KIG4068 MACHINE LEARNING

GROUP ASSIGNMENT: PROGRESS REPORT

PREDICTIVE MAINTENANCE OF TURBOFAN JET ENGINE

GROUP MEMBERS	MATRICS NO
MUHAMMAD SHAHRIL BIN ZAINOL ABIDIN	17205720/2
CHNG ZHEN HUO	S2131858/1
NIK IRWAN ISKANDAR BIN NIK AHMAD KAMAL	17204356/2

GITHUB: https://github.com/ZhenHao03/KIG4068-Assignment

DR. MEOR MOHD FAISAL BIN MEOR ZULKIFLI

DEPARTMENT OF MECHANICAL ENGINEERING
FACULTY OF ENGINEERING
UNIVERSITI MALAYA

1. SUMMARY OF WORK DONE

1.1 Project Objectives

1.1.1 Predictive Maintenance in Aerospace Industry

In an era dominated by rapid technological advancements and intricate machinery, the maintenance of critical systems has emerged as a paramount concern across various industries. Among these, the aerospace industry stands as a pinnacle of innovation and precision, where the reliability of its assets, equipment, and systems plays a pivotal role in ensuring safety and operational efficiency. According to the International Air Transport Association (IATA), 2023 has been a year where air transportation very nearly returned to its pre-pandemic pace of activity and in the long run, global passenger traffic is forecasted to double by 2040. These statistics highlight the importance of adopting more advanced maintenance strategies to sustain the growing consumers of the aerospace industry.

Predictive maintenance represents a paradigm shift from traditional reactive or preventive maintenance strategies. By harnessing the power of data analytics, machine learning (ML) algorithms, and sensor technology, predictive maintenance endeavors to forecast equipment failures before they occur (Carvalho et al., 2019). This proactive approach enables organizations to address issues in a timely manner, thereby averting costly unplanned downtime, mitigating safety risks, and optimizing resource utilization.

1.1.2 Predicting Remaining Useful Life of Turbofan Jet Engine

Turbofan is a type of airbreathing jet engine that is widely used in aircraft propulsion. It consists of a gas turbine engine that achieves mechanical energy from combustion and a ducted fan that uses the mechanical energy from the gas turbine to force air rearwards (NASA, 2021). Turbofan jet engine is commonly used for slower and long-distance flights such as commercial airline aircraft. Figure 1.1 below shows the operation and components of a turbofan jet engine.

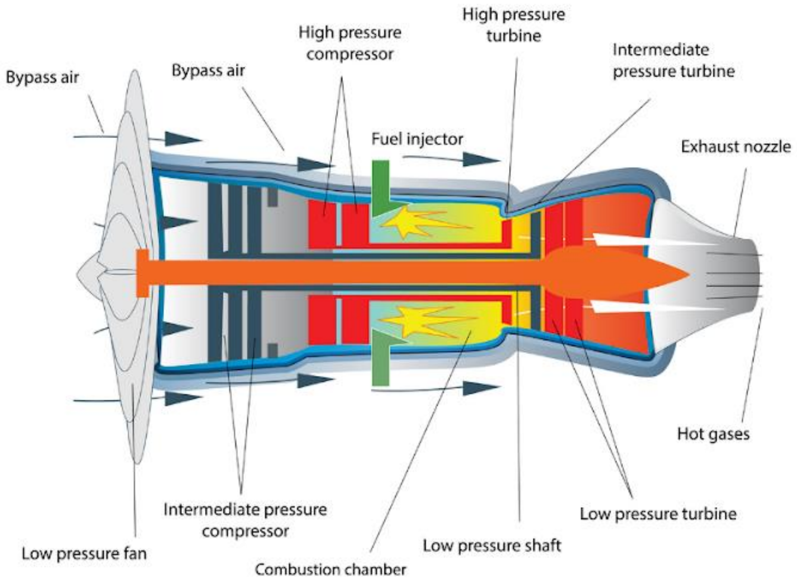


Figure 1.1: Turbofan jet engine

At the heart of predictive maintenance lies the concept of Remaining Useful Life (RUL) prediction. RUL refers to the estimated duration of time that a component, machine, or system will remain operational before it becomes unreliable or fails to meet performance requirements (Ferreira & Gonçalves, 2022). By accurately predicting RUL, maintenance activities can be strategically planned and executed, maximizing the operational lifespan of assets while minimizing the likelihood of unexpected failures.

In the context of aviation, predicting the RUL of aircraft engines using advanced data analytics techniques such as ML algorithms and statistical models helps to optimize resource allocation, enhance safety, and prolong the operational lifespan, ultimately contributing to the reliability and efficiency of the aviation industry as a whole. This process involves analyzing a myriad of operational data, including sensor measurements, operational parameters, maintenance records, and environmental factors. Once a large volume of data has been collected, a machine learning model can be built by applying proper data preprocessing techniques, selecting an appropriate machine learning model, and hyperparameters tuning.

1.1.3 Objectives

The main objective of this project is to develop a machine learning model for turbofan jet engine multiclass classification and RUL prediction. To achieve this, several goals have been established.

- To establish a comprehensive Exploratory Data Analysis (EDA) on the chosen dataset
- To determine the appropriate data preprocessing techniques for the chosen dataset.
- To determine the best-performing ML models by hyperparameters tuning and evaluating various performance metrics.

1.2 Member Contributions

1.2.1 Shahril

Shahril has done the exploratory data analysis (EDA) on a dataset related to the predictive maintenance of turbofan jet engines. There are 5 main points in the EDA :

1. Data Inspection: Provides information about the dataset, including the number of train and test trajectories, conditions, and fault modes for each dataset. Describes the dataset structure and column definitions.
2. Distribution of Engine Lifetime: Calculates and visualizes the distribution of engine lifetime in the train and test sets, showing that the test set engines have shorter lifetimes on average compared to the train set.
3. Distribution of Remaining Useful Life (RUL): Calculates RUL for each data point and plots its distribution in the train and test sets, highlighting the difference in RUL distributions between the two sets.
4. Wear/Degradation Patterns: Analyzes wear or degradation patterns in sensor measurements over time, identifying sensors with constant values that can be dropped due to low variance.
5. Difference between Engine Units: Selects specific sensors, calculates average sensor values for each engine unit, and compares sensor time series between engines with the highest and lowest average values for each selected sensor.

Overall, Shahril performs a comprehensive exploratory analysis of the turbofan jet engine dataset, including data inspection, visualization of engine lifetime and RUL distributions, analysis of wear/degradation patterns in sensor data, and comparison of sensor time series between different engine units.

1.2.2 Nik

Nik has provided a utility module for the data. The Utility modules contain functions or constants used across multiple scripts or projects to perform specific tasks, such as data preprocessing, file reading, or calculations. In this project, it provides functions for reading and preprocessing data relevant to predictive maintenance tasks, making it a utility module customized for that purpose.

Util modules provided help streamline development by centralizing common tasks, promoting code reuse, and improving maintainability. Team members can import and use the functions defined in this module in various scripts or projects without rewriting the same code, enhancing efficiency and consistency across the codebase.

1.2.3 Zhen Huo

Zhen Huo has done the data preprocessing steps for the predictive maintenance of turbofan jet engines. It begins by loading the dataset and ensuring there are no null values. Remaining Useful Life (RUL) is calculated for each engine, and features with low correlation to RUL are removed. Additionally, constant features are identified and eliminated. The sensor time series data is scaled per engine, and a rolling window technique is applied to transform the time series into sliding windows. Features are then engineered using TSFresh, extracting a wide range of features from the time series data. Principal Component Analysis (PCA) is employed to reduce the dimensionality of the extracted features, followed by feature selection to retain only the most relevant ones. The resulting dataset is visualized for feature correlation, ensuring independence among selected features.

Overall, these preprocessing steps aim to prepare the data for machine learning modeling, ensuring that it's properly formatted, scaled, and contains relevant features for predicting the Remaining Useful Life of turbofan jet engines.

2. CURRENT RESULTS

2.1 Exploratory Data Analysis (EDA)

The dataset chosen for this project is FD001, which is a part of the NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset. The dataset is designed for predictive maintenance of turbofan jet engines and it contains multivariate time series data collected from a fleet of turbofan engines.

2.1.1 Data Inspection

There are three files in FD001 dataset which are training data, test data, and true value of Remaining Useful Life (RUL). The training data consists of 20631 rows, each representing a snapshot of data captured during operational cycles of 100 different engine units. These snapshots span from the initial operational cycles to the last operational cycle before engine failure. On the other hand, the test data comprises 13096 rows, capturing snapshots of operational cycles from the same 100 engine units. However, unlike the training data, the test data includes only snapshots up to a certain time point, stopping before the engine reaches failure. The purpose of the test data is to evaluate the performance of predictive maintenance models trained on the training data. Model predictions are compared against the true RUL values provided separately to assess the accuracy of the predictions.

Table 2.1: Columns name

Column Index	Column Name
1	Unit number of engine
2	Time, in cycles
3	Operational setting 1
4	Operational setting 2
5	Operational setting 3
6	Sensor 1, Fan inlet temperature
7	Sensor 2, LPC outlet temperature
8	Sensor 3, HPC outlet temperature
9	Sensor 4, LPT outlet temperature
10	Sensor 5, Fan inlet pressure
11	Sensor 6, Bypass-duct pressure
12	Sensor 7, HPC outlet pressure
13	Sensor 8, Physical fan speed
14	Sensor 9, Physical core speed
15	Sensor 10, Engine pressure ratio
16	Sensor 11, HPC outlet static pressure
17	Sensor 12, Ratio of fuel flow to
18	Sensor 13, Corrected fan speed
19	Sensor 14, Corrected core speed
20	Sensor 15, Bypass ratio
21	Sensor 16, Burner fuel-air ratio
22	Sensor 17, Bleed enthalpy
23	Sensor 18, Required fan speed
24	Sensor 19, Required fan conversion speed
25	Sensor 20, High-pressure turbines cool air flow
26	Sensor 21, Low-pressure turbines cool air flow

Table 2.1 above shows the columns name in both of the training dataset and test dataset. For the true RUL dataset, it consists of the RUL of the 100 engines in the test dataset. All the data are numerical type (int64 and float64) and zero null values.

2.1.2 Comparison of Engine Lifetime per Unit Distribution for Training Dataset and Test Dataset

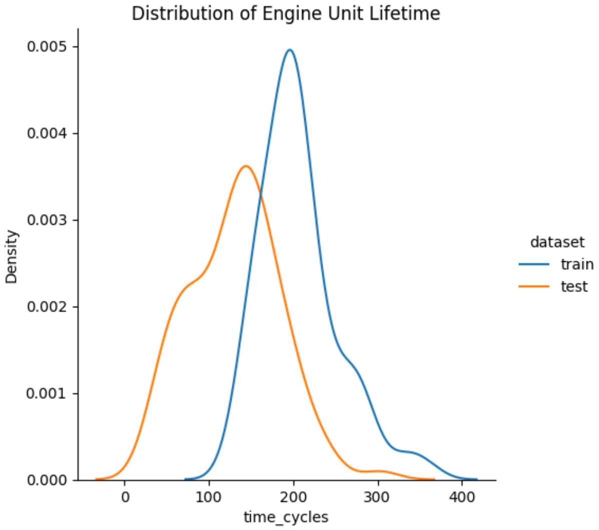


Figure 2.1: Distribution of engine lifetime per unit

From Figure 2.1, it is observed that the engine units in the test set have shorter lifetimes than the train set. From the describe() function, on average, test set lifetimes are 70 time cycles shorter compared to the train set.

2.1.3 Comparison of RUL Distribution in Training Dataset and True RUL Dataset

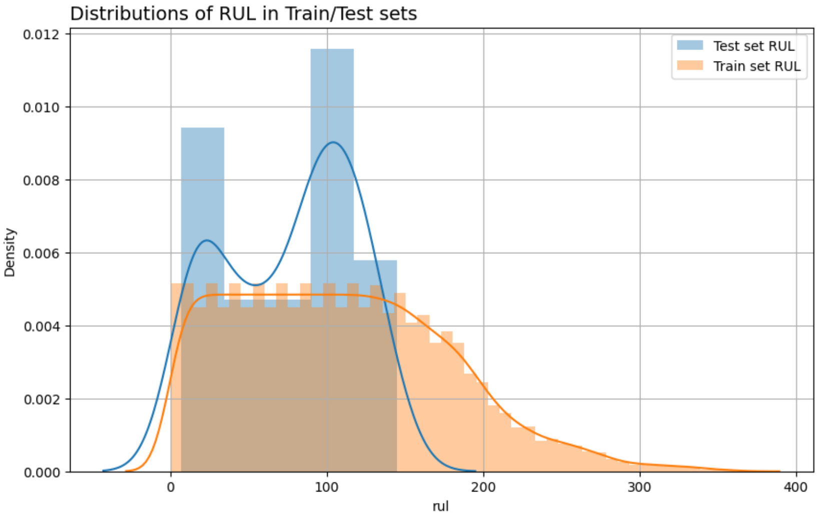


Figure 2.2: Distribution of RUL

As expected, the train set contains engine units with RUL way higher than the true RUL. This is because the train set have the full lifetime data of an engine while the test set only contains data until a period of time before its end of life. Hence, the target is to predict the RUL at the last cycle of each engine unit in the test set.

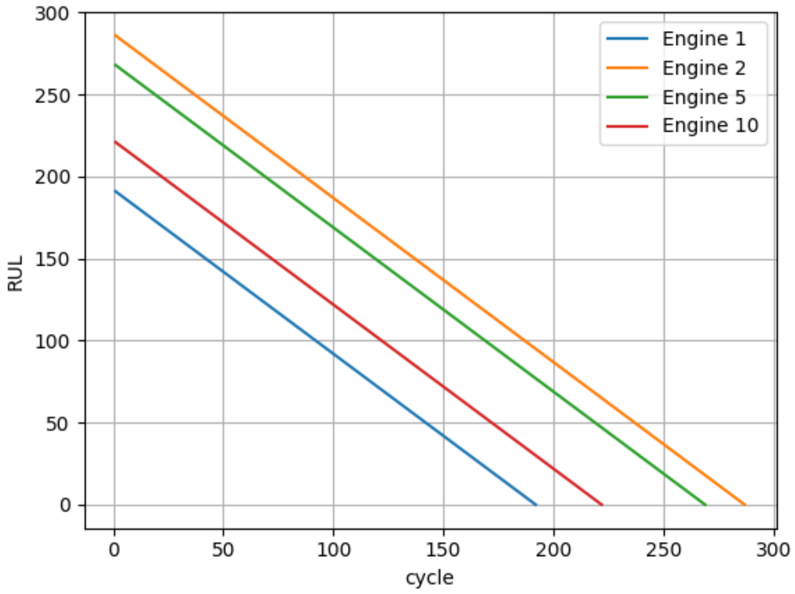


Figure 2.3: Plot of RUL against time cycles

For Figure 2.3, engine 1, 2, 5, and 10 have been chosen to visualize the correlation between RUL and time cycles. It is obvious that as the engines have been running for many time cycles, the RUL will be reduced.

2.1.4 Identifying Wear and Degradation Patterns from Sensors Data

Figure 2.4 shows the time series data of each sensor (sensor 1-21). It is observed that sensor 1, 5, 10, 16, 18, and 19 are constant throughout the time cycles. This rendered these sensors useless as they have zero variance and can be dropped during data preprocessing. For other sensors, it can be seen that the sensor values will change with respect to time cycles, whether increasing or decreasing. Thus, this indicates that they will have correlation to the target. In addition, feature scaling is needed in data preprocessing due to the different range of sensor values.

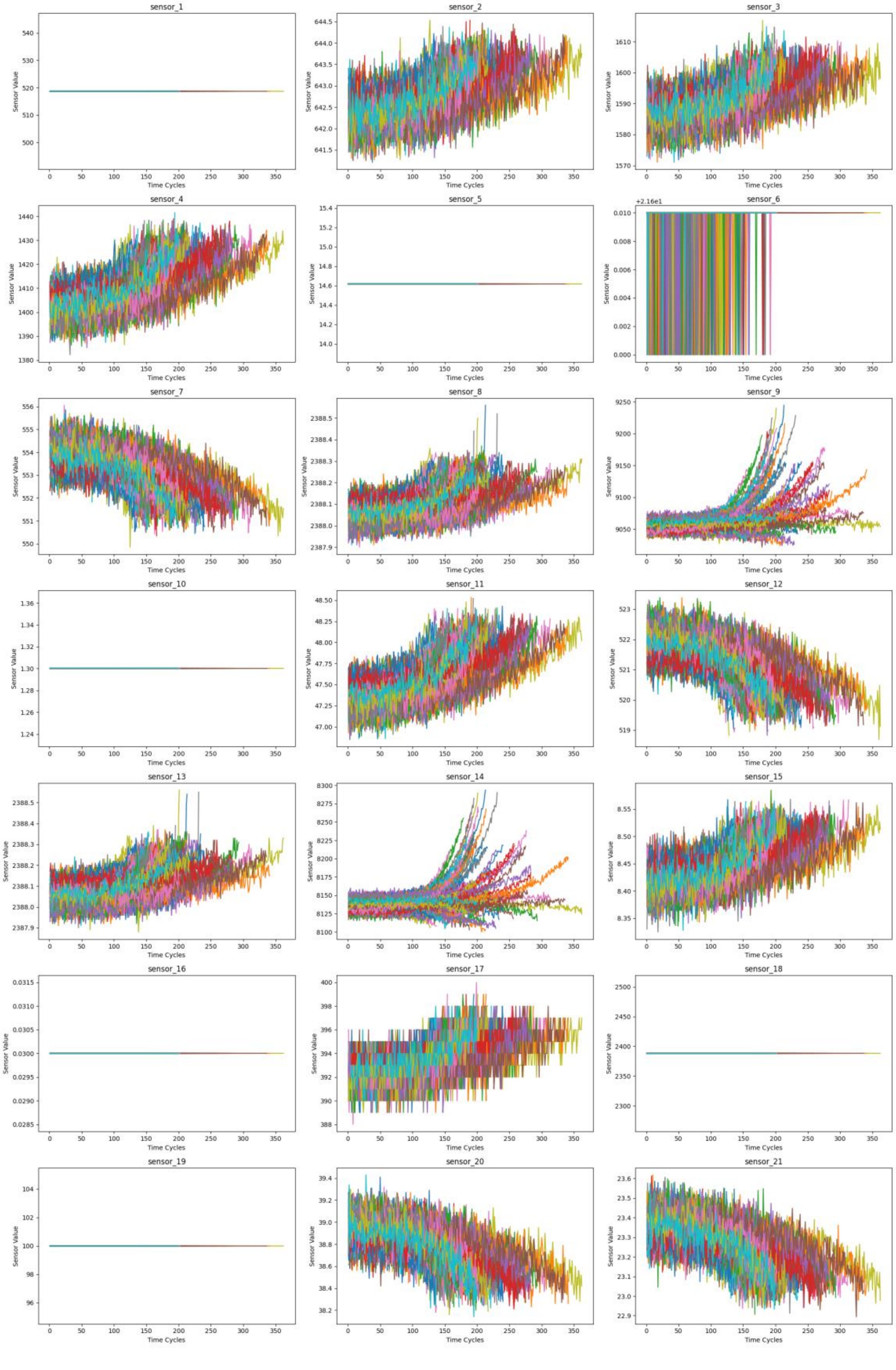


Figure 2.4: Plot of time series data of sensors

2.1.5 Difference between Each Unit of Engines

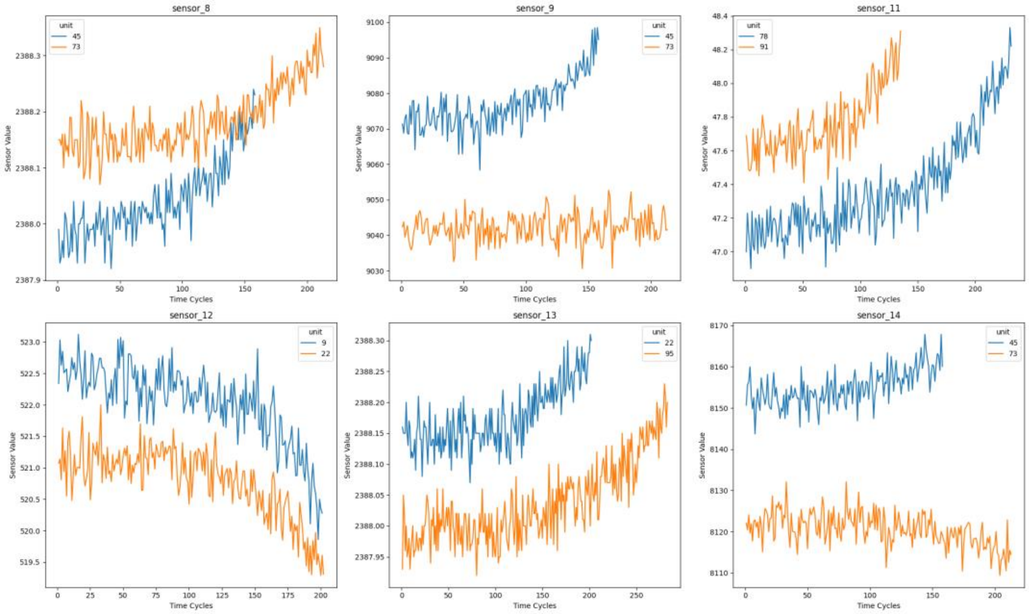


Figure 2.5: Difference of sensor value for each unit of engine

Figure 2.5 shows the plots of two units of engine with the biggest difference in mean values for sensor 8, 9, 11, 12, 13, and 14. Each unit have different starting point and the scaling is off. Thus, this suggests that we might need to scale the sensor time series with respect to the start of every individual engines time series. Scaling with respect to the individual engines starting values allows us to bring all the engines time series to the same scale.

2.2 Data Preprocessing

2.2.1 Adding RUL with Upper Threshold

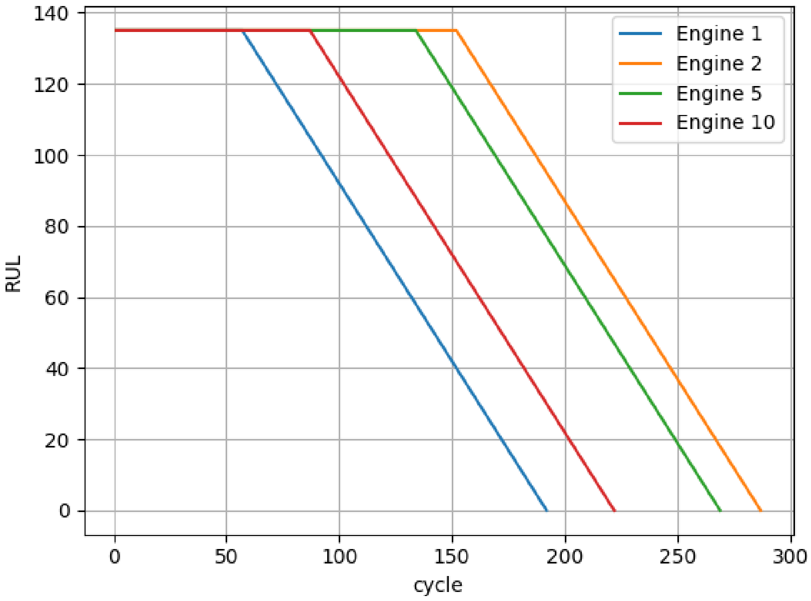
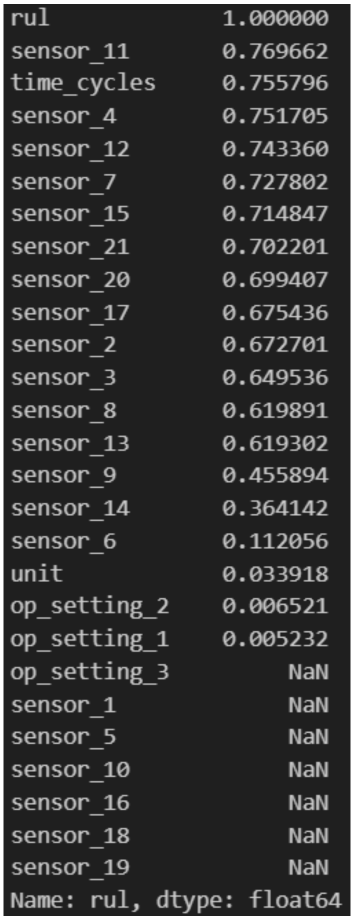


Figure 2.6: Adding RUL with upper threshold

Based on Heimes (2008), it is best to set an upper threshold for the RUL in the train dataset as the solution to solve the problem encountered in Section 2.1.3 and the motivation for this is that a degradation process will only be noticeable in the data after a unit has been operating for some time. The author decided to limit a maximum value of RUL with 135 cycles. By applying the upper threshold, Figure 2.6 is obtained.

2.2.2 Removing Features with Zero Correlation to RUL



```
rul             1.000000
sensor_11       0.769662
time_cycles     0.755796
sensor_4        0.751705
sensor_12       0.743360
sensor_7        0.727802
sensor_15       0.714847
sensor_21       0.702201
sensor_20       0.699407
sensor_17       0.675436
sensor_2        0.672701
sensor_3        0.649536
sensor_8        0.619891
sensor_13       0.619302
sensor_9        0.455894
sensor_14       0.364142
sensor_6        0.112056
unit            0.033918
op_setting_2    0.006521
op_setting_1    0.005232
op_setting_3         NaN
sensor_1             NaN
sensor_5             NaN
sensor_10            NaN
sensor_16            NaN
sensor_18            NaN
sensor_19            NaN
Name: rul, dtype: float64
```

Figure 2.7: Feature correlation to target

Figure 2.7 shows the absolute value of feature correlation to the target (RUL) in descending order. As expected, operational setting 3 and sensor 1, 5, 10, 16, 18, 19 have zero correlations with RUL since they are constant throughout the time cycles. By applying Scikit-Learn's VarianceThreshold, features with zero correlations to the target will be automatically dropped.

2.2.3 Scaling per Engine

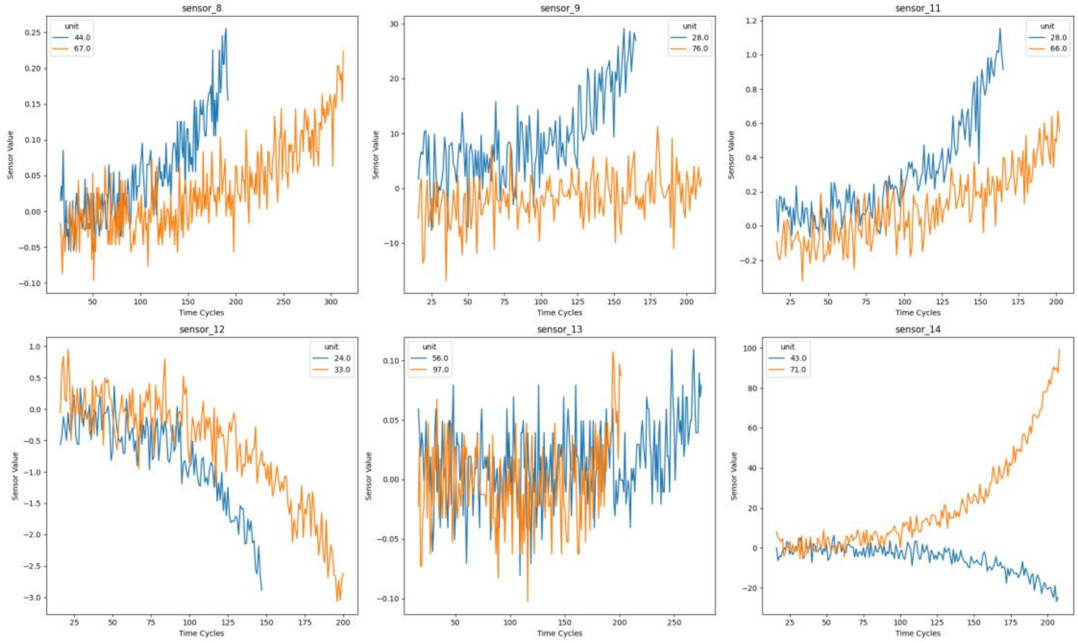


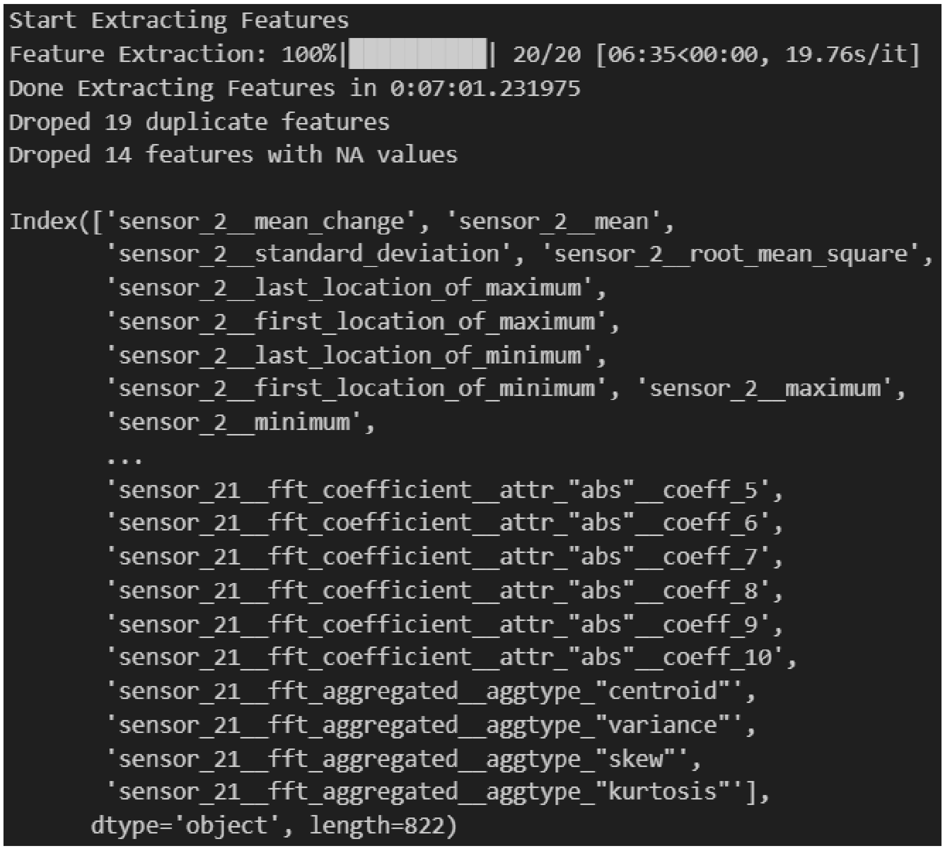
Figure 2.8: Scaled sensor time series data

As suggested from Section 2.1.5, the sensor time series data needs to be scaled with respect to the start of every individual engines. Figure 2.8 shows the result of performing scaling with respect to the start of each engine unit.

2.2.4 Applying Rolling Window

Rolling window is a technique where you use a fixed-size subset of the most recent observations to train your model and make predictions. The window moves forward through the time series, discarding the oldest observation and including the next new observation at each step. We will be transforming the original sensor time series into sliding windows of length 30 using a popular time series analysis library called TSFresh.

2.2.5 Features Engineering using TSFresh



```
Start Extracting Features
Feature Extraction: 100%|██████████| 20/20 [06:35<00:00, 19.76s/it]
Done Extracting Features in 0:07:01.231975
Droped 19 duplicate features
Droped 14 features with NA values

Index(['sensor_2__mean_change', 'sensor_2__mean',
       'sensor_2__standard_deviation', 'sensor_2__root_mean_square',
       'sensor_2__last_location_of_maximum',
       'sensor_2__first_location_of_maximum',
       'sensor_2__last_location_of_minimum',
       'sensor_2__first_location_of_minimum', 'sensor_2__maximum',
       'sensor_2__minimum',
       ...
       'sensor_21__fft_coefficient__attr_"abs"__coeff_5',
       'sensor_21__fft_coefficient__attr_"abs"__coeff_6',
       'sensor_21__fft_coefficient__attr_"abs"__coeff_7',
       'sensor_21__fft_coefficient__attr_"abs"__coeff_8',
       'sensor_21__fft_coefficient__attr_"abs"__coeff_9',
       'sensor_21__fft_coefficient__attr_"abs"__coeff_10',
       'sensor_21__fft_aggregated__aggtype_"centroid"',
       'sensor_21__fft_aggregated__aggtype_"variance"',
       'sensor_21__fft_aggregated__aggtype_"skew"',
       'sensor_21__fft_aggregated__aggtype_"kurtosis"'],
      dtype='object', length=822)
```

Figure 2.9: Feature engineering to the train dataset

In order to obtain a higher-quality features for model training and make robust predictions, feature engineering is needed. For time series analysis, TSFresh automates the process of extracting a wide range of features from time series data. This includes statistical measures, model-based features, and others, without requiring manual intervention. This helps in capturing the underlying patterns and characteristics of the time series data efficiently. By providing a comprehensive set of features that capture various aspects of the time series data, tsfresh enhances the ability of machine learning models to make accurate predictions. Figure 2.9 shows that the number of features have increased from 21 to 822 that span across sensor 1 to sensor 21.

2.2.6 Applying Scikit-Learn's PCA and TSFresh's Feature Selection

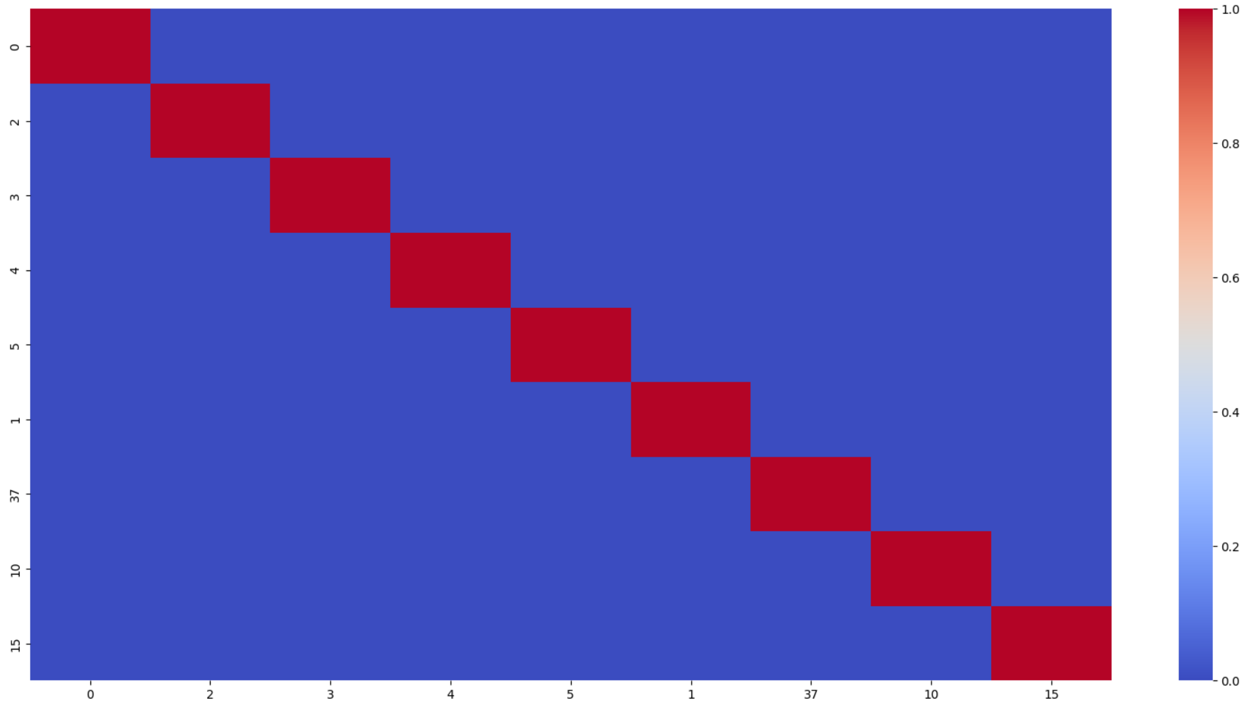


Figure 2.10: Feature correlation after PCA and Feature Selection

Since we have a huge number of features after applying TSFresh, we will have highly correlated features which needs to be remove. In order to eliminate this problem, Principal Component Analysis (PCA) will be performed to reduce the dimensionality. First, we will apply PCA with number of components set to 40, thus giving us 40 principal components. Then, we will use TSFresh's feature selection function to automatically choose the best features in terms of dependency towards other features. In the end, we were left with 9 features that are highly independent of each other (refer Figure 2.10) as well as high correlation to the target. This should gives us a more robust ML models.

3. FUTURE PLANS

Table 3.1: Gantt chart for future plans

26/3	15/4	16/5	23/5	26/5	30/5	2/6
Explore datasets	Milestone 1					
	Deep explore on the dataset chosen	Milestone 2				
		Explore Data preprocessing , build baseline for preprocessing	Milestone 3			
			Explore Regression on RUL prediction	Train Regression models	Evaluate and perform cross validations to fin out the best model	Milestone 4
			Explore Multiclassification on health predictions	Train Classification models	Evaluate Models and perform hyperparameter tuning	Milestone 5

Milestone 1: Proposal

- Explore on different datasets to find the suitable one to perform various method of supervised machine learning
- Deadline: 15 April 2024
- By: All members

Milestone 2: Perform EDA

- Deep Explore on the dataset chosen
- Find out the correlations between the features and targets
- Explore the info in the datasets using Seaborn, NumPy etc.
- Deadline: 16 May 2024
- By: Shahril

Milestone 3: Data Preprocessing

- Find out the suitable codes for pipeline for both regression and multiclassification problems
- Find out if there are any methods to improve the training of the models, for example applying rolling window and data reduction etc.
- Deadline: 23 May 2024
- By: Shahril & Chng Zhen Huo

Milestone 4: Perform Regression for RUL prediction

- Train data using different models using scikit learn
- Regression models: Linear Regression, SVR, XGBoost, Random Forest, Decision Tree, etc.
- Perform cross validations
- Find out the best models for regression and perform hyperparatuning
- Deadline: 2 June 2024
- By: Nik Irwan

Milestone 5: Perform Multiclassification for health conditions predictions

- Train data using different models using scikit learn
- Classification models: SVC, XGBoost, Random Forest, Decision Tree, etc.
- Perform cross validations
- Find out the best models for classification and perform hyperparatuning
- Deadline: 2 June 2024
- By: Chng Zhen Huo

4. CODES AND RESOURCES

4.1 Datasets

4.1.1 NASA C-MAPSS Jet Engine Simulated Data

The dataset chosen for this project was provided by the Prognostics Center of Excellence (PCoE) at NASA Ames and involves engine degradation simulation using the Commercial Modular Aero Propulsion System Simulation (C-MAPSS) (Saxena & Goebel, 2008). C-MAPSS is software that provides a transient simulation of a large commercial turbofan engine with a realistic engine control system. The dataset comprises four distinct sets, each simulating engines under different combinations of operational conditions and fault modes as shown in Table 4.1. For this assignment, we will be using FD001 dataset since it is widely researched, thus it is suitable for benchmarking purposes.

Table 4.1: Details of the C-MAPSS dataset

Dataset	FD001	FD002	FD003	FD004
Train size (# of engine units)	100	260	100	249
Test size (# of engine units)	100	259	100	248
Operating conditions	1	6	1	6
Fault conditions	1	1	2	2

4.2 Repositories and Codes

Below are the repositories and codes used to benchmark our ML models:

- https://github.com/Ali-Alhamaly/Turbofan_usefull_life_prediction
- https://www.kaggle.com/code/wassimderbel/nasa-predictive-maintenance-rul#Models-Implementation-and-instantiation
- https://www.mathworks.com/help/predmaint/ug/remaining-useful-life-estimation-using-convolutional-neural-network.html
- https://www.mathworks.com/help/predmaint/ug/similarity-based-remaining-useful-life-estimation.html

4.3 Tools

4.3.1 TSFresh

TSFresh is a python package. It automatically calculates a large number of time series characteristics, the so called features. Further the package contains methods to evaluate the explaining power and importance of such characteristics for regression or classification tasks. It is used for feature engineering of our sensor time series data to generate higher quality features.

4.3.2 Scikit-Learn

Scikit-Learn is a standard python package for building machine learning models and statistical modelling. Through scikit-learn, we can implement various machine learning models for regression, classification, clustering, and statistical tools for analyzing these models. It also provides functionality for dimensionality reduction, feature selection, feature extraction, ensemble techniques, and inbuilt datasets.

5. CONCLUSION

In summary, the team has made significant strides in preparing for predictive maintenance of turbofan jet engines. Shahril conducted a comprehensive Exploratory Data Analysis (EDA), uncovering insights into engine lifetime distribution, Remaining Useful Life (RUL), and wear patterns in sensor data. Nik's utility module streamlines data handling, while Zhen Huo's preprocessing steps ensure data readiness for machine learning modeling. With clear milestones set for dataset exploration, model training, and evaluation, this team can deliver effective predictive maintenance solutions for the Turbofan Jet Engine.

6. REFERENCES

Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. d. P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, *137*, 106024. https://doi.org/https://doi.org/10.1016/j.cie.2019.106024

Ferreira, C., & Gonçalves, G. (2022). Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods. *Journal of Manufacturing Systems*, *63*, 550-562. https://doi.org/https://doi.org/10.1016/j.jmsy.2022.05.010

Heimes, F. O. (2008, 6-9 Oct. 2008). Recurrent neural networks for remaining useful life estimation. 2008 International Conference on Prognostics and Health Management,

IATA. (2023). *Global Outlook for Air Transport - A local sweet spot*. https://www.iata.org/en/iata-repository/publications/economic-reports/global-outlook-for-air-transport---december-2023---report/

NASA. (2021). *Turbofan Engine*. https://www.grc.nasa.gov/www/k-12/airplane/aturbf.html

Saxena, A., & Goebel, K. (2008). *Turbofan Engine Degradation Simulation Data Set*. https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6/about_data