Project title: "Prediction of the Remaining Useful Life (RUL) of aircraft engine"

Course - DS-203 Machine Learning and Artificial Intelligence with Python

22nd April 2023

Team (AeroML) members:

Mr. Vijay Kothari
Dr. Debabrata Adhikari
Dr. Keshava Kumar S

Faculty - Professor Shashikumar Ganesan

Teaching Assistant: Thivin Anandh, Aravind Vallal

Centre for Continuing Education

Indian Institute of Science, Bangalore, INDIA

GitHub link: https://github.com/VijayAerospace/ML_IISC_CCE_RUL_FD003_V1_2023

Code name: ML_IISC_CCE_RUL_FD003_V1_2023.ipynb

Index

Acknowledgement

- We would like to express our gratitude to our Professor **Shashikumar Ganesan** and the **TA's** for their guidance, feedback, and support throughout the course.
- Their valuable insights and suggestions have been instrumental in helping us develop skills and knowledge in machine learning. Thank you for sharing your expertise and for being available to answer our questions.
- Also, thank you Centre for Continuing Education, Indian Institute of Science, Bangalore, INDIA for this opportunity.

AeroML Team Members

Mr. Vijay Kothari (MSc Aerospace Vehicle Design, Cranfield University, UK)
GKN Aerospace, Bangalore, INDIA
vijayaeronautics@gmail.com

Responsibility: Project Leader, developed project proposal, initial and final presentation, Python code: data - loading, pre-processing, clean-up and visualization, feature importance analysis and feature selection, RUL clipping, ML modelling, final code clean-up/maintenace, interpretation of results set-up GitHub repository, GitHub read me, context for IEE report write up, etc.

Dr. Debabrata Adhikari (PhD in Aerospace Eng, IISc Bangalore, IN)
Postdoctoral Researcher, Bernal Institute, University of Limerick, Ireland
deb1729@gmail.com

Responsibility: IEEE report write up, Python code - ML modelling, Grid search CV, final code clean-up, different ML plots, interpretation of results, code flowchart, update final ppt, commit, pull, push to GitHub etc.

Dr. Keshava Kumar S (PhD in Aerospace Engineering, IISc Bangalore, IN)
Blades Technology, Siemens Gamesa Renewable Energy
keshav@iitbombay.org

Responsibility: IEEE report write up, brainstorming of ML/code ideas, conceptualization etc.

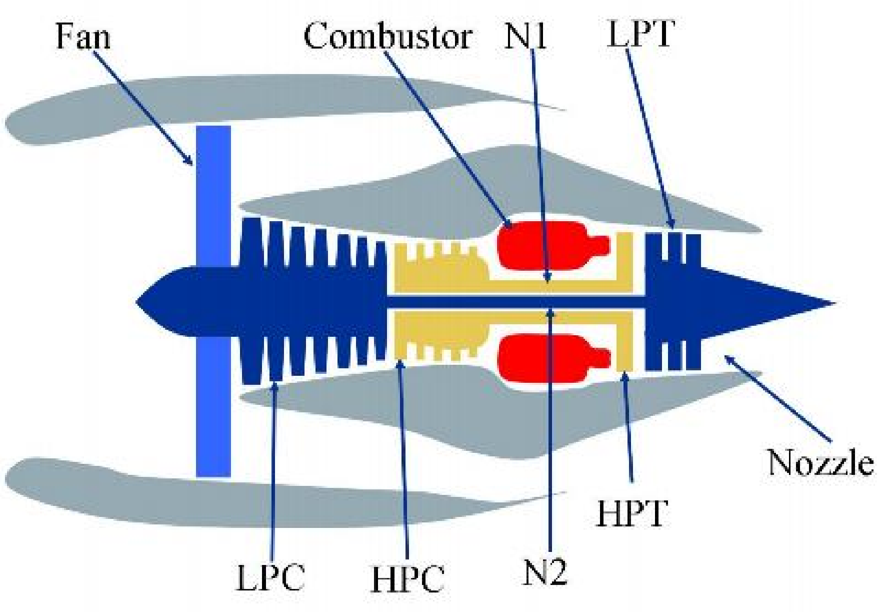
Key Success Factors for Project Team Collaboration:

- Efficient communication via a dedicated "WhatsApp" group.
- Consistent use of a "Google Meet“ for virtual meetings with different continent.
- Shared "Google Drive" and GitHub repository to track commit history
- Use of “Overleaf" online latex to writeup IEEE documents.

Acronyms

Abbreviation	Meaning
RUL	Remaining Useful Life
ML	Machine Learning
HPC	High Pressure Compressor
C-MAPPS	Commercial Modular Aero-Propulsion System Simulation
NaN	Not a Number
CORR	Correlation regression
RF	Random Forest regression
LR	Linear regression
PR	Polynomianl Regression
GB	Gradient Boosting regression
RMSE	Root Mean Square Error
R2	Regression error metric

Problem Description



Simplified diagram of aircraft engine, Ref [2]

High-Pressure Turbine (HPT)
Low-Pressure Turbine (LPT)
High-Pressure Compressor (HPC)
Low-Pressure Compressor (LPC)

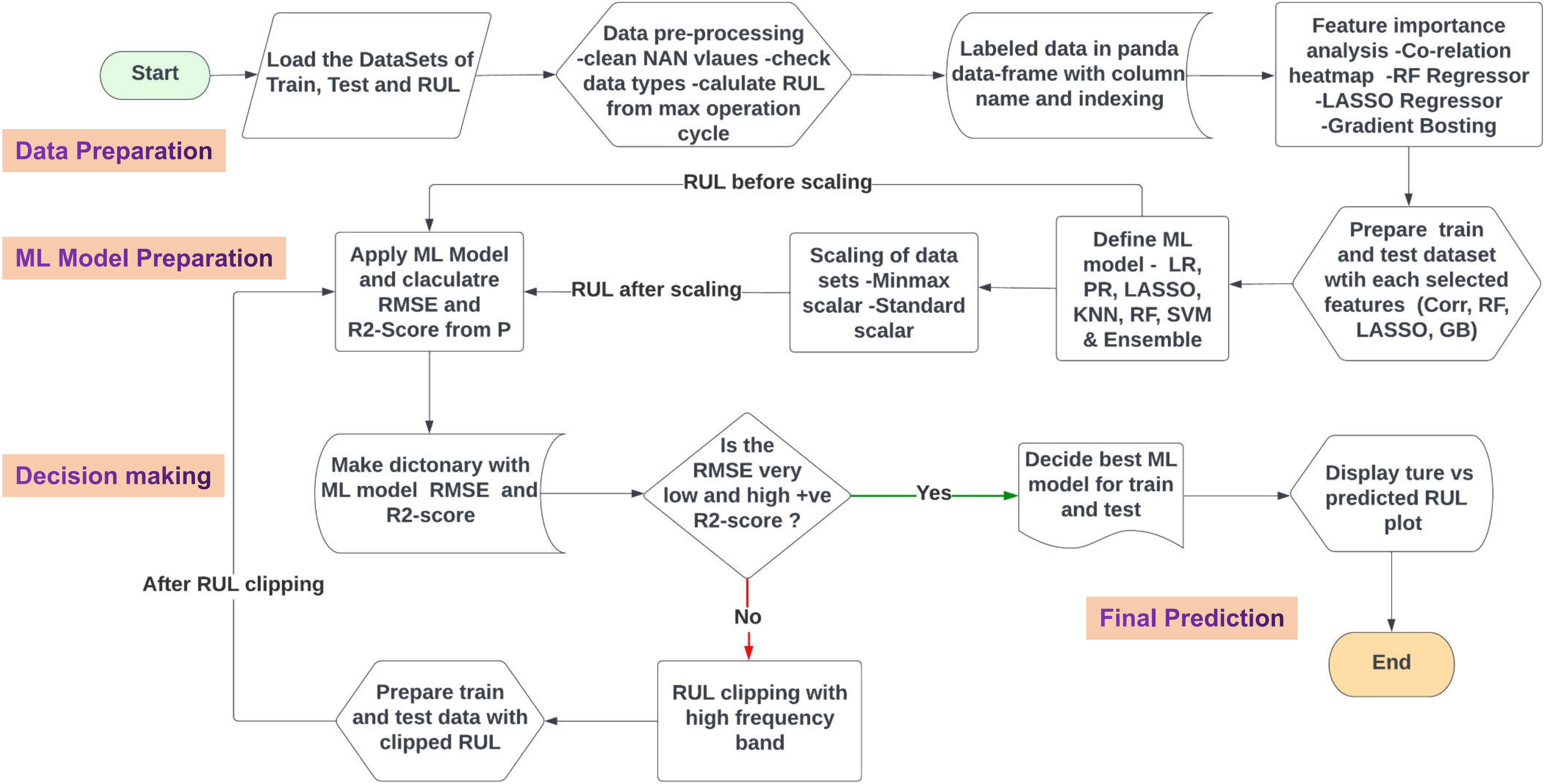
- **Problem statement:** During their lifetime, aircraft engine components are susceptible to degradation, which affects directly their reliability and performance.

- **Objective:** The objective of the project is to develop a machine learning model that can predict the number of remaining useful life before failure for each engine in the test set. This involves analyzing the training data to identify patterns in the sensor measurements and operational settings that are associated with engine failure and using this information to make accurate predictions for the test set.

- **Conditions:** ONE (Sea Level)

- **Fault Modes**: TWO (HPC Degradation, Fan Degradation)

Flowchart of the ML project



Data collection and understanding (FD003)

NASA's C-MAPSS Tool (**Source**)[1]

train_FD003.txt

test_FD003.txt

RUL_FD003.txt

- Dataset “train_FD003.txt” is used for training ML model
- It contains time-series data from multiple sensors for 100 turbofan engines.
- The goal is to train a machine learning model to predict the RUL of new engines based on the sensor readings.

- Dataset “test_FD003.txt” is used for testing the trained ML model.
- It contains time-series data from multiple sensors for a new set of 100 turbofan engines.

- Dataset “RUL_FD003.txt” contains the actual RUL of from test data.
- It contains true RUL of 100 turbofan engine.
- The goal is to compare the ML model predicted RUL with this true RUL and measure the RMSE and R2-Score goodness of fit.

[1] Data source: https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6

Data Preprocessing and Cleaning

Load train dataset (train_FD003.txt) from a text file using the panda's library read_csv() function

	0	1	2	3	4	5	6	7	8	9	...	18	19	20	21	22	23	24	25	26	27
0	1	1	-0.0005	0.0004	100.0	518.67	642.36	1583.23	1396.84	14.62	...	8145.32	8.4246	0.03	391	2388	100.0	39.11	23.3537	NaN	NaN
1	1	2	0.0008	-0.0003	100.0	518.67	642.50	1584.69	1396.89	14.62	...	8152.85	8.4403	0.03	392	2388	100.0	38.99	23.4491	NaN	NaN
2	1	3	-0.0014	-0.0002	100.0	518.67	642.18	1582.35	1405.61	14.62	...	8150.17	8.3901	0.03	391	2388	100.0	38.85	23.3669	NaN	NaN
3	1	4	-0.0020	0.0001	100.0	518.67	642.92	1585.61	1392.27	14.62	...	8146.56	8.3878	0.03	392	2388	100.0	38.96	23.2951	NaN	NaN
4	1	5	0.0016	0.0000	100.0	518.67	641.68	1588.63	1397.65	14.62	...	8147.80	8.3869	0.03	392	2388	100.0	39.14	23.4583	NaN	NaN

5 rows × 28 columns

Data cleaning process before train and test ML model

- Clean "NaN (not a number) " value from the last two column 26 and 27 and check for others
- removing unnecessary columns, scaling the data
- check for null values, empty column
- look at head and tail data
- look at "Descriptive statistics"

```
0.03 is not a number
0ž.03 is not a number

print(train_data[20].unique())

['0.03' '0ž.03']
```

- Converted the column to float data type using the astype() function of the pandas Series.

Data Visualization and RUL calculation

- **Renamed the column names of dataset**

	engineID	cycles	operational_setting_1	operational_setting_2	operational_setting_3	sensor_1	sensor_2	sensor_3	sensor_4	sensor_5	...	sensor_13	senso
0	1	1	-0.0005	0.0004	100.0	518.67	642.36	1583.23	1396.84	14.62	...	2388.01	814
1	1	2	0.0008	-0.0003	100.0	518.67	642.50	1584.69	1396.89	14.62	...	2388.03	815
2	1	3	-0.0014	-0.0002	100.0	518.67	642.18	1582.35	1405.61	14.62	...	2388.00	81

- **RUL = (maximum cycle of an engine) - (current cycle of the engine)**

```
#Calculating the RUL based on the available data in the data set
train_data['RUL'] = train_data.groupby('engineID')['cycles'].transform(max) - train_data['cycles']
```

- **Describe datasets**

```
train_data.loc[:,['engineID','cycles']].describe()
```

	engineID	cycles
count	24720.000000	24720.000000
mean	48.631877	139.077063
std	29.348985	98.846675
min	1.000000	1.000000
25%	23.000000	62.000000
50%	47.000000	124.000000
75%	74.000000	191.000000
max	100.000000	525.000000

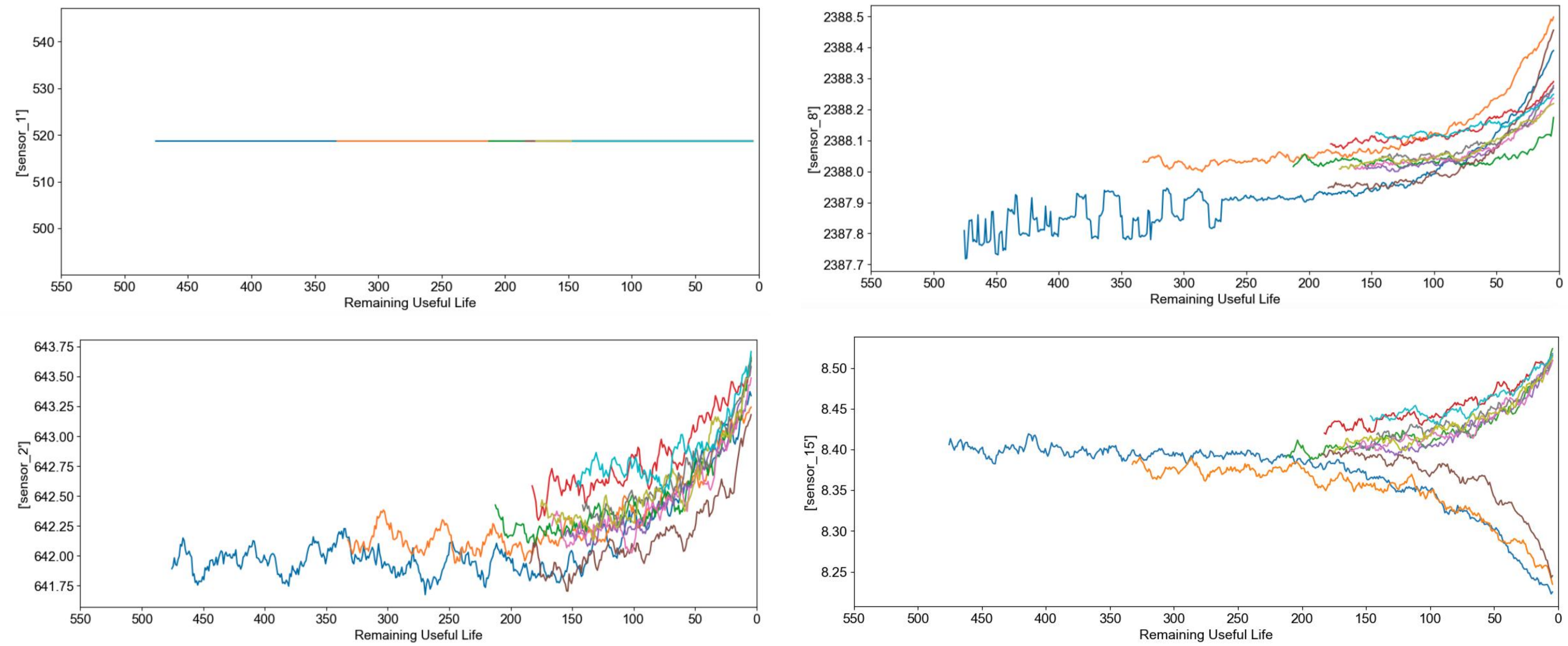
Takeaway from the descriptive statistics:

- There are 24,720 rows of data in this dataset.
- The engine numbers start at 1 and end at 100.
- The mean and quantiles don’t align neatly means each engine has a different maximum cycles and thus a different number of rows.
- The maximum value for ‘engine' is 100 and for 'cycles' is 525.

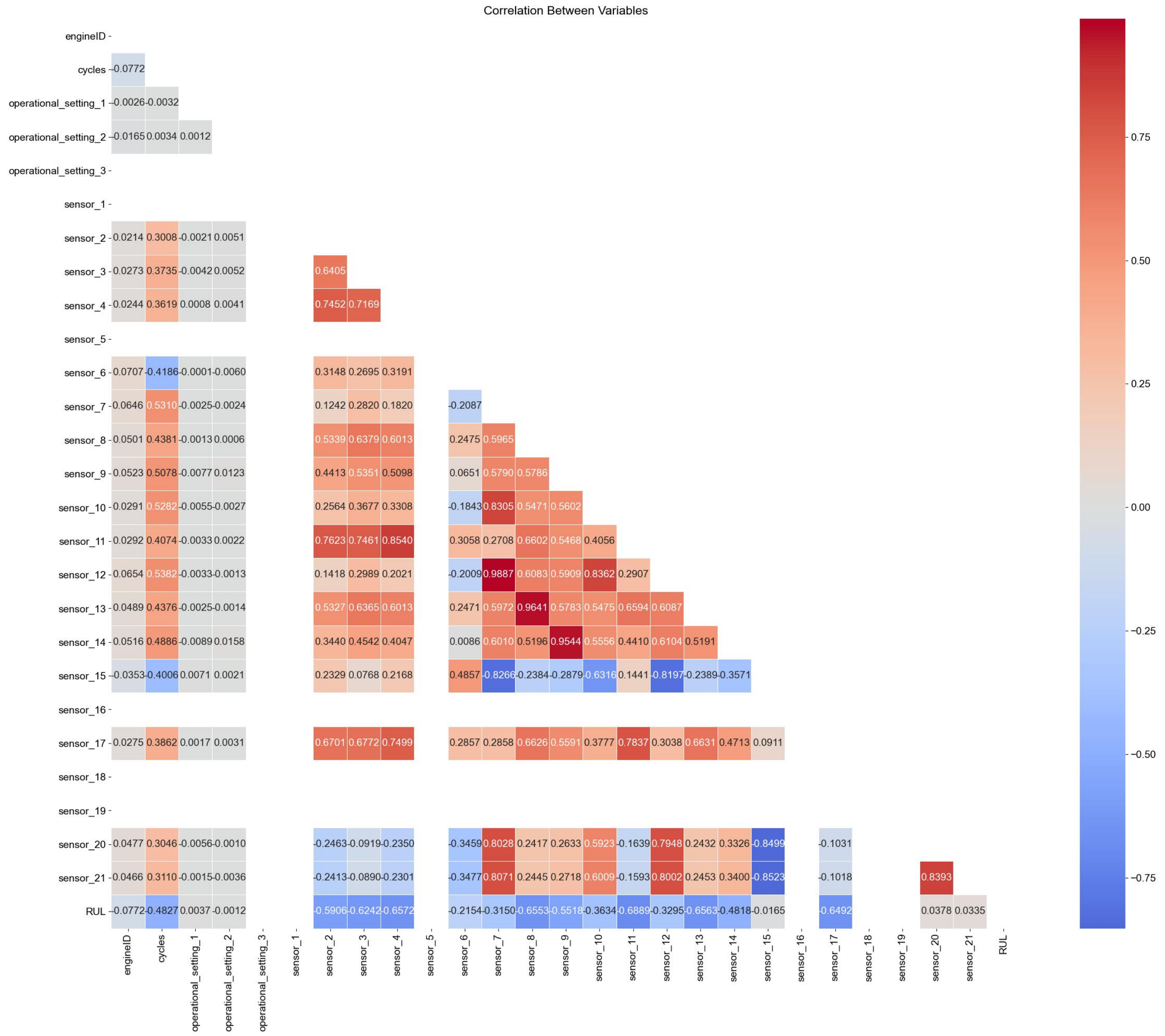
Sensor Signal Visualization for Predictive Maintenance Analysis

Types of sensors signal with respect to RUL for every 10th turbofan engine:

- Time series signal plots for sensors 1 remain constant with RUL
- Sensor 2, 8 and 15 shown nonlinear behavior with RUL



Feature Importance analysis



Heatmap correlation

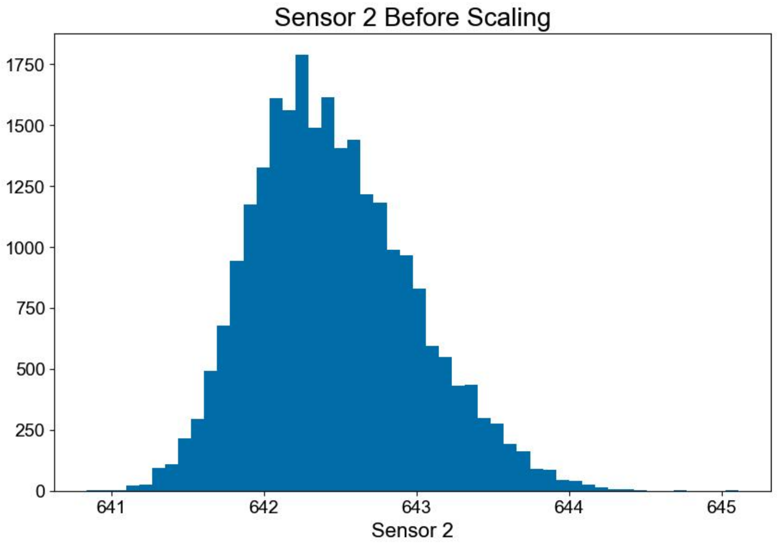
- The white space for the sensors do not have any co-relation with RUL.
- A positive correlation coefficient indicates that as one variable increases, the other variable also increases.
- The negative correlation coefficient indicates the other way.

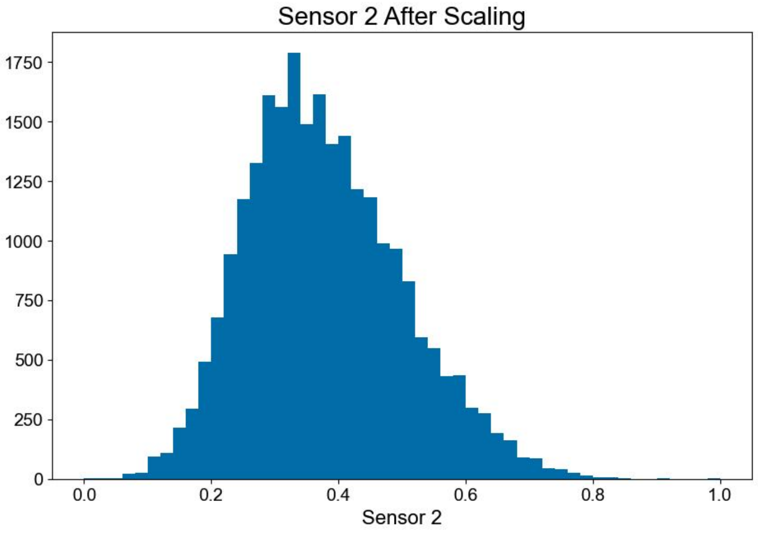
Feature Importance analysis

- Number of important features from different methods are (based on relation with RUL)
  - **Correlation (CORR): 13 (correlation threshold 0.1)**
  - **Random Forest (RF): 12**
  - **Lasso: 18**
  - **Gradient Boosting (GB): 17**
- The feature importance ranking is calculated based on the predictive importance for RUL determined by different methods and the ranks are stored in the merged dataframe.
- Four iterations of the ML model were performed using different feature selections, and the error metric results revealed that the features chosen through the **Random Forest regression** feature importance analysis yielded the **best performance**.
- The computational time for running the program with different feature selection is shown below.

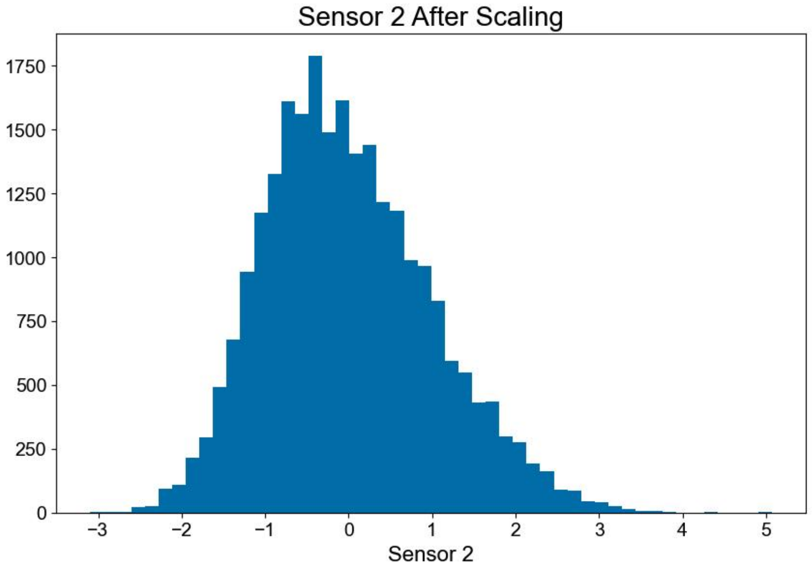
Features selected from	*Code Running Time (in [s])
RF	554
CORR	606
GB	1663
LASSO	2727

Types of feature scaling – MinMax Scaler and Standard Scaler





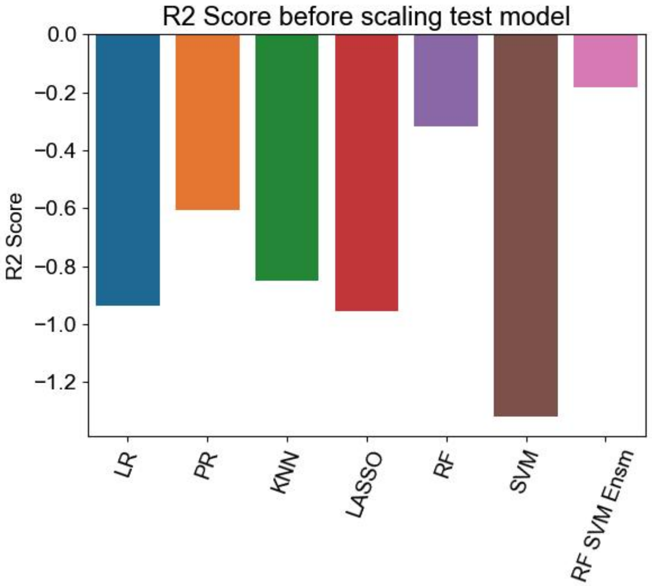
MinMax Scaler



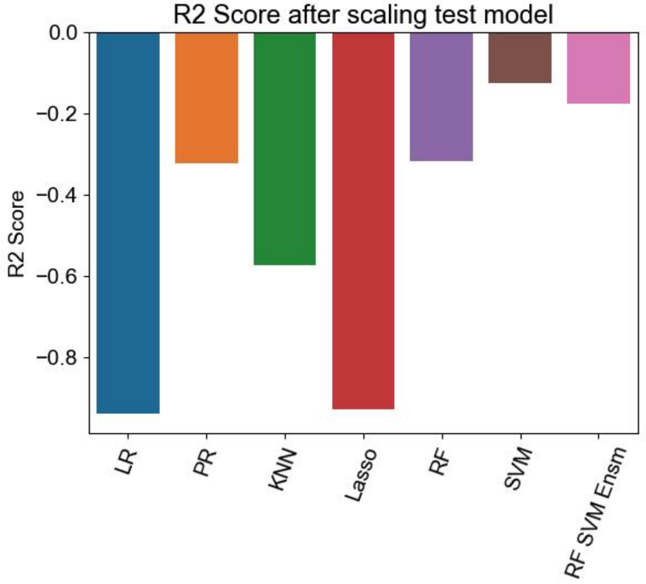
Standard Scaler

- MinMax Scaler – Scaled the data sets into [0, 1] close interval
- Standard Scaler - Normalize the data sets into probability scaling between min and max values
- Evaluate RMSE and R2-score before and after scaling.
- Datasets follow a normalized probability distribution from both the scaling
- Prefer MinMax Scaler method as the distribution above is in the form of probability distribution.
- We evaluated our models on both the original unscaled data and data scaled using MinMax Scaler and Standard Scaler, we observed that the R2 score improved significantly on the scaled data while the RMSE remained relatively unchanged, indicating that scaling enhanced the **performance and stability** of the **models without significantly affecting prediction errors.**

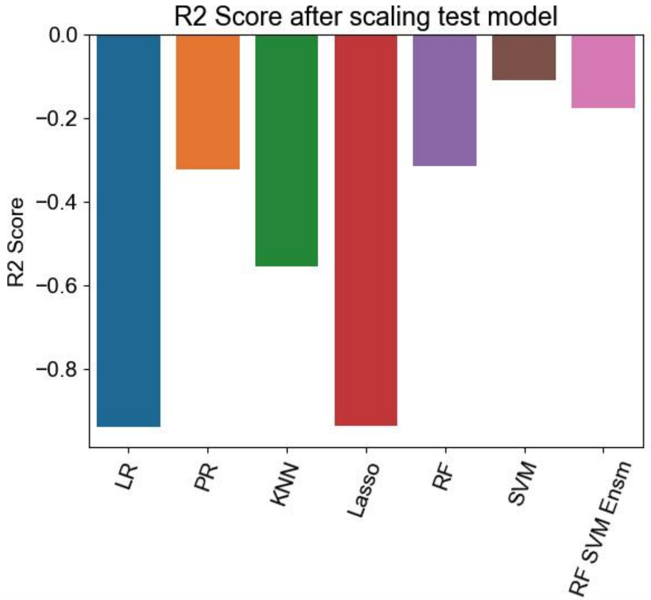
Feature Scaling - MinMax Scaler and Standard Scaler

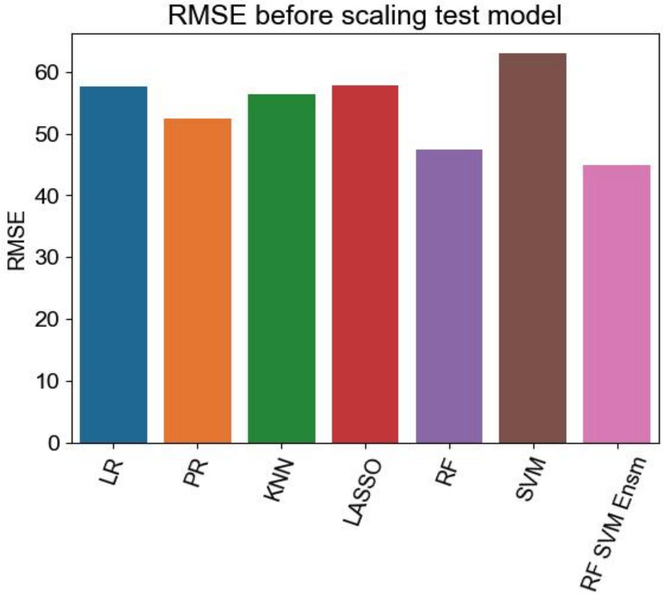


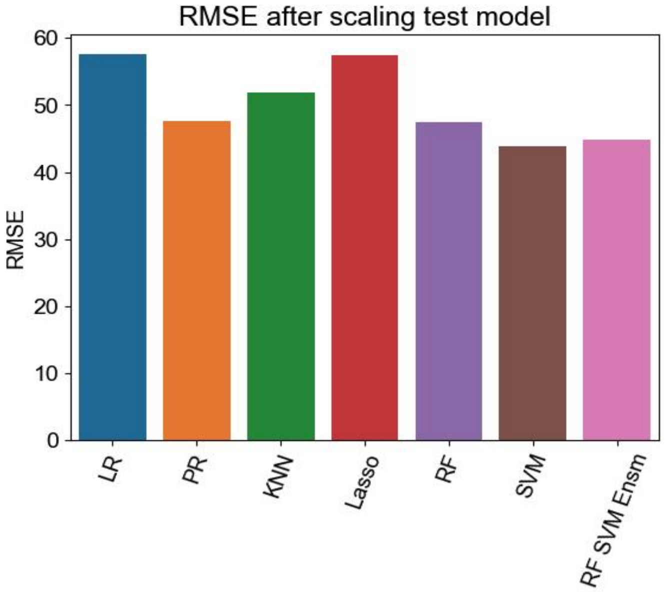
MinMax Scaler

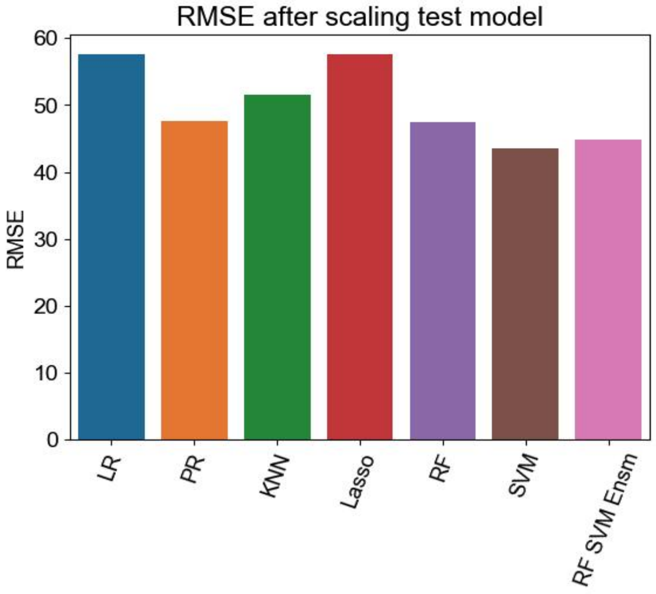


Standard Scaler









- Both StandardScaler and MinMaxScaler have shown similar performance in conjunction with the tested regression algorithms.
- R2-score has increase after scaling

Proposed ML Model

- **ML Methods used**:

  Linear Regression (LR),

  Polynomial Regression (PR) – 4^{th} order polynomial,

  Random Forest (RF),

  Support Vector Machine (SVM),

  Random Forest Support Vector

  Machine (RF SVM) Ensemble,

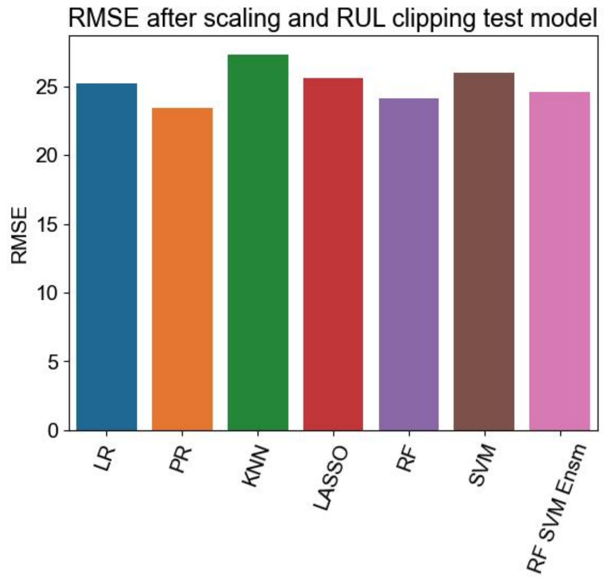
  K Nearest Neighbours (KNN) – (n_neighbors =5)

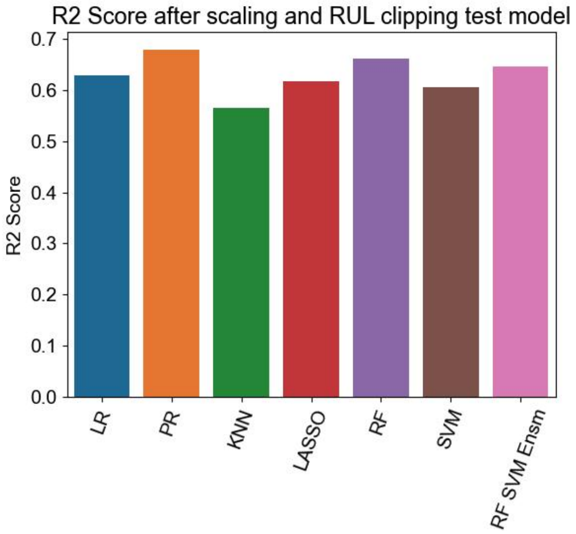
  Least Absolute Shrinkage and Selection Operator (LASSO).

  Error metric used: Root Mean Square Error (RMSE) and R2 Score.

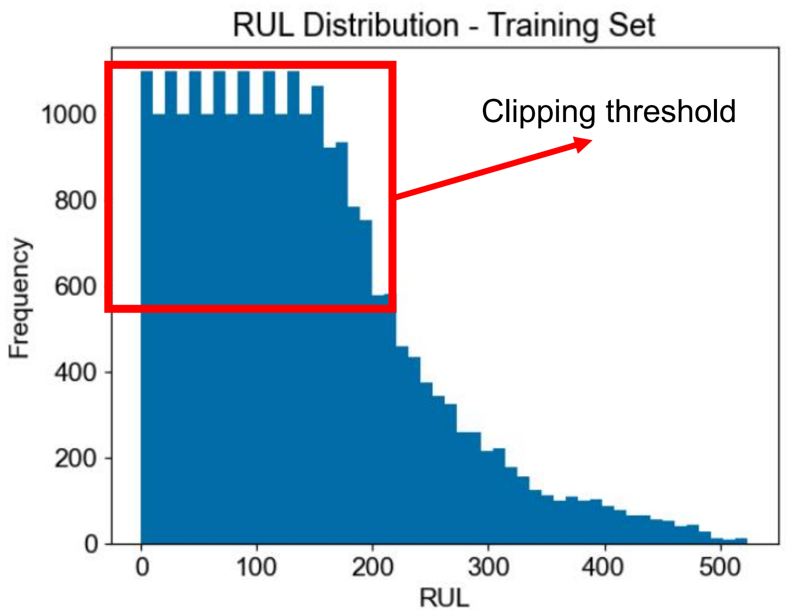
Grid search CV (for hyperparameters tuning)

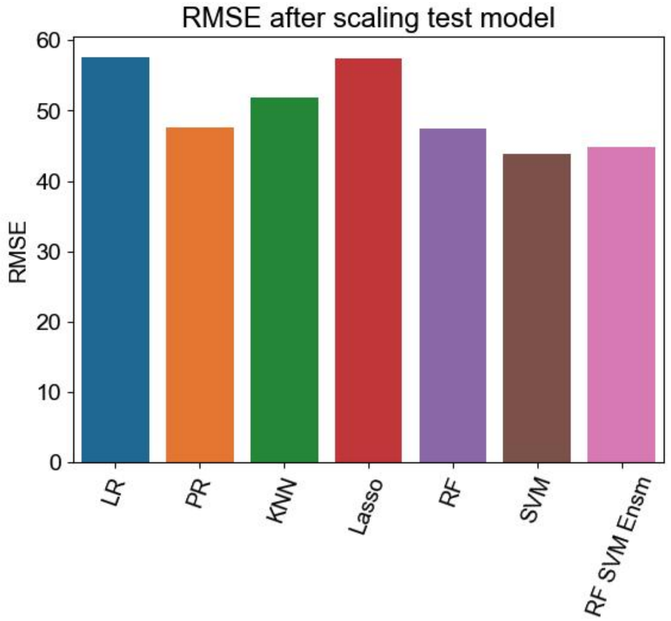
- For **KNN regression**, n_neighbors hyperparameter varied from 1 to 20, and model performance evaluated using RMSE score. Increasing n_neighbors improved performance, but difference in RMSE score between ranks 16 and 17 was only 0.75. Therefore, n_neighbors restricted to 5 for KNN regression model.
- For **Lasso regressor,** grid search CV performed on 'alpha' values of [0.1, 0.2, 0.3] and 'max_iteration' values of [10, 1000, 10000]. RMSE score used to rank hyperparameters, and alpha = 0.1 and max_iteration = 1000 provided best performance with rank 1 for the dataset.
- We will talk about RUL clipping in the next slide.

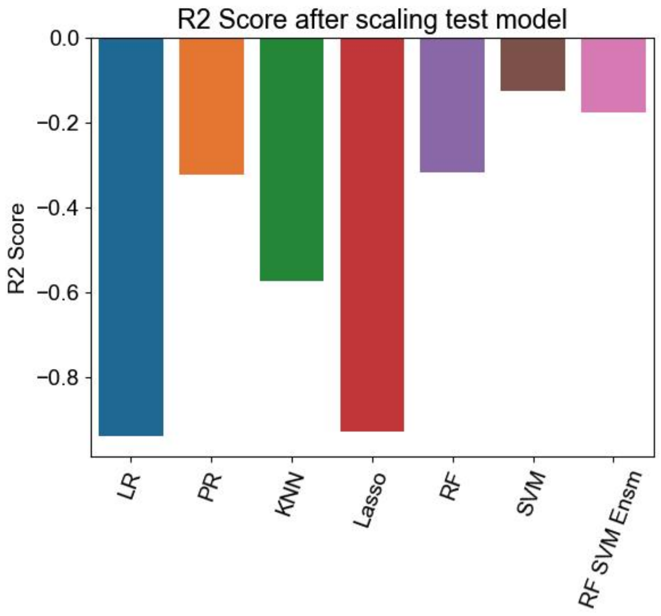




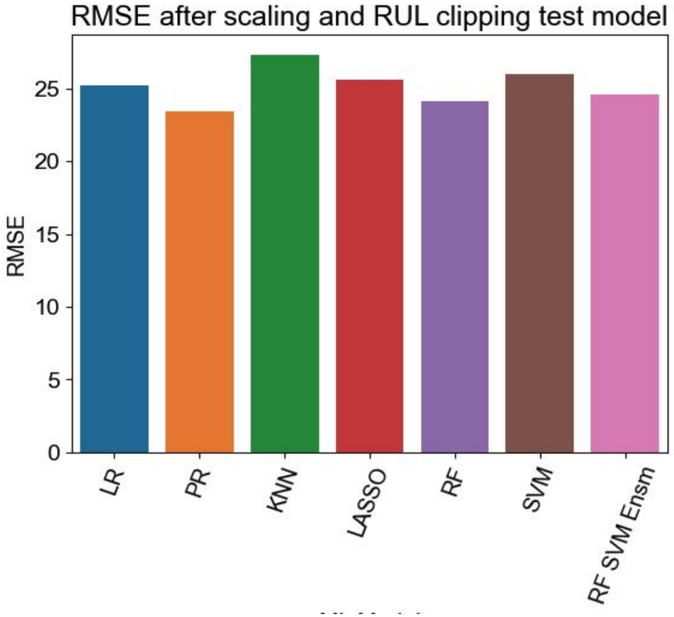
RUL Clipping

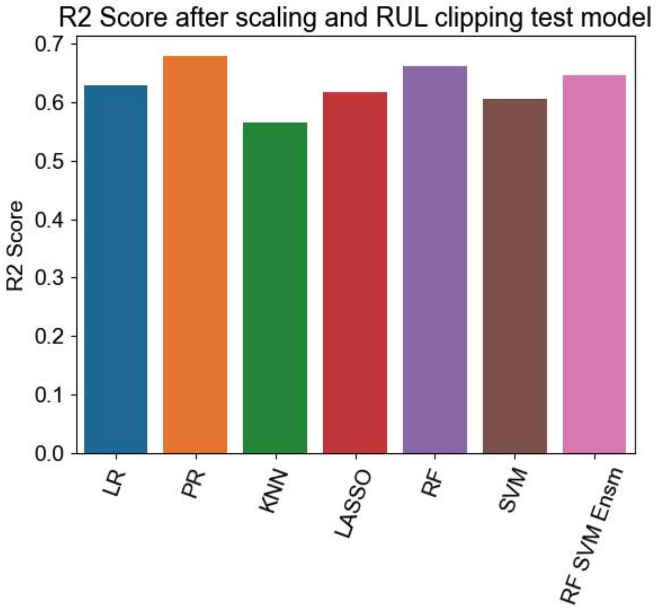






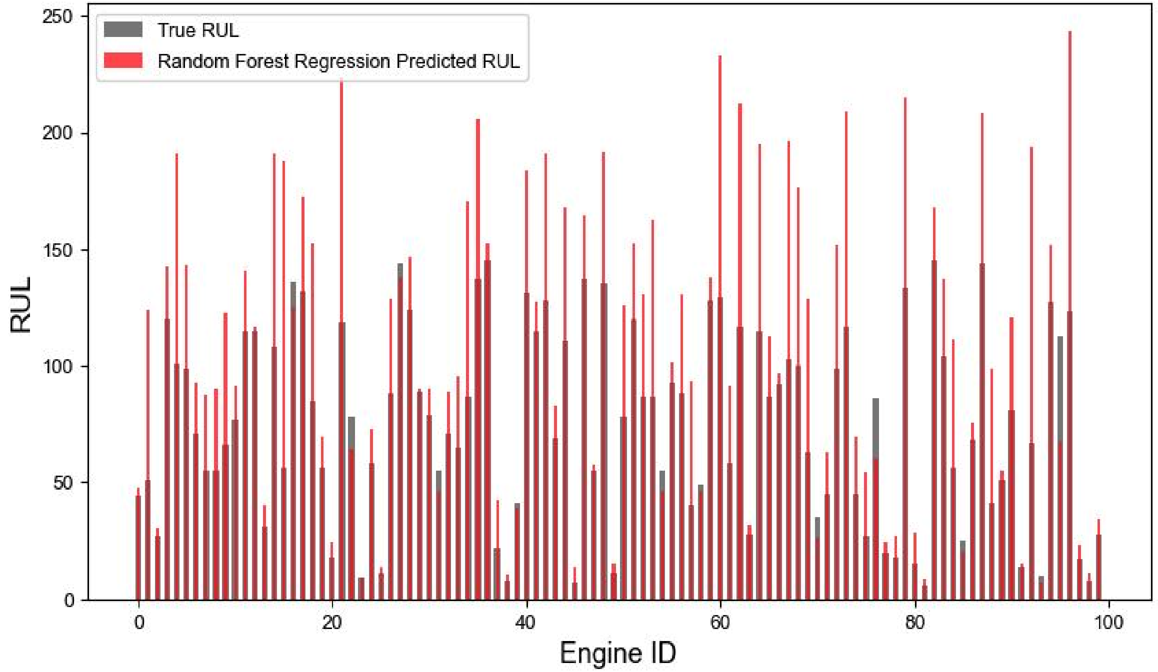
- RUL clipping was performed to address overestimation in our algorithm’s by restricting the RUL threshold values.
- The threshold of 145 for RUL clipping was determined based on more frequent occurrences in the train datasets
- RUL values above 145 were set to 145 to reduce overhead without affecting ML model performance.
- RMSE and R2-score has significantly improved



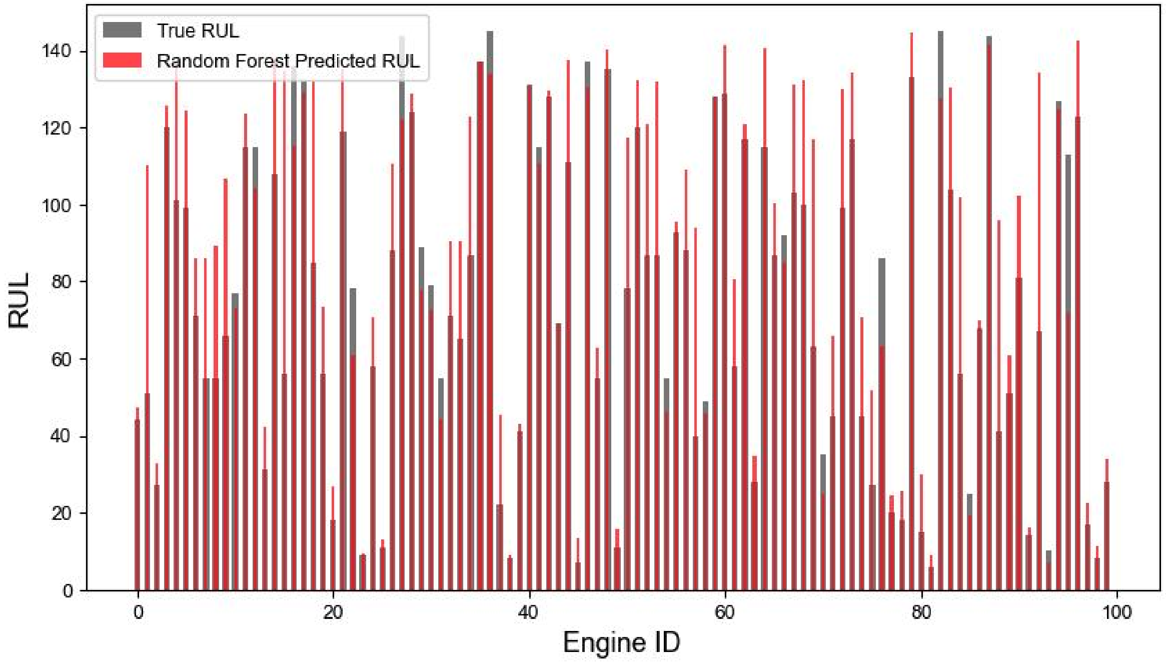


True vs Predicted RUL after clipping

BEFORE RUL CLIPPING – Test data



AFTER RUL CLIPPING – Test data



- Significant improvement in the true vs predicted model after clipping
- Predicted RUL are closer to true values of each engine

ML Model Prediction Summary

TRAINING DATA SET

Selected Features	RF		CORR		GB		LR	
ML Model	**RMSE**	**R2**	**RMSE**	**R2**	**RMSE**	**R2**	**RMSE**	**R2**
Linear Regression (LR)	23.82	0.76	23.82	0.76	23.88	0.75	23.65	0.76
Polynomial Regression (PR)	19.48	0.84	19.48	0.84	18.13	0.86	17.48	0.87
K Nearest Neighbours (KNN)	18.12	0.86	18.12	0.86	18.39	0.85	18.44	0.85
Lasso Regression(LR)	24.12	0.75	24.13	0.75	24.13	0.75	23.91	0.75
Random Forest Regression(RF)	7.52	0.97	7.53	0.98	7.51	0.98	7.51	0.98
Support Vector Machine(SVM)	21.59	0.79	22.05	0.79	21.44	0.80	21.52	0.80
Ensemble(E)	14.27	0.91	14.47	0.91	14.20	0.91	14.24	0.91
Low RMSE and Highest R2 Score	7.52	0.97	7.53	0.98	7.51	0.98	7.51	0.98
Best Model	RF	RF	RF	RF	RF	RF	RF	RF

TEST DATA SET

Selected Features	RF		CORR		GB		LR	
ML Model	**RMSE**	**R2**	**RMSE**	**R2**	**RMSE**	**R2**	**RMSE**	**R2**
Linear Regression (LR)	25.22	0.63	26.56	0.59	25.17	0.63	25.93	0.61
Polynomial Regression (PR)	23.44	0.68	24.25	0.66	25.96	0.61	28.64	0.52
K Nearest Neighbours (KNN)	27.33	0.56	28.05	0.54	30.33	0.46	30.00	0.47
Lasso Regression(LR)	25.62	0.62	26.23	0.60	25.88	0.61	26.20	0.60
Random Forest Regression(RF)	24.09	0.66	23.98	0.66	23.60	0.67	23.66	0.67
Support Vector Machine(SVM)	26.00	0.61	27.37	0.56	26.13	0.60	26.74	0.58
Ensemble(E)	24.60	0.65	25.10	0.63	24.46	0.65	24.79	0.64
Low RMSE and Highest R2 Score	23.44	0.68	23.98	0.66	23.60	0.67	23.66	0.67
Best Model	PR	PR	RF	RF	RF	RF	RF	RF

RF Regression model shows best performance in both the train and test dataset by ranking

Conclusion

- Propose predictive ML model for predicting remaining useful life of aircraft engine.
- Employed feature selection, scaling, and model training to analyze the data and build the model.
- Evaluated model performance error metrics: RMSE and R2 Score.
- Utilized four different feature selection techniques to identify the most relevant features for the prediction.
    - Four iterations of ML model were performed with different feature selections, and Random Forest regression feature importance analysis yielded the best performance.
- Tuned hyperparameters using techniques like GridSearchCV for improved model performance.
    - Top-performing models were determined to be Random Forest, Polynomial Regression, and Ensemble Learning, in that order, after comprehensive analysis of different ML models.
- Findings may have practical implications for real-world applications.
- Overall, successfully developed and evaluated a predictive model, providing valuable insights and recommendations for addressing the problem statement and achieving project objectives.

Key Differentiators of the Project: Uncovering Unique Insights and Opportunities

- **Use of FD003 Data:** Utilized FD003 data, presenting unique challenges and opportunities for analysis, leading to novel findings and recommendations.
- **Feature Importance Analysis:** Conducted comprehensive feature importance analysis to identify key drivers of model performance, uncovering previously unrecognized patterns and relationships.
- **Study of Different ML Models:** Explored multiple machine learning algorithms to provide a broader understanding of model strengths and weaknesses.
- **Potential future work opportunities** is mentioned in the next slide to provide a brief introduction of what can be done further.

Potential Future Work Opportunity

- **Anomaly Detection:** Time series data in the FD003 dataset can be analyzed using anomaly detection techniques to identify unusual patterns in sensor readings, which can help identify engines that may require additional maintenance or investigation.
- **Predictive Maintenance Optimization:** In addition to predicting Remaining Useful Life (RUL), optimizing the maintenance schedule based on predicted RUL can potentially reduce maintenance costs by identifying the optimal time for maintenance activities.
- **Clustering Analysis:** Clustering analysis can be performed on the data to identify different groups of engines with similar behavior patterns, allowing for tailored maintenance activities for each group to improve efficiency and effectiveness. (Type of degradation)
- **Further Hyperparameter Tuning:** More detailed hyperparameter tuning can be explored to optimize the performance of the machine learning models.
- **Signal Smoothing:** Applying signal smoothing techniques can potentially improve the predictive power of the algorithm by reducing noise in the data.
- **Adding Features:** Generating new features from existing features or incorporating additional data such as lagged features, derivatives of features, geographic data, etc., can potentially enhance the performance of the models.
- **Using a Different Loss Function:** Choosing a loss function that penalizes overestimates more than underestimates can be prudent in problems like RUL prediction to mitigate potential risks and costs associated with overestimations.

Inspirational link (Reference)

1. Data: https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6
2. Research Paper: Saxena, A., Goebel, K., Simon, D. and Eklund, N., 2008, October. Damage propagation modeling for aircraft engine run-to-failure simulation. In 2008 international conference on prognostics and health management (pp. 1-9). IEEE.
3. Kaggle: https://www.kaggle.com/code/wassimderbel/nasa-predictive-maintenance-rul
4. Scikit_learn: https://scikit-learn.org/
5. Tutorial Github: https://github.com/thivinanandh/Teaching-Python
6. Lecture notes: https://www.zenteiq.com/

Thank You …

Any Questions?

APPENDIX (SUPPORTING SLIDES)

Important definitions

- **NASA's C-MAPSS** (Commercial Modular Aero-Propulsion System Simulation) is a software program that is used to simulate and model the behavior of gas turbine engines used in commercial aviation. The C-MAPSS software is designed to simulate a wide range of engine conditions and failure modes, including those related to the combustion system, compressor, and turbine sections of the engine.
- **Remaining Useful Life (RUL)** is used to provide an early indication of failures that required performing maintenance and/or replacement of the system in advance. This is the amount of time they have left before they need to be replaced or serviced or in other words it refers to the estimated amount of time or number of operational cycles that an engine can continue to operate safely and reliably before it needs to be overhauled or replaced.
- **High Pressure Compressor (HPC)** – The HPC module is made up of a series of rotor and stator assemblies whose main function is to raise the pressure of the air supplied to the combustor.
- **HPC (High-Pressure Compressor) degradation:** It refers to the gradual deterioration of the compressor's performance due to various factors, such as erosion, corrosion, fouling, and thermal stresses.

Challenges

- Data understanding / representation / downselection of features.
- Team members new to Python
- Time to complete the project.
- More details/challenges to be observed when we start working on it in details.

Why ML?

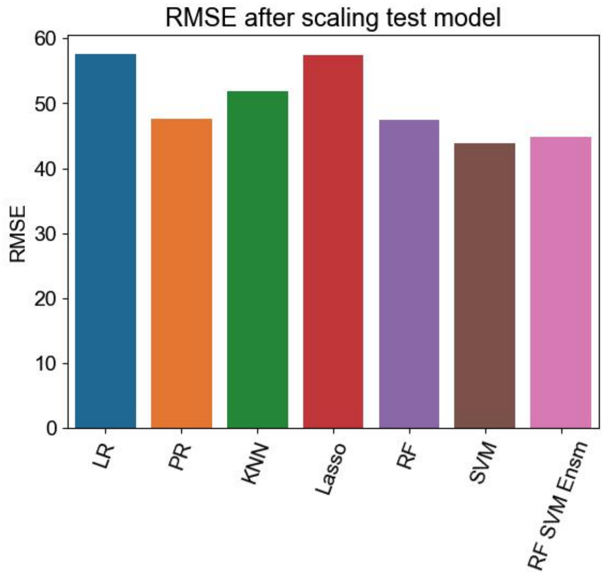
- The main purpose of the NASA C-MAPSS dataset is to provide a benchmark for developing and evaluating machine learning algorithms for predicting the remaining useful life (RUL) of turbofan engines. The idea is to use the sensor measurements and operational settings from the engine to predict the RUL, which would allow for better maintenance planning and prevent engine failures.
- In practice, predicting the RUL of a turbofan engine can be challenging because of the complex relationships between the sensor measurements, operational settings, and the engine's health. Machine learning algorithms can be used to learn these relationships from historical data, and then use this knowledge to predict the RUL of new engines.
- Therefore, the ultimate goal of the NASA C-MAPSS dataset is to enable the development and evaluation of machine learning models that can accurately predict the RUL of turbofan engines, which can help reduce maintenance costs, minimize downtime, and improve overall safety and reliability.

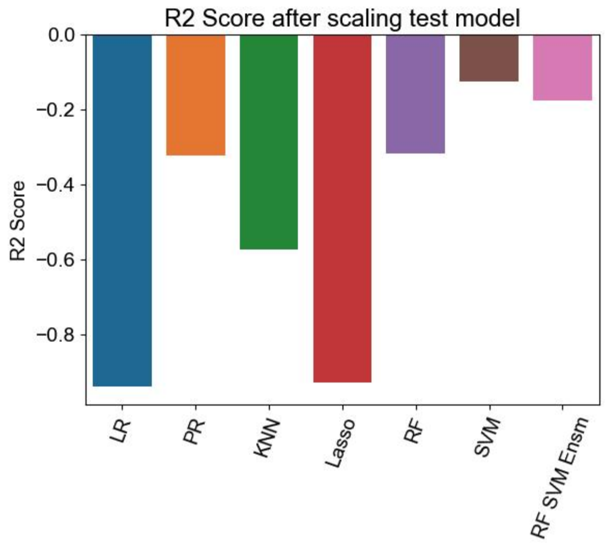
Data Collection and initial understanding

- **NASA C-MAPSS-2 (Turbofan Engine Degradation Simulation Data Set-2)** is a publicly available dataset created by NASA for research on prognostics and health management (PHM) of aircraft engines. Ref [1]
- It consists of simulated data for turbofan engines that have been degraded under various operating conditions, with the aim of developing models to predict when and how these engines will fail.
- The dataset includes sensor measurements such as **temperature, pressure, and vibration**, as well as information about the **operating conditions** and the **degree of degradation of the engine.**
- It is commonly used as a benchmark dataset in the field of PHM and provides an opportunity for researchers to develop and test their algorithms for predicting engine failures.
- **"id":** This column represents the **ID number** of each observation or measurement.
- **"cycle"**: This column represents the **cycle number or time step** of the measurement. The cycle number is incremented by 1 for each new observation.
- **"setting1", "setting2", "setting3":** These columns represent the operational settings of the engine**.** These settings are inputs to the engine and affect its performance. **The specific meaning of each setting is not provided by NASA due to proprietary reasons.**
- "**s1" to "s21":** These columns represent the **sensor measurements of the engine.** The sensors measure various physical quantities such as **temperature, pressure, and vibration.** Each column corresponds to a different sensor. **The specific meaning of each sensor measurement is not provided by NASA due to proprietary reasons.**
- **"ttf":** This column represents the **time to failure** of the engine in units of cycles. It is the target variable that we want to predict using machine learning algorithms. The "ttf" value is not provided for the test set because it is the value that we are trying to predict.

RUL CLIPPING

BEFORE RUL CLIPPING
(Test data)





AFTER RUL CLIPPING
(Test data)