Institute of Engineering & Technology (IET)
J.K. Lakshmipat University

CS1138: MACHINE LEARNING

# RUL PREDICTION FOR TURBO ENGINES

FACULTY GUIDE:

Dr ARPAN GUPTA

GROUP MEMBERS:

ANURAG SUTHAR (2022BTECH015)

NIKHIL PAREEK (2022BTECH065)

AMAN BHARTI (2022BTECH121)

GOVIND SHARMA (2022BTECH126)

# INDEX

## Introduction

In recent years, the aviation industry has seen a paradigm shift towards predictive maintenance strategies aimed at improving operational efficiency, reducing downtime and ensuring safety for passengers. One of the main challenges in this field is the accurate prediction of the remaining useful life (RUL) of aircraft components, such as turbofan engines. The ability to predict the remaining operating time of critical components enables maintenance teams to proactively plan maintenance activities, optimize resource allocation, and minimize the risk of breakdowns unexpected.

## Problem-solving motivation

The motivation behind this project stems from the important role of turbofan engines in aviation and the potential benefits of predictive maintenance techniques. Traditional reactive maintenance methods that rely on predetermined maintenance schedules or incidents are often costly and ineffective. In contrast, predictive maintenance leverages machine learning algorithms and sensor data to predict the RUL of engine components, enabling proactive and cost-effective maintenance activities.

## Applications

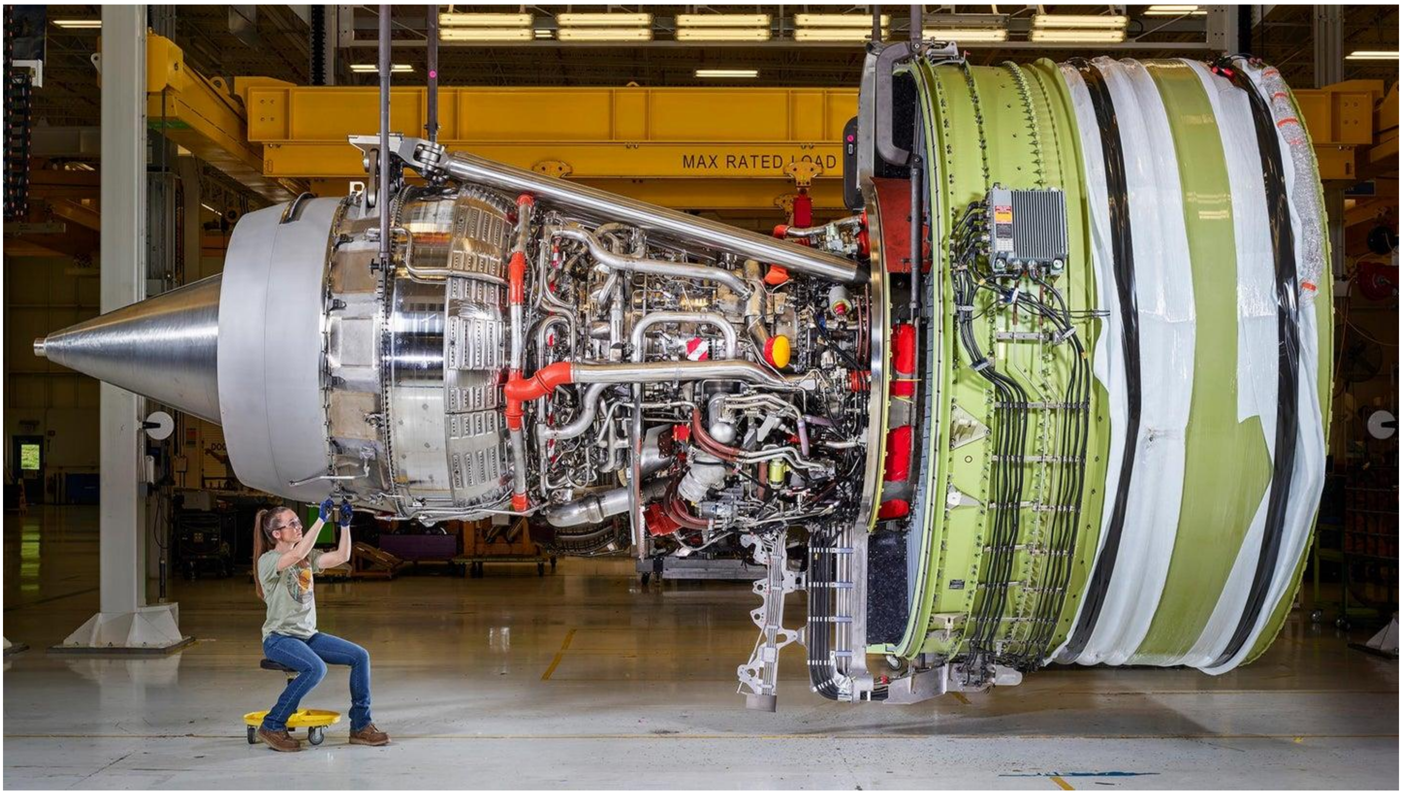
The results of this project have wide applications in a variety of fields, including aeronautics, aerospace engineering, and industrial machinery. In addition to turbofan engines, the predictive maintenance techniques developed in this project can be adapted to predict the life of a variety of equipment and assets, from wind turbines to machinery. manufacture. RUL prediction facilitates resource optimization, reduces maintenance costs, and improves reliability of critical systems.

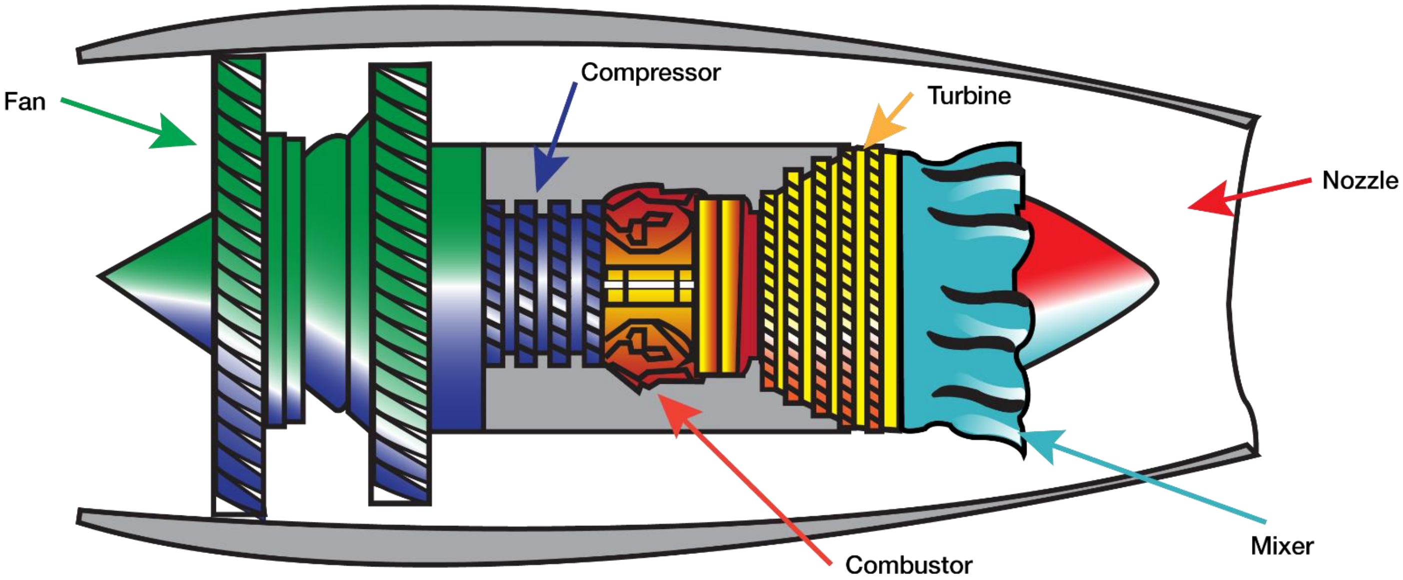
## Survey of Related Work

Previous research in the field of predictive maintenance has explored various machine learning models, data sources, and feature techniques to predict RUL. This project builds on existing literature by leveraging NASA's C-MAPSS dataset and deploying advanced machine learning algorithms to predict the RUL of turbofan engines. By comparing and evaluating different models, this project aims to bring new knowledge and methods to the field of predictive maintenance.

# Problem Statement

Develop a predictive maintenance model for turbofan engines using machine learning techniques. By analysing historical sensor data from turbofan engines, the goal is to predict the Remaining Useful Life (RUL) of engines, allowing for proactive maintenance scheduling and improved reliability.





# Literature Review

### Remaining Useful Life (RUL) Estimation of Turbofan Engines with Deep Learning Using Change-Point Detection Based Labelling and Feature Engineering" by Kıymet Ensarioğlu, Tülin İnkaya and Erdal Emel

This study proposes a novel prognostic approach for accurate remaining useful life (RUL) prediction of turbofan engines. Key innovations include feature engineering techniques like filtering, normalization, dimension reduction, and constructing a new "difference" feature from initial sensor data to capture early degradation. A piece-wise linear target labelling method using change point detection for engine-specific RUL labeling is employed. The approach leverages a hybrid 1D-CNN-LSTM deep learning model, with CNNs extracting spatial sensor features and LSTMs modeling temporal patterns. Extensive hyperparameter tuning optimized performance. Results demonstrated superior RUL prediction accuracy, especially using the "difference" feature, while reducing reliance on very deep neural architectures. Effective feature engineering diminished model complexity requirements.

### Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation by Abhinav Saxena, Member IEEE, Kai Goebel,

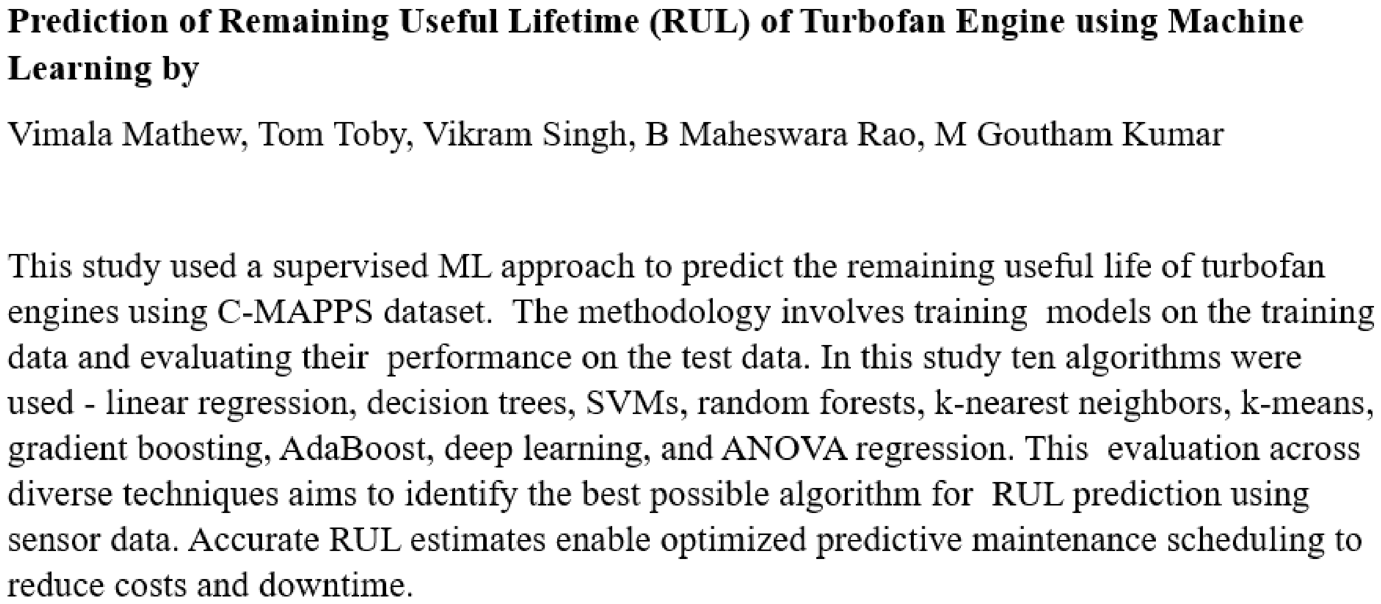
**Don Simon, Member, IEEE, Neil Eklund, Member IEEE**

The research paper "Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation" by Abhinav Saxena et al. presents a data-driven approach to model damage propagation in aircraft gas turbine engines for developing prognostic algorithms. The authors utilized NASA's C-MAPSS simulation to generate run-to-failure data for turbofan engines, modeling degradation by varying flow and efficiency parameters with an exponential rate of change. They incorporated factors like initial wear, process noise, and measurement noise to mimic real-world scenarios. The generated datasets, with varying operational conditions, health indices based on stall margins, and remaining useful life values, were used for the Prognostics and Health Management data competition. This work enabled the evaluation of prognostic techniques and contributed novel insights into predictive maintenance strategies for critical systems across aviation, aerospace, and industrial sectors. The health index, a metric representing overall system health, was computed as the minimum of superimposed operational margins for fan, high-pressure compressor (HPC), low-pressure compressor (LPC), and exhaust gas temperature (EGT) stall margins. These margins were calculated from response surfaces generated by varying the flow and efficiency parameters in the C-MAPSS simulation.

### Genetically Optimized Prediction of Remaining Useful Life by

**Shaashwat Agrawal, Sagnik Sarkar,Gautam Srivastava,Praveen Kumar Reddy Maddikunta,Thippa Reddy Gadekallu**

Accurate prediction of critical life (RUL) has become a significant challenge in many fields such as aviation, energy optimization and risk mitigation. The most traditional approaches are to use deep learning techniques such as Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) models, which have shown good results in capturing the human environment in connected products. However, these models rely on efficient operators such as Adam and stochastic gradient descent (SGD) for training, which may not produce consistent and good predictions, especially in situations where uncertainty can lead to large losses. To address this limitation, Agrawal et al. A new method is proposed to be combined with genetic methods to optimize the hyperparameters of LSTM and GRU models, especially training speed and batch size. The authors used a neural network architecture training method in which a group of individuals (samples) with initial weights go through an iterative process of training, competing, and changing. The safety of each individual is evaluated based on changes in accepted error, and the best individuals spread their genes to offspring. This approach focuses on tuning individual hyperparameters beyond the capability of the grid and potentially improving the consistency and accuracy of the RUL estimate. The method was evaluated on NASA's turbofan aircraft dataset, and the results showed a better genetic reconstruction than the LSTM and GRU models.



### Prediction of Remaining Useful Lifetime (RUL) of Turbofan Engine using Machine Learning by

Vimala Mathew, Tom Toby, Vikram Singh, B Maheswara Rao, M Goutham Kumar

This study used a supervised ML approach to predict the remaining useful life of turbofan engines using C-MAPPS dataset. The methodology involves training models on the training data and evaluating their performance on the test data. In this study ten algorithms were used - linear regression, decision trees, SVMs, random forests, k-nearest neighbors, k-means, gradient boosting, AdaBoost, deep learning, and ANOVA regression. This evaluation across diverse techniques aims to identify the best possible algorithm for RUL prediction using sensor data. Accurate RUL estimates enable optimized predictive maintenance scheduling to reduce costs and downtime.

# Methodology

- **Data Cleaning and Filtering:**

  1. Filtered out features with extremely low correlation (-0.001 to 0.001) to reduce noise and enhance model clarity.

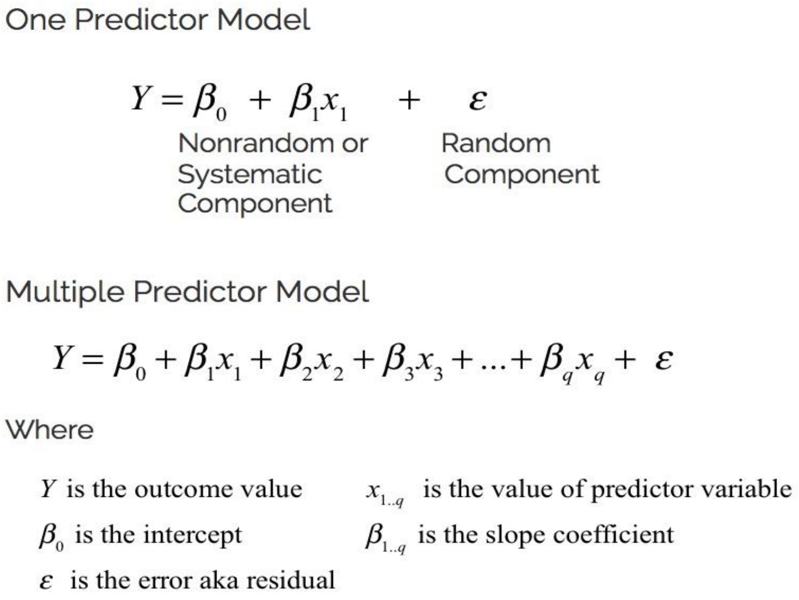
  2. Removed highly correlated features (correlation > 0.9) to mitigate multicollinearity and improve model generalization.

  3. Columns having 2 unique values with a ratio greater or equal to 0.95 will be removed from the dataset for a better training model.

- **Regression Methods:**

**Linear Regression** - It is one of the fundamental algorithms in machine learning and statistics used for predictive modeling. It is a supervised learning algorithm that models the relationship between a dependent variable (often called the target or output variable) and one or more independent variables (also known as predictors or features).

The main objective of linear regression is to find the best-fitting straight line (or hyperplane in higher dimensions) that minimizes the sum of squared differences between the predicted values and the actual values in the dataset.



One Predictor Model

$$Y = \underbrace{\beta_0 + \beta_1 x_1}_{\text{Nonrandom or Systematic Component}} + \underbrace{\varepsilon}_{\text{Random Component}}$$

Multiple Predictor Model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_q x_q + \varepsilon$$

Where

$Y$ is the outcome value

$x_{1..q}$ is the value of predictor variable

$\beta_0$ is the intercept

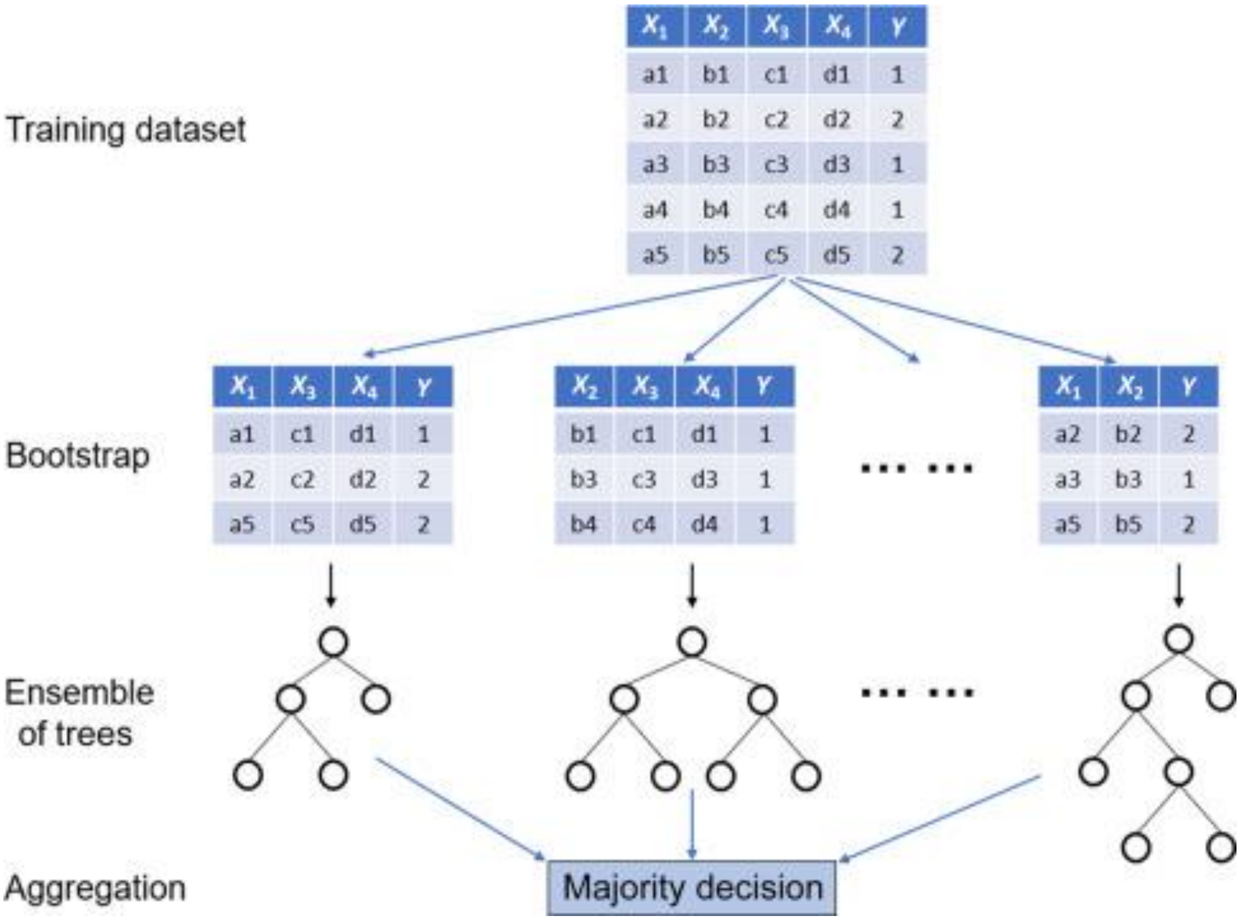
$\beta_{1..q}$ is the slope coefficient

$\varepsilon$ is the error aka residual

- **Random Forrest Regression:**

It is a powerful ensemble learning algorithm that extends the concept of decision trees to a collection of trees, known as a forest. It belongs to the family of tree-based methods and is widely used for both regression and classification tasks. In the context of regression, random forests are employed to predict a continuous numerical target variable based on one or more input features.

The key idea behind random forest regression is to construct multiple decision trees independently from random subsets of the training data and features. During the training process, each tree in the forest is built using a different bootstrap sample (a random sampling with replacement) of the original data. Additionally, at each node of a tree, a random subset of features is evaluated for the best split, rather than considering all available features. This randomization process helps to reduce the correlation between individual trees, thereby decreasing the overall variance and mitigating the risk of overfitting.



| $X_1$ | $X_2$ | $X_3$ | $X_4$ | $Y$ |
|---|---|---|---|---|
| a1 | b1 | c1 | d1 | 1 |
| a2 | b2 | c2 | d2 | 2 |
| a3 | b3 | c3 | d3 | 1 |
| a4 | b4 | c4 | d4 | 1 |
| a5 | b5 | c5 | d5 | 2 |

| $X_1$ | $X_3$ | $X_4$ | $Y$ |
|---|---|---|---|
| a1 | c1 | d1 | 1 |
| a2 | c2 | d2 | 2 |
| a5 | c5 | d5 | 2 |

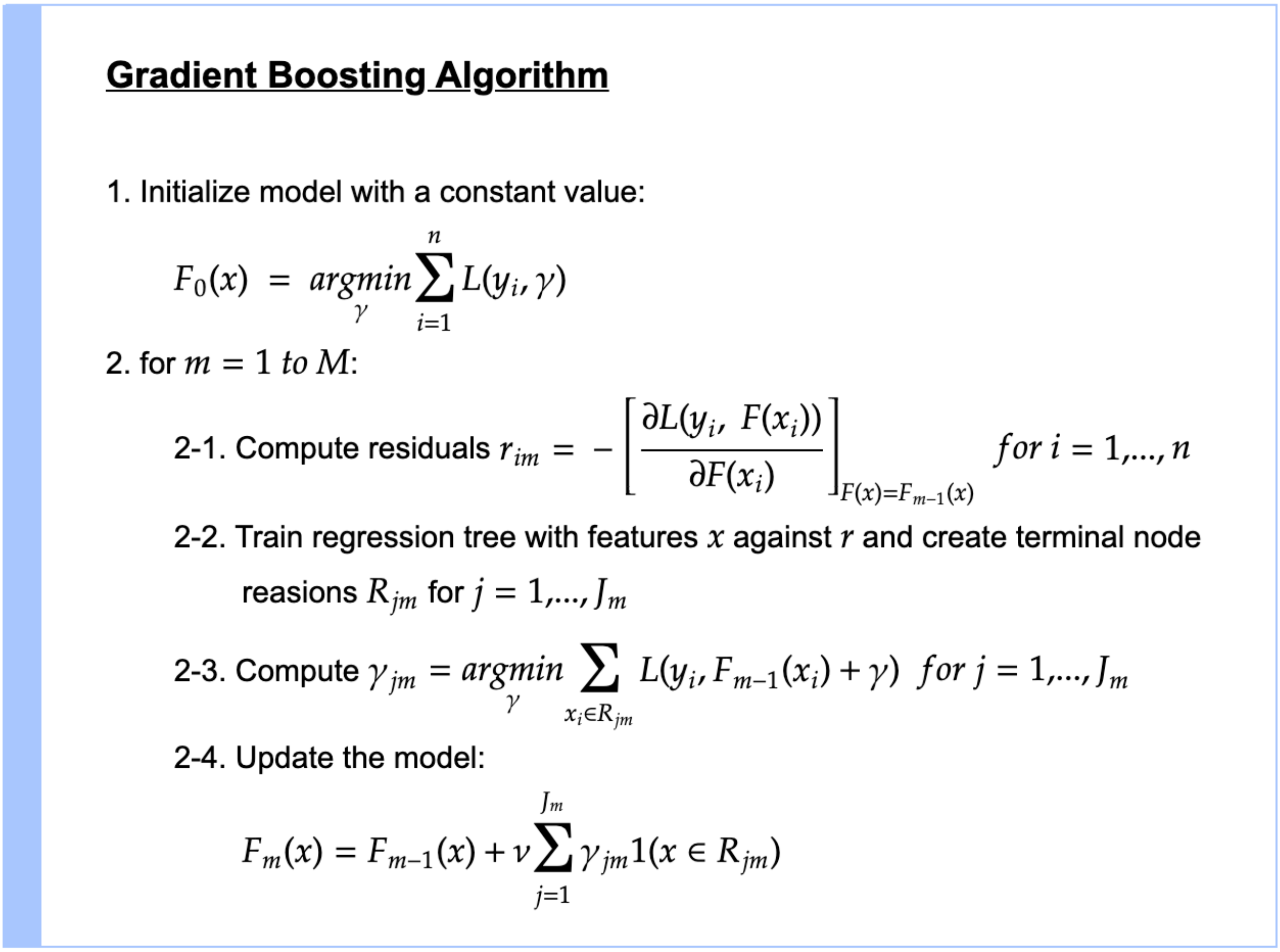
| $X_2$ | $X_3$ | $X_4$ | $Y$ |
|---|---|---|---|
| b1 | c1 | d1 | 1 |
| b3 | c3 | d3 | 1 |
| b4 | c4 | d4 | 1 |

| $X_1$ | $X_2$ | $Y$ |
|---|---|---|
| a2 | b2 | 2 |
| a3 | b3 | 1 |
| a5 | b5 | 2 |

- **Gradient Boosting:**

It is a powerful machine learning technique that combines multiple weak predictive models (usually decision trees) to create strong predictive models. It is an ensemble learning method that builds models sequentially, where each new model is trained to improve the error of the previous model.

The main idea of gradient boosting is to reduce redundancy by training weak models on the remnants of previous models. Each time a new weak model is added to the assembly, its predictions are weighted by how much they improve the performance of the overall model.



## Gradient Boosting Algorithm

1. Initialize model with a constant value:

$$F_0(x) = \underset{\gamma}{argmin} \sum_{i=1}^{n} L(y_i, \gamma)$$

2. for $m = 1 \; to \; M$:

2-1. Compute residuals $r_{im} = -\left[\frac{\partial L(y_i,\ F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad for\ i = 1,\ldots,n$

2-2. Train regression tree with features $x$ against $r$ and create terminal node reasions $R_{jm}$ for $j = 1,\ldots,J_m$

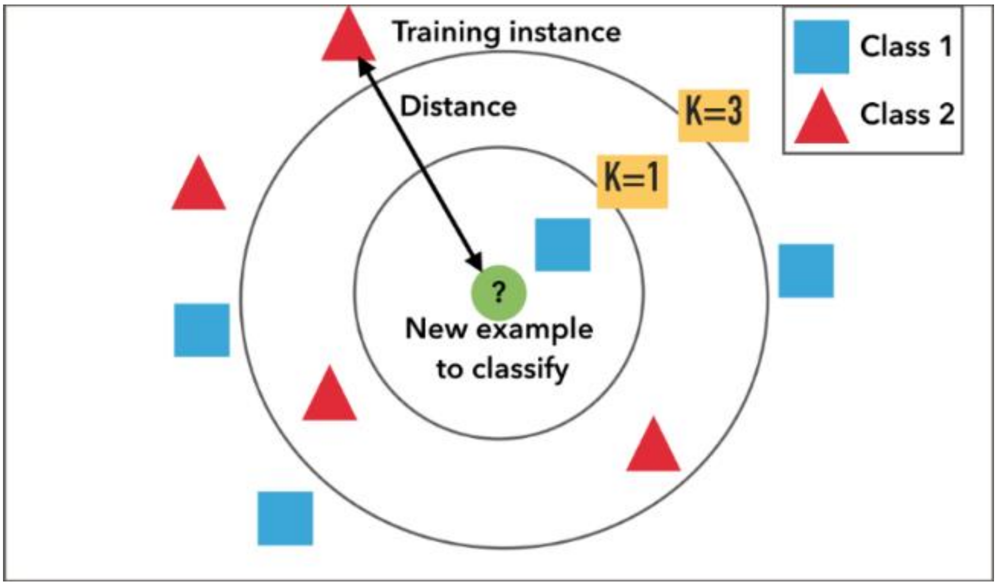
2-3. Compute $\gamma_{jm} = \underset{\gamma}{argmin} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \quad for\ j = 1,\ldots,J_m$

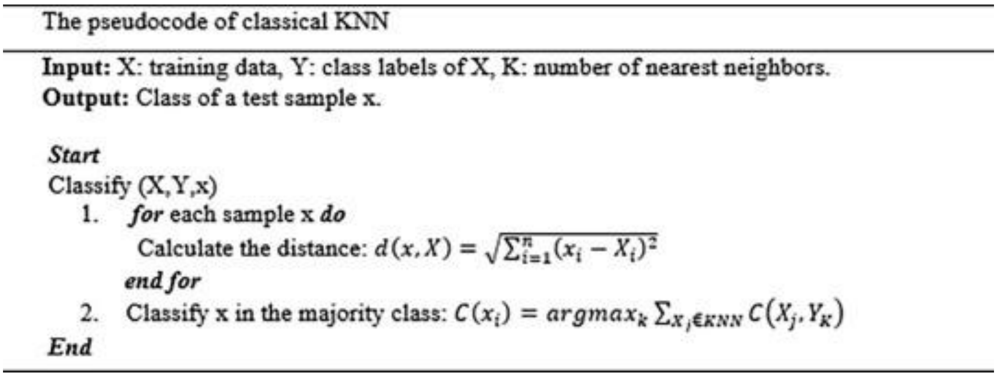
2-4. Update the model:

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} 1(x \in R_{jm})$$

- **K Nearest Neighbours**:

In K-Nearest Neighbors (KNN) methodology, we classify or regress data points based on the majority of their k nearest neighbors in the feature space. We begin by calculating the distances between the test data and all other data points in the training set. Next, we select the k nearest neighbors and determine their weighted average (for regression) or majority class (for classification) to predict the outcome for the test data. This non-parametric algorithm's simplicity and flexibility make it effective for various datasets and applications.





The pseudocode of classical KNN

**Input:** X: training data, Y: class labels of X, K: number of nearest neighbors.
**Output:** Class of a test sample x.

```
Start
Classify (X,Y,x)
  1.  for each sample x do
        Calculate the distance: $d(x, X) = \sqrt{\sum_{i=1}^{n}(x_i - X_i)^2}$
      end for
  2.  Classify x in the majority class: $C(x_i) = argmax_k \sum_{X_j \in KNN} C(X_j, Y_K)$
End
```

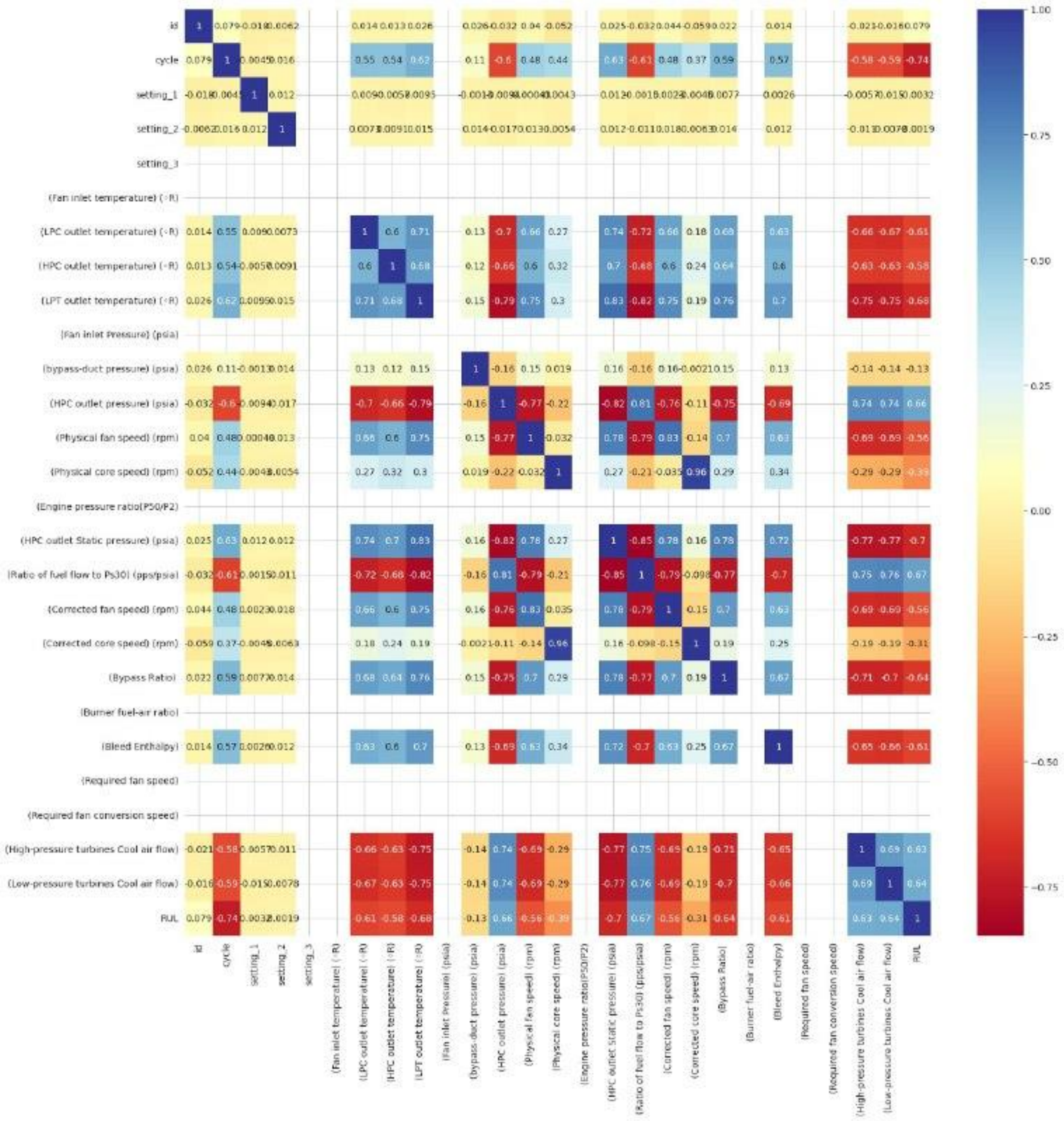
# Experiments

## Details of Dataset :

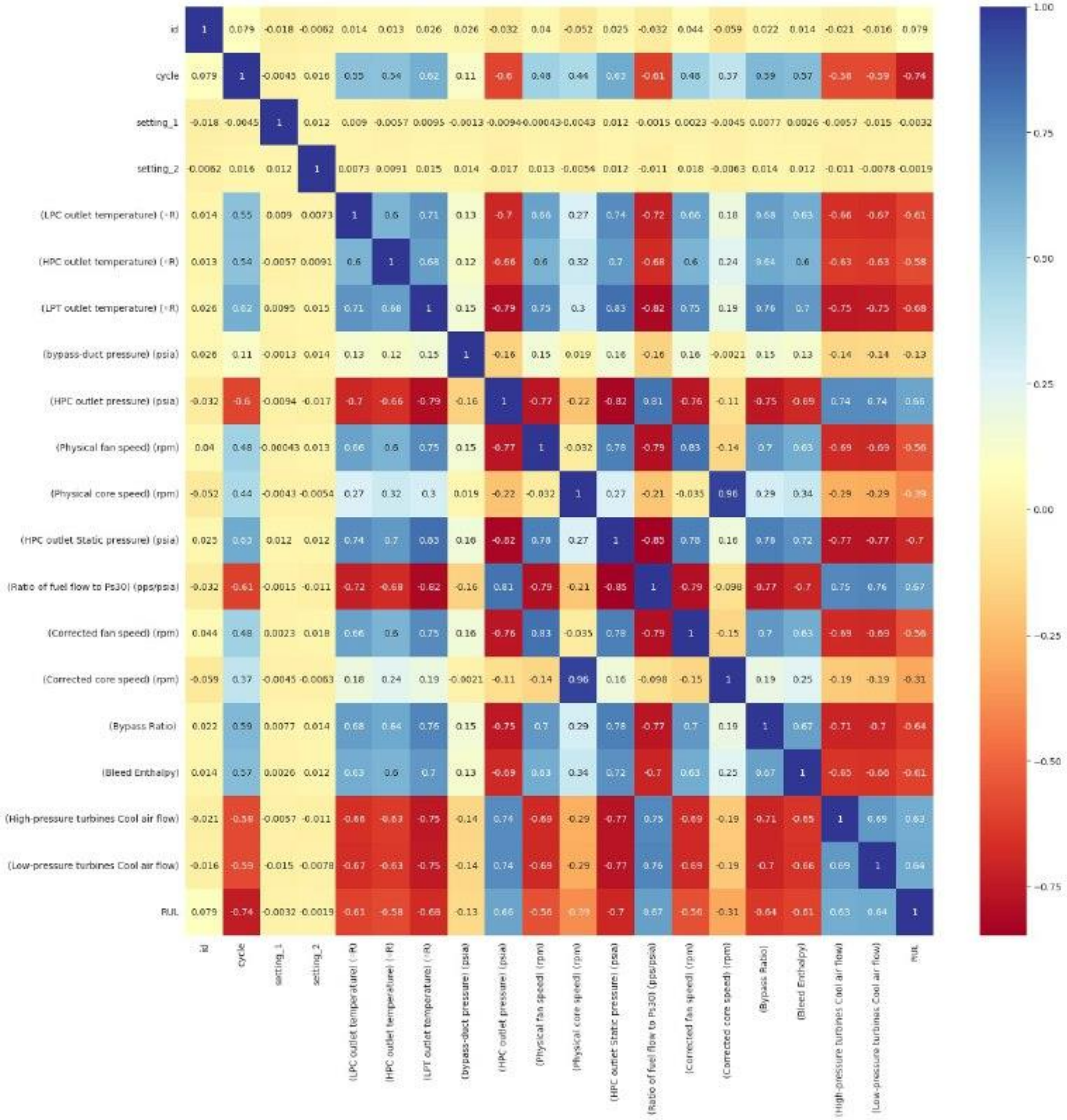
## Data filteration and cleaning :

- Filtered out features with extremely low correlation (-0.001 to 0.001) to reduce noise and enhance model clarity.

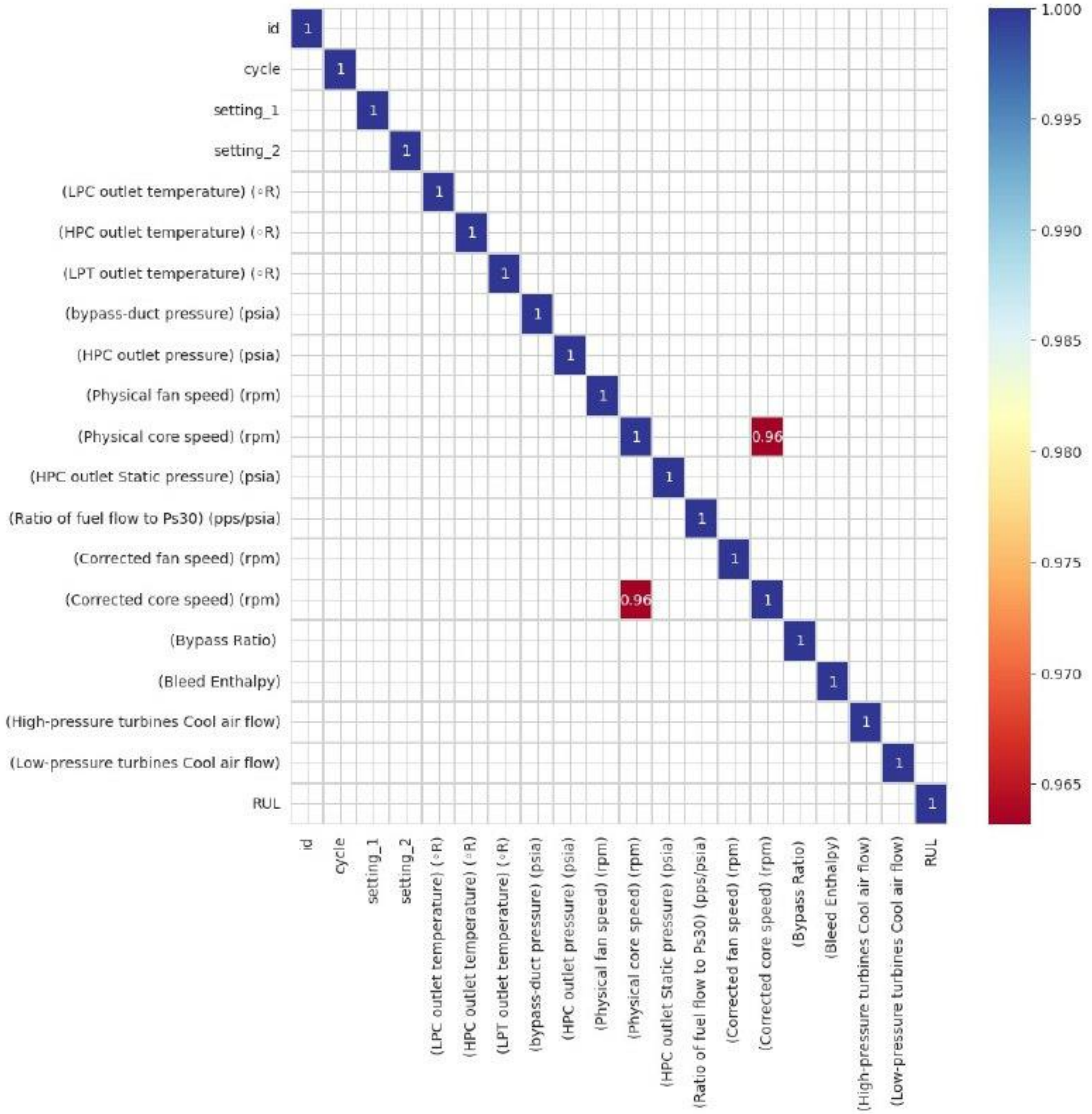
Before removing features with low correlations -



After removing features with low correlations -



-Removed highly correlated features (correlation > 0.9) to mitigate multicollinearity and improve model generalization.



-Columns having 2 unique values with a ratio greater or equal to 0.95 will be removed from the dataset for a better training model.

# Results and Discussion

## Visualization of results:

- Visualization and comparison of actual labels and model predicted labels

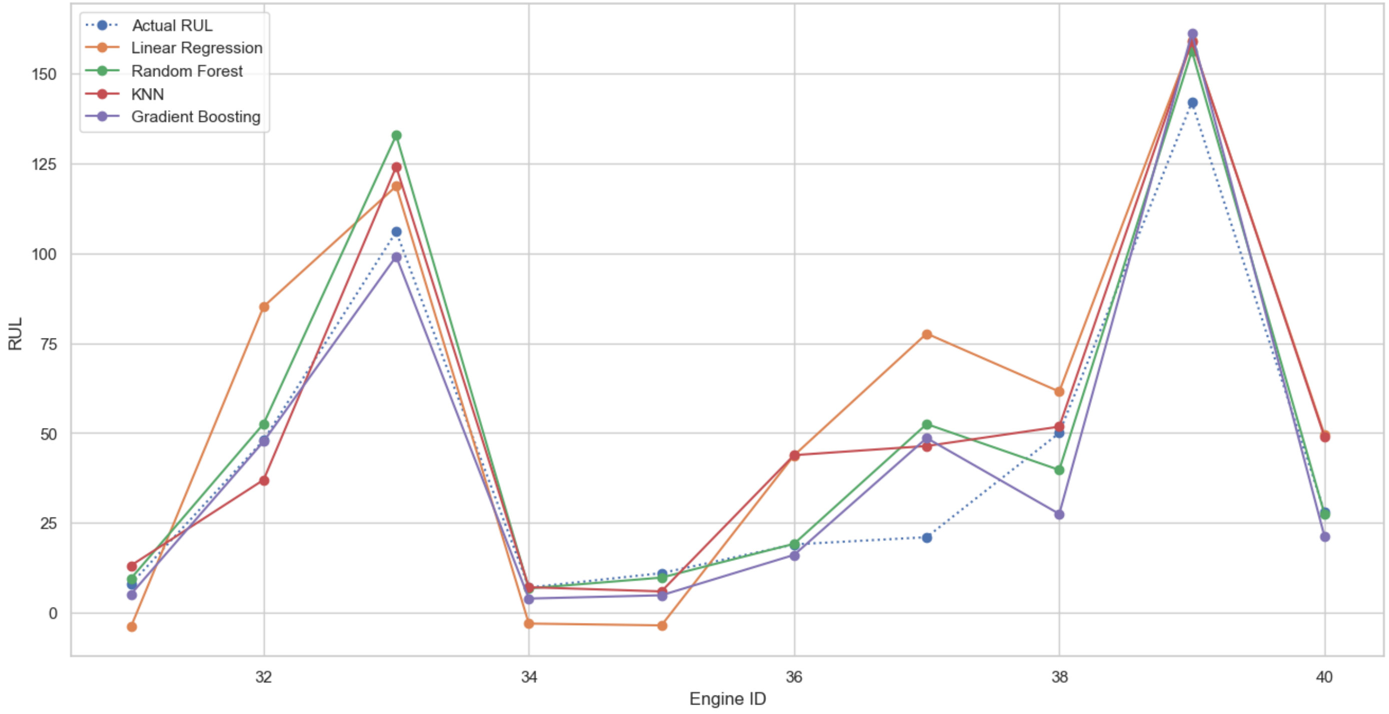
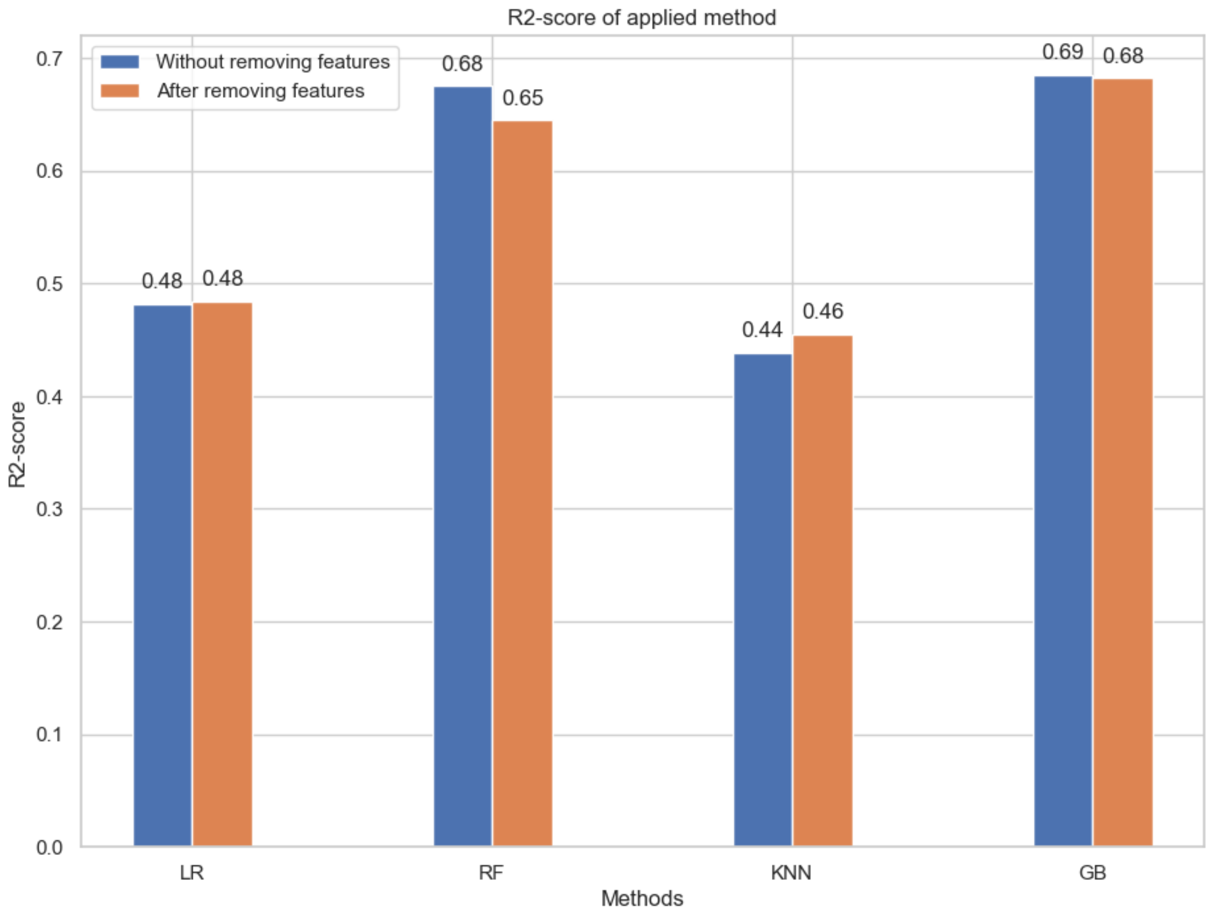


*Image :- here we have shown image containing actual labels with model predicted labels*

➔ **R2 – score for every applied method :**

| S. No. | Applied Model | R2 score | |
|---|---|---|---|
| **1.** | **Linear Regression** | W/O removing features | **0.48** |
| | | Removing features | **0.48** |
| **2.** | **Random Forest** | W/O removing features | **0.68** |
| | | Removing features | **0.65** |
| **3.** | **K Nearest Neighbor** | W/O removing features | **0.44** |
| | | Removing features | **0.46** |
| **4.** | **Gradient Boost** | W/O removing features | **0.69** |
| | | Removing features | **0.68** |



➔ RMSE for every applied method-

| S. No. | Applied Model | RMSE | |
|---|---|---|---|
| **1.** | **Linear Regression** | W/O removing features | **29.91** |
| | | Removing features | **29.84** |
| **2.** | **Random Forest** | W/O removing features | **23.67** |
| | | Removing features | **24.75** |
| **3.** | **K Nearest Neighbor** | W/O removing features | **31.12** |
| | | Removing features | **30.66** |
| **4.** | **Gradient Boost** | W/O removing features | **23.30** |
| | | Removing features | **23.42** |

**FINAL RESULT: -**

After all work, implementation and from above visualisation it can be concluded that **GRADIENT BOOST** is the most suitable method for the RUL Prediction Project with highest R2 – score (0.68) and lowest RMSE (23.30).

# Conclusion and Future Scope

Conclusion:

In summary, this project aims to create a predictive maintenance model for turbofan engines using machine learning techniques. Through data quality management, feature selection, and testing with a variety of regression methods, including linear regression, random forest, K-nearest neighbour (KNN), and gradient boosting, we gain a better understanding of prediction performance. The results showed that gradient boost emerged as the most suitable method for RUL estimation, with the highest R2 score of 0.68 and the lowest RMSE value of 23.3. This demonstrates its effectiveness in predicting the service life of the turbofan engine, ensuring good control, and improving reliability. Additional research and development to improve our predictive model.

Future Scope:

Looking ahead, there are exciting opportunities to further improve our predictive maintenance model for turbofan engines:

- ➔ Broadening Data Sources: By including a wider range of data types and sources, such as operational data and environmental factors, we can enhance the model's accuracy and reliability.

- ➔ Refining Model Parameters: Fine-tuning the settings and configurations of our model can lead to better performance, ensuring that it can adapt to different engine types and operating conditions.

- ➔ Real-time Application: Developing ways to integrate our model into real-time monitoring systems would enable immediate action based on predictive insights, leading to more proactive maintenance practices, and minimizing downtime

- ➔ Exploring New Techniques: Investigating emerging techniques like deep learning and reinforcement learning could uncover new approaches for RUL prediction, potentially improving our model's effectiveness and efficiency.

These steps promise to not only refine our model for turbofan engine maintenance but also pave the way for predictive maintenance advancements across various industries, ultimately saving costs and enhancing safety.

# References

Data set: CMAPSS Jet Engine Simulated Data | NASA Open Data Portal. (2022, June 30). https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6/about_data

Literature Survey:

Saxena, A., Goebel, K., Simon, D., Eklund, N., Research Institute for Advanced Computer Science, NASA Ames Research Center, NASA Glenn Research Center, & GE Global Research. (2008). Damage propagation modeling for Aircraft Engine Prognostics. In Prognostics and Health Management (PHM) Data Competition [Journal-article]. https://ntrs.nasa.gov/api/citations/20090029214/downloads/20090029214.pdf

Agrawal, S., ∗, Sarkar, S., ∗, Srivastava, G., †, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India, Department of Mathematics and Computer Science, Brandon University, Manitoba, Canada, & School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India. (2021). Genetically optimized prediction of remaining useful life [Journal-article]. arXiv. https://research.vit.ac.in/pdf/postprint-genetically-optimized-prediction-of-remaining-useful-life

lEnsarioğlu, K., İnkaya, T., & Emel, E. (2023). Remaining Useful Life Estimation of Turbofan Engines with Deep Learning Using Change-Point Detection Based Labeling and Feature Engineering. Applied Sciences, 13(21), 11893.

# Appendix

Here is the Google Colab File for the Code - Code File