# NASA Turbofan Jet Engine Remaining Useful Life Prediction

Ali Ahangarpour
Aghil Zeinalzadeh

April 8, 2024

Data mining Course Project
University of Victoria

# Abstract

Accurate prediction of the Remaining Useful Life of turbo-machinery is important in the domain of predictive maintenance, providing significant cost savings and reliability in aviation industries. This paper presents a comprehensive machine-learning framework designed to predict the Remaining Useful Life of critical components with good precision. The proposed framework integrates multi-sensor time-series data sampled by C-MAPSS from NASA. We evaluate different data-driven techniques, with a core focus on advanced algorithms such as the Gradient Boosting Machine for its robustness in dealing with high-dimensional and noisy data. This study also explores the interpretability of classification machine learning models, evaluating the potential of various classification methods suitable for practical usability for maintenance decision-making. The results demonstrate a good potential of the Gradient Boosting Machine to model the Remaining Useful Life, with 0.018 MSE and 0.83 R2 score. Regarding classification, we explore Support Vector Machines with linear and polynomial kernels, K-nearest neighbours, Logistic Regression, Random Forest Classifier, and Gaussian Naive Bayes focusing on the recall metric. Recall measures the model's ability to correctly identify all relevant instances of a particular class. This can remarkably reduce the cost of missing a true positive case, a scenario in which a machine nearing failure is incorrectly classified as operating normally, which can be significantly higher than that of a false positive. The results showed that the SVM-Poly with rolling means demonstrates exceptional balance and accuracy (0.704) in capturing complex patterns in turbofan operational data, while the Logistic Regression without rolling means stands out with a high recall for DANGER class (0.89) and robust overall performance (accuracy: 0.655).



**Keywords:** Prognostics · Turbo Fan Engine · Classification · Regression · Remaining Useful Life

# Contents



# List of Tables

# List of Figures

Chapter 1

# Introduction

## 1.1 Motivation

Prognostics and health management is vital in industries reliant on complex machinery such as turbofan engines. This domain focuses on predictive maintenance to anticipate and mitigate potential failures. A notable example is the adaptation of NASA's public data set on Kaggle, which features simulated Run-to-Failure data from turbofan engines using the C-MAPSS tool. The data, encompassing various operating scenarios and fault conditions, is recorded across multiple sensors to monitor fault progression and was provided by NASA Ames' Prognostics Center of Excellence.

In aviation, turbofan engines are critical, powering a wide range of aircraft under demanding conditions. Ensuring their operational reliability is a top priority, and prognostics serves as a proactive measure for engine health management. This literature review highlights the progress and challenges in the field of turbofan engine prognostics, examining various methodologies and considering future developments that could further its impact throughout an engine's operational life.

### 1.1.1 Prediction Goal

In this project, the primary objective is to predict the Remaining Useful Life (RUL) of each engine within a given test dataset. RUL is defined as the number of operational cycles an engine can perform before it requires maintenance or replacement, starting from the last recorded data point in the dataset. To achieve a comprehensive analysis, the project is structured to address both regression and classification tasks.

For the regression aspect, the focus is on accurately predicting the exact RUL in terms of operational cycles. This quantitative measure allows for precise maintenance scheduling, optimizing engine use, and reducing the risk of unexpected failures.

The classification task is designed to categorize each engine into one of three states: 'Danger', 'Not Danger Yet', and 'Safe'. These categories are intended to provide a qualitative assessment of engine health, offering a quick reference for maintenance prioritization. 'Danger' indicates that an engine is at immediate risk of failure and requires urgent attention. 'Not Danger Yet' suggests that the engine is approaching a critical condition but can operate for a limited period. 'Safe' signifies that the engine is operating within normal parameters and is not in immediate need of maintenance.

Given the critical nature of ensuring operational safety and reliability, the project emphasizes the importance of the recall metric for the 'Danger' category in the classification task. The rationale for prioritizing recall (the ability of the model to identify all actual instances of the 'Danger' class) over other metrics is rooted in the imperative to minimize the risk of overlooking engines that are close to failure. A high recall rate for the 'Danger' class ensures that nearly all engines at immediate risk are correctly identified and addressed, thereby significantly reducing the potential for catastrophic failures. This approach aligns with the overarching goal of the project to enhance safety and reliability through timely and accurate engine health assessment.

## 1.2 Methodologies in Prognostics

Prognostics in turbofan engines encompass a diverse array of methodologies ranging from statistical models to physics-based simulations and machine learning algorithms. Statistical models utilize historical data to predict future failure probabilities based on statistical analysis techniques such as regression and time series forecasting. Physics-based simulations leverage domain knowledge and engineering principles to model the degradation mechanisms and predict component health degradation over time. Machine learning algorithms, including neural networks, support vector machines, and random forests, enable data-driven prognostic models capable of learning complex patterns and relationships from large-scale sensor data.

## 1.3 Literature Review

The evolution of prognostics in turbofan engines traces back to the early developments in condition monitoring and fault detection systems. Initial approaches relied on simplistic threshold-based methods, which gradually evolved into more sophisticated prognostic models integrating sensor data, machine learning algorithms, and physics-based simulations. Over the years, advancements in sensor technology, computing power, and data analytics have catalyzed the refinement of prognostic techniques, enabling real-time monitoring and predictive maintenance strategies.
The recent literature on prognostics in turbofan engines highlights a significant trend towards employing advanced data-driven methods, particularly deep learning techniques, for predicting the Remaining Useful Life (RUL) of engines. They have pointed out that traditional machine learning methods might fall short due to their reliance on single feature extraction, which often leads to less accurate RUL predictions. To address this, a combination of one-dimensional convolutional neural networks (1-FCLCNN) and Long Short-Term Memory (LSTM) networks has been proposed[5]. This approach aims to extract both temporal and spatial features, demonstrating higher prediction accuracy compared to older models[3].
Additionally a study in 2023 outlined a prognostic procedure that incorporates difference-based feature construction, change-point-detection-based labeling, and a 1D-CNN-LSTM hybrid neural network model for RUL prediction. The new approach has shown clear superiority in prediction capability over traditional machine learning and some deep learning models[2]. Some methods show also a promising result over existing approaches when validated against standard datasets like the C-MAPSS dataset[1].These studies collectively underscore the progressive shift toward integrating multiple data characteristics and leveraging machine learning methods for enhanced predictive performance in turbofan engine prognostics. This reflects the growing complexity of turbofan engines and the need for

more sophisticated predictive maintenance strategies.Li et.al have introduced the Light Gradient Boosting Machine (LightGBM) algorithm, known for handling high-dimensional data and being robust to noise. To improve degradation information capture, they use normalized time window and engine run time data as inputs for the LightGBM model[4]. In this study we are trying to evaluate different machine learning methods to make regression and classification models over the NASA turbofan engine dataset. Data sets consists of multiple multivariate time series. Each data set is further divided into training and test subsets. Each time series is from a different engine i.e., the data can be considered to be from a fleet of engines of the same type. Each engine starts with different degrees of initial wear and manufacturing variation which is unknown to the user. This wear and variation is considered normal, i.e., it is not considered a fault condition. There are three operational settings that have a substantial effect on engine performance. These settings are also included in the data. The data is contaminated with sensor noise.
The engine is operating normally at the start of each time series, and develops a fault at some point during the series. In the training set, the fault grows in magnitude until system failure. In the test set, the time series ends some time prior to system failure. The objective of the competition is to predict the number of remaining operational cycles before failure in the test set, i.e., the number of operational cycles after the last cycle that the engine will continue to operate. Also provided a vector of true RUL values for the test data.
The data are provided as a zip-compressed text file with 26 columns of numbers, separated by spaces, as shown in figure 1.1. Each row is a snapshot of data taken during a single operational cycle, each column is a different variable.

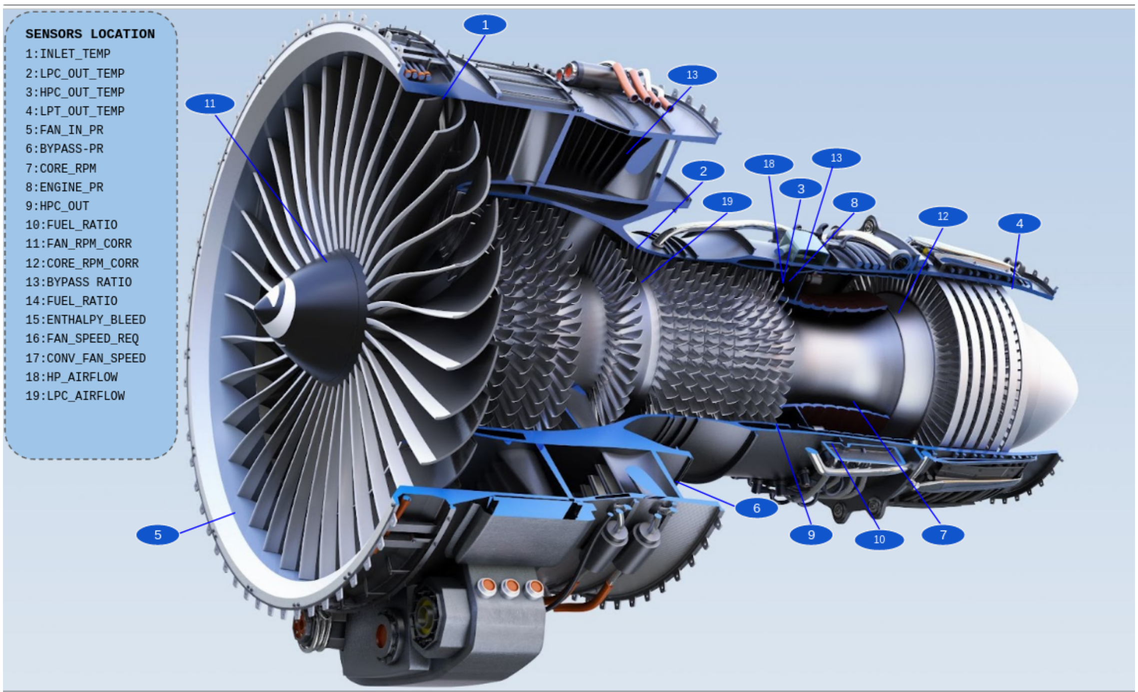



**Fig. 1.1:** Location of important sensors across each turbofan engine

Chapter **2**

# Data Mining

In this chapter, we are going to perform a data mining procedure over NASA's dataset. Alongside this document, we've included two notebooks corresponding to the key areas of this project: regression and classification, which contain end-to-end machine learning procedures. First, we initiate our analysis with data visualization to establish a preliminary understanding of the dataset. Second, we engage in regression analysis, focusing on modeling the relationships between continuous variables. Third, our exploration transitions to classification, to categorize the operational states of machinery into distinct groups. For additional code details, extended visualizations, and comprehensive analyses, please refer to the accompanying notebooks.

## 2.1 Data Visualization

Table 2.1gives the information of all features in dataset, including operational conditions and sensors information located in each turbofan engine. It is apparent that data does not include any null data, and all features have numeric values.

**Table 2.1:** Dataset Columns and Data Types

| # | Feture | Non-Null Count | Dtype |
|---|---|---|---|
| 0 | ENGINE | 20631 non-null | int64 |
| 1 | CYCLE | 20631 non-null | int64 |
| 2 | SET1 | 20631 non-null | float64 |
| 3 | SET2 | 20631 non-null | float64 |
| 4 | SET3 | 20631 non-null | float64 |
| 5 | INLET_TEMP | 20631 non-null | float64 |
| 6 | LPC_OUT_TEMP | 20631 non-null | float64 |
| 7 | HPC_OUT_TEMP | 20631 non-null | float64 |
| 8 | LPT_OUT_TEMP | 20631 non-null | float64 |
| 9 | FAN_IN_PR | 20631 non-null | float64 |
| 10 | BYPASS-PR | 20631 non-null | float64 |
| 11 | HPC_OUT_PR | 20631 non-null | float64 |
| 12 | FAN_RPM | 20631 non-null | float64 |
| 13 | CORE_RPM | 20631 non-null | float64 |
| 14 | ENGINE_PR | 20631 non-null | float64 |
| 15 | HPC_OUT | 20631 non-null | float64 |
| 16 | FUEL_RATIO | 20631 non-null | float64 |
| 17 | FAN_RPM_CORR | 20631 non-null | float64 |
| 18 | CORE_RPM_CORR | 20631 non-null | float64 |
| 19 | BYPASS RATIO | 20631 non-null | float64 |

Continued on next page

**Table 2.1 continued from previous page**

| # | Column | Non-Null Count | Dtype |
|---|---|---|---|
| 20 | FUEL_RATIO_BURNER | 20631 non-null | float64 |
| 21 | ENTHALPY_BLEED | 20631 non-null | int64 |
| 22 | FAN_SPEED_REQ | 20631 non-null | int64 |
| 23 | CONV_FAN_SPEED | 20631 non-null | float64 |
| 24 | HP_AIRFLOW | 20631 non-null | float64 |
| 25 | LPC_AIRFLOW | 20631 non-null | float64 |

We have provided a scatter plot matrix, also known as a pairs plot. Each individual plot in the matrix shows the relationship between two variables in our DataFrame. Scatter plot matrices are very useful for a preliminary examination of the potential relationships between variables. However, they are exploratory in nature; further statistical analysis is typically required to draw precise conclusions about the relationships between variables.

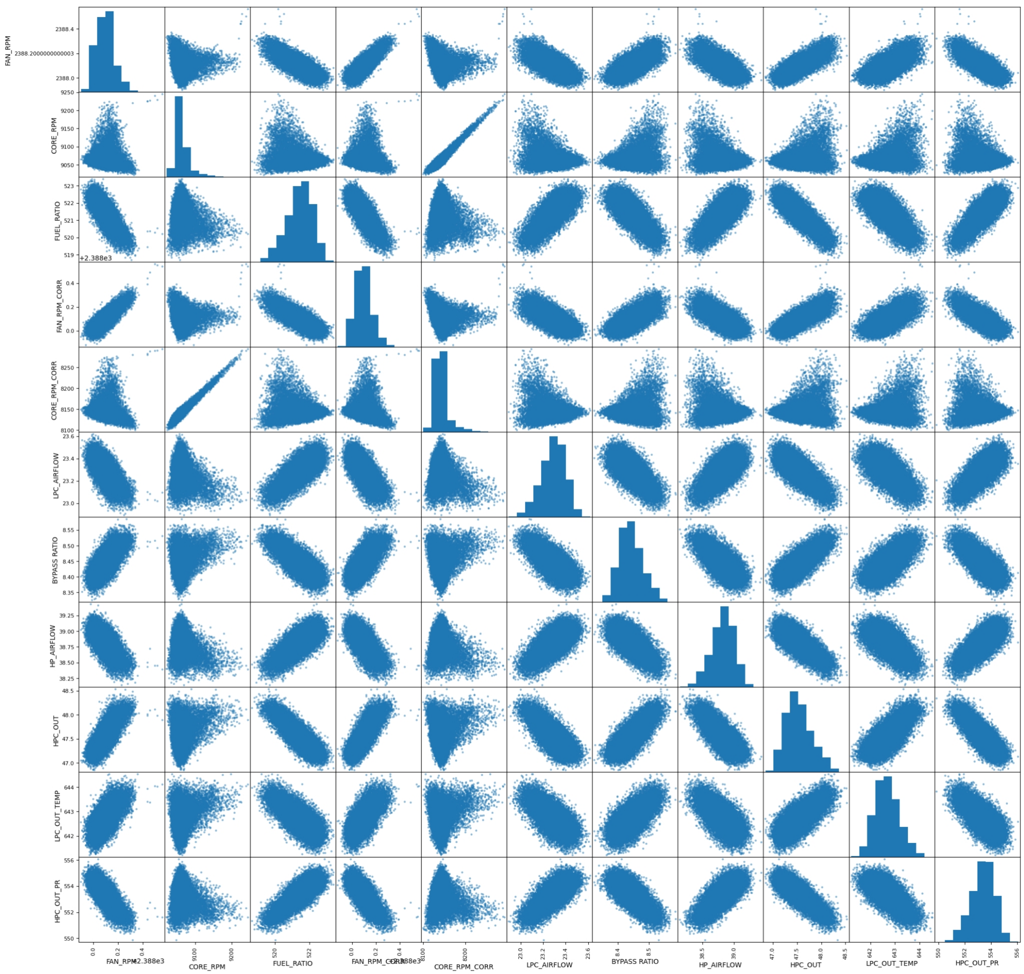



**Fig. 2.1:** The scatter plot of features in dataset

On the diagonal, where the row and column indices are the same, we typically have histograms or kernel density estimates, which show the distribution of a single variable. It looks like histograms are used in your plot. Each bar in a histogram represents the frequency (count) of data points that fall into a corresponding range of values. Almost all features in dataset have normal distribution. We could use np.log() to normalize features, however, it is not necessary. The plots below and above the diagonal represent the pairwise relationships between different variables. Each plot in these positions shows the scatter plot of two different variables, with one variable on the x-axis and the other on the y-axis. The pattern of points can give insights into the correlation between the two variables. For instance, an upward-sloping trend signifies a positive correlation between variables. A downward-sloping trend indicates a negative correlation. A scattered cloud of points without a clear trend points to a lack of linear correlation. The density or spread of the points in the off-diagonal plots can also indicate the strength of the correlation. Points that lie close to a straight line suggest a strong correlation, while points that are widely spread out suggest a weaker correlation. Any points that fall far away from the majority of the other points might be outliers. These are data points that have a markedly different value from others and could be due to variability in the measurement or may indicate an experimental error.

In order to remove unwanted features, we demonstrate the unique features among dataset, shown in Table.2.2. As a result, some variables like SET3, FAN_IN_PR, and ENGINE_PR etc. are removed from the dataset at first step.

**Table 2.2:** Unique values for each feature

| **Feature** | **Unique Values** |
|---|---|
| ENGINE | 100 |
| CYCLE | 362 |
| SET1 | 158 |
| SET2 | 13 |
| SET3 | 1 |
| INLET_TEMP | 1 |
| LPC_OUT_TEMP | 310 |
| HPC_OUT_TEMP | 3012 |
| LPT_OUT_TEMP | 4051 |
| FAN_IN_PR | 1 |
| BYPASS-PR | 2 |
| HPC_OUT_PR | 513 |
| FAN_RPM | 53 |
| CORE_RPM | 6403 |
| ENGINE_PR | 1 |
| HPC_OUT | 159 |
| FUEL_RATIO | 427 |
| FAN_RPM_CORR | 56 |
| CORE_RPM_CORR | 6078 |
| BYPASS_RATIO | 1918 |
| FUEL_RATIO_BURNER | 1 |
| ENTHALPY_BLEED | 13 |
| FAN_SPEED_REQ | 1 |

*Continued on next page*

Table 2.2 – *Continued from previous page*

| Feature | Unique Values |
|---|---|
| CONV_FAN_SPEED | 1 |
| HP_AIRFLOW | 120 |
| LPC_AIRFLOW | 4745 |

### 2.1.1 Features Correlation

Essentially, a correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. The value of a correlation coefficient ranges between -1 and 1. A value close to 1 implies a strong positive correlation: as one variable increases, the other variable tends to also increase. A value close to -1 implies a strong negative correlation: as one variable increases, the other variable tends to decrease. A value close to 0 implies no correlation: changes in one variable do not predict changes in another.
In the heatmaps, each square shows the correlation between the variables on each axis. Correlation values are color-coded according to the color bar on the right side, which ranges from blue (negative correlation) to red (positive correlation). Diagonal cells, which compare a variable to itself, always have a correlation coefficient of 1 (shown in red). The image 2.2 shows the heatmaps of correlation matrices before and after the removal of some features from a dataset.
We started with a full set of features(left), calculated the correlation matrix, and then visualized it using a heatmap. After analyzing the initial heatmap with respect to the RUL, we removed some features from the dataset using a threshold (0.6). The final heatmap (right) represents the correlation matrix after this feature selection process. The remaining features might show a reduced redundancy (less high correlation between features) and potentially a stronger correlation with the target variable, RUL. By removing unnecessary features based on the correlation analysis, you aim to simplify the model, reduce the risk of multicollinearity (which can adversely affect model performance), and potentially enhance the generalization capability of the predictive model.

## 2.2 Regression

Before preparing any machine learning method, the dataset has split to training, 80% and testing dataset,20% using the train_test_split command. All regression models are built over the training dataset and are evaluated over the unseen dataset in the training procedure.
Table 2.3 presents the performance metrics of different regression models on a particular dataset provided by a Jupyter notebook shown in A.1,. In this table MSE (Mean Squared Error) reflects the average squared difference between the observed actual outturns and the values predicted by the model. Lower values are better; they indicate that the model's predictions are closer to the actual values. RMSE (Root Mean Squared Error) is in the same units as the outcome variable. Like MSE, lower values indicate a better fit. R2 Score (Coefficient of Determination) represents the proportion of the variance in the dependent

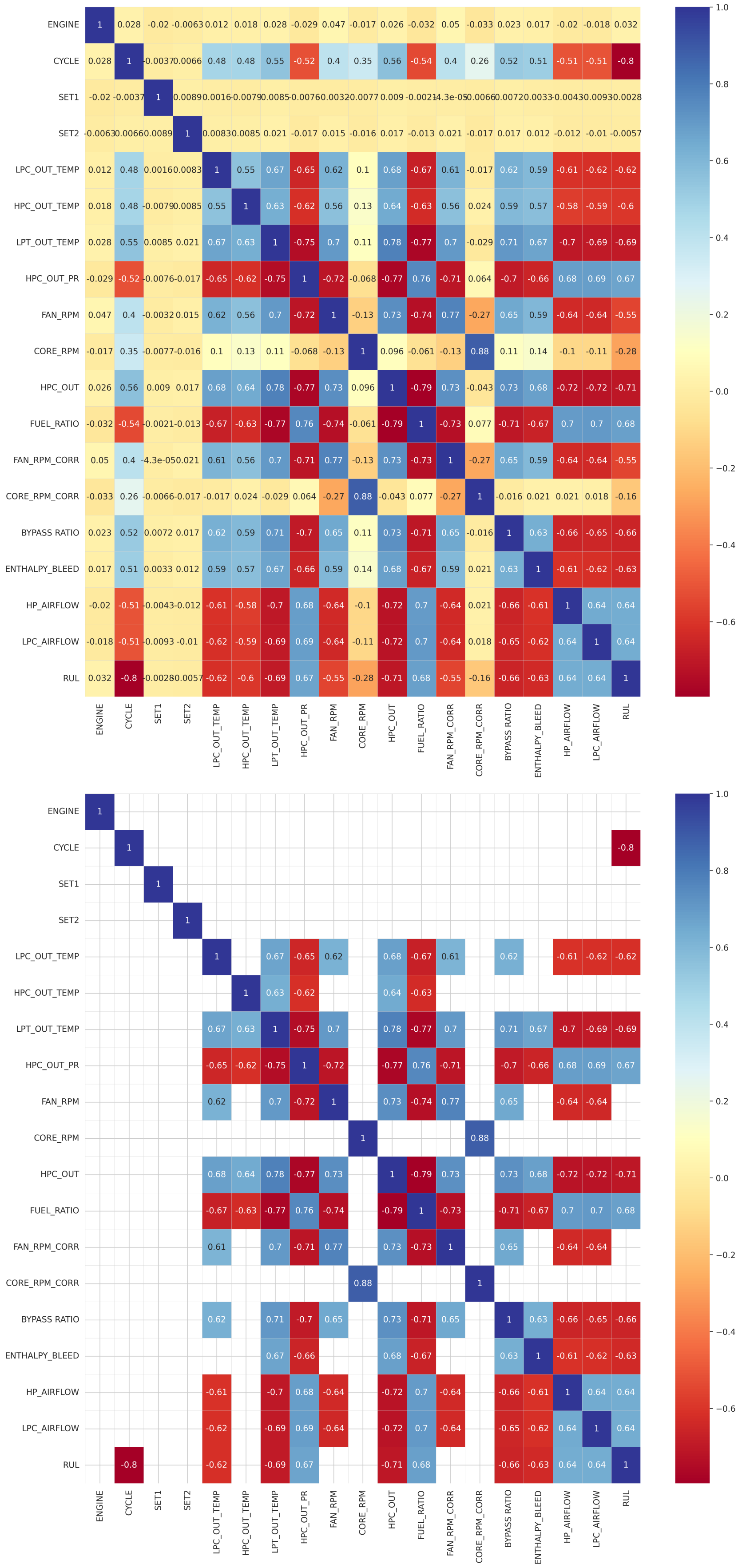



**Fig. 2.2:** Feature correlation and dimensionality reduction

variable that is predictable from the independent variables, with values ranging from 0 to 1. Higher values indicate a better fit, with 1 being a perfect fit.

Linear Regression has an MSE of 0.0256, RMSE of 0.1601, and an R2 score of 0.7631. This model performs reasonably well, with just over 76% of the variance in the target variable being predictable from the features. Random Forest Regressor shows a better performance with the lowest MSE (0.0185) and RMSE (0.1362) among all the models and an R2 score of 0.8286, suggesting it predicts over 82% of the variance. Decision Tree Regressor exhibits the highest MSE (0.0382) and RMSE (0.1955), with an R2 score of 0.6467, indicating it performs less effectively compared to the other models. KNeighbors Regressor has an MSE of 0.0211, RMSE of 0.1453, and an R2 score of 0.8048, making it a relatively strong model with just over 80% of the variance being explained. Gradient Boosting Regressor offers the best R2 score (0.8300) suggesting it can explain 83% of the variance, and it has low MSE (0.0184) and RMSE (0.1356) values, indicating precise predictions and the most effective performance among the models tested. Finally, SVR (Support Vector Regressor) presents a middle ground with an MSE of 0.0195, RMSE of 0.1398, and an R2 score of 0.8194, indicating it is a strong model, though not the best among those evaluated.

**Table 2.3:** Comparison of Different Regression Models

| Model | MSE | RMSE | R2 Score |
|---|---|---|---|
| Linear Regression | 0.0256 | 0.1601 | 0.7631 |
| Random Forest Regressor | 0.0185 | 0.1362 | 0.8286 |
| Decision Tree Regressor | 0.0382 | 0.1955 | 0.6467 |
| KNeighbors Regressor | 0.0211 | 0.1453 | 0.8048 |
| **Gradient Boosting Regressor** | **0.0184** | **0.1356** | **0.8300** |
| SVR | 0.0195 | 0.1398 | 0.8194 |

In summary, based on these metrics, the Gradient Boosting Regressor is the top-performing model for this particular dataset, closely followed by the Random Forest Regressor. The Decision Tree Regressor is the least effective model according to these results. It's important to tune these metrics in the context of the problem using GridSearchCV, shown in A.2. Different metrics might be more or less important depending on the specific objectives and constraints of the regression task. The results of tunning parameter is shown in table 2.4.

**Table 2.4:** Tuned Gradient Boosting Regressor Model Result

| Model | MSE | RMSE | R2 Score |
|---|---|---|---|
| **Gradient Boosting Regressor** | **0.0184** | **0.1356** | **0.8300** |

## 2.3 Classification

In our project, classification task is designed to categorize the operational states of Turbofans into distinct classes based on their Remaining Useful Life, enhancing our predictive maintenance framework. This section details the methodology, models used, evaluation metrics, and key findings from the classification aspect of our study.

Preprocessing included normalization, feature engineering to extract meaningful attributes from raw time-series data, and the application of rolling means to capture temporal trends. The target variable for classification was discretized into three categories: 'Safe', 'Not Danger Yet', and 'Danger', representing the operational state of the machinery based on thresholds defined by domain expertise.

The primary evaluation metric was recall, particularly for the 'DANGER' class, due to the high cost of false negatives in predictive maintenance. Precision, F1-score, and overall accuracy were also considered to provide a comprehensive view of model performance.

In table 2.5, we showcase the correlation of each feature with the target feature. These features constituted the initial set utilized for classification tasks. However, after models with these features, we opted to enhance our dataset by incorporating rolling means of these features. This augmentation aimed to better capture temporal trends and dependencies within the multi-sensor time-series data, an essential aspect given the dynamic nature of turbo-machinery operations.

**Table 2.5:** Correlation with RUL

| Feature | Correlation with RUL |
|---|---|
| HPC_OUT | -0.696228 |
| LPT_OUT_TEMP | -0.678948 |
| FUEL_RATIO | 0.671983 |
| HPC_OUT_PR | 0.657223 |
| BYPASS RATIO | -0.642667 |
| LPC_AIRFLOW | 0.635662 |
| HP_AIRFLOW | 0.629428 |
| LPC_OUT_TEMP | -0.606484 |
| ENTHALPY_BLEED | -0.606154 |
| HPC_OUT_TEMP | -0.584520 |
| FAN_RPM | -0.563968 |
| FAN_RPM_CORR | -0.562569 |
| CORE_RPM | -0.390102 |
| CORE_RPM_CORR | -0.306769 |
| BYPASS-PR | -0.128348 |
| ENGINE | 0.078753 |
| SET1 | -0.003198 |
| SET2 | -0.001948 |

A variety of machine learning models known for their efficacy in classification tasks were employed:

- **Support Vector Machines (SVM):** Linear and polynomial kernels were used to find the hyperplane that best separates the classes in the feature space.
- **K-Nearest Neighbors (KNN):** This model was chosen for its simplicity and effectiveness in classification by analyzing the labels of the 'K' closest data points in the feature space.
- **Logistic Regression:** Used for its ability to provide probabilistic outputs for binary and multi-class classification problems. This model leverages a logistic function to model the probability of a default class.
- **Random Forest Classifier:** Selected for its robustness and ability to handle high-dimensional data without overfitting, by constructing multiple decision trees.
- **Gaussian Naive Bayes:** Known for its efficiency and performance in classification tasks, especially under the assumption of feature independence.

In the following subsections, we first train and evaluate the selected machine learning models using the initial set of features. Then, to establish a benchmark for our evaluation, we employ Dummy classifiers. Following this, we introduce an updated dataset, augmented with rolling means, and retrain the models.

**Table 2.6:** Correlation of Rolling Features with RUL

| **Feature** | **Correlation with RUL** |
|---|---|
| LPT_OUT_TEMP_rolling | -0.732868 |
| LPC_AIRFLOW_rolling | 0.729701 |
| HPC_OUT_rolling | -0.728727 |
| ENTHALPY_BLEED_rolling | -0.728268 |
| BYPASS RATIO_rolling | -0.727228 |
| HPC_OUT_TEMP_rolling | -0.726007 |
| HP_AIRFLOW_rolling | 0.723255 |
| LPC_OUT_TEMP_rolling | -0.721540 |
| HPC_OUT_PR_rolling | 0.711189 |
| FUEL_RATIO_rolling | 0.710306 |
| FAN_RPM_rolling | -0.592877 |
| FAN_RPM_CORR_rolling | -0.591497 |
| CORE_RPM_rolling | -0.393039 |
| CORE_RPM_CORR_rolling | -0.305386 |
| BYPASS-PR_rolling | -0.301141 |

### 2.3.1 Evaluation of Different Classifiers

#### SVM-linear

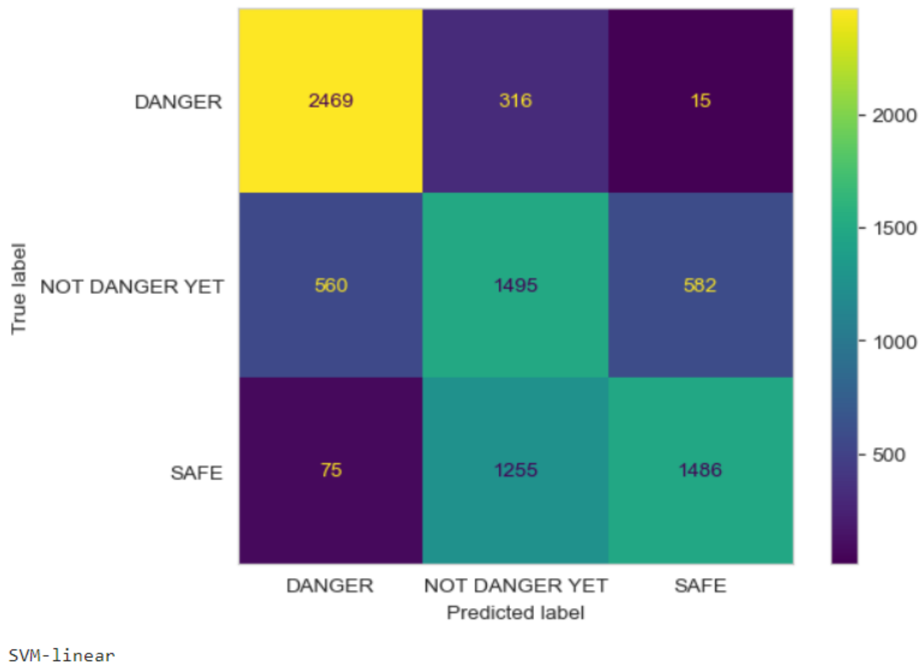



**Fig. 2.3:** SVM-linear without rolling means

**Table 2.7:** SVM-linear performance metrics without rolling means

| Metric | Training | Test |
|---|---|---|
| Accuracy Score | 0.686 | 0.660 |
| Error Rate | 0.181 | 0.168 |

**Table 2.8:** SVM-linear classification report without rolling means

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.80 | 0.88 | 0.84 | 2800 |
| 2 | 0.49 | 0.57 | 0.52 | 2637 |
| 3 | 0.71 | 0.53 | 0.61 | 2816 |
| Accuracy | | | 0.66 | 8253 |
| Macro Avg | 0.67 | 0.66 | 0.66 | 8253 |
| Weighted Avg | 0.67 | 0.66 | 0.66 | 8253 |

### Random Forest Classifier


**Fig. 2.4:** Random Forest Classifier without rolling means

**Table 2.9:** Random Forest Classifier performance metrics without rolling means

| Metric | Training | Test |
|---|---|---|
| Accuracy Score | 0.686 | 0.660 |
| Error Rate | 0.181 | 0.168 |

**Table 2.10:** Random Forest Classifier classification report without rolling means

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.76 | 0.87 | 0.81 | 2800 |
| 2 | 0.47 | 0.54 | 0.50 | 2637 |
| 3 | 0.68 | 0.48 | 0.57 | 2816 |
| Accuracy | | | 0.63 | 8253 |
| Macro Avg | 0.64 | 0.63 | 0.63 | 8253 |
| Weighted Avg | 0.64 | 0.63 | 0.63 | 8253 |

## Gaussian Naive Bayes


**Fig. 2.5:** Gaussian Naive Bayes without rolling means

**Table 2.11:** Gaussian Naive Bayes performance metrics without rolling means

| Metric | Training | Test |
|---|---|---|
| Accuracy Score | 0.647 | 0.633 |
| Error Rate | 0.191 | 0.184 |

**Table 2.12:** Gaussian Naive Bayes classification report without rolling means

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.73 | 0.89 | 0.80 | 2800 |
| 2 | 0.47 | 0.52 | 0.50 | 2637 |
| 3 | 0.70 | 0.48 | 0.57 | 2816 |
| Accuracy | | | 0.63 | 8253 |
| Macro Avg | 0.64 | 0.63 | 0.62 | 8253 |
| Weighted Avg | 0.64 | 0.63 | 0.63 | 8253 |

### KNeighbors Classifier


KNeighborsClassifier

**Fig. 2.6:** KNeighbors Classifier without rolling means

**Table 2.13:** KNeighborsClassifier performance metrics without rolling means

| Metric | Training | Test |
|---|---|---|
| Accuracy Score | 0.696 | 0.662 |
| Error Rate | 0.187 | 0.174 |

**Table 2.14:** KNeighborsClassifier classification report without rolling means

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.85 | 0.84 | 0.84 | 2800 |
| 2 | 0.49 | 0.62 | 0.55 | 2637 |
| 3 | 0.70 | 0.52 | 0.59 | 2816 |
| Accuracy | | | 0.66 | 8253 |
| Macro Avg | 0.68 | 0.66 | 0.66 | 8253 |
| Weighted Avg | 0.68 | 0.66 | 0.66 | 8253 |

### Logistic Regression


**Fig. 2.7:** Logistic Regression without rolling means

**Table 2.15:** Logistic Regression performance metrics without rolling means

| Metric | Training | Test |
|---|---|---|
| Accuracy Score | 0.684 | 0.655 |
| Error Rate | 0.179 | 0.170 |

**Table 2.16:** Logistic Regression classification report without rolling means

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.77 | 0.89 | 0.82 | 2800 |
| 2 | 0.49 | 0.55 | 0.51 | 2637 |
| 3 | 0.72 | 0.52 | 0.61 | 2816 |
| Accuracy | | | 0.65 | 8253 |
| Macro Avg | 0.66 | 0.65 | 0.65 | 8253 |
| Weighted Avg | 0.66 | 0.66 | 0.65 | 8253 |

### 2.3.2 Evaluation of Dummy Classifiers

#### Stratified Classifier

**Table 2.17:** Stratified Classifier performance

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.36 | 0.34 | 0.35 | 2800 |
| 2 | 0.34 | 0.35 | 0.34 | 2637 |
| 3 | 0.34 | 0.34 | 0.34 | 2816 |
| Accuracy | | | 0.344 | 8253 |
| Macro Avg | 0.34 | 0.34 | 0.34 | 8253 |
| Weighted Avg | 0.34 | 0.34 | 0.34 | 8253 |

#### Most Frequent Classifier

**Table 2.18:** Most Frequent Classifier performance

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.00 | 0.00 | 0.00 | 2800 |
| 2 | 0.00 | 0.00 | 0.00 | 2637 |
| 3 | 0.34 | 1.00 | 0.51 | 2816 |
| Accuracy | | | 0.341 | 8253 |
| Macro Avg | 0.11 | 0.33 | 0.17 | 8253 |
| Weighted Avg | 0.12 | 0.34 | 0.17 | 8253 |

#### Uniform Classifier

**Table 2.19:** Uniform Classifier performance

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.33 | 0.33 | 0.33 | 2800 |
| 2 | 0.32 | 0.33 | 0.32 | 2637 |
| 3 | 0.33 | 0.32 | 0.33 | 2816 |
| Accuracy | | | 0.329 | 8253 |
| Macro Avg | 0.33 | 0.33 | 0.33 | 8253 |
| Weighted Avg | 0.33 | 0.33 | 0.33 | 8253 |

The clear superiority of other models' results over the dummy classifiers suggests that our classifiers are capable of making meaningful predictions.

### 2.3.3 Adding Rolling Means and More Results

#### SVM-linear

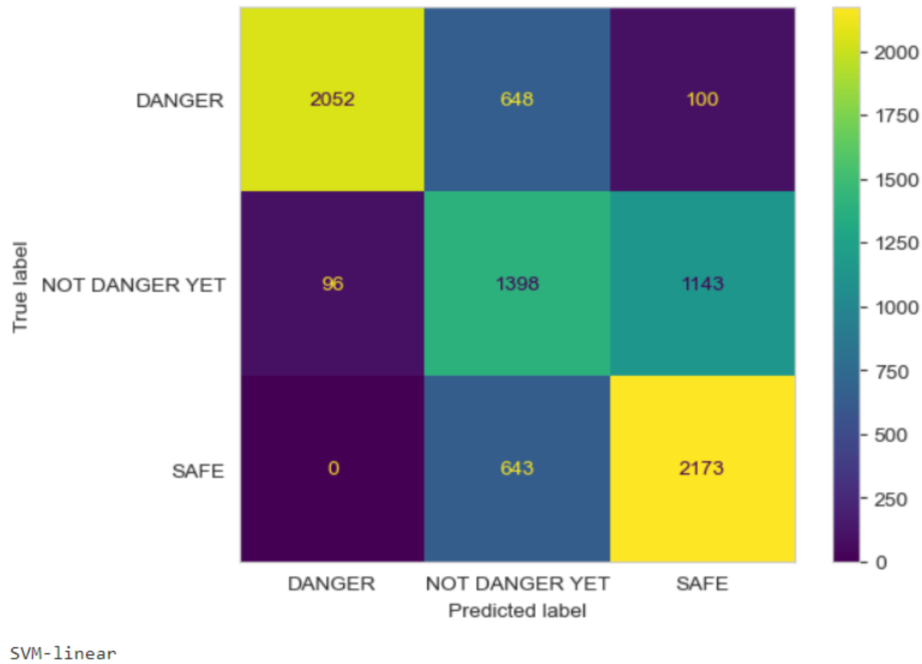



**Fig. 2.8:** SVM-linear with rolling means

**Table 2.20:** SVM-linear performance metrics with rolling means

| Metric | Training | Test |
|---|---|---|
| Accuracy Score | 0.691 | 0.681 |
| Error Rate | 0.178 | 0.204 |

**Table 2.21:** SVM-linear classification report with rolling means

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.96 | 0.73 | 0.83 | 2800 |
| 2 | 0.52 | 0.53 | 0.52 | 2637 |
| 3 | 0.64 | 0.77 | 0.70 | 2816 |
| Accuracy | | | 0.681 | 8253 |
| Macro Avg | 0.70 | 0.68 | 0.68 | 8253 |
| Weighted Avg | 0.71 | 0.68 | 0.69 | 8253 |

### SVM-Poly


SVM-Ploy

**Fig. 2.9:** SVM-Poly with rolling means

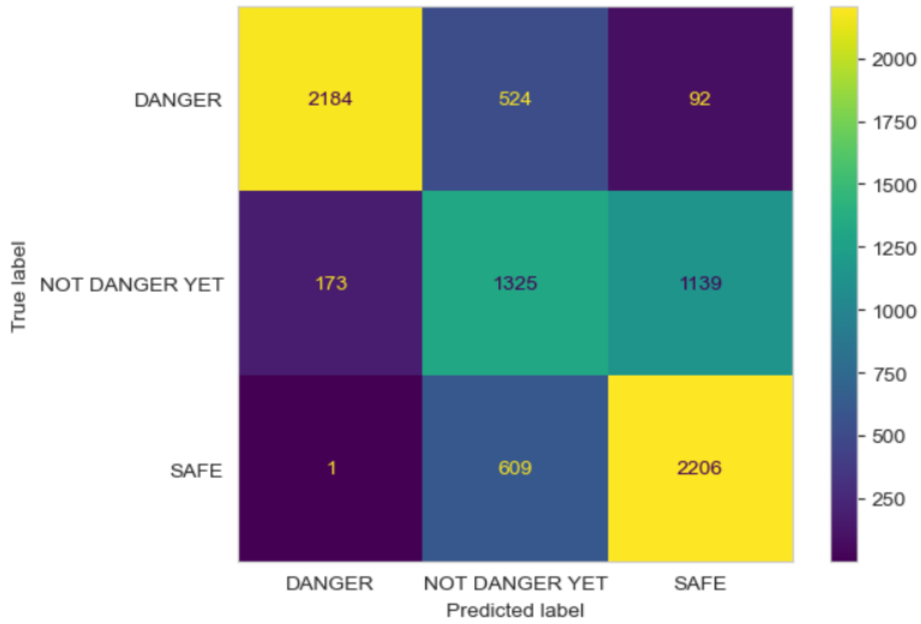
**Table 2.22:** SVM-Poly performance metrics with rolling means

| Metric | Training | Test |
|---|---|---|
| Accuracy Score | 0.725 | 0.704 |
| Error Rate | 0.172 | 0.196 |

**Table 2.23:** SVM-Poly classification report with rolling means

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.96 | 0.76 | 0.85 | 2800 |
| 2 | 0.57 | 0.52 | 0.54 | 2637 |
| 3 | 0.64 | 0.82 | 0.72 | 2816 |
| Accuracy | | | 0.704 | 8253 |
| Macro Avg | 0.72 | 0.70 | 0.70 | 8253 |
| Weighted Avg | 0.72 | 0.70 | 0.71 | 8253 |

### Logistic Regression




LogisticRegression

**Fig. 2.10:** Logistic Regression with rolling means

**Table 2.24:** LogisticRegression performance metrics with rolling means

| Metric | Training | Test |
|---|---|---|
| Accuracy Score | 0.690 | 0.692 |
| Error Rate | 0.174 | 0.190 |

**Table 2.25:** LogisticRegression classification report with rolling means

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.93 | 0.78 | 0.85 | 2800 |
| 2 | 0.54 | 0.50 | 0.52 | 2637 |
| 3 | 0.64 | 0.78 | 0.71 | 2816 |
| Accuracy | | | 0.692 | 8253 |
| Macro Avg | 0.70 | 0.69 | 0.69 | 8253 |
| Weighted Avg | 0.71 | 0.69 | 0.69 | 8253 |

### Random Forest Classifier


**Fig. 2.11:** Random Forest Classifier with rolling means

**Table 2.26:** RandomForestClassifier performance metrics with rolling means

| Metric | Training | Test |
|---|---|---|
| Accuracy Score | 0.993 | 0.678 |
| Error Rate | 0.004 | 0.189 |

**Table 2.27:** Random Forest Classifier classification report with rolling means

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.86 | 0.82 | 0.84 | 2800 |
| 2 | 0.53 | 0.51 | 0.52 | 2637 |
| 3 | 0.65 | 0.70 | 0.67 | 2816 |
| Accuracy | | | 0.678 | 8253 |
| Macro Avg | 0.68 | 0.67 | 0.68 | 8253 |
| Weighted Avg | 0.68 | 0.68 | 0.68 | 8253 |

### Gaussian Naive Bayes

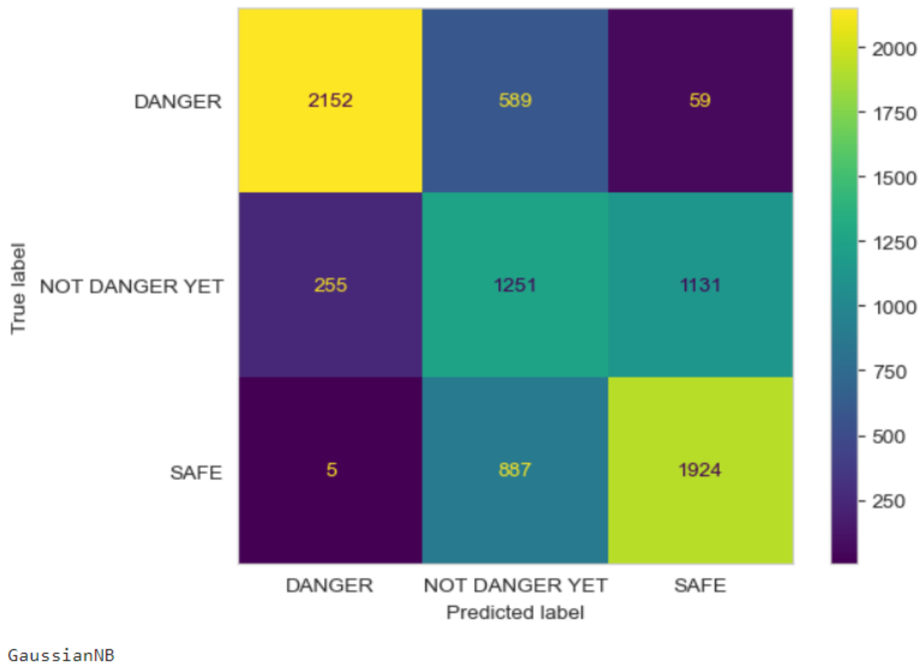



**Fig. 2.12:** Gaussian Naive Bayes with rolling means

**Table 2.28:** Gaussian Naive Bayes performance metrics with rolling means

| Metric | Training | Test |
|---|---|---|
| Accuracy Score | 0.638 | 0.645 |
| Error Rate | 0.200 | 0.206 |

**Table 2.29:** Gaussian Naive Bayes classification report with rolling means

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.89 | 0.77 | 0.83 | 2800 |
| 2 | 0.46 | 0.47 | 0.47 | 2637 |
| 3 | 0.62 | 0.68 | 0.65 | 2816 |
| Accuracy | | | 0.645 | 8253 |
| Macro Avg | 0.66 | 0.64 | 0.65 | 8253 |
| Weighted Avg | 0.66 | 0.65 | 0.65 | 8253 |

### KNeighbors Classifier

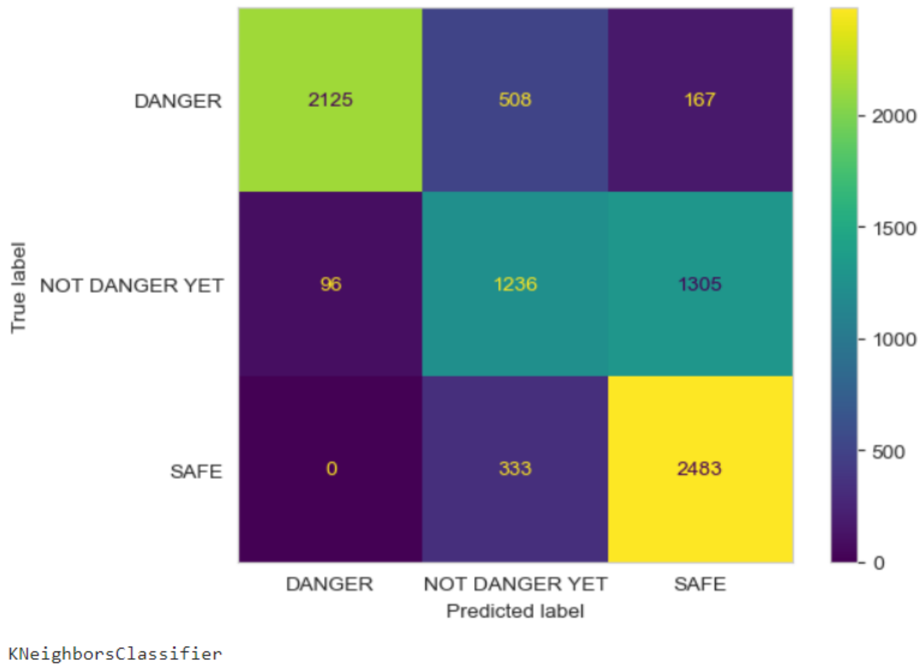



**Fig. 2.13:** KNeighbors Classifier with rolling means

**Table 2.30:** KNeighborsClassifier performance metrics with rolling means

| Metric | Training | Test |
|---|---|---|
| Accuracy Score | 0.723 | 0.708 |
| Error Rate | 0.172 | 0.200 |

**Table 2.31:** KNeighbors Classifier classification report with rolling means

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 1 | 0.96 | 0.76 | 0.85 | 2800 |
| 2 | 0.60 | 0.47 | 0.52 | 2637 |
| 3 | 0.63 | 0.88 | 0.73 | 2816 |
| Accuracy | | | 0.708 | 8253 |
| Macro Avg | 0.73 | 0.70 | 0.70 | 8253 |
| Weighted Avg | 0.73 | 0.71 | 0.70 | 8253 |

### 2.3.4 Evaluation Analysis

Given the detailed metrics provided for each model in the previous subsections, including their performance on training and test datasets, as well as precision, recall, and F1-score across classes, we can now conduct a more informed analysis to identify the best model both from the perspective of recall for the "Danger" class (class 1) and overall performance. Here we dive deeper into the results and analyze them from a statistical and model performance standpoint.

- **SVM and Polynomial SVM:**
  - **Before Rolling Means:** The liner SVM showed a great balance of precision and recall for the first class, however did not perform well in terms of accuracy.
  - **After Rolling Means:** Both variants exhibited marked improvement in terms of accuracy, significantly with polynomial SVM, indicating an enhanced ability to exploit non-linear patterns brought out by rolling means.
- **Random Forest Classifier:**
  - **Training vs. Testing Performance:** Exhibited high training accuracy with a noticeable drop in testing accuracy, indicating overfitting.
  - **Impact of Rolling Means:** Slight improvement in testing performance with rolling means, suggesting value in temporal features but also a need for addressing overfitting.
- **Gaussian Naive Bayes:**
  - **Consistency in Performance:** Slight improvement with the addition of rolling means, indicating robustness and efficiency, especially for computational simplicity.
- **K-Nearest Neighbors (KNN):**
  - **Improvement with Rolling Means:** Noticeable improvement highlights the model's dependency on local similarity, with rolling means providing a nuanced view of temporal structure. This model holds the highest accuracy among all the models.
- **Logistic Regression:**
  - **Stable Performance:** Slight improvement with rolling means in terms of precision and accuracy however the recall for the DANGER class reduced.

However, the improvements after adding the rolling mean future were in terms of accuracy and precision and didn't improve the recall of the DANGER class.

The recall metric is particularly crucial for the "Danger" class (Class 1), as it ensures all potentially dangerous states are accurately identified, minimizing the risk of false negatives. **Logistic Regression** without rolling means exhibits a recall for the "Danger" class (Class 1) of 0.89, making it highly effective at identifying potentially dangerous operational states.

When considering overall performance, which includes accuracy, precision, recall, and F1-scores across all classes, the model demonstrating the highest overall accuracy and balanced performance metrics is deemed the best.

- The **SVM-Poly with rolling means** emerges as the top performer across several metrics, demonstrating an exceptional balance of precision, recall, and F1-scores. With an overall accuracy of 0.704, it not only showcases its strength in capturing complex, nonlinear patterns inherent in the operational data of turbofans but also illustrates its versatility in generalizing well from training to testing datasets.

- The **Logistic Regression without rolling means** model, with a recall of 0.89 for the "Danger" class and an overall accuracy of 0.655, stands out for its ability to accurately identify the critical "Danger" operational state while also maintaining robust overall performance.

Chapter 3

# Conclusion

Despite significant advancements, prognostics in turbofan engines face several challenges. Data quality and availability remain a primary concern, as sensor data may be noisy, incomplete, or unreliable. Additionally, the complexity of turbofan engines poses challenges in developing accurate and robust prognostic models capable of capturing the intricate interdependencies among various components. Furthermore, transitioning from research prototypes to practical implementations necessitates addressing issues related to scalability, interpretability, and integration with existing maintenance systems.

Regarding the regression models, different machine learning methods including SVR, Linear Regression, Random Forest Regression, and Gradient boosting were evaluated over suitable preprocessed features. The gradient boosting method showed a remarkable score compared to the other methods. In our classification task aimed at enhancing the predictive maintenance of turbofan engines, we meticulously evaluated various machine learning models to categorize operational states into distinct classes. Throughout this process, the two models distinguished themselves for their specific strengths. The SVM-Poly with rolling means was identified as the superior model overall, notable for its adept handling of the dataset's complex patterns and achieving high accuracy. Its success underscores the value of incorporating temporal trends via rolling means to better understand the dynamic nature of turbofan operations. On the other hand, Logistic Regression without rolling means proved exceptional in its ability to detect the critical "Danger" state, boasting an impressive recall that highlights its importance in preventive maintenance scenarios where identifying potential failures is paramount. Together, these models form a robust foundation for a predictive maintenance framework, with SVM-Poly offering broad classification capabilities and Logistic Regression ensuring acute detection of the most critical operational risks.

# Bibliography


[1] Author, U., et al.: Predicting the remaining useful life of turbofan engines using fractional lévy stable motion with long-range dependence. Fractal and Fractional **8**(1), 55 (2024). https://doi.org/10.3390/fractalfract8010055

[2] Emel, E., et al.: Remaining useful life estimation of turbofan engines with deep learning using change-point detection based labeling and feature engineering. Applied Sciences **13**(21), 11893 (2023). https://doi.org/10.3390/app132111893

[3] Gui, W., et al.: A remaining useful life prognosis of turbofan engine using temporal and spatial feature fusion. Sensors **21**(2), 418 (2021). https://doi.org/10.3390/s21020418

[4] Li, F., Zhang, L., Chen, B., Gao, D., Cheng, Y., Zhang, X., Yang, Y., Gao, K., Huang, Z., Peng, J.: A light gradient boosting machine for remainning useful life estimation of aircraft engines. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). pp. 3562–3567 (2018). https://doi.org/10.1109/ITSC.2018.8569801

[5] Peng, Cheng, C.Y.: Remaining useful life prognosis of turbofan engines based on deep feature extraction and fusion **12** (2022). https://doi.org/10.1038/s41598-022-10191-2