Preventive Maintenance for Marine Engines

Olivier Manthey, Axel Lo Schiavo, Salma Fourati

HEC Lausanne

Machine Learning in Business Analytics

Prof. Marc-Olivier Boldi

2025

Table of Contents

Abstract

The maritime industry relies heavily on the operational integrity of marine engines, where unexpected failures can lead to costly downtime and safety risks. This project explores the application of supervised and unsupervised machine learning techniques to anticipate maintenance needs based on operational sensor data collected from marine engines. The analysis followed a three-step approach: Exploratory Data Analysis (EDA), was performed to understand the key variables across maintenance classes, Unsupervised learning using K-means clustering was applied to uncover behavioral patterns based on numerical engine performance features and Supervised classification models, including Multinomial Logistic Regression, Random Forest and Support Vector Machines were trained to predict engine maintenance status. K-means clustering showed distinct operational profiles driven by features such as fuel consumption, engine load, and running period. However, these clusters did not align with categorical variables like engine type or failure mode. In the supervised phase, all models achieved low overall accuracy (33–35%), with the Requires Maintenance class being the most consistently predicted. The remaining classes, Normal and Critical, were often misclassified. The findings showed that machine learning can capture certain maintenance-related patterns, but the synthetic nature of the dataset and the exclusion of the temporal and categorical variables limited model performance. Future research should focus on time-series modeling and real-world data. Overall, this project highlights the challenges of predictive maintenance in data-limited environments, while underscoring the value of operational features in uncovering engine behavior patterns. Future work should consider temporal modeling and real-world data for improved accuracy.

1 Introduction

1.1 Overview and Motivation

The marine industry plays a crucial role in global logistics, with over 80% of international trade transported by sea. [7] At the heart of these ships are marine engines, complex mechanical systems responsible for propulsion and power generation. Given the demanding operating environments they face, marine engines are prone to wear and failure over time. Unexpected failures can lead to serious safety risks, increased operating costs and delays in supply chains. Traditional maintenance strategies in this field rely heavily on corrective (post-failure) and preventive (scheduled) maintenance. This creates an opportunity for predictive maintenance, which uses real-time and historical data to predict failures before they occur.

Machine learning enables such predictive strategies by learning patterns from sensor data to identify early warning signs of failure. With the rise of on-board data acquisition systems, modern marine engines now generate a wealth of operational data, such as temperature, pressure, fuel consumption, vibration and fault codes, which can be used to build intelligent diagnostic and prognostic systems.

This project explores the potential of machine learning to detect early warning signs of marine engine maintenance needs by analyzing operational data, combining both unsupervised learning (to explore hidden structures in the data) and supervised learning (to classify engine states or predict failures). The motivation is twofold: to reduce unplanned engine downtime, and to improve data-driven decision-making in marine asset management.

1.2 Data

The dataset used in this study comes from Kaggle owned by Fijabi J. Adekunle, entitled “Preventive Maintenance for Marine Engines”. It gathers weekly time series of simulated data collected during the operation of 50 marine engines totalizing 5200 instances, including performance measurements such as engine temperature or pressure, as well as operational conditions such as ambient temperature or load. The database also provides labeled failure modes, with each line corresponding to an observation of engine condition x over the weeks from 2023 to 2024. The dataset is available for access on Kaggle [3]. It is provided in CSV format and will be used for the aimed analysis of modeling maintenance status for marine engines [Table 1B].

1.3 Related Work

Predictive maintenance has become an important approach in industry to reduce unexpected equipment failures and improve efficiency. Unlike traditional maintenance methods which either wait for failures to occur or schedule maintenance at fixed intervals predictive maintenance uses data from sensors to monitor the condition of machines in real time. With the help of technologies like the internet of things and cyber-physical systems, it is now possible to detect early signs of wear or malfunction and predict when maintenance should be performed.

This allows companies to act before a breakdown happens, reducing downtime and extending the life of equipment [5].

In this context, machine learning (ML) plays a key role. Machine learning techniques are well suited to predictive maintenance because they can analyze large volumes of complex data, identify patterns, and make accurate predictions. As a result, in recent years, there has been increasing research into the application of machine learning techniques to predictive maintenance in a variety of sectors, including automotive, aerospace and marine. Each domain brings unique operational conditions and failure modes, but many of the underlying modeling approaches are transferable.

For instance, in the automotive sector, Tessaro et al (2020) [8] demonstrated the efficacy of supervised ML models such as Random Forests, Support Vector Machines (SVMs), and Artificial Neural Networks (ANNs) for forecasting maintenance requirements in engine components. Their research emphasizes the trade-off between predictive accuracy and model interpretability, which is critical for informed decision-making in maintenance operations. In the aerospace sector, Adryan and Wijaya (2022) [2] and Adryan and Sastra (2021) [1] explored predictive maintenance approaches for aircraft engines using ML models. These studies applied techniques such as Naïve Bayes, decision trees and K-Nearest Neighbors (KNN), highlighting how different algorithms respond to time-series sensor data and operational parameters. Their findings underline the role of data pre-processing and feature engineering in improving model performance, both of which are at the heart of the first stages of this project focused on the marine engines project.

Zhu et al (2024) [10] offer a broader perspective, categorizing predictive maintenance approaches into supervised learning, unsupervised clustering (e.g., K-Means), and dimensionality reduction techniques (e.g., PCA). Their review advocates for hybrid learning frameworks that leverage unsupervised methods to uncover hidden patterns and anomalies in exploratory data analysis, which can then be used to enhance the performance of supervised models trained to predict failure data. This combined approach addresses key predictive maintenance challenges such as limited data, heterogeneous sensor inputs, and dynamic operational conditions. For instance, clustering algorithms can detect operational modes or isolate noisy data, improving the reliability and accuracy of subsequent supervised learning models. In brief, their analysis bridges the gap between supervised learning, unsupervised clustering and dimensionality reduction methods, which is directly align to this project's dual focus on exploratory (unsupervised) and predictive (supervised) analysis.

The maritime sector exemplifies the practical benefits of such hybrid methodologies. Indeed, Rehman et al (2023) [6] focused specifically on the marine context, proposing a deep learning-based fault detection framework for marine diesel engines. Their use of convolutional neural networks (CNNs) for fault classification highlighted the potential of deep architectures for modelling complex nonlinear relationships in sensor data. However, their approach also requires large datasets and significant computational resources. As a more practical alternative, lightweight machine learning models such as Random Forests and Support Vector Machine are an option to adopt due to their lower computational requirements and robustness on smaller datasets. The integration of unsupervised techniques for anomaly detection with supervised models for fault classification provides a balanced, resource-efficient, and interpretable solution

for real-time predictive maintenance at sea. Therefore, it prompts us to explore those alternatives on a smaller and more suited dataset.

In summary, the reviewed literature establishes a robust foundation for applying machine learning techniques in predictive maintenance. Drawing from these insights, this project combines unsupervised learning (to explore patterns in sensor behavior) with supervised modeling (to predict failures and operational anomalies), targeting the specific dynamics of marine engine systems.

1.4 Research Question

Our study, "Preventive Maintenance for Marine Engines", focuses on building a predictive model to detect early signs of technical failure in marine engines, aiming to support timely maintenance and prevent accidents or delays in the supply chain. Our approach begins with exploring the data to understand the structure of the data and to identify potentially relevant features. Subsequently, unsupervised learning techniques are used to uncover patterns and groupings in the sensor and operational data, aiding in feature selection and anomaly detection. Finally, supervised machine learning models are developed to predict maintenance needs. A multinomial logistic regression model is serving as the baseline, and its performance is then compared against more complex algorithms such as Random Forest and Support Vector Machines (SVM). The central research question guiding this project is the following:

How can machine learning models predict early signs of maintenance needs in marine engines using engine performance metrics, operational conditions, and failure modes data?

1.5 Assumptions

In this study, we do not account for the temporal component (*timestamp*) or engine-specific identifiers (*engine_id*) in the analysis. The objective is to focus solely on the relationship between engine performance features and maintenance status, treating each observation as an independent instance. This simplification enables a clearer evaluation of how operational variables influence maintenance needs, without introducing the complexity of time-series or individualized modeling. While this limits the ability to capture degradation trends or engine-specific behaviors, it aligns with the exploratory and feature-driven nature of the current work.

2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a critical step in understanding the structure, patterns, and relationships within a dataset. By combining statistical summaries and visualizations, EDA reveals trends, anomalies, and key insights that inform subsequent modeling efforts. In this section, we analyze the dataset related to the maintenance status of marine engines, aiming to uncover the specific characteristics and behaviors of relevant features. The analysis will focus on how various features influence maintenance outcomes, providing a solid foundation for predictive modeling.

2.1 Data Distribution

In this section, we will examine the data distribution of the features relative to each maintenance status. This analysis aims to elucidate the interactions between the various features and the predicted features. Specifically, for the purposes of this study, the predicted variable is the maintenance status (*maintenance_status*). The objective is to identify which features demonstrate significant behavior in relation to the predicted maintenance categories, thereby enhancing our understanding of the dataset.

2.1.1 Distribution of Predictive Variables

The dataset comprises marine engines categorized into three distinct maintenance status groups: Normal, Critical, and Requires Maintenance. The distribution of engine statuses, detailed numerically in Appendix A [Figure 1A], reveals a relatively balanced representation across these categories. A marginally higher number of engines falls into the Requires Maintenance category, closely followed by those categorized as Critical and Normal.

This balanced distribution is beneficial for subsequent predictive modeling efforts, as it reduces concerns related to class imbalance. The notable proportion of engines classified under Critical and Requires Maintenance categories highlights the operational significance of timely maintenance prediction. Moreover, the substantial representation across all categories facilitates the development of robust predictive models, thereby potentially increasing the external validity and generalizability of the analysis outcomes.

2.1.2 Distribution of Numerical Variables

Following the initial assessment of maintenance status distribution, further analysis explored the relationships between key numerical variables and engine maintenance conditions. Visual details are provided in Appendix A.

Engine temperature [Figure 2A] showed a near-normal distribution centered around 85°C, with higher values in the Requires Maintenance category, suggesting temperature as a meaningful indicator of mechanical stress. Similarly, vibration levels [Figure 3A] and RPM

[Figure 4A] were slightly higher in Critical and Requires Maintenance cases, suggesting possible predictive potential. Fuel consumption [Figure 3A] was notably right-skewed, especially in Requires Maintenance engines, pointing to inefficiency or stress for increase in fuel usage. Coolant temperature [Figure 5A] also stood out, with a higher, slightly bimodal distribution in the same category. In contrast, oil pressure [Figure 2A], engine load [Figure 4A], exhaust temperature [Figure 5A], fuel consumption per hour [Figure 6A], and running period [Figure 6A] showed minimal variation across maintenance statuses, limiting their individual predictive value.

In summary, engine temperature, vibration, RPM, and coolant temperature are the most informative variables. Others like exhaust temperature and fuel consumption per hour may be excluded from modeling due to limited relevance. However, the data seems to be distributed nearly everywhere evenly meaning that clear distinctions between maintenance status can be difficult to determine.

2.1.3 Distribution of Categorical Variables

The distribution of key categorical features in relation to engine maintenance status was also explored, as illustrated in the bar charts provided in Appendix A [Figure 7A]. The variables analyzed included failure mode, engine type, fuel type, and manufacturer, each demonstrating varying degrees of relevance to maintenance status classification.

Failure modes were fairly evenly distributed, with Oil Leakage being the most common, followed by overheating and mechanical wear. No failure was the least reported. Although their frequency varied, the distribution of maintenance statuses within each failure type remained quite similar, making it difficult to use failure mode as a reliable predictor. Most engines were either 4-stroke High-Speed or 2-stroke Medium-Speed, but the maintenance status appeared consistent across both types. The same applies to fuel type, with around 60% of engines using Diesel and 40% using Heavy Fuel Oil. There was no clear connection between fuel type and maintenance condition. In terms of manufacturers, MAN B&W and Yanmar were the most represented, but there were no strong patterns in maintenance status linked to brand.

Overall, understanding categorical variable distributions supports effective feature selection and model development. Nevertheless, even if some categories were more frequent than others, the distribution of maintenance statuses in each class provide limited value information for the upcoming modeling assessment.

2.2 Correlation Between Variables of Interest

The correlation analysis [Figure 8A] highlights a strong internal consistency among core operational variables. Notably, engine temperature exhibits a high degree of correlation with both coolant temperature ($r = 0.92$) and engine load ($r = 0.72$), suggesting that these features may jointly reflect the thermal stress and workload experienced by the engine. Oil pressure also demonstrates a strong positive correlation with engine load ($r = 0.83$), aligning with expected mechanical responses under increased operational demand. Fuel consumption shows a substantial correlation with the running period ($r = 0.75$), indicating a close relationship that,

while anticipated, may introduce redundancy in modeling. Vibration levels present moderate correlations with engine load and oil pressure, implying their partial role in capturing physical stress, albeit less directly. Conversely, RPM exhibits minimal correlation with other variables, and both fuel consumption per hour and exhaust temperature show limited variability and weak associations with the rest of the dataset. These patterns suggest that these variables may offer limited predictive value for subsequent modeling.

Overall, the heatmap offers a comprehensive overview of inter-variable relationships, contributing to a better understanding of underlying engine dynamics. Such insights are essential for refining feature selection and enhancing the predictive modeling of marine engine maintenance status.

2.3 Outliers Analysis

2.3.1 IQR Method Univariate

To detect outliers in numerical features, we used the Interquartile Range (IQR) method, a non-parametric approach well-suited for skewed sensor data in engine monitoring systems. We began by visualizing each variable using boxplots to understand data spread, central tendency, and variability, aiding in the detection of extreme values and asymmetries.

Boxplot analysis showed that fuel consumption [Figure 9A] had numerous high outliers, especially in engines needing maintenance, suggesting inefficiencies. RPM [Figure 9A] was symmetric but had lower-end outliers, indicating occasional abnormal behavior. Running period [Figure 10A] showed slightly more variability in the maintenance group, while engine load [Figure 10A] had increased variability in the Critical category but stable medians, limiting its standalone predictive value. Coolant temperature [Figure 10A] had tight, consistent distributions with minimal outliers, showing limited discriminative power. Engine temperature [Figure 10A], however, showed more outliers in maintenance-related categories, hinting at thermal irregularities. Oil pressure [Figure 11A] remained stable across statuses with minimal outliers, while vibration levels [Figure 11A] showed moderate variability and possible mechanical wear in maintenance cases. Both fuel consumption per hour [Figure 11A] and exhaust temperature [Figure 11A] had dense clusters of outliers, raising concerns about their reliability.

Using the IQR method, we quantified outliers across features. The highest counts were found in fuel consumption per hour (528), exhaust temperature (161), and running period (161). No outliers were found in oil pressure, coolant temperature, or engine load, supporting earlier visual insights and highlighting the need for caution when interpreting features with extreme values.

2.3.2 Multivariate Outliers Detection

While univariate analysis helps identify extreme values within individual features, it may overlook observations that are only anomalous when multiple variables are considered

jointly. In complex systems like marine engines, operational faults often result from unusual combinations of variables rather than isolated deviations.

To address this, we perform a multivariate outlier analysis to detect data points that diverge from the overall multivariate structure of the dataset. This approach enables the identification of hidden anomalies that may indicate early-stage failures or inconsistent sensor behavior, thus improving the reliability of subsequent modeling and clustering tasks. Using a multivariate approach, we identified 202 outliers among the 5,200 observations, representing approximately 3.9% of the dataset. These points exhibit unusual combinations of feature values that are not apparent in univariate analyses. Their detection is essential for ensuring the reliability of downstream modeling, as such anomalies may correspond to rare operational conditions, sensor noise, or early signs of failure. With this, the exploratory data analysis provides a comprehensive understanding of the dataset's structure, variability, and potential data quality issues, laying a solid foundation for the subsequent modeling phase.

3 Unsupervised Learning

3.1 K-mean Clustering

In this study, we employ K-means clustering as an unsupervised learning technique to investigate whether natural groupings exist within the engine data based solely on operational and performance features. The primary aim is to determine if patterns in variables such as RPM, fuel consumption, vibration, and load can form distinct clusters that reflect underlying maintenance conditions without using the maintenance status during training. This approach allows us to assess how combinations of engine features may inherently relate to different health states or usage profiles, offering insight into potential risk patterns and early indicators of maintenance needs.

3.1.1 Number of Clusters

To identify the most appropriate number of clusters for the K-means algorithm, we employed both the Elbow Method and the Silhouette Analysis. The Elbow Method evaluates the Total Within Sum of Squares (TWSS) and helps determine where adding additional clusters no longer significantly reduces intra-cluster variance. Although the TWSS plot [Figure 12A] shows a steady decline without a sharply defined elbow, a moderate inflection point appears around K = 3, suggesting a balance between complexity and within-cluster cohesion. Similarly, the silhouette score [Figure 13A] reaches its maximum at K = 2, but K = 3 remains competitive, offering slightly more structure while avoiding excessive fragmentation.

Together, these two metrics support the choice of three clusters as a statistically robust solution, balancing within-cluster compactness and between-cluster separation for our engine performance dataset.

3.1.2 Clusters & Principal Component Analysis (PCA)

To enhance the interpretability of the K-means clustering results, we applied Principal Component Analysis (PCA) as a dimensionality reduction technique. PCA allows us to project the data into a lower-dimensional space while preserving most of the variance. Based on the EDA, we selected variables most likely to discriminate between operational states and maintenance needs, namely: fuel consumption, RPM, engine temperature, vibration level, engine load, coolant temperature and running period. Variables such as fuel consumption per hour, oil pressure, and exhaust temperature were excluded due to low variability or weak discriminative value across maintenance statuses.

The PCA projection [Figure 14A] shows that the first two principal components explain approximately 66% of the total variance (PC1: 43.6%, PC2: 22.5%) [Table 2B]. The direction and length of the arrows indicate each variable's contribution to the components: for instance, fuel consumption and running period strongly influences PC2, while the other features

contribute more to PC1. The clusters show good separation in the PCA space, with Cluster 1 and Cluster 3 appearing as the most distinct groups. To assess the relationship between the groups obtained and our objective variable *maintenance_status*, we examined the distribution of maintenance labels across the three groups. The contingency table [Table 3B] reveals that each cluster contains a relatively balanced proportion of the three maintenance classes: Normal, Critical and Requires Maintenance and that no cluster is strongly dominated by a single category. For example, cluster 2 contains slightly more observations in all categories, but the overall distribution remains fairly even. This suggests that the clustering reflects underlying operational models that are not strictly related to the predefined maintenance labels. While K-means operate independently of any label information, the overlap between clusters and maintenance classes supports the interpretation that selected engine performance features capture general behavioral groupings rather than directly classifying maintenance states.

To better understand the characteristics that define each cluster, we visualized the distribution of key engine variables using boxplots. These visualizations help clarify which features vary most between clusters. The distribution of key engine features across the three clusters [Figure 15A] reveals distinct operational profiles. Cluster 3 is characterized by the highest values of fuel consumption, engine load and running period, indicating a group of engines operating under sustained high performance or load conditions. Cluster 1, in contrast, exhibits lower running time and fuel consumption, and high coolant and engine temperatures, suggesting engines that may be active but used in shorter, possibly high-intensity cycles. Cluster 2 shows lower engine load, coolant temperature profiles, and a low fuel consumption range, potentially representing lighter-duty or intermittently used engines. Vibration levels and RPM are relatively similar across clusters but show slight variations. Cluster 1 tends to have slightly higher vibration level, while RPM remains fairly consistent, indicating that rotational speed may not be a strong differentiator between groups.

3.1.3 Cluster & Categorical Variables Analysis

Beyond numerical features, we examined whether categorical attributes, namely: engine type, fuel type, manufacturer and failure mode are associated with the cluster structure produced by the K-means algorithm. The objective was to assess whether certain categories of engines, fuels, or brands tend to exhibit similar operational behavior, potentially revealing embedded patterns related to maintenance or design characteristics.

To evaluate the relationship between categorical variables and cluster assignments, we applied Pearson's Chi-squared test of independence. This statistical test determines whether the distribution of categories differs significantly across clusters, with low p-values indicating dependence (i.e., a likely association). As shown in the output, none of the tests yielded statistically significant results:

- *engine_type vs. cluster*: $\chi^2 = 3.87$, df = 6, p = 0.6947
- *fuel_type vs. cluster*: $\chi^2 = 0.59$, df = 2, p = 0.7434
- *manufacturer vs. cluster*: $\chi^2 = 5.52$, df = 10, p = 0.8539
- *failure_mode vs. cluster:* $\chi^2 = 7.81$, df = 6, p = 0.253

These high p-values suggest that there is no strong statistical association between the categorical variables and the cluster structure. In other words, the clustering seems to be driven more by operational variables (e.g., fuel consumption, temperature, vibration) than by categorical identifiers such as engine model or fuel type. Visualizations [Figure 16A] further confirm this result. The barplots display a relatively uniform distribution of engine types, fuel types, manufacturers and failure mode across clusters, with no obvious dominance or segregation of specific categories in any one group.

In summary, the unsupervised clustering revealed meaningful groupings based on operational characteristics rather than categorical distinctions. These findings reinforce the importance of sensor-driven metrics in capturing the true behavioral patterns of marine engines and highlight that similar maintenance profiles may occur across different equipment types and manufacturers.

4 Supervised Learning

Teke and Depci (2023) [7], applied three machine learning models to predict maintenance needs in maritime logistics, the models where: Random Forest, K Nearest Neighbors, and Support Vector Machine. These results show that all three models provided strong predictive outcomes, with Random Forest performing the best in terms of overall accuracy and reliability. Moreover, but used in another field, Tessaro, Mariani, and Coelho (2020) [8], applied random forest and support vector machine learning method in predicting maintenance in automative engine components. Those techniques effectively predicted faults in automotive engine components using simulated data, which is very close to our case. Lastly, Adryan (2022) [2] concluded through a study on predictive maintenance that Random Forest and Support Vector Machine were very effective in this field. The results that were under covered in those studies have driven us to use the same methods, respectively Random Forest (RF) and Support Vector Machine (SVM) to predict the maintenance status of a marine engine, additionally we constructed a Logistic Regression Model as our baseline model.

4.1 Data Processing

4.1.1 Removed Unwanted Columns

Before building the supervised learning models, we made the decision to focus exclusively on the seven numerical variables identified during our exploratory data analysis and cluster analysis. These variables including fuel consumption, engine temperature, running period, engine load, RPM, coolant temperature and vibration level demonstrated clear variance across maintenance classes and were most informative in distinguishing between operational states. In contrast, the categorical variables such as engine type, fuel type, failure mode and manufacturer showed no significant association with the cluster structure and displayed nearly uniform distributions across groups, as confirmed by chi-squared tests. Including them would likely introduce noise rather than predictive value. Therefore, to ensure model interpretability and to reduce dimensionality and overfitting risk, we retained only the numerical features for all supervised models.

4.1.2 Data Splitting

To guarantee the robustness and credibility of our models, we divided the dataset into two distinct subsets: one for training and one for testing. This separation enables us to fit the models using the training data while assessing their predictive performance on data they have never encountered. By doing so, we can better estimate how well the model generalizes real-world, unseen data rather than simply memorizing the training patterns. We opted for an 80/20 split, which provides a solid foundation for learning while reserving enough data to perform a reliable evaluation of the model's accuracy and avoid overfitting.

4.1.3 Class Imbalance

Before training our models, we examined the class distribution to check for potential imbalance issues. Class imbalance can negatively affect model performance by causing it to favor the majority class. However, in our case, the data was sufficiently balanced, and no corrective measures were needed. This verification step remains important, as it ensures that the model's predictions are not biased due to uneven class representation. The distributions of the proportion of each maintenance status from both the full dataset and training set [Table 4B] are consistent across both sets and indicate that no significant class imbalance is present. Consequently, no resampling or balancing techniques were required. This observation was already highlighted during the earlier exploratory data analysis (EDA) phase, confirming the representativeness and stability of the class distribution.

4.2 Models

4.2.1 Multinominal Logistic Regression (MLR)

To establish a baseline for supervised classification, we implemented a multinomial logistic regression model. This algorithm is an extension of binary logistic regression, designed to handle classification tasks involving more than two classes which makes it a natural fit for our target variable, *maintenance_status*, composed of three categories: Normal, Critical, and Requires Maintenance. Before training, the target variable was encoded as a factor, allowing the model to interpret the classes correctly. All models were trained using 10-fold cross-validation, which divides the training data into 10 subsets, trains the model on 9 of them, and validates it on the 10th. This process helps reduce variance in performance estimates and limits overfitting by ensuring the model generalizes across different segments of the data.

4.2.2 First Model – Standard MLR

The first model was trained without regularization. On the training set, it achieved an accuracy of 36.2%, and a relatively balanced performance across classes with balanced accuracy scores of approximately 52% for each class. However, sensitivity was notably low particularly for the Normal class (22.1%), indicating poor ability to identify actual normal conditions. On the test set, the model yielded a slightly lower accuracy of 34.1%, with similar sensitivity and specificity patterns. These results suggest that the model struggles especially with correctly identifying normal engine behavior, while specificity remains higher, indicating it is better at identifying what does not belong to each class. The model may favor the majority class Requires Maintenance during optimization, leading to underrepresentation of the Normal class in predictions. However, the small gap between training and test performance suggests no strong overfitting, but rather an overall limited predictive capacity of the base model.

4.2.2.1 Second Model – Penalized MLR (LASSO & RIDGE)

To address the limitations of the baseline model, we trained a second model using regularization, combining L1 (LASSO) and L2 (RIDGE) penalties. This approach introduces a constraint on the size of the model coefficients to prevent overfitting and to improve generalization by reducing complexity and potentially discarding less informative variables. The regularized model produced similar performance to the unpenalized one, with a training accuracy of 36.1% and a test accuracy of 34.5%. Balanced accuracy, sensitivity, and specificity remained largely consistent across both models, with a slight improvement in the detection of the Requires Maintenance class (test sensitivity: 50.1%). This indicates that the regularization did not significantly alter model behavior but may have contributed to a slightly more stable generalization on unseen data.

4.2.2.2 MLR Results Summary

Overall, multinomial logistic regression offers a simple and interpretable baseline but shows limited predictive performance in this multi-class setting. While penalization helped marginally improve class balance and generalization, the model still struggled to differentiate between maintenance statuses, especially for the Normal class. The feature importance graph [Figure 17A] highlights the most influential predictors in the different classes. For example, running period is a key variable in distinguishing the Normal class, although the other characteristics are indistinguishable from each other. Engine load and vibration level contribute more to the classification of Critical motors. On the other hand, coolant and engine temperature appear to be of greater importance in identifying the Requires Maintenance class. These trends are consistent with our exploratory results and confirm to some extent the model's alignment with operational behavior.

Despite these advantages in terms of interpretability, the limited sensitivity and accuracy of the model between classes warrants the exploration of more flexible and nonlinear models, such as Random Forest and Support Vector Machine in the subsequent sections.

4.2.3 Random Forest (RF)

Although multinomial logistic regression provided a useful baseline, its limited capacity to model non-linear relationships made it less effective in capturing the underlying structure of the data. To build on this initial model, and guided by findings in the literature, we proceeded to evaluate the Random Forest algorithm. Random Forest is an ensemble method that constructs multiple decision trees and aggregates their outputs to improve predictive accuracy and model stability. Its ability to handle complex feature interactions makes it a strong candidate for addressing the challenges of multi-class classification in our context.

4.2.3.1 Baseline RF Model

The baseline random forest model, used as a reference, yielded an overall accuracy of 35.2% on the test set. This performance is only marginally better than the no-information rate of 33.8%, suggesting limited predictive value. The evaluation of class-wise prediction reveals substantial performance variability. The model achieves a relatively high sensitivity of 77.8% for the Requires Maintenance category but performs poorly on the Normal and Critical classes, with sensitivities of 7.1% and 19.8%, respectively. These discrepancies highlight the model's inability to generalize effectively across all target categories. Balanced accuracy near 50% implies that, on average, the model is performing just slightly better than flipping a coin. This shows that the model struggles to differentiate between the classes in a meaningful and consistent way. The baseline model was performed using a random set of hyperparameters, the next part of the modelling will focus on tunning them.

4.2.3.2 Hyperparameters Tuning

To optimize the performance of the Random Forest model, we conducted a manual hyperparameter tuning process based on classification accuracy evaluated on a separate validation set using cross-validation to avoid information leakage which can cause overfitting. To optimize model performance, we systematically tuned four key hyperparameters, each influencing the model in distinct ways. We began with the number of trees (ntree), which enhances stability and accuracy, though gains tend to plateau beyond a certain point. Next, we adjusted mtry, the number of features considered at each split, to promote tree diversity and reduce correlation among them. We then fine-tuned nodesize, the minimum number of observations required in a terminal node, smaller values allow the model to capture finer patterns but increase the risk of overfitting. Finally, we optimized maxnodes, which caps the number of terminal nodes and helps regulate tree complexity.

We jointly evaluated the number of trees (ntree) and the number of variables considered at each split (mtry) while holding all other parameters constant. The grid search was conducted over five values for ntree (100, 300, 500, 800, 1000) and across the seven possible mtry values derived from the dataset. As shown in Appendix A [Figure 18A], classification accuracy does not increase monotonically with the number of trees. Accuracy peaked at 300 trees, with partial recovery at 1000 trees. The response to ntree varied across mtry values, with two combinations standing out: 300 trees with mtry = 2, and 500 trees with mtry = 3. To maintain model simplicity, the configuration of 300 trees and mtry = 2 was selected. This choice was further supported by an analysis of the Out-Of-Bag (OOB) error rate [Figure 19A], which confirmed that error began increasing beyond 300 trees.

The two last hyperparameters where also jointly measured with the two best other hyperparameters evaluated in the first tunning part. A grid search over selected values (nodesize $\in \{1, 3, 5, 10, 15, 20\}$ and maxnodes $\in \{5, 10, 20, 30, 50\}$) revealed that the highest accuracy was achieved with nodesize = [10] and maxnodes = [10]. A heatmap was used to visualize the interaction between these two parameters [Figure 20A]. This systematic tuning led to a fix of all hyperparameters which allowed us to find the apparent optimal tuned model.

4.2.3.3 Best Tuned Parameters Model

The final tuned random forest model achieved an apparent accuracy of 37.83% on the training set and 35.23% on the test set. Those results show a low overfitting behavior. Nevertheless, these results are effectively identical to those obtained from the untuned baseline model. Despite extensive hyperparameter tuning, the model exhibited similar sensitivity and specificity values across classes, indicating that it fails to generalize effectively to unseen data. This suggests that the model converges toward the same decision boundaries regardless of tuning, likely due to limitations in the data or feature space, and highlights a low potential for generalization.

4.2.3.4 Variable Importance

After evaluating the final tuned random forest model, we analyzed variable importance to understand which features drive predictions. Both Mean Decrease in Accuracy and Gini measures identified *engine_load* and *coolant_temp* as the most influential variables [Figure 21A]. The use of Partial dependence plots [Figure 22A] reveals that the probability of predicting Requires Maintenance remains high at moderate engine load levels but drops sharply beyond 70, indicating that lower engine loads are associated with higher maintenance risk. For coolant temperature, the probability increases steeply between 70°C and 80°C, suggesting that elevated temperatures are strong indicators of maintenance needs.

While overall model performance remains limited, these results highlight key predictors that contribute most to the model's decision-making and offer valuable insights for monitoring system health.

4.2.3.5 RF Results Summary

To sum up, the baseline random forest model reached a test accuracy of 35.2%, with low sensitivity for the Normal (7.1%) and Critical (19.8%) classes, where most predictions were concentrated in the Requires Maintenance class, reflecting poor class separation. To improve performance, key hyperparameters were tuned, but the final model produced the same test accuracy (35.23%) with low overfitting, showing no improvement in classification performance. This indicates that the model's capacity to generalize is limited by the data rather than the hyperparameter configuration. Furthermore, variable importance analysis revealed *engine_load* and *coolant_temp* as the top predictors and the partial dependence plots showed that the probability of predicting Requires Maintenance increases at low engine loads and coolant temperatures above 75°C, confirming their relevance to the target outcome. Nevertheless, the overall result of the random forest is a sign that it is likely that more complex model will not necessarily improve the accuracy of our current case.

4.2.4 Support Vector Machine (SVM)

To explore the performance of non-linear classification, we implemented support vector machines (SVMs), a supervised learning algorithm that searches for optimal hyperplanes to separate classes in a multi-dimensional space. Given the complexity and potential overlap between maintenance classes, SVMs are particularly suited to handling non-linear relationships in data when combined with a radial-based function (RBF) kernel. In order to configure the model correctly, we coded the *maintenance_status* variable as a factor and scaled all numerical features using normalization to ensure that the SVM model was not biased by differences in variable scaling.

4.2.4.1 Baseline SVM Model

As a reference, we trained a model using the RBF kernel with randomly chosen hyperparameters (cost = 1, gamma = 0.1). Over the training set, the model achieved an accuracy of 40.4% and a relatively balanced performance between classes (balanced accuracy ~ 55%). However, sensitivity for the normal class remained low (15.6%), and this trend continued in the test set, where accuracy fell to 35%, and sensitivity for the Normal class dropped further (13.2%). This indicates a poor ability to assign observations to Normal status, so in the context of the study, low values indicate that the model misclassifies Normal status rather than Critical (38.2%) or Requires Maintenance (53%) status. This tends to result in a high false-negative rate. Specificity values were consistently higher (88% for the Normal class), suggesting that the model excludes non-targeted classes more reliably than it correctly identifies them. However, the difference between sensitivity and specificity for the Requires Maintenance class was the best prediction (sens = 52.99%, spec = 50.87%).

4.2.4.2 Hyperparameters Tuning

These previous results highlight a limited generalization and a potential bias in favor of the Requires Maintenance classes, motivating the search for better SVM model by tuning the model. To improve the baseline, we performed 10-fold cross-validation on a grid of hyperparameters. We tested cost values {0.1, 1, 10, 100, 200, 300, 400, 500, 1000} and sigma values {0.001, 0.01, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 1}, where sigma is used with the "caret" package in R to automatically determine the corresponding gamma value based on the formula $\text{sigma} = 1 / (2 \times \text{gamma}^2)$.

The resulting cross-validation plot [Figure 23A] shows that Cost = 10 and Sigma = 0.01 gave the highest cross-validated accuracy, making them the best candidates for our next model. However, it would be interesting to determine whether there is a trade-off between the best model determined by the grid, the second best with close values (C = 1, Sigma = 0.001) and also to observe whether the accuracy of classifying observations into classes improves better if the overall accuracy is reduced a little but the sigma is increased, as in the model using the following values (C = 0.1, Sigma = 0.35). In each of these three cases, the cost varies but

remains rather low (0.1, 1, 10), which tends to widen the margin and leave room for greater tolerance of error in assessment. This choice of hyperparameters shows a tendency to follow a rather simple model, without too much complexity. This result seems counter-intuitive given the low proportion of well-classified observations. As far as the best choice of sigma parameter is concerned, there is again no great difference between the two main models (0.001 and 0.01) and so testing a third model with a higher sigma (0.35) could be interesting. Indeed, the higher sigma has more local influence and is stricter for the more complex observations that can be added. On the other hand, the low sigma tends to have a wider influence that may underfit the data, so comparing these three "best models" may be useful in our case.

4.2.4.3 Best Tuned Parameters Model

With the hyperparameters adjusted (C = 10 and Sigma = 0.01), the second model achieved an accuracy of 49.9% in training and 33.4% in testing. The sensitivity of Requires Maintenance improved to 59.2% during training versus 38.2% with the test set, which no longer represents the best predicted class. Indeed, Critical status achieves a slightly higher prediction with the test set (~40%). As for the Normal class, this is the highest improvement based on the basic model (22% vs. 13%). However, the specificity of each of these three predicted classes remains very high (Normal = 76%, Critical = 62%, Requires Maintenance = 61%) and therefore shows a large difference between the measurements. This can also be explained by the fact that we are in a multi-class model, so the prediction of true negatives is shared between the two remaining classes, which tends to increase the coefficient of specificity. Overall, this model, which the algorithm considers to be the best, is debatable, although it does vaguely equalize the classification between the different statuses. Thus, it motivates comparison with other models using different parameters.

4.2.4.4 Others SVM Model Comparison

As we saw earlier, we want to explore the tuning space further by testing the second-best grid parameter (C = 1, Sigma = 0.001). We want to see if a lower sigma, which can under-fit our data, can be more robust with our multi-class model.

This model obtained a lower overall learning accuracy than the previous model (42.4%), but this time the accuracy on the test set was higher than the previous model (34.8%). The prediction of true positives between classes is less balanced, with a further drop in the sensitivity of the Normal class (15.8%). The Critical class with the test set achieves a slightly lower sensitivity than the previous model (36% vs. 39%). However, the Requires Maintenance class gains significantly in predictive accuracy (52%), while reducing the difference between sensitivity and specificity (0.83% vs. 22.58% in the second model). This balance allows us to determine that the Requires Maintenance class is the best predicted of the three. The generally similar specificity values suggest that it remains more reliable for rejecting false positives than for detecting good positives. This means that the third model improves the overall accuracy of

the test set, especially for the Requires Maintenance class. Moreover, balanced accuracy remains in the low average (~ 51%) for each class.

We then seek to compare the models obtained so far with a model taking a higher sigma (C = 0.1, Sigma = 0.35) to see whether making the model slightly more complex by reducing the overall accuracy could lead to a better balance in classification. The overall accuracy of the test set is close to the previous one (34%), however the prediction obtained on the Normal class is even worse (7%) with a particularly high specificity (94%), which supports the idea that making the model more complex does not improve the prediction of this class in particular. The Critical (39%) and Requires Maintenance (54.7%) statuses are not improved by these parameters.

4.2.4.5 SVM Results Summary

The four SVM models performed more or less similarly on the test set, with accuracy stabilizing around 33-34%. Hyperparameter adjustment led to minor improvements in sensitivity and class balance, but did not substantially improve generalization and prediction accuracy. If we were to choose one model from the four tested, and despite the moderate accuracy, we would not follow the conclusion of the best-tuned model but would choose the second-best according to this analysis (C = 1, Sigma = 0.001). This is the one that best predicts the class between the three cases, while minimizing the gap between sensitivity and specificity. This choice is arbitrary and may be debatable given the low accuracy predicted by this implementation.

4.3 Results comparison

To evaluate and compare the performance of our three supervised learning models, Multinomial Logistic Regression (MLR), Random Forest (RF) and Support Vector Machine (SVM), we calculated several key measures on the test set, including precision, balanced precision and the macro-average F1 score [Table 5B]. The F1 score, which harmonizes precision and recall across all classes, was chosen as the main criterion because of its robustness in multi-class evaluation. Both the MLR and SVM models achieved a macro F1 score of 0.33, indicating a more balanced ability to identify all maintenance classes, despite relatively moderate overall precision. In contrast, the Random Forest model, although performing well on the training set, achieved a lower F1 macro score of 0.285, reflecting a decline in sensitivity and generalization on minority classes in the test set. These results highlight that, despite their lower complexity, the MLR and SVM models were more effective in maintaining class balance, particularly in distinguishing less frequent or more nuanced motor conditions.

Based on these results, the SVM model appears to offer the best compromise between generalizability, interpretability and balanced predictive performance, making it the most appropriate choice for early maintenance classification in this study.

5 Conclusion

5.1 Limitations

Throughout this project, we encountered several limitations that impacted the models' ability to produce strong predictive outcomes. First, the synthetic nature of the dataset introduces uncertainty about how the maintenance labels were generated. It's difficult to verify whether operational variables are truly informative of maintenance status or if the labeling is partially random or based on hidden, unobserved variables. This could explain why no model significantly outperformed a random baseline, and why hyperparameter tuning yielded only marginal improvements.

Another limitation relates to data simplification: we excluded temporal identifiers (*timestamp*, *engine_id*). This choice was made for exploratory clarity, while this was necessary given our exploratory focus, it also removed potentially valuable information on degradation patterns over time, which were key in studies like Adryan & Wijaya (2022) [2] where LSTM (a time-series model) outperformed traditional methods. Future work could explore hybrid approaches that reinstate categorical variables through feature engineering or embedding techniques. Lastly, despite model tuning, the overall predictive performance remained low, indicating that the selected features may not contain enough signal to effectively separate the three maintenance classes. This calls for a reconsideration of either the dataset quality or the modeling approach.

5.2 Discussion

The study initially aimed to explore whether operational variables derived from marine engines could predict the maintenance status of an engine, using machine learning models trained on selected numerical features. Our initial hypothesis was that performance metrics such as engine temperature, vibration level, RPM, and fuel consumption would hold predictive power in anticipating the need for maintenance. This assumption was grounded in both domain knowledge where these variables are typically associated with mechanical stress and prior literature emphasizing the potential of data-driven maintenance approaches.

As part of our exploratory data analysis (EDA), we tried to look at the distribution of each predictive variable across the three maintenance classes (Normal, Critical, Requires Maintenance). This separation allowed us to detect patterns specific to each status and revealed that the Requires Maintenance class displayed the most distinguishable behavior across several key indicators (engine load, coolant temperature and fuel consumption), while the other classes were far less separable. This raised doubts about whether the available features alone could meaningfully capture the underlying maintenance logic. In retrospect, we might have benefited from exploring more complex feature engineering techniques or integrating domain-driven thresholds to better distinguish between states.

In the unsupervised learning phase, we implemented K-means clustering using only numerical variables and analyzed the cluster structure in relation to categorical variables. Despite a thorough comparison, including chi-squared independence tests and visual inspections, we found no statistically significant association between cluster structure and any of the categorical variables, and that led us to drop all the categorical variables from supervised models to simplify modeling and focus on sensor-based numerical signals. However, this choice may have led to the omission of contextual signals (e.g some engine models being more prone to failure) that could have improved prediction accuracy. These results also revealed a deeper issue: that the dataset's structure may not reflect realistic degradation trajectories.

Finally, our supervised learning showed a more nuanced outcome. Despite the implementation of multiple supervised learning models, including Random Forest, Support Vector Machine, and Multinomial Logistic Regression, all models showed relatively low predictive performance, with overall, test accuracy ranging between 33% and 35%. Among all these models, the Support Vector Machine (SVM) achieved the best overall performance, with a test accuracy comparable to the others 33-34%. It achieved a moderate sensitivity of 52% for the Requires Maintenance class while also performing relatively better on the Normal and Critical classes compared to the other models. This suggests that the SVM model was more capable of capturing subtle distinctions across operational profiles, even within the limitations of the dataset. Nevertheless, its predictive power remains insufficient for reliable deployment, reinforcing the need for improved feature design, temporal modeling, and real-world data integration.

5.3 Conclusion

In summary, our study shows that machine learning can identify certain maintenance patterns, especially for engines already requiring maintenance, using operational and sensor-based data. While the results from both supervised and unsupervised models were modest, they offer meaningful insights into the challenges and potential predictive maintenance in the maritime context.

In response to our research question, we found that machine learning models failed in identifying the maintenance statuses. While the class labeled as Requires Maintenance showed more detectable patterns based on features such as fuel consumption, engine load and coolant temperature, this was not sufficient to validate the productive power of any model tested, the models struggled to differentiate reliably between engines in Normal and Critical states. This suggests that the current features, in the context of this synthetic dataset are not sufficient for robust early-stage maintenance prediction. Moreover, the exclusion of temporal and categorical variables likely limited the models' ability to have more complex degradation patterns.

Despite these limitations, the study confirms that data-driven approaches are valuable tools in uncovering behavioral profiles among marine engines. With more transparent, real-world datasets and the inclusion of time-based modeling, machine learning has the potential to significantly improve preventive maintenance strategies, reduce costs, and enhance safety in maritime operations

6 Appendix

Appendix A – Figures

Figure 1A: Distribution of the predicted variable (*maintenance_status*)

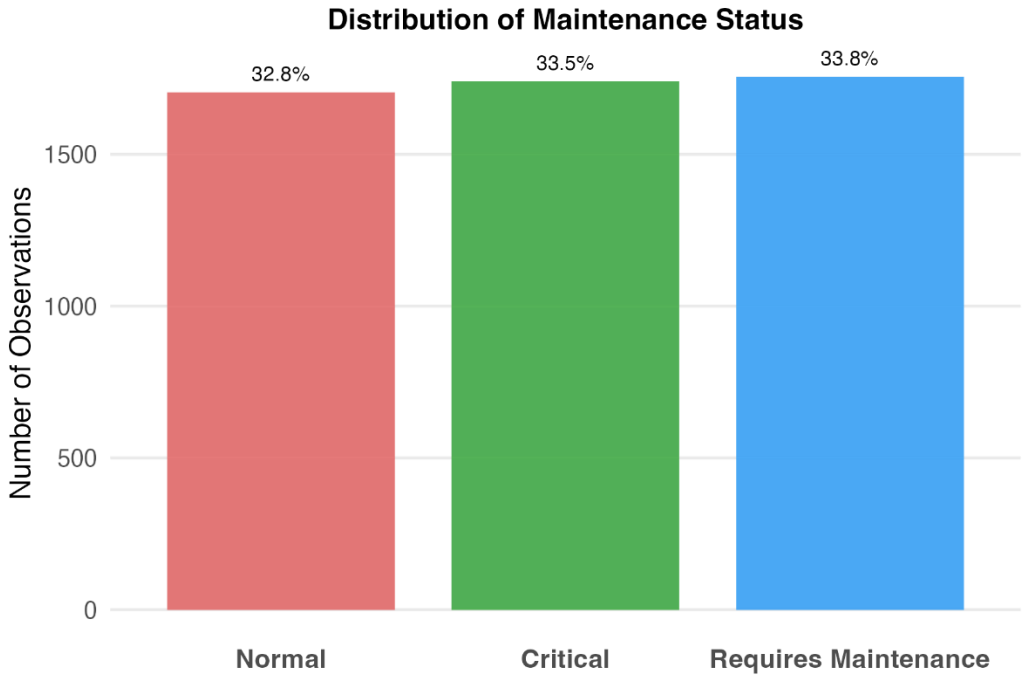

Figure 1A

Figure 2A: Distribution of *engine_temp* & *oil_pressure* by maintenance status

Figure 2A

Figure 3A: Distribution of *fuel_consumption* & *vibration_level* by maintenance status

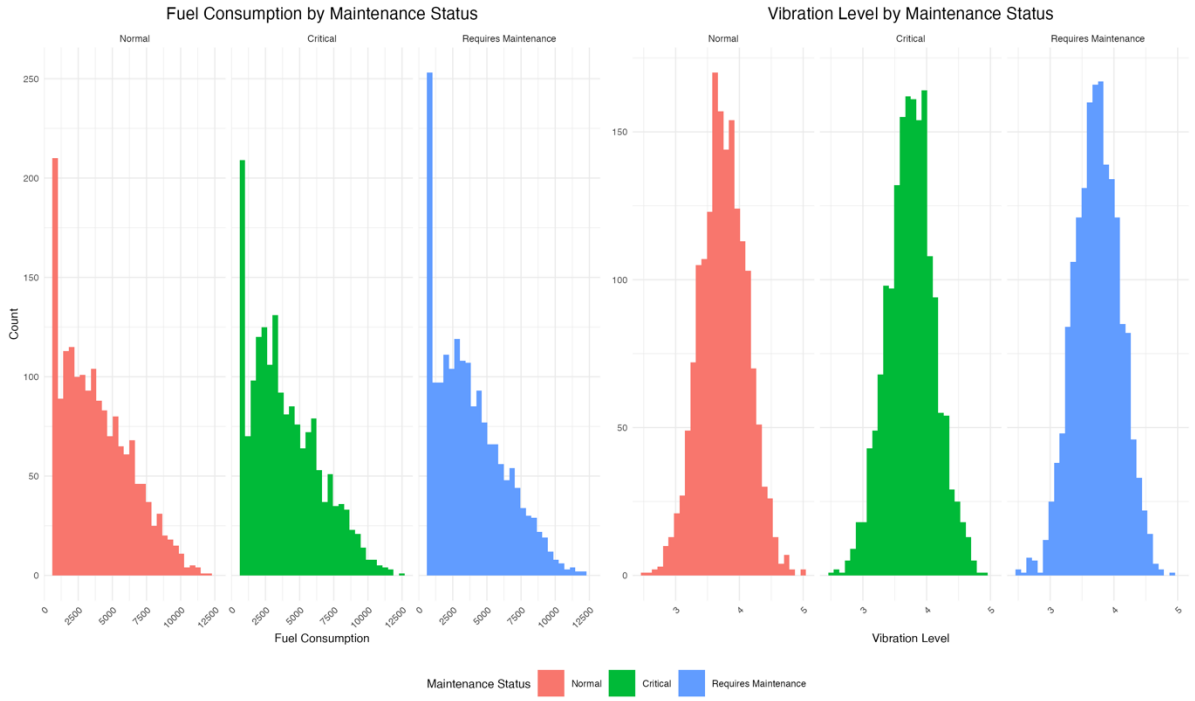

Figure 3A

Figure 4A: Distribution of *rpm* & *engine_load*, by maintenance status

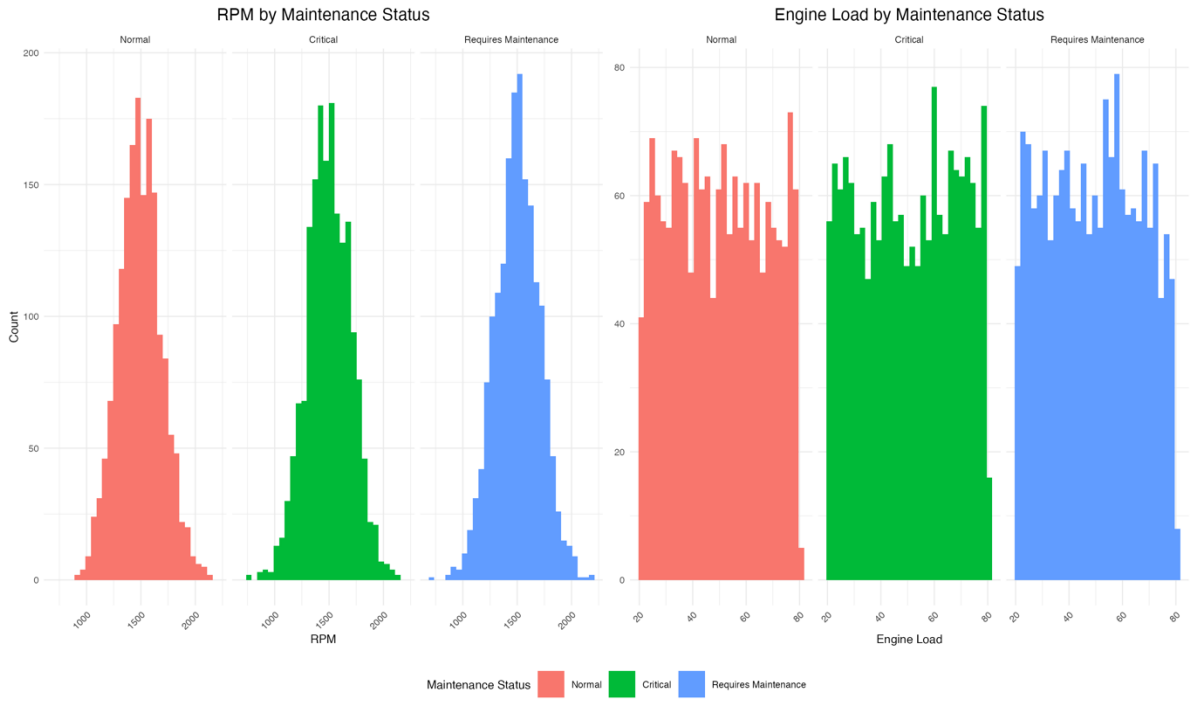

Figure 4A

Figure 5A: Distribution of *coolant_temp* & *exhaust_temp* by maintenance status

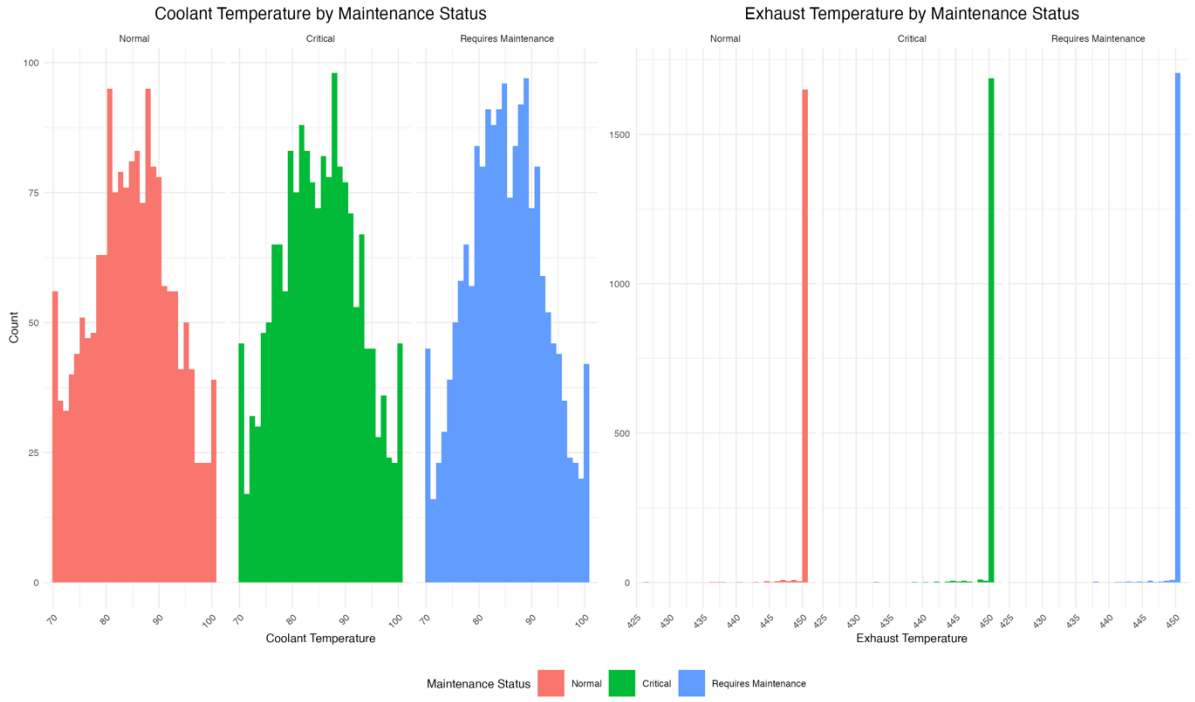

Figure 5A

Figure 6A: Distribution of *running_period* & *fuel_consumption_per_hour* by maintenance status

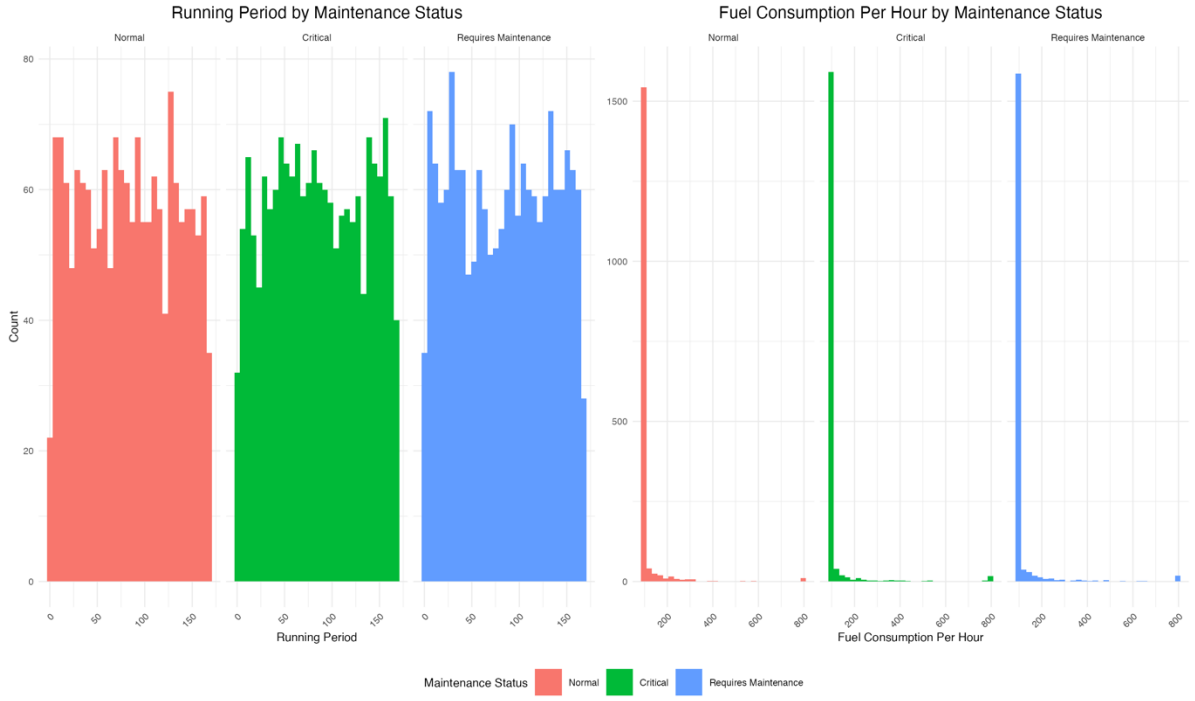

Figure 6A

Figure 7A: Distribution of each categorical predictive variables by maintenance status

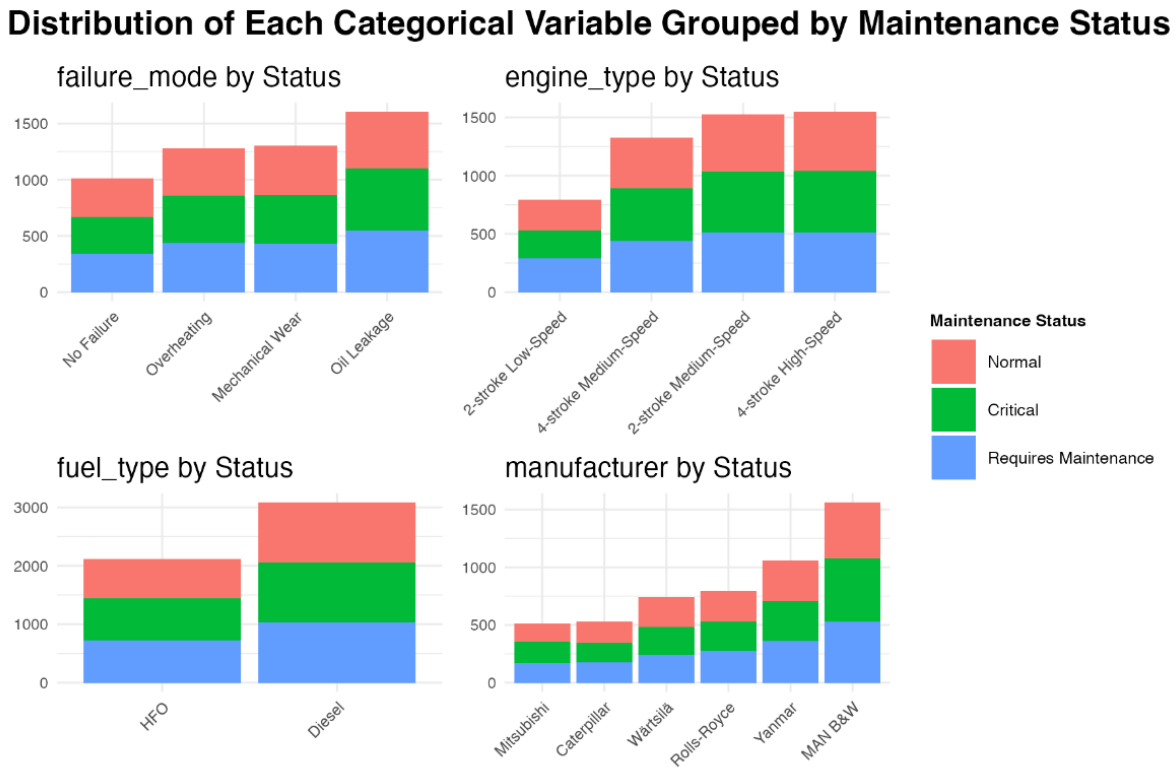

Figure 7A

Figure 8A: Correlation heatmap between numerical variables

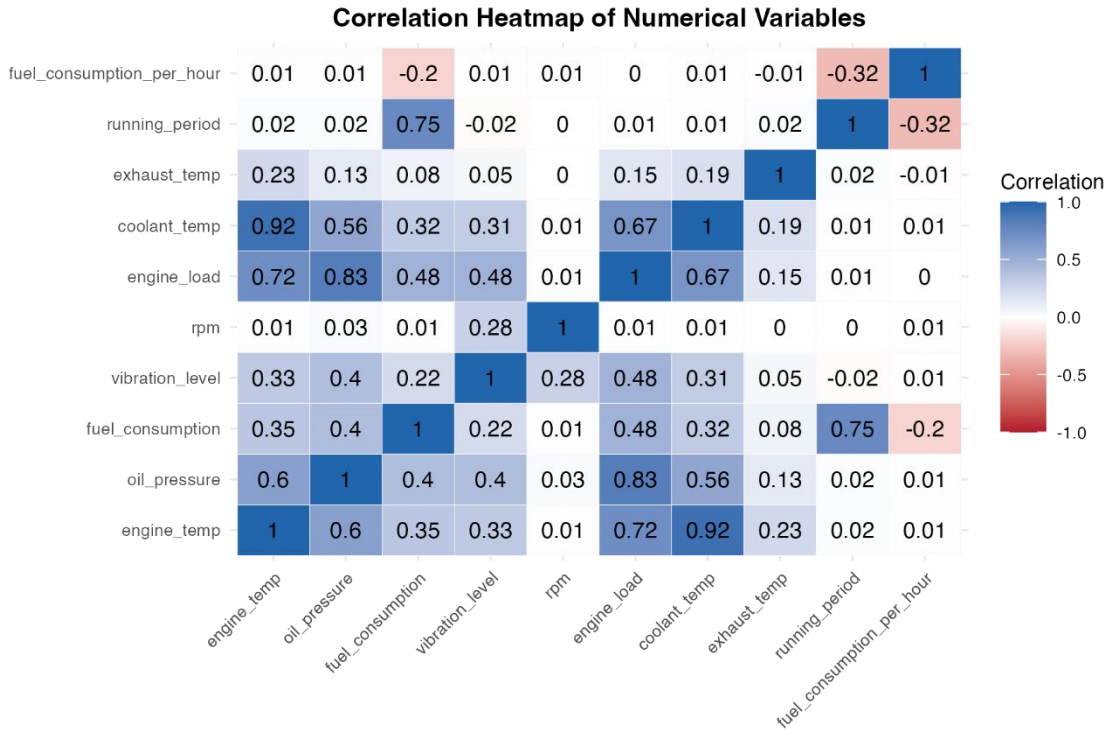

Figure 8A

Figure 9A: Boxplots illustrating the distribution of Fuel Consumption and RPM across the three maintenance statuses

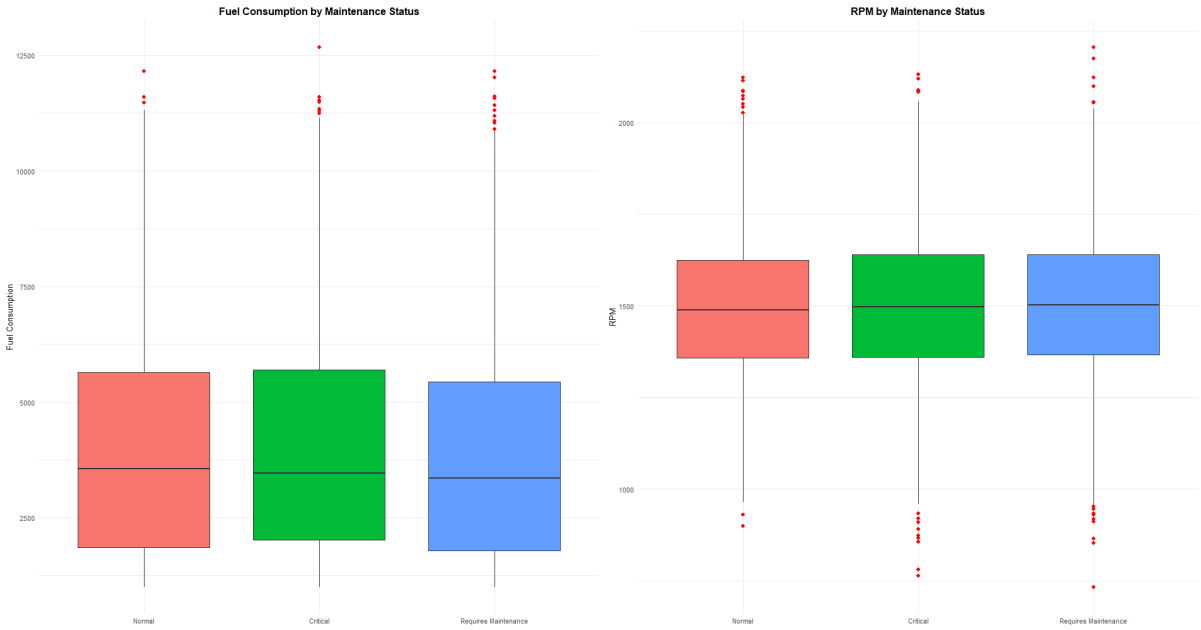

Figure 9A

Figure 10A: Boxplots illustrating the distribution of Running Period, Engine Load, Coolant Temperature, and Engine Temperature across the three maintenance statuses

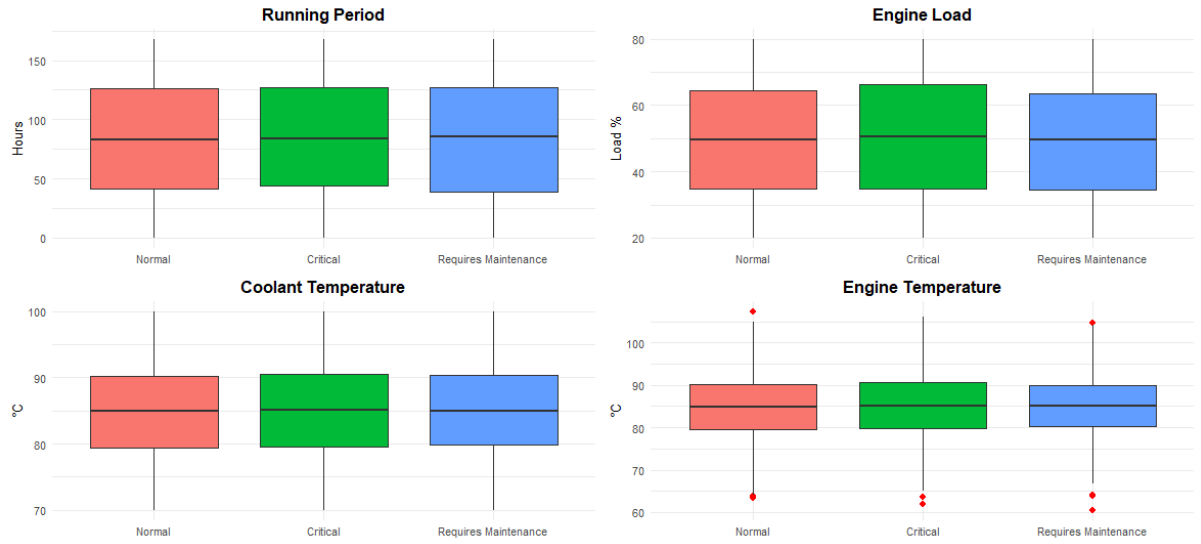

Figure 10A

Figure 11A: Boxplots illustrating the distribution of Oil Pressure, Vibration Level, Fuel Consumption per hour, and Exhaust Temperature across the three maintenance statuses.

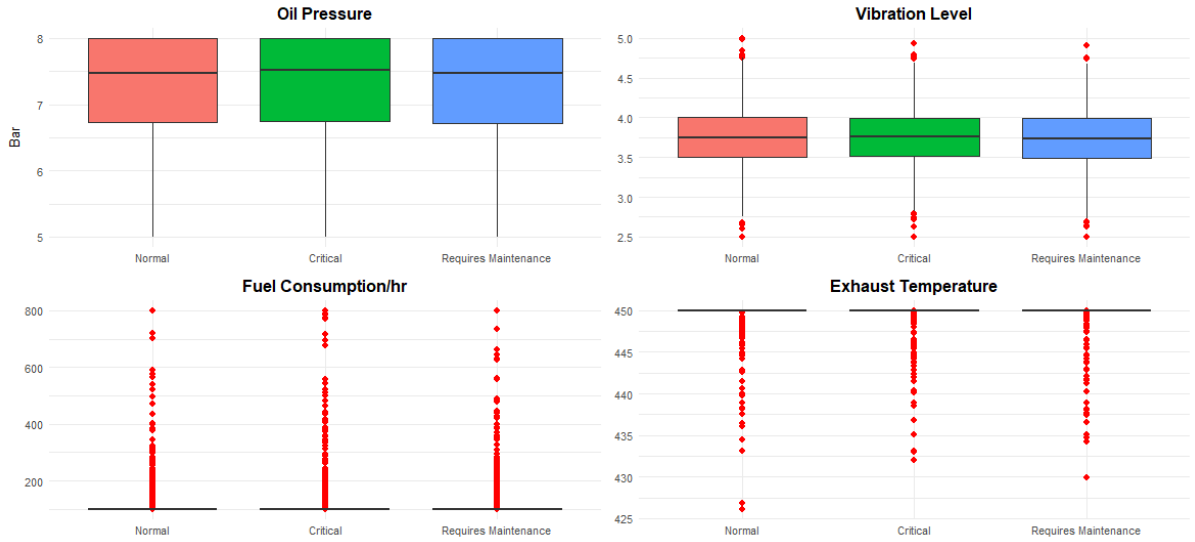

Figure 11A

Figure 12A: Optimal number of clusters using Elbow Method

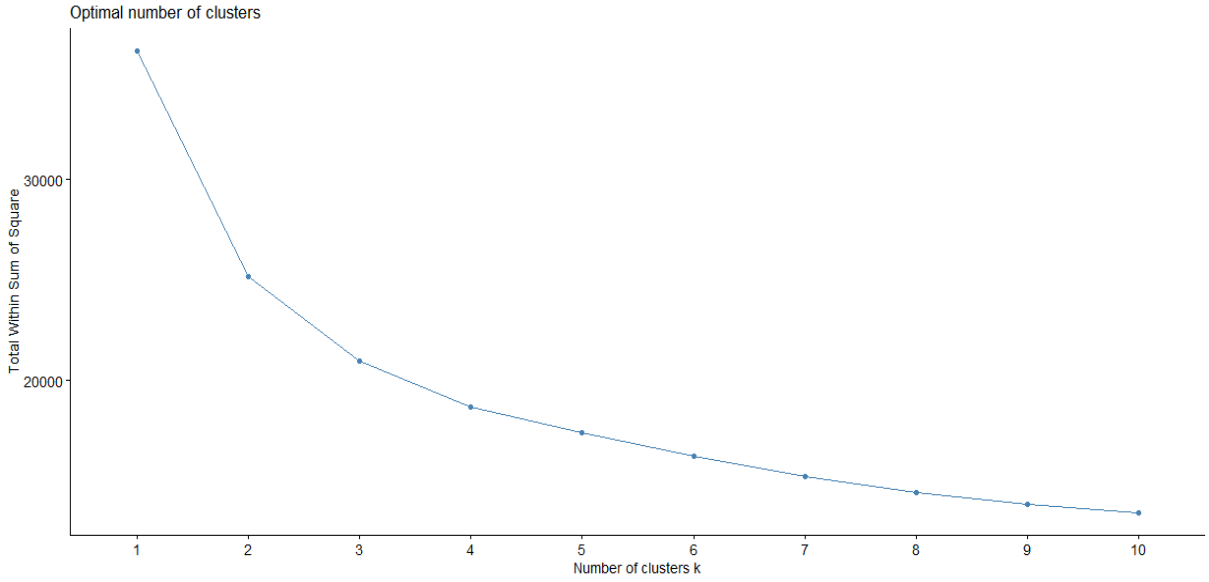

Figure 12A

Figure 13A: Optimal number of clusters using Silhouette Method

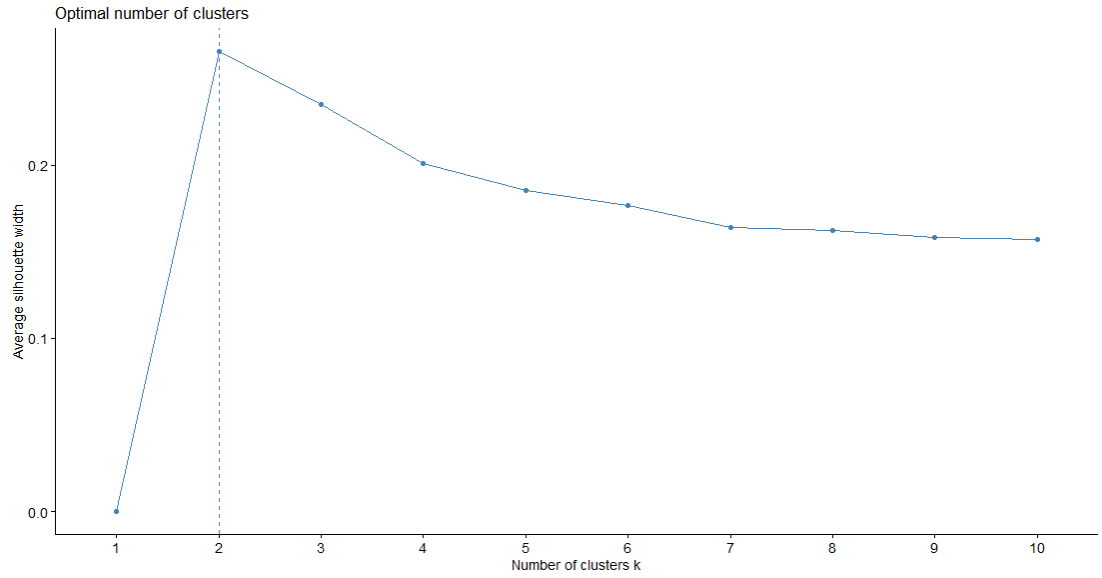

Figure 13A

Figure 14A: PCA projection of engine features compared with three K-means clusters

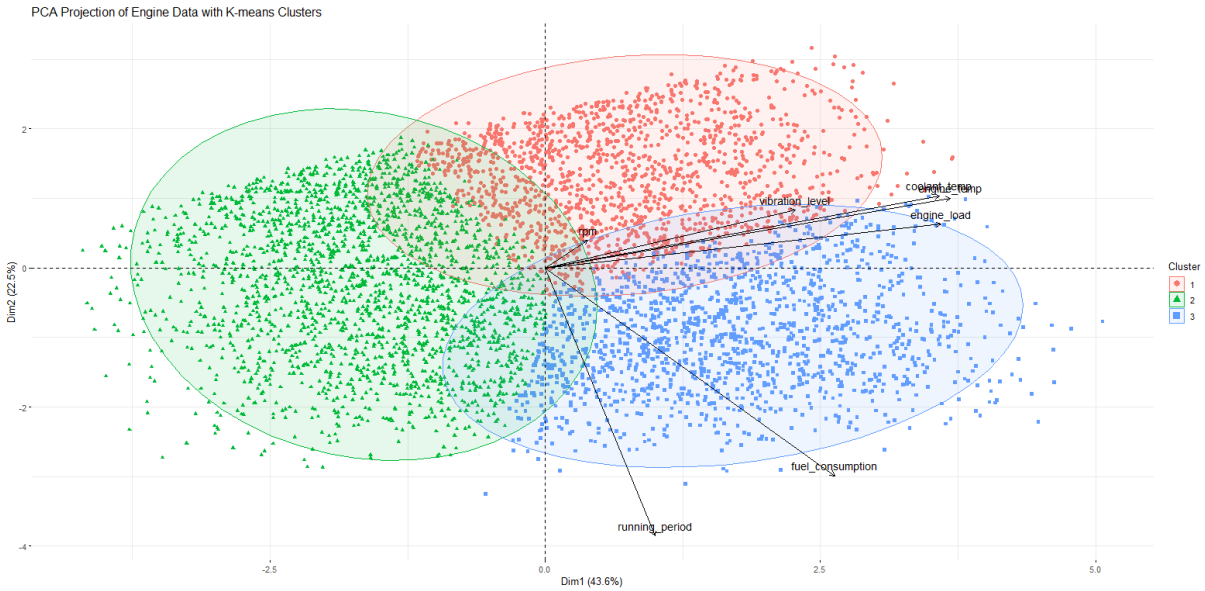

Figure 14A

Figure 15A: Distribution of key features across each cluster

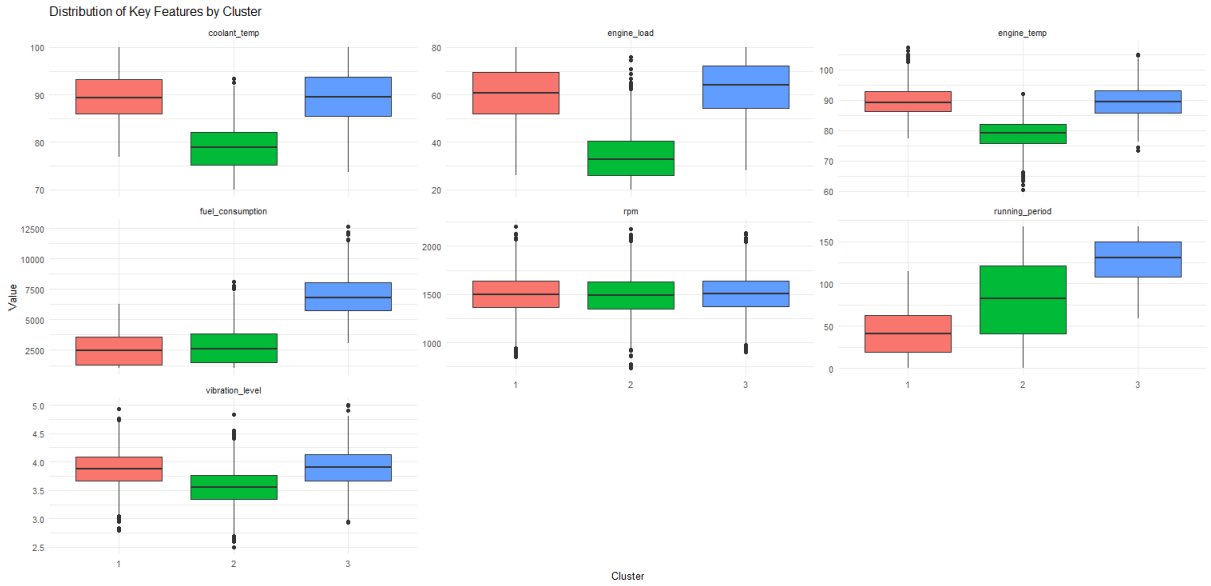

Figure 15A

Figure 16A: Distribution of engine types, fuel types, and manufacturer across K-means clusters

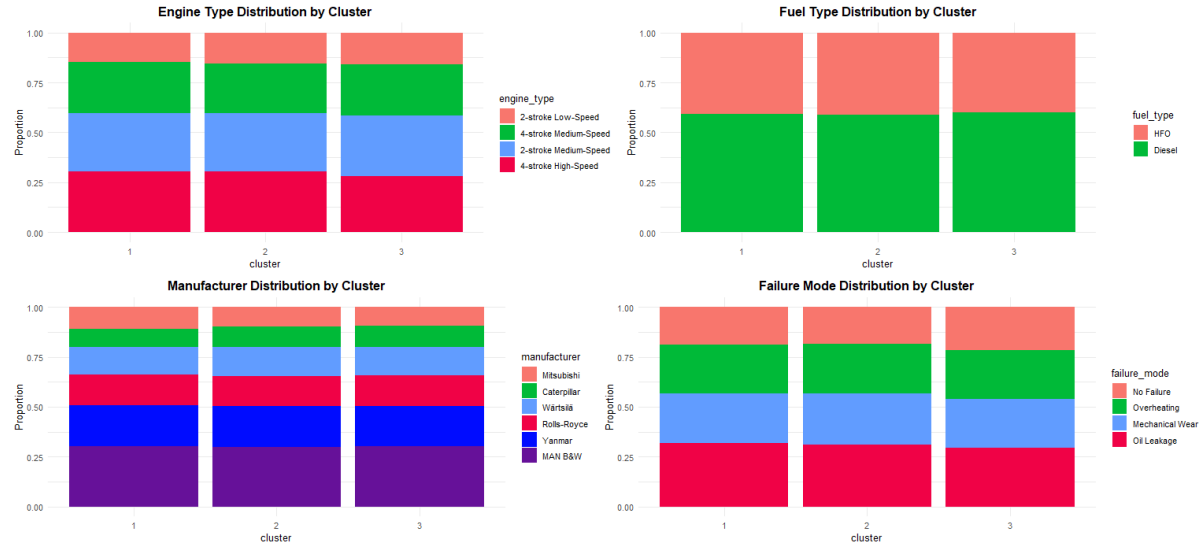

Figure 16A

Figure 17A: Features importance of the penalized MLR model

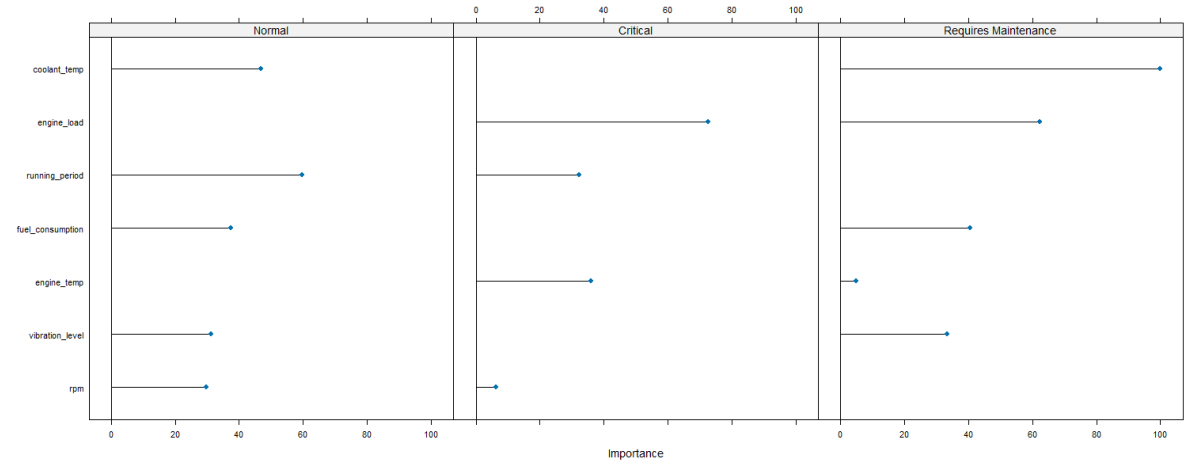

Figure 17A

Figure 18A: Simultaneous tuning of mtry and ntree

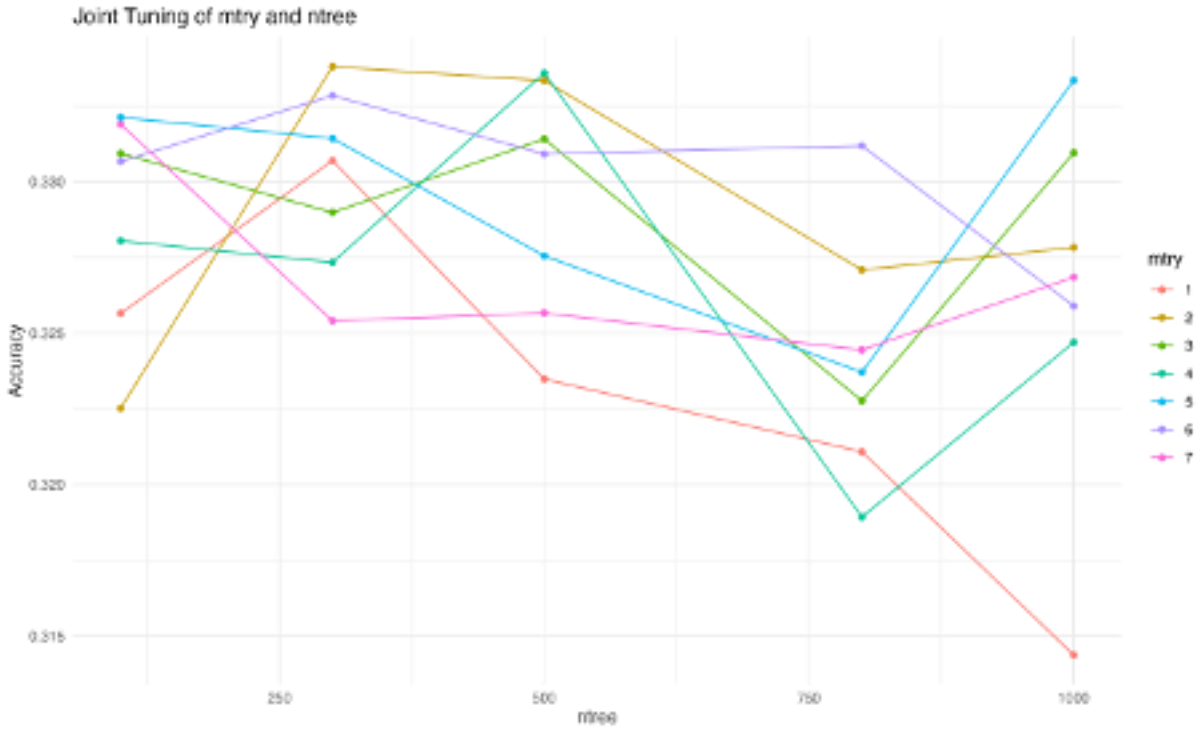

Figure 18A

Figure 19A: Out-of-bag (OOB) error rate for different ntree values

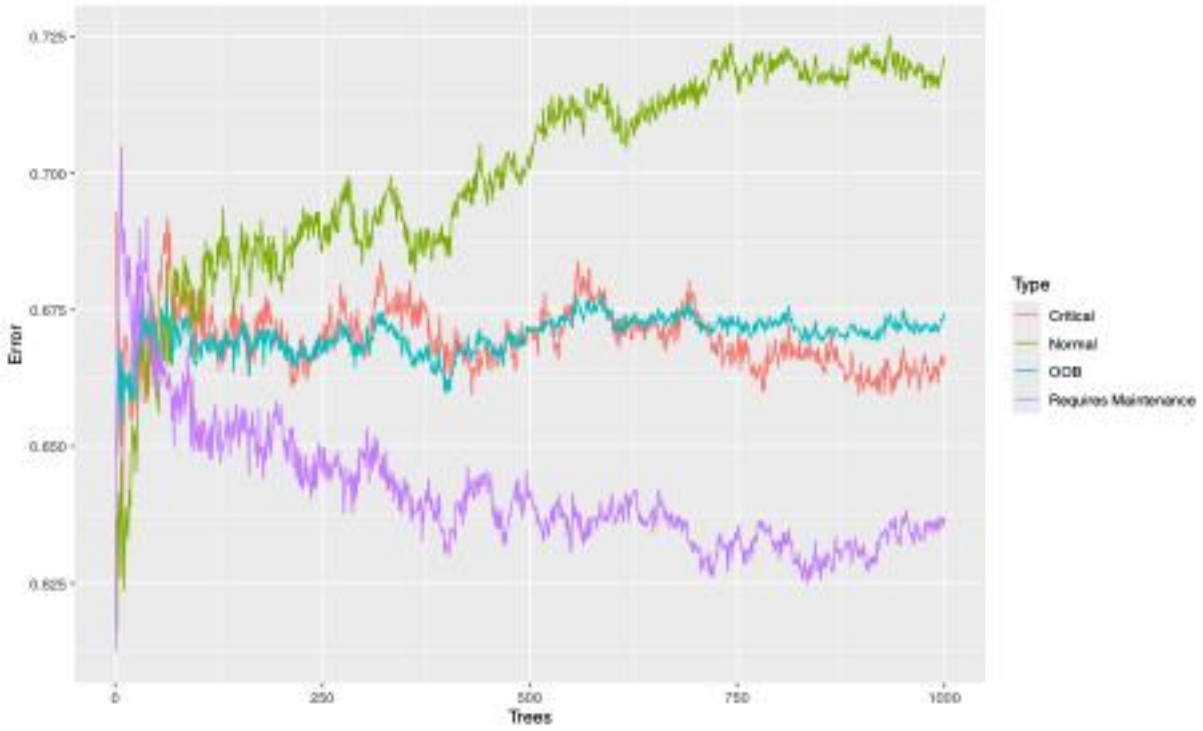

Figure 19A

Figure 20A: Simultaneous tuning of nodesize and maxnodes

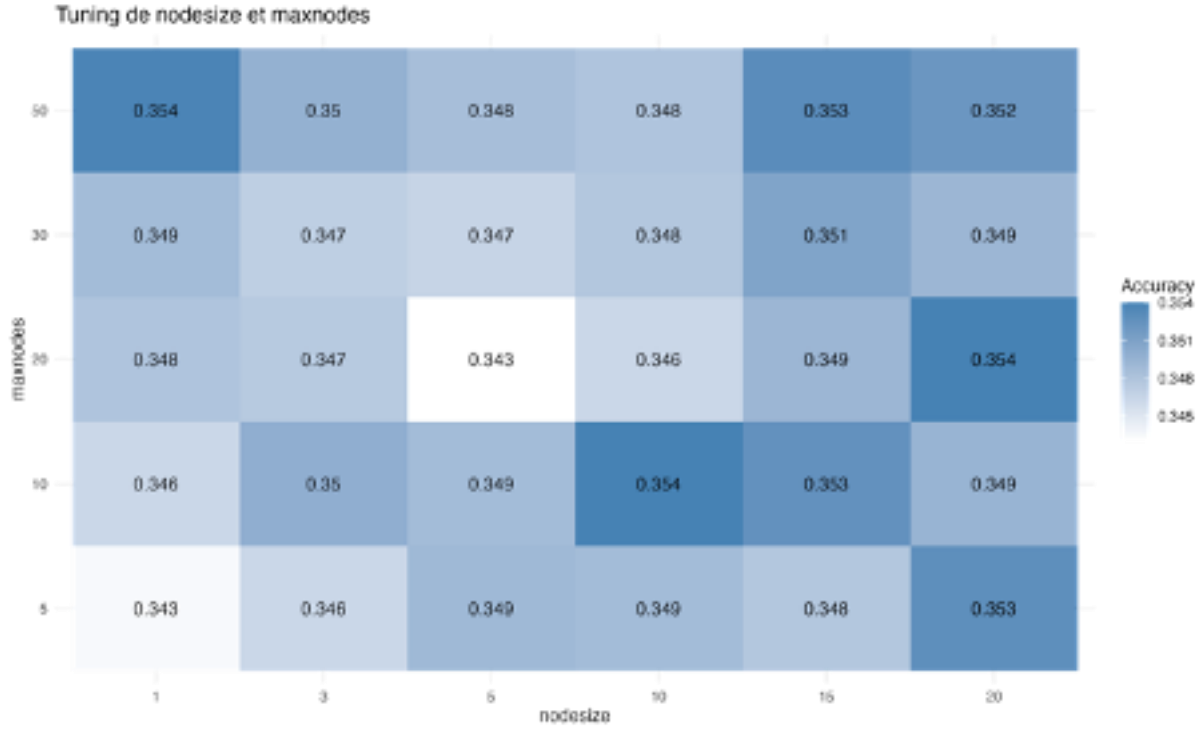

Figure 20A

Figure 21A: Variable importance in the model

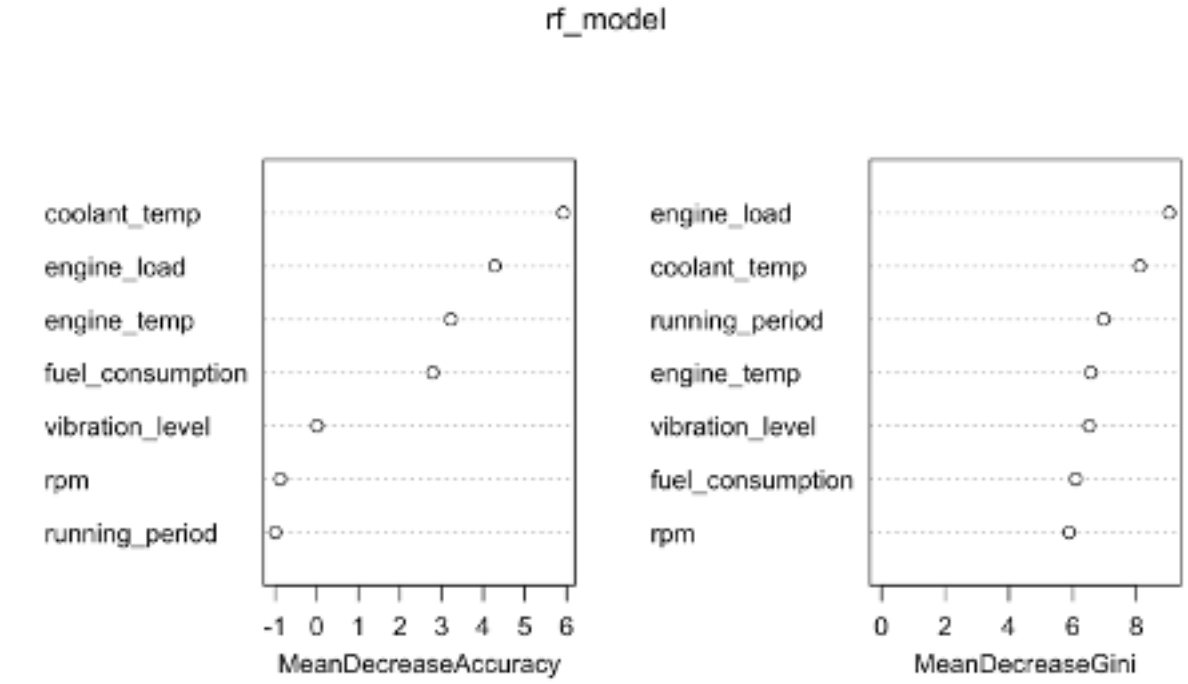

Figure 21A

Figure 22A: Partial Dependence Plot (PDP) for the two main contributing variables

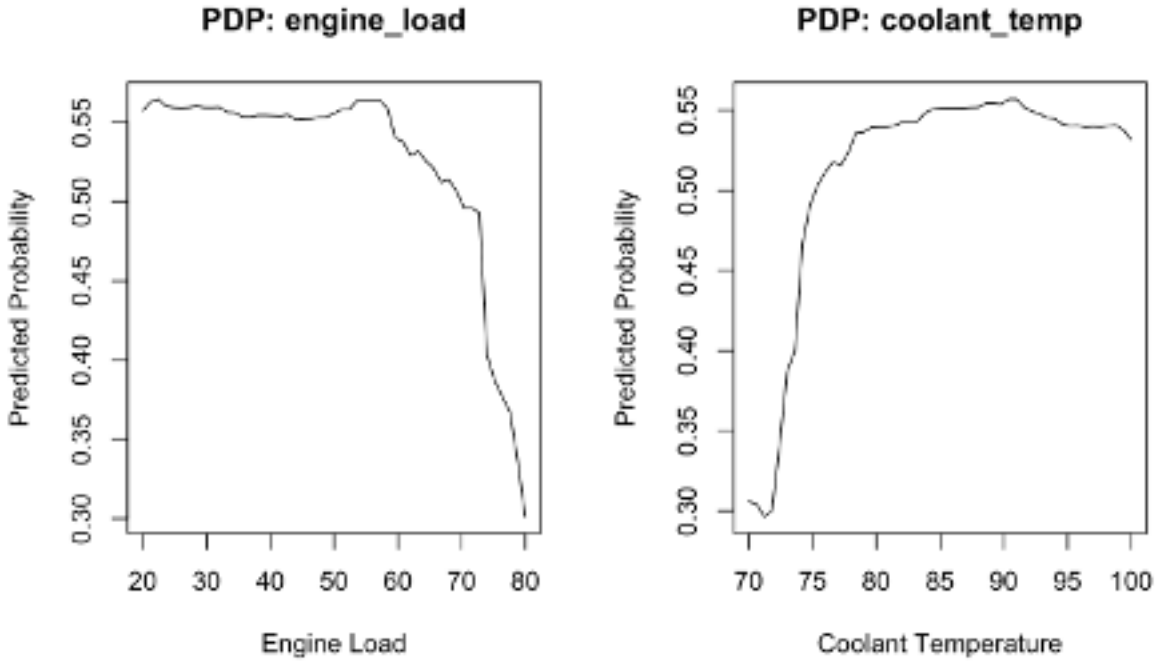

Figure 22A

Figure 23A: SVM hyperparameters tunning grid

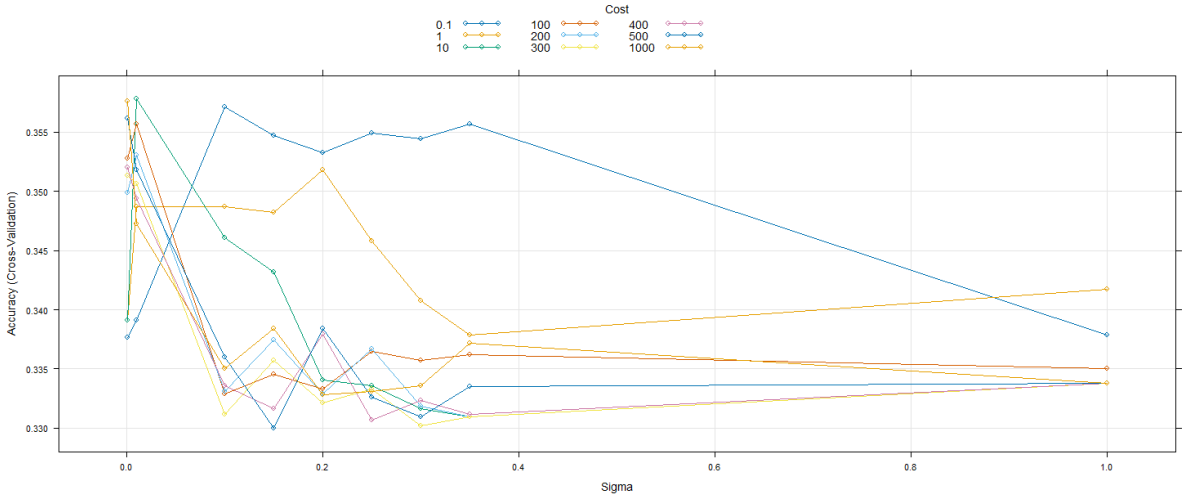

Figure 23A

Appendix B – Tables

Table 1B - Variables Description

Variables Name	Description
Timestamp	Records the weekly time of each observation.
Engine_id	Unique identifier assigned to each marine engine.
Engine_temp	Operating temperature of the engine, measured in degrees Celsius.
Oil_pressure	Pressure of the engine's lubricating oil, in bars.
Fuel_consumption	Total fuel consumed over the weekly observation period.
Vibration_level	Level of mechanical vibration detected in the engine.
Rpm	Revolutions per minute, indicating engine speed.
Engine_load	Percentage representing how much of the engine's capacity is utilized.
Coolant_temp	Temperature of the engine coolant fluid.
Exhaust_temp	Temperature of the exhaust gases emitted by the engine.
Running_period	Duration (in hours) that the engine was operational.
Fuel_consumption_per_hour	Average hourly fuel consumption.
Maintenance_status	Indicates whether the engine requires maintenance, is in normal condition, or is critical.
Failure_mode	Categorical variable describing the cause of engine failure (e.g., Mechanical Wear, Oil Leakage, No Failure).
Engine_type	Classification of the engine (e.g., 2-stroke, 4-stroke, high-speed, low-speed).
Fuel_type	Type of fuel used by the engine (e.g., Diesel).
manufacturer	Manufacturer of the engine (e.g., MAN B&W, Wärtsilä).

Table 1B

Table B2: Importance of the components provided by PCA

Importance of component	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	1.7478	1.2563	1.0793	0.8173	0.5884	0.3351	0.2572
Proportion of the variance	0.4364	0.2255	0.1664	0.0954	0.0495	0.0160	0.0108
Cumulative proportion	0.4364	0.6619	0.8283	0.9237	0.9731	0.9892	1.0000

Table 2B

Table 3B: Contingency matrix of Maintenance Status across clusters

Maintenance Status	Cluster 1	Cluster 2	Cluster 3
Normal	473	727	504
Critical	493	733	514
Requires Maintenance	532	744	480

Table 3B

Table B4: Count and frequency of maintenance class throughout the full dataset and training set

	Full dataset		Training dataset	
Maintenance Status	Count	Frequency	Count	Frequency
Normal	1704	0.328	1364	0.328
Critical	1740	0.335	1392	0.335
Requires Maintenance	1756	0.338	1405	0.338

Table 4B

Table B5: Metrics comparison between all best models

Model	F1 Score	Accuracy	Balanced Acc. (mean)	Sens (N)	Sens (C)	Sens (RM)
MLR	0.33	0.3407	0.5044	0.1971	0.3448	0.4758
RF	0.285	0.3523	0.5150	0.0706	0.1983	0.7778
SVM	0.33	0.3484	0.5099	0.1588	0.3592	0.5214

Table 5B

7 References

[1] Adryan, F. A., & Sastra, K. W. (2021). Predictive maintenance for aircraft engine using machine learning: Trends and challenges. *AVIA*, *3*(1), 37–44. https://avia.ftmd.itb.ac.id/index.php/jav/article/view/45/29

[2] Adryan, F. A., & Wijaya, S. K. (2022). Determining the method of predictive maintenance for aircraft engine using machine learning. *Journal of Computer Science and Technology Studies*, *4*(1), 1–8. https://doi.org/10.32996/jcsts.2022.4.1.1

[3] Jeleel, A. F. (2023). Preventive maintenance for marine engines [Data set]. Kaggle. https://www.kaggle.com/datasets/jeleeladekunlefijabi/preventive-maintenance-for-marine-engines/data

[4] OpenAI. (2025). ChatGPT conversation on predictive maintenance in marine engines. Retrieved May 8, 2025, from https://chat.openai.com/

[5] P. Nunes, J. Santos, E. Rocha, Challenges in predictive maintenance – A review, CIRP Journal of Manufacturing Science and Technology, Volume 40, 2023, Pages 53-67, ISSN 1755-5817, https://doi.org/10.1016/j.cirpj.2022.11.004.

[6] Rehman, A., Khan, M. A., Saba, T., Tariq, U., & Alharbi, R. (2023). Energy-efficient fault detection framework for marine diesel engines using deep learning. *Energy*, *281*, 128676. https://doi.org/10.1016/j.energy.2023.128676

[7] Teke, Orkun & Depci, Tolga. (2023). Zonguldak Bulent Ecevit University Maritime Faculty I. International Maritime and Logistics Congress Proceedings Book/Predictive Maintenance in Maritime Logistics: A Machine Learning Approach (pp.1-9).

[8] Tessaro, I., Mariani, V. C., & Coelho, L. d. S. (2020). Machine learning models applied to predictive maintenance in automotive engine components. *Proceedings*, *64*(1), 26. https://doi.org/10.3390/IeCAT2020-08508

[9] UNCTAD. (2023). *Review of Maritime Transport 2023*. United Nations Conference on Trade and Development. https://unctad.org/webflyer/review-maritime-transport-2023

[10] Zhu, T., Ran, Y., Zhou, X., & Wen, Y. (2024). A survey of predictive maintenance: Systems, purposes and approaches. *arXiv preprint arXiv:1912.07383*. https://doi.org/10.48550/arXiv.1912.07383