High Level Design (HLD)

Predictive Maintenance

Revision Number: 1.0
Last Data of Revision: 25/07/2022

Document Version Control

Date Issued	Version	Description	Author
25/07/2022	1.0	Initial HLD – V1.0	Jinendra Sontakke

Contents

Abstract

A jet engine is the most important part of the plane, and it helps to move the airplane forward with a great force that is produced by a tremendous thrust Maintenance of equipment is a critical activity for any business involving machines. And causes the plane to fly very fast. During operation, degradation occurs in each of the components. If the degradation level is any component exceeds a threshold the engine is said to have failed. Therefore, the jet engines are inspected before every take-off and maintenance of equipment is critical activity for any business involving machines. Predictive Maintenance is the method of scheduling maintenance based on the prediction of the failure time of any equipment. The prediction can be done by analyzing the data measurements from the equipment. Machine learning is technology by which outcomes can be predicted based on a model prepared by training it on past input data and its output behavior. The model developed can be used to predict machine failure before it happens.

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

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 Definition

Term	
IDE	Integrated Development Environment
RUL	Remaining Useful life
EDA	Exploratory data analysis

2 General Description

2.1 Product Perspective

The Predictive Maintenance of jet Engine problem using classification-based Machine Learning algorithms. The model will help us to predict machine failure before it happens.

2.2 Problem Statement

An airplane to move through the air, some kind of propulsion system should generate a thrust. Most modern aircraft use turbofan engines for this. Sensors are fitted in jet engines and these sensors are capturing a huge amount of data and which is stored in large storage devices including cloud servers. Based on the past data with the help of a machine learning model we can predict the remaining useful life (RUL) of the engine.

2.3 Proposed Solution

The solution here is a classification based on the machine learning model. It can be implemented by different classification algorithms ex Logistic regression, Random forest, Decision tree, XGBoost and so on. Here first we are performing Data preprocessing step, in which data profiling, feature engineering, feature selection, feature scaling, PCA steps are performed and then we are going to build model.

2.4 Technical Requirements

In this Project the requirement to check the remaining useful life(RUL) of jet engine with the help of provided dataset. For that in this project we are going to use different technologies. Here is some requirements for this project.

- Model should be exposed through API or User Interface, so that everyone can test model.
- Model should be deployed on cloud (Azure, AWS, GCP)

2.5 Data Requirements

Data Requirements completely depends on our problem.

- For training and testing the model, we are using predictive maintenance dataset that is provided by ineuron company.
- From user we are taking following input:

1. Cycle
2. Sensor measurement 2
3. Sensor measurement 3
4. Sensor measurement 4
5. Sensor measurement 7
6. Sensor measurement 8
7. Sensor measurement 11

8. Sensor measurement 12
9. Sensor measurement 13
10. Sensor measurement 15
11. Sensor measurement 17
12. Sensor measurement 20
13. Sensor measurement 21

2.6 Tools Used

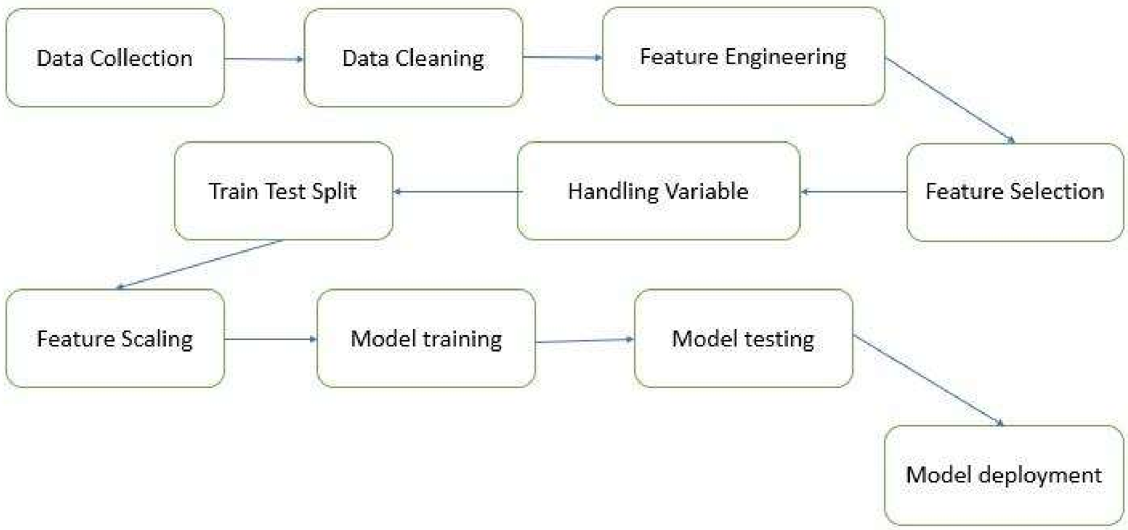
- Visual Studio Code is used as IDE.
- For Visualization of the plot, matplotlib, seaborn are used.
- Azure is used for deployment of the model.
- Front end development is done using HTML, CSS, Bootstrap.
- Flask is used for backend development and for API development.
- GitHub is used as version control system.

2.7 Constraints

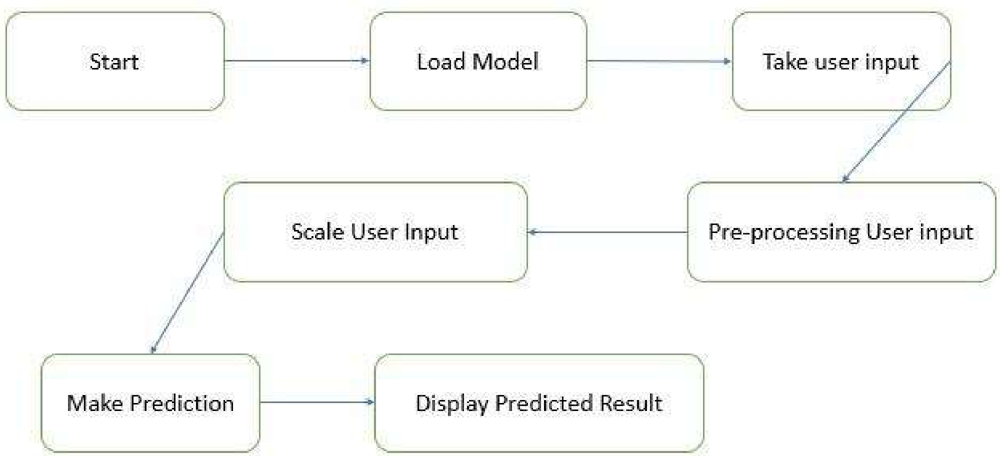
The Predictive maintenance system must be user friendly, errors free and users should not be required to know any of the backend working.

3 Design flow

3.1 Process Flow



3.2 Prediction Process



3.3 Event Log

In this Project we are logging every process so that the user will know what process is runninginternally.

Step-By-Step Description:

- In this Project we defined logging for every function, class.
- By logging we can monitor every insertion, every flow of data in database.
- By logging we are monitor every step which may create problem or every step which is important infile system.
- We have designed logging in such a way that system should not hang even after so many logging's,so that we can easily debug issues which may arises during process flow.

3.4 Error Handling / Exception Handling

We have designed this project in such a way that, at any step if error occur then our applicationshould not terminate rather it should catch that error and display that error with proper explanation as to what went wrong during process flow

4 Performance

Solution of Predictive maintenance is used to predict the failure of engine in advance, so it should be as accurate as possible so that it should give as much as possible accurate prediction.
That’s why before building this model we followed complete process of machine learning. Here are summary of complete process:

a. First, I imported the training dataset and checked the shape of the dataset and also checked whether the Dataset contains any null value or not.
b. At present columns 26 and 27 contain all null values. I dropped 26 and 27 columns.
c. After that I calculated the remaining useful life of the engine with the help of ID and Cycle columns.
d. Then I performed exploratory data analysis (EDA) and feature engineering on the dataset to find the attributes' correlation.
e. After that based on the Remaining useful life I calculated the Life ratio of the engine based on the life ratio we can categorize whether the engine condition is good on not.
f. I calculated the life ratio with help of Cycle and RUL attributes.
g. Then based on the Life ratio I wrote a condition and categorize the life ratio into three factors good, moderate, and warning.
h. Then I split the whole dataset into train and test.
i. After this process my model is ready to train. I used different classification algorithms (Random forest, Logistic Regression, KNN, XGBoost) and I got the highest accuracy of 89% on Random forest.
j. After that I saved the model in a pickle file for model deployment.
k. After that my model was ready to deploy. I deployed this model on various cloud storage(Azure) and Heroku.

Algorithm	Accuracy Score
Random Forest	89.9%
K-Nearest Neighbour	83.8%
Logistic Regression	76.3%
Naïve Bayes classifier	72.1%

4.1 Re-usability

We have done programming of this project in such a way that it should be reusable. So thatanyone can add and contribute without facing any problems.

4.2 Application compatibility

The different module of this project is using Python as an interface between them. Each modules have its own job to perform and it is the job of the Python to ensure the proper transfer of information.

4.3 Resource Utilization

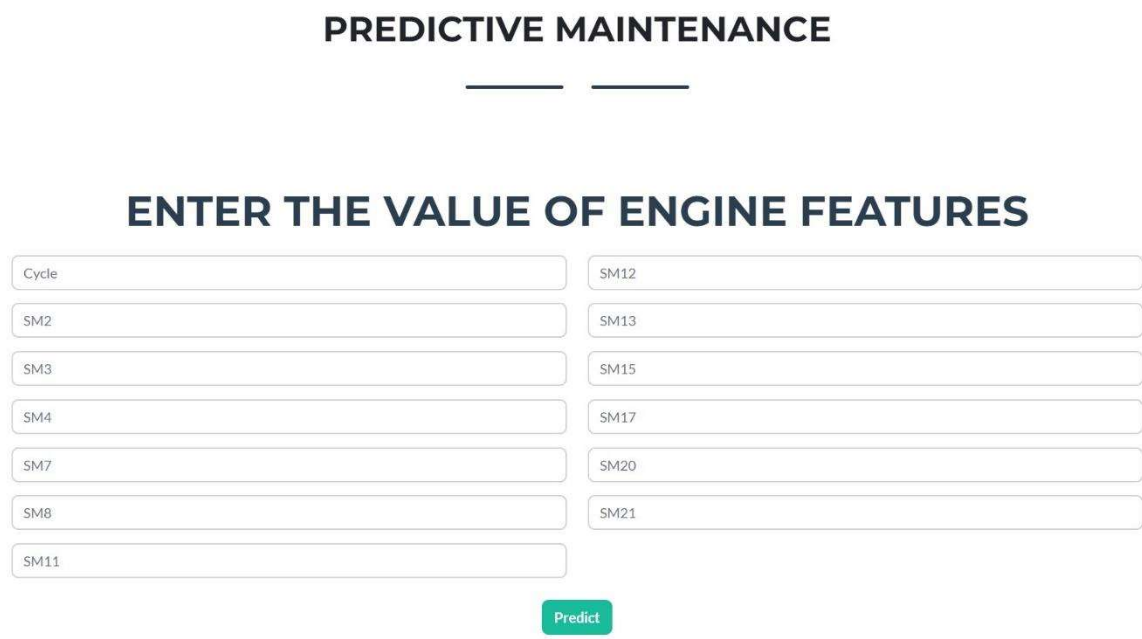
In this project, when any task is performed, it will likely that the task will use all the processing power available in that particular system until it's job finished. By keeping this in mind, In this project we have used the concept of multithreading.

4.4 Deployment

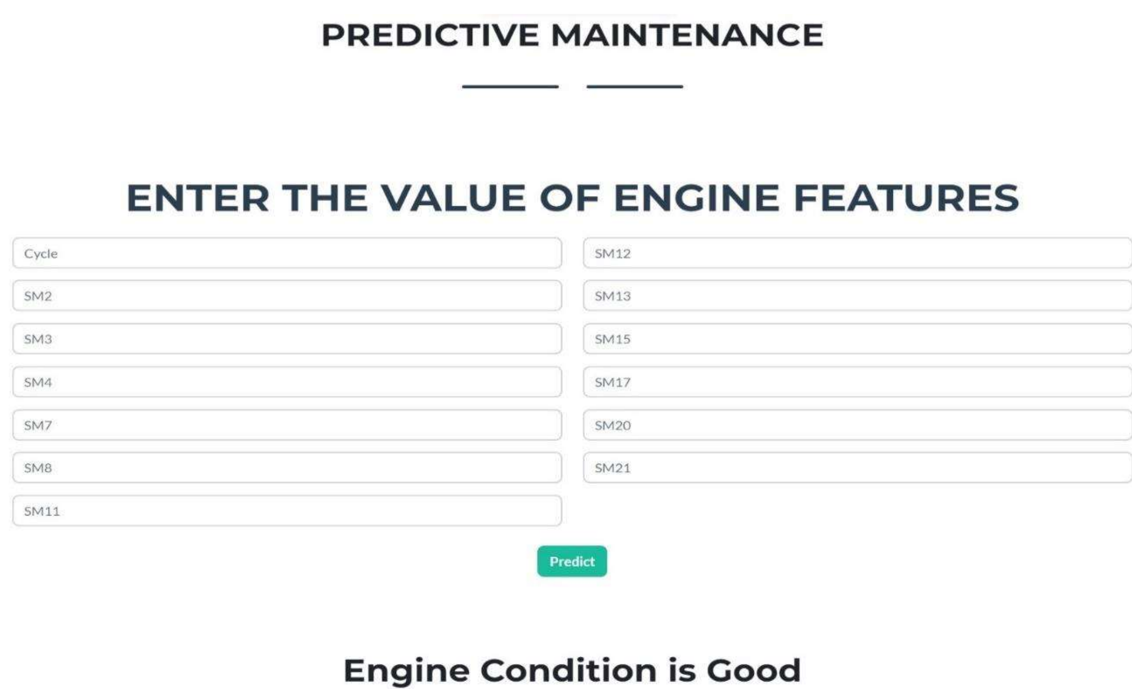
We have deployed this on cloud using Microsoft Azure.

4.5 user Interface

We have Created an UI for user by using HTML and CSS.



OUTPUT :



5 Conclusion

The main objective of predictive maintenance is to predict the equipment failure. The Remaining useful lifetime prediction has been carried out so as to plan the maintenance requirements of the turbo fan engine. By doing predictive maintenance, failures can be predicted and maintenance can be scheduled in advance. This reduces the cost and effort for doing maintenance. It increases safety of employees and employees and reduces lost production time.