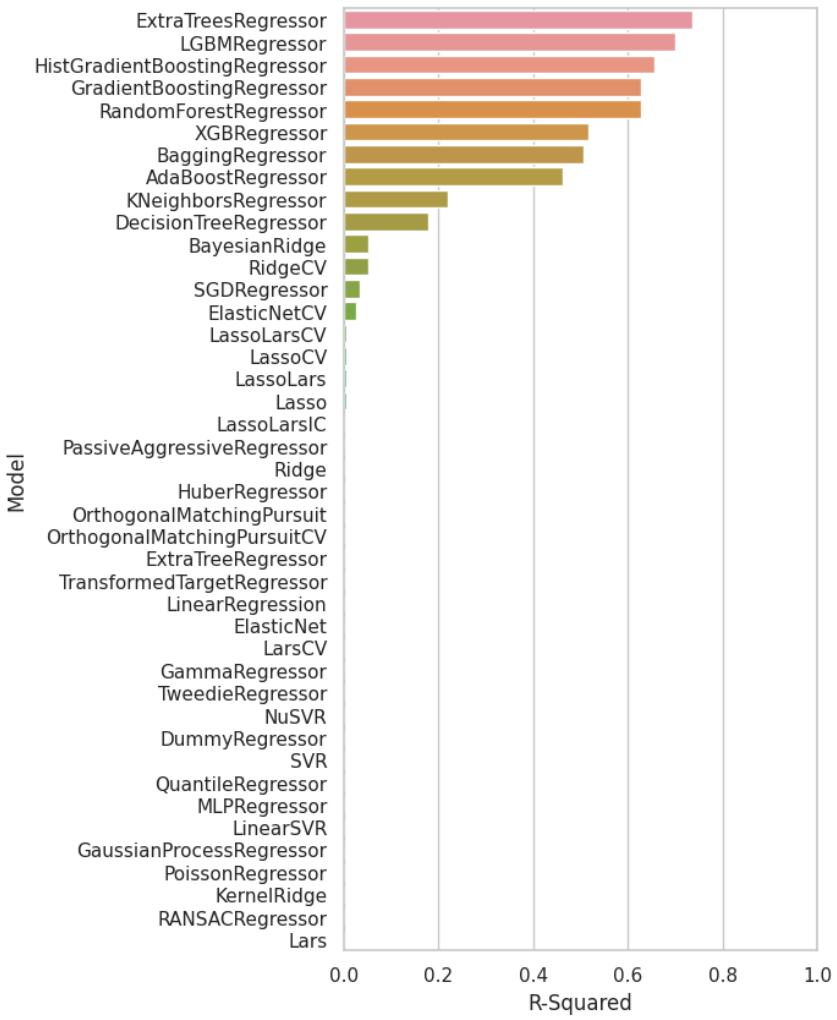


Fig.no: 4.12 R Squared

Adjusted R-squared

The Adjusted R-squared takes into account the number of independent variables used for predicting the target variable. In doing so, we can determine whether adding new variables to the model actually increases the model fit. Let's have a look at the formula for adjusted R-squared to better understand its working.

$$Adjusted\ R^2 = \{1 - [\frac{(1 - R^2)(n-1)}{(n-k-1)}]\}$$

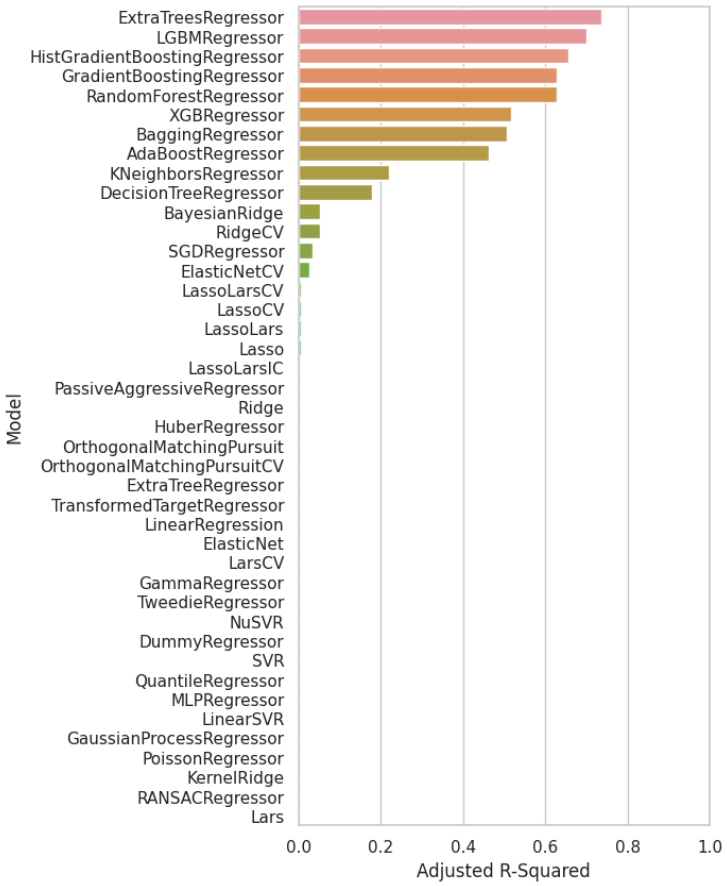


Fig No 4.13 Adjusted R Squared

Based on the R Squared values and Adjusted R Squared values we have estimated the values in the plots. Following Models seems to be performing better than other models

ExtraTreesRegressor

LGBMRegressor

Based on the R Squared values and Adjusted R Squared values the ExtraTreesRegressor is the winner model that we will be recommending to the company to use in production.

4.8 Results

The accuracy of the model can be represented using different evaluation metrics and those error values should be minimized. It is important to discuss the utility of reduction results to achieve the business goals which were defined in the beginning stage. Below is the graph that explains that more than 95% of times both test and train values overlap each other.

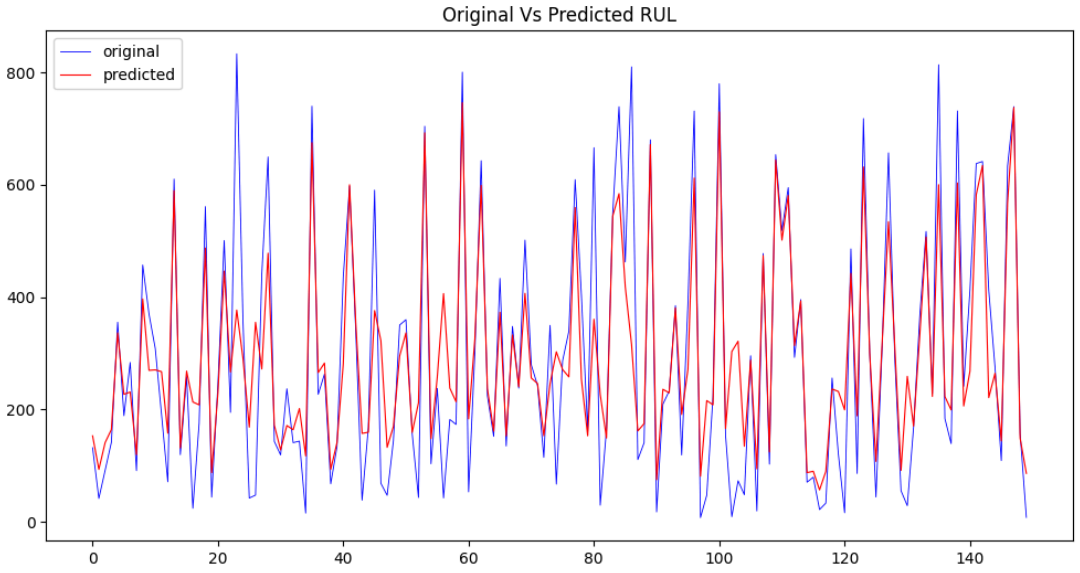


Fig No: 4.14 Original Vs Predicted RUL

5.CONCLUSION

5.1 Conclusion

- The project methodology utilized for this thesis was an enhanced CRISP-DM, as the data collection, data understanding, data preparation, modelling and evaluation was an iterative process due to non-ideal real time scenarios at industries.
- Several data visualizations were preformed and many insights were obtained about the trends and patterns inside the data. These visualizations were quintessential for data pre-processing and formulating assumptions.
- The data preparation was carried out to merge relevant data from different data sources and to transform it in to suitable formats for ML. Since the auto recorded breakdown data was unavailable,the triggering of severe alarms were assumed as breakdowns.
- In this thesis project ExtraTreesRegressor model and LGBM model were used for prediction of multivariate time-series data from multiple sources. The best results was given by the ExtraTreesRegressor model.
- In conclusion the ExtraTreesRegressor model proposed was able to predict the RUL of the manufacturing plant based on

parameters like temperature, vibration and also the machine parameters like RPM.

- These parameters are recorded by the sensors of the tubing machine over a period of time, and hence maintenance can be scheduled as per the requirements.
- This model will prove to be extremely useful maintaining the productivity and minimizing the cost of maintenance.This will reduce the time for which the unit remains idle and maintenance.
- The model was very effective in predicting the RUL of the plant as a whole based on the previous historical sensor data that it received as this model is known for its fast training speed and can effectively handle high-dimensional data.

5.2 Recommendations

While this machine learning algorithm achieved success at predicting the RUL of the manufacturing plant as a whole, there is still a lot of room for improvement in its predictive accuracy.

Improvisation of data resources

In the data side of things ,having a lot of data is essential for project like this, however, more data is not always better data. Feeding a machine learning algorithm data entails a lot of time and work that must go into pre-processing and cleaning the data so that a machine learning algorithm can make proper sense of it.

6.REFERENCES

[1] Brad Cline, Radu Stefan Niculescu, Duane Huffman, and Bob Deckel. Predictive maintenance applications for machine learning. In 2017 Annual Reliability and Maintainability Symposium (RAMS), pages 1–7, 2017.

[2] Marcia Baptista, Shankar Sankararaman, Ivo P de Medeiros, Cairo Nascimento Jr, Helmut Prendinger, and Elsa MP Henriques. Forecasting fault events for predictive maintenance using data-driven techniques and arma modeling.Computers & Industrial Engineering, 115:41–53, 2018.

[3] Shashidhar Kaparthi and Daniel Bumblauskas. Designing predictive maintenance systems using decision tree-based machine learning techniques. International Journal of Quality & Reliability Management, 2020.

[4] Ebru Turanoglu Bekar, Per Nyqvist, and Anders Skoogh. An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study. Advances in Mechanical Engineering,12(5):1687814020919207, 2020

[5] Olga Fink. Data-driven intelligent predictive maintenance of industrial assets.In Women in Industrial and Systems Engineering, pages 589–605. Springer,2020.

6 Predictive analytics: Transforming data into future insights CIO, article John Edwards

7 Predictive Modelling Analytics through Data Mining", International Research Journal of Engineering & Technology (IRJET) Volume: 04 Issue: 09| Sep-2017, e-ISSN: 2395 -0056, P-ISSN: 2395-0072. Lakshay Swani, Pratika Tyagi

8 Predictive Analytics: A study of its Advantages and Applications , IARS - International Research Journal, Mitanshi Rustagi, Neha Goel

9 Predictive Analytics: A Review of Trends and Techniques International Journal of Computer Applications (0975 – 8887) Vaibhav Kumar, M. L. Garg

10 The Application of Predictive Analytics : Benefits, Challenges and how it can be improved, International Journal of Scientific and Research Publications, Volume 7, Issue 5, May 2017 Vfatimetou Zahra Mohamed Mahmoud.