HIGH LEVEL DESIGN (HLD)

Predictive Maintenance

Prajwal Krishna

Document Version Control

Date Issued	Version	Description	Author
01.10.2023	V1.0	Initial HLD- V1.0	Prajwal Krishna

Contents

Abstract

This document presents a comprehensive overview of a predictive maintenance classification machine learning project aimed at determining the likelihood of machine failure. In an era where minimizing downtime and maximizing operational efficiency are paramount, predictive maintenance systems have emerged as indispensable tools for industries reliant on machinery. Leveraging advanced machine learning algorithms, our project endeavors to accurately classify whether a machine is likely to experience a failure, enabling proactive maintenance interventions to mitigate potential disruptions and costly downtime. The abstract encapsulates the essence of our project, highlighting its significance in enhancing operational efficiency and reducing maintenance costs.

1.0 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level document is to add necessary details to current project description to represent a suitable model for coding. This document is used as a reference manual for how the model interacts at a high-level.

The HLD will

- Presents all design aspects and defines 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 document presents the structure of the system, such as the database architecture, application architecture, and technology architecture. The HLD uses non-technical to middle-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	Description
Database	Collection of all the information
IDE	Integrated Development Environment
API	Application Programming Interface
KPI	Key Performance Indicator
VSCode	Visual Studio Code
EDA	Exploratory Data Analysis

2.0 General Description

2.1 Product Perspective

The problem addressed in this project lies in the development of an effective machine learning-based classification model that can accurately predict the likelihood of machinery failure. By harnessing historical data on machine health, performance, and maintenance records, the objective is to train a predictive model capable of distinguishing between normal operational conditions and impending

2.2 Problem Statement

In industrial settings, the occurrence of machinery failures can lead to significant financial losses, operational disruptions, and safety hazards. Traditional reactive maintenance practices, where repairs are conducted only after a breakdown occurs, are not only costly but also inefficient, often resulting in unplanned downtime and production delays. Hence, there is a critical need for predictive maintenance systems capable of preemptively identifying potential failures before they occur.

2.3 Proposed Solution

Data Collection and Preprocessing

Model Selection and Training

Imbalanced Data Handling

Model Validation and Tuning

Deployment and Integration

Continuous Monitoring and Improvement

2.4 Further Improvements

2.5 Technical Requirements

The solution can be a cloud-based or application hosted on an internal server or even be hosted on a local machine. For accessing this application below are the minimum requirements:

- Good internet connection.
- Browsers such as Brave, chrome, Mozilla firefox

For training model, the system requirements are as follows:

- +4 GB RAM preferred
- Operation System: Windows, Linux, Mac
- vim / Visual Studio Code / Jupyter notebook

2.6 Data Requirements

- Comma separated values (CSV) file.
- Input file feature/field names and its sequence should be followed as per decided.

2.7 Tools Used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Plotly, Flask are used to build the whole model.

- Pandas is an open-source Python package that is widely used for data analysis and machine learning tasks.

- NumPy is the most commonly used package for scientific computing in Python.
- Plotly is an open-source data visualization library used to create interactive and quality charts/graphs.
- Scikit-learn is used for machine learning.
- Flask is used to build API.
- VS Code is used as IDE (Integrated Development Environment)
- GitHub is used as a version control system.
- Front end development is done using HTML/CSS.
- Heroku is used for deployment of the model.

2.8 Constraints

2.9 Assumptions

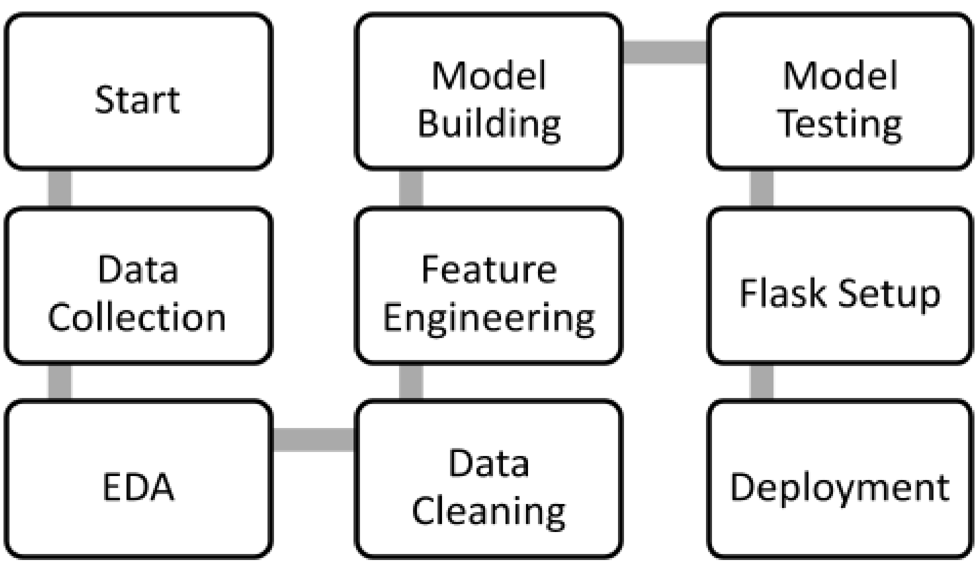
Sufficient Data Availability: It is assumed that there is access to a comprehensive dataset containing historical records of machine health, performance metrics, sensor readings, and maintenance logs.

Data Quality: It is assumed that the collected data is of sufficient quality, free from significant errors, missing values, and inconsistencies that could adversely affect model performance.

Relevance of Features: It is assumed that the selected features for training the predictive maintenance model accurately capture the underlying patterns indicative of potential machinery failures.

3.0 Design Details

3.1 Process Flow



3.2 Event Log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

- The system identifies at what step logging is required.
- The system should be able to log each and every system flow.
- Developers can choose logging methods. You can choose database logging.

System should not hang out even after using so many loggings.

4.0 Performance

4.1 Reusability

The entire solution will be done in modular fashion and will be API oriented. So, in the case of the scaling of the application, the components are completely reusable.

4.2 Application Compatibility

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

4.3 Deployment

5.0 Conclusion

In conclusion, the development of a predictive maintenance classification model offers significant potential benefits for industries reliant on machinery by enabling proactive identification of potential failures and preemptive maintenance interventions. Through the execution of the proposed solution outlined in this document, organizations can effectively leverage machine learning techniques to optimize maintenance schedules, minimize downtime, and enhance operational efficiency.

By collecting and preprocessing relevant data, selecting appropriate models, handling imbalanced datasets, and validating and fine-tuning the model's performance, organizations can deploy robust predictive maintenance systems capable of accurately identifying impending machinery failures. Furthermore, the seamless deployment and integration of the predictive maintenance system into existing workflows ensure its practical applicability and utility in real-world industrial settings.

However, it is essential to recognize that the successful implementation of predictive maintenance relies on continuous monitoring, feedback incorporation, and iterative improvement processes. By embracing a culture of continuous improvement and adapting to evolving operational requirements, organizations can maximize the long-term benefits of predictive maintenance systems and maintain a competitive edge in today's dynamic industrial landscape.

In summary, the development and deployment of a predictive maintenance classification model represent a proactive approach to asset management, offering substantial cost savings, operational efficiencies, and enhanced safety for industries operating in machinery-intensive environments.