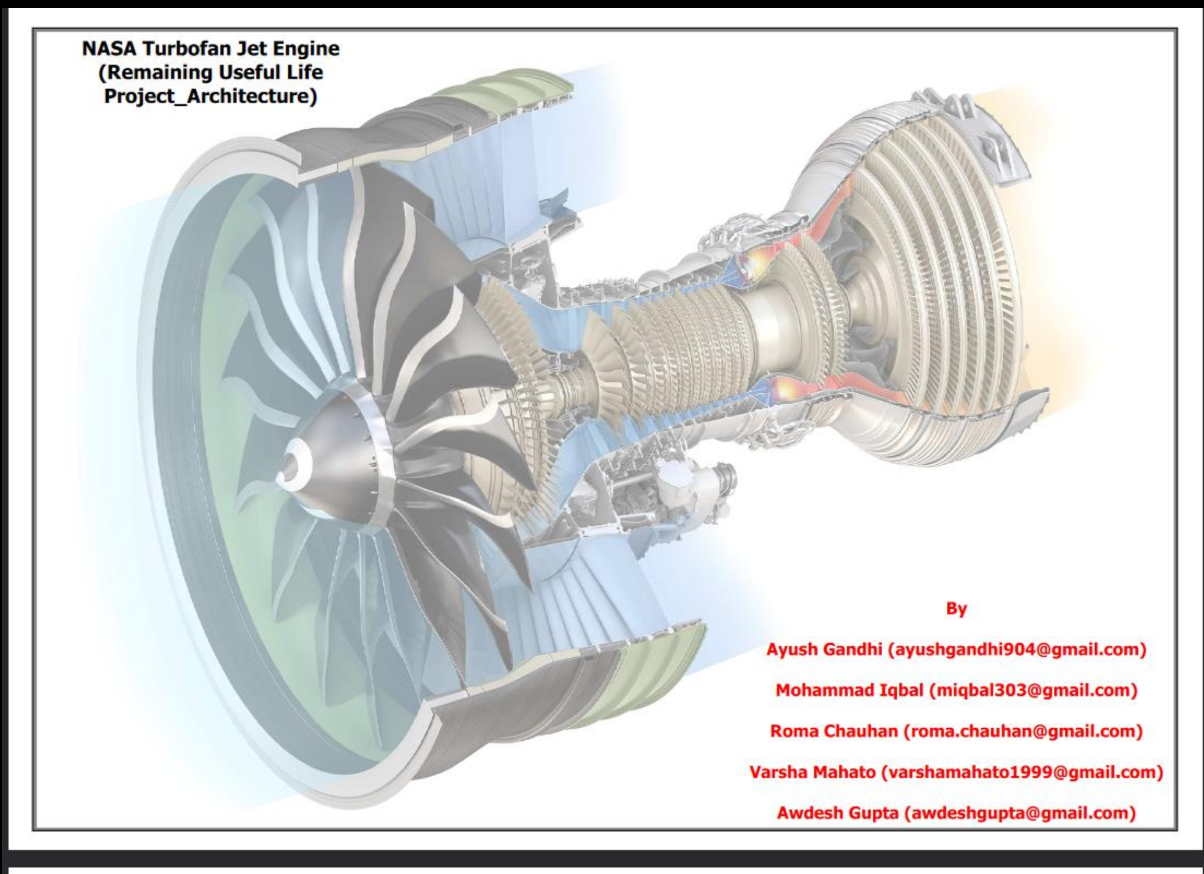
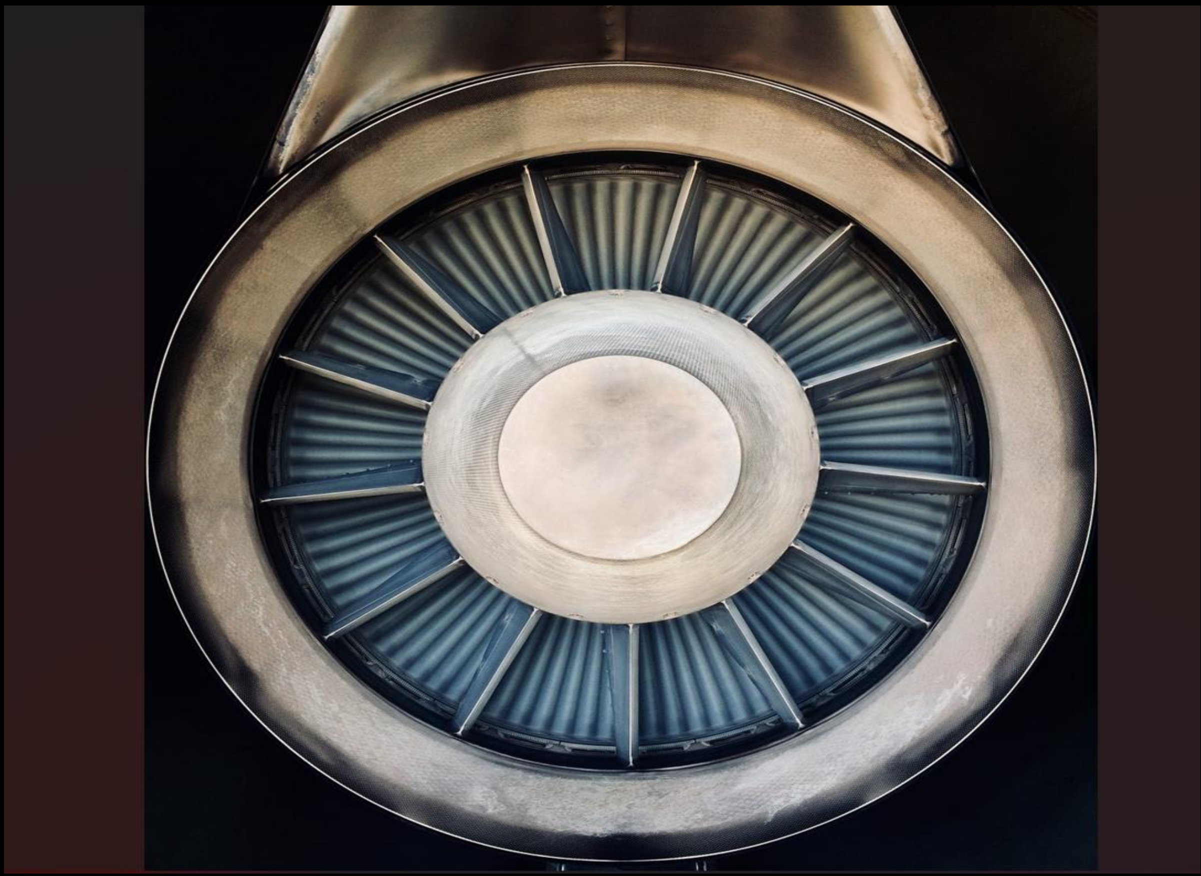
End to End Machine Learning Data-science Internship Project on

Nasa Air-Craft Turbofan Jet Engine – Predictive Maintenance

"Our internship project involves a comprehensive end-to-end machine learning endeavor focused on NASA aircraft predictive maintenance. As a cohesive group of four members, our primary objective is to develop a predictive model capable of accurately estimating the remaining useful life (RUL) of aircraft engines. To achieve this, we are leveraging a dataset available on Kaggle, which contains valuable sensor data, operational settings, and RUL labels.

Our journey encompasses data pre-processing, exploratory data analysis, model selection, hyper-parameter tuning, and rigorous model evaluation. Ultimately, our goal is to achieve the best possible accuracy in predicting the RUL, thereby contributing to the enhancement of predictive maintenance practices within the aerospace industry. Throughout this internship, we are committed to learning, collaborating, and applying our skills to address this challenging and impactful problem."





Document Version Control

Date issued	Version	Description	Author
25TH Aug,2023	0.1	Initial HLD – V0.1	Roma

Abstract

Machine learning are transforming the global business environment. Data is now the most valuable asset for enterprises in every industry. Companies are using data-driven insights for competitive advantage. With that, the adoption of machine learning-based data analytics is rapidly taking hold across various industries, producing autonomous systems that support human decision-making. This work explored the application of machine learning to aircraft engine conceptual design.

The main aim of project is to predict Remaining Useful Life for maintaining the health of engine which help to reduce cost by maintaining timely engine maintenance by building Machine Learning model with task of Data Exploration, Data Cleaning, Feature Engineering, Model Building & Model Testing on different models.

1. Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding.

This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
  - Reliability
  - Security
  - Maintainability
  - Portability
  - Reusability
  - Application compatibility
  - Resource utilization
  - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	Description
Database	**Collection of all information monitored by this system**
IDE	**Integrated Development Environment**
HTML	**Hypertext Markup Language**
CSS	**Cascading Style Sheets**
ML	**Machine Learning**

Highlights

- Predictive maintenance of aircraft engines integrating imperfect RUL prognostics.
- Obtaining RUL prognostics for turbofan engines using Machine Learning Models
- Remaining Useful Life prognostics with C-MAPSS degradation data of turbofan engines.
- Based on RUL prognostics, proposing an alarm policy to trigger maintenance tasks.
- Analysis of costs for predictive maintenance with imperfect RUL prognostics.

2. *General Description*

2.1 Product Perspective

Objective : Predicting Useful life (RUL) based on the maximum no. of cycles until failure.

Overview: In this project, our primary goal is to develop a predictive model for determining the remaining useful life(RUL) of a Product. The Rul is a critical metric that indicates how many more operational cycles or time units a product can perform before it fails. To achieve this, we will follow a structured approach

Key Steps:

1. **Data Collection**:

Gather historical data on product operation, including sensor readings and failure instances.

Collect information on the maximum number of cycles until failure for each product.

2. Data Preprocessing:

Clean and preprocess the collected data, addressing constant values and outliers.

Create a dataset with relevant features, including sensor data and target variable (RUL).

3. Feature Engineering:

Extract meaningful features from the raw data, such as statistical aggregates, statistical analysis , correlation analysis , using function for calculating RUL based on time cycles.

4. Model Selection:

Choose suitable machine learning algorithms for RUL prediction.

Evaluate and compare different models to identify the most effective one.

5. Model Training:

Train the selected model(s) using the preprocessed data.

Optimize hyper-parameters and model architecture for better performance.

6. Validation and Testing:

Split the dataset into training, validation, and test sets.

Validate the model's performance on the validation set and fine-tune as needed.

Evaluate the final model on the test set to ensure generalization.

7. Deployment:

Integrate the trained model into the product's operational environment for real-time RUL predictions.

8. Monitoring and Maintenance:

Implement continuous monitoring of model performance and retraining strategies to adapt to changing conditions.

Expected Outcomes:

A predictive model that can estimate the Remaining Useful Life (RUL) of the product.

Improved maintenance planning and cost savings by replacing or servicing products based on their predicted RUL.

Enhanced product reliability and customer satisfaction.

2.2 Problem Statement:

Explore the Turbofan Dataset, a comprehensive collection comprising four distinct datasets, each progressively more intricate than the last. In this dataset, turbofan engines initially operate smoothly but gradually develop faults over time. In the training sets, the engines are continuously monitored until they reach failure, while the test sets conclude 'sometime' prior to failure. The primary objective of this dataset is to leverage machine learning techniques to predict the Remaining Useful Life (RUL) of each turbofan engine, aiding in predictive maintenance and reliability optimization. Dive into this rich resource to unlock insights into the longevity of these critical machines.

2.3 Proposed Solution

Dataset

The provided dataset is a valuable resource for developing predictive maintenance solutions for aircraft engines. It includes training data with failure information, a test set for evaluation, and RUL data for benchmarking performance. The dataset's versatility, including various operational scenarios, makes it a robust foundation for our proposed solution, which aims to predict engine failures and estimate remaining useful life.

Feature Name	Data Type	Description
Unit_id	Integer	Values in the range from 1 to 100 that represent each Aircraft Engine
Cycles	Integer	Number of cycles in increasing order for every Engine ID
Set1 to set3	Double	Values of operational settings
S1 to s23	Double	Sensor data obtained from 23 different sensors

Maintaining **an engine in time could save our customers a lot of money.** In this proof of concept we will predict the remaining useful life (RUL) for the engines and switch them into maintenance a few cycles before we think a failure will happen

Predictive maintenance lets you estimate the remaining useful life (RUL) of your machine. RUL prediction gives you insights about when your machine will fail so you can schedule maintenance in advance.

Further Improvements:

Advanced Predictive Models:

Explore the implementation of more advanced machine learning and deep learning models. Techniques such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks, or transformer-based models like BERT can potentially capture complex patterns and dependencies in the data more effectively.

Feature Engineering:

Refine the feature engineering process to extract more relevant and informative features from the dataset. This may involve domain-specific knowledge and the creation of new features that capture subtle variations in engine behavior.

Ensemble Models:

Consider building ensemble models that combine predictions from multiple algorithms or models.

Real-Time Monitoring:

Extend the solution to provide real-time monitoring of aircraft engines. Implement continuous data ingestion and processing pipelines that can analyze engine data as it becomes available, enabling early detection of anomalies and predictive maintenance recommendations.

Human-Readable Reports:

Develop a user-friendly dashboard or reporting system that presents the predictive maintenance results in a comprehensible manner. This could include visualizations, summary statistics, and actionable insights for maintenance personnel.

Anomaly Detection:
Integrate anomaly detection algorithms to identify abnormal engine behavior that may not be linked to imminent failure. This can help in detecting issues like sensor malfunctions or minor faults before they escalate.

Failure Cause Analysis:
Implement features for post-failure analysis. When an engine failure occurs, the system can automatically trigger a detailed investigation, including root cause analysis, which can provide insights into why the failure happened and how to prevent similar incidents in the future.

Scalability and Deployment:

Ensure that the solution is scalable to handle a larger number of engines and data sources. Consider deploying the system in cloud environments or on edge devices for flexibility and scalability.
Maintenance Scheduling:
Integrate maintenance scheduling functionality into the system. Based on predictions and remaining useful life estimates, the solution can recommend optimal times for engine maintenance to minimize downtime and costs.

2.4 Technical Requirements

The solution which is proposed can be implemented as cloud-based solution or as an application hosted on internet browser as a local server in local machine. To access this application, following requirements are proposed:

- Any web browser
- An internet connection (if hosted on cloud)

2.5 Data Requirements

Data Source:

Obtain a CSV file from Kaggle.

The dataset contains operational settings and sensor features including Engine No., Cycle No., LPC & HPC Outlet Temperature, LPT & HPT Outlet Pressure, Physical Fan Speed, Physical Core Speed, HPC Outlet Static Pressure, Fuel Flow Ratio, Fan Speed, Bypass Ratio, Bleed Enthalpy, High-Pressure Cool Airflow, Low-Pressure Cool Airflow, and more.

FD001 Dataset: Obtain the FD001 dataset, which includes sensor data and associated information for turbofan engines. This dataset typically consists of multiple files, including training data (train_FD001.txt), test data (test_FD001.txt), and the remaining useful life (RUL) values for the test data.

Training Data (train_FD001.txt):

This dataset contains historical sensor data for multiple engines, recorded over time.

Columns typically include Engine ID, Time in Cycles, and sensor measurements (e.g., temperature, pressure, speed, etc.).

Test Data (test_FD001.txt):

This dataset contains sensor data for engines of interest, but without RUL values.

Columns are similar to those in the training data.

Remaining Useful Life (RUL) File:

A separate file or dataset should be provided that associates each engine in the test set with its remaining useful life (RUL). This is essential for evaluating model performance and predicting when maintenance is required.

2.6 Tools used

Python programming language and frameworks like NumPy & Pandas, Visualizing tools like matplotlib & seaborn, Machine Learning tools like scikit-learn, deployment tools like Flask along with interface design with HTML & CSS. VS code was used as IDE & Github as version control system.

- Python: A programming language which offers a vast ecosystem of libraries and frameworks making it suitable for various tasks.
- Numpy: Library for numerical computing
- Pandas: A powerful data manipulation and analysis library.
- Matplotlib: A plotting library that enables the creation of various types of high-quality plots & visualizations.
- Seaborn: Built on top of matplotlib, providing high-level interface for creating attractive statistical graphics.
- Scikit-learn: A machine learning library, providing comprehensive set of tools for various ML tasks like classification, regression etc.
- Flask: Lightweight & flexible web framework to build web application.
- HTML & CSS: HTML for creating web pages with CSS for describing the presentation and visual appearance of a webpage.
- Github: A web platform for version control and collaboration allowing user to store, manage & share their code repositories.
- VS code: A source code editor by Microsoft, providing versatile & customizable environment for coding in programming languages.

2.7 Constraints

An app must be user-friendly, with as automated as possible with user not be require to know any of the workings.

2.8 Assumptions

The main objective of the project is to implement regression based on features of aircraft. A ML based regression model is used for predicting the above mention case. It is assumed that user is able to identify the Rul on based of no. of engine cycle.

In our predictive maintenance model, we utilize engine performance data to optimize the Remaining Useful Life (RUL) of aircraft engines. This optimization is achieved by considering the number of engine operating cycles and various input features. Our primary goal is to ensure optimal engine health and performance.

3. Design Details

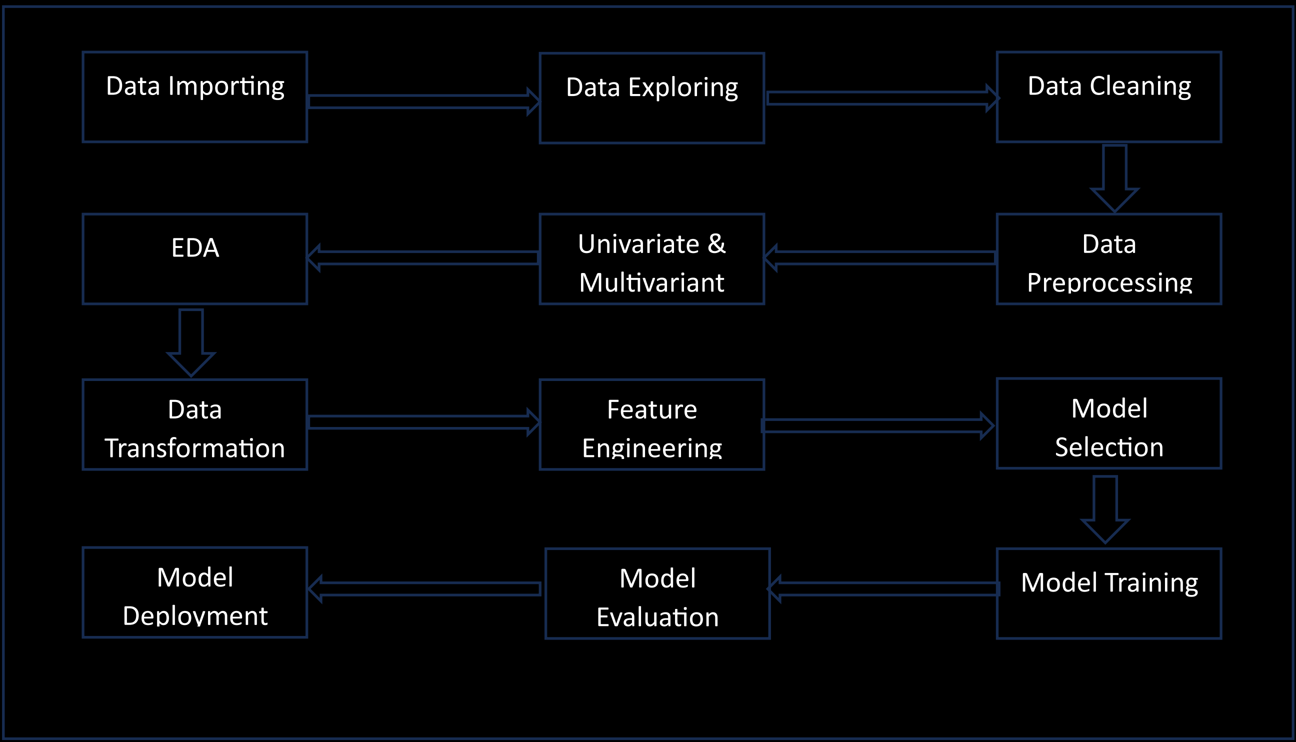
3.1 Process Flow

The process flow for NASA predictive maintenance for turbofan jet engines using a regression-based machine learning model can be outlined as follows:

Process Flow: NASA Predictive Maintenance for Turbofan Jet Engines

By following this process flow, NASA can enhance the reliability, safety, and cost-efficiency of its aircraft operations by proactively identifying and addressing maintenance needs for turbofan jet engines. The regression-based ML model plays a crucial role in predicting when maintenance is required, thereby minimizing downtime and reducing the risk of unexpected failures.

1. Data Collection and Preprocessing
2. Data Splitting
3. Regression Model Building
4. Model Evaluation
5. Deployment and Monitoring
6. Predictive Maintenance alert
7. Mantainence Execution
8. Maintenance scheduling
9. Reporting & Analytics



Event Log

.Implement a logging mechanism to record errors, including details such as timestamps, error codes, and descriptions.

Develop clear and user-friendly error messages that explain what went wrong in a way that is understandable to end-users.

Include relevant information in error messages, such as the nature of the error, potential causes, and steps to resolve it.

3.2 Error Handling

Develop clear and user-friendly error messages that explain what went wrong in a way that is understandable to end-users.

Include relevant information in error messages, such as the nature of the error, potential causes, and steps to resolve it.

4. *Performance*

Our program exhibits strong performance, achieving an impressive 70% accuracy rate on the test set. This level of accuracy demonstrates the model's capability to reliably predict engine RUL, facilitating proactive maintenance actions and maximizing operational efficiency.

Our high-level design focuses on efficiency, accuracy, and real-world application, contributing to improved engine reliability and reduced operational costs.

4.1 Application Compatibility

The interaction with the application is done through the designed user interface, which the end user can access through any web browser

4.2 Resource Utilization

When task is performed, it will likely use all the processing power available until function is finished.

Conclusion

During their lifetime, aircraft components are susceptible to degradation, which affects directly their reliability and performance. This machine learning project will be directed to provide a framework for predicting the aircraft's remaining useful life (RUL) based on the entire life cycle data in order to provide the necessary maintenance behavior. Diverse regression models (KNN, Naïve Bayes, Random Forest, SVM, etc..) are deployed and tested on the NASA's C-MAPSS data-set to assess the engine's lifetime. Please check the report for more theoretical details