Architecture

Predictive Maintenance

Revision Number: 1.0
Last Data of Revision: 24/07/2022

Jinendra Sontakke

Document Version Control

Date Issued	Version	Description	Author
24/07/2022	V1.0	HLD – V1.0	Jinendra Sontakke

Approved status:

Version	Review date	Reviewed by	Approach by	comments

Contents

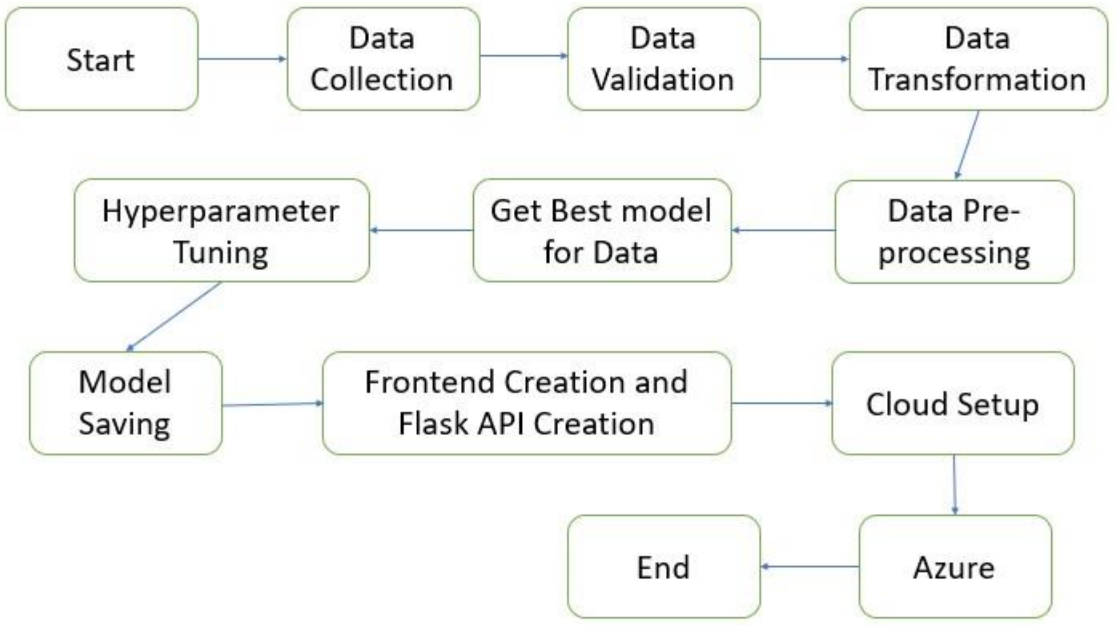
1 Introduction

1.1 Why this Architecture Design Document?

The purpose of this document is to provide a detailed architecture design of the Back Order Prediction Project by focusing on four key quality attributes:
usability, availability, maintainability, testability.

This document will address the background for this project, and the architecturally significant function requirements. The intension of this document is to help the development team to determine how the system will be structured at the highest level. Finally, the project coach can use this document to validate that the development team is meeting the agreed-upon requirements during the evaluation of the team’s efforts.

2 Architecture



3 Architecture Description

3.1 Data Description

We have 1.6 lakh Dataset row columnar data includes Engine ID, Cycle, Operational Setting1, Operational Setting2, Operational Setting3, Sensor Measure 1, Sensor Measure 2, Sensor Measure 3, Sensor Measure 4, Sensor Measure 5, Sensor Measure 6, Sensor Measure 7, Sensor Measure 8, Sensor Measure 9, Sensor Measure 10, Sensor Measure 11, Sensor Measure 12, Sensor Measure 13, Sensor Measure 14, Sensor Measure 15, Sensor Measure 16, Sensor Measure 17, Sensor Measure 18, Sensor Measure 19, Sensor Measure 20, Sensor Measure 21, Sensor Measure 22, Sensor Measure 23. This is given in the text format we converted it into comma separated value format (.csv) . these data is given by company team which contains both the test data and train data.

3.2 Data Cleaning / Data Transformation

In this cleaning process, we have cleaned up all the data because data which contain null value because of this bad format the machine will not recognized it. So dara cleaning is after that data transformation process come where we will convert our original dataset which is in text format to CSV format.

3.3 Exploratory Data Analysis

In EDA we have seen various insights from the data so we have selected which column is most important and dropped some of the columns by observing them spearman rank correlation and plotting their heatmap.

3.4 Event Log

The system should log every event so that the user will know what process is running Internally. Logging is implemented using python's standard logging library. Initial step-by-step description:-

- The system should be able to log each and every system flow.
- System must be able to handle logging at greater scale because it helps debugging the issue and hence it is mandatory to do.

3.5 Data Pre-Processing

Data Pre-processing steps we could use are Null value handling, Categorical to Numerical Transformation of columns , Splitting Data into Dependent and Independent Features , Robust Scaling , Remove those columns which are does not participate in model building Processes , Imbalanced data set handling, Handling columns with standard deviation zero or below a threshold, etc.

3.6 Model Creation / Model Building

After cleaning the data and completing the feature Engineering/ data Per processing. we have done splitted data in the train data and test data using method build in pre-processing file and implemented various Classification Algorithm like RandomForestClassifier and XgBoostClassifier also calculated their accuracies on test data and train data.

3.7 Hyperparameter Tuning

In hyperparameter tuning we have implemented randomized search cv or grid search cv and from that we also implemented cross validation techniques for that. From that we have choose best parameters according to hyperparameter tunning and best score from their accuracies.

3.8 Model Dump

After comparing all accuracies I have choose hyper parameterized random forest classifier as our best model by their result so I have dumped this model in a pickle file format.

3.9 Data from User

Here we will collect user's requirement to predict whether a engine is good or not their Engine Cycle, Sensor Measure 2, Sensor Measure 3, Sensor Measure 4, Sensor Measure 7, Sensor Measure 8, Sensor Measure 11, Sensor Measure 12, Sensor Measure 13, Sensor Measure 15, Sensor Measure 17, Sensor Measure 20 and Sensor Measure 21.

3.10 Data Validation

Here Data Validation will be done, given by the user

3.11 Model Call for specific input

Based on the User input will be throwing to the backend in the variable format then it converted into pandas data frame then we are loading our pickle file in the backend and predicting whether engine condition is good or not as an output and sending to our html page.

3.12 User Interface

In Frontend creation we have made a user interactive page where user can enter their input values to our application. In these frontend page we have made a form which has beautiful styling with CSS and bootstrap. These HTML user input data is transferred in variable format to backend. Made these html fully in a decoupled format

3.13 Deployment

We will be deploying the model with the help of Azure cloud platforms.