Low Level Document (LLD)

Predictive Maintenance

Rajesh Gottoju

Contents

Abstract

Abstract

In industry, prognostics and health management are key topics for anticipating asset state and avoiding downtime and breakdowns. Run-to-Failure simulation data from turbofan jet engines is included. The C-MAPSS software was used to simulate engine degradation. Four separate sets of operational conditions and fault modes were simulated in four different ways. To characterize fault progression, record numerous sensor channels. The Prognostics CoE at NASA Ames provided the data set. The main goal is to predict the remaining useful life (RUL) of each engine. RUL is equivalent of number of flights remained for the engine after the last data point in the test dataset.

1.Introduction

Why this Low-Level Design Documentation?

The purpose of this documentation is detailed description of restaurant rating prediction system which will explain the purpose and the feature of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will perform under different parameters. This document is intended for both the stack holders and developers of the system and will be proposed for the higher management for its approval.

This project can be delivered in three phases

Phase 1: Building Machine learning model depending on the requirements.

Phase 2: Integration of UI and database to all the functionalities.

Phase 3: Deployment of project on cloud.

Scope

This software system will be a web application, this system will be designed to predict the RUL based on user's input.

Constraints

This project is based on Aero space domain, this system can get excepted results.

Out Of Scope

System will not perform correctly if the data in good format

2. Technical Specifications

Data: Predictive Maintenance

Finalized: Yes

Data Set overview

20631 rows

26 columns

	id	cycle	op1	op2	op3	sensor1	sensor2	sensor3	sensor4	sensor5	...	sensor12	sensor13	sensor14	sensor15	sensor16	sensor17
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	521.66	2388.02	8138.62	8.4195	0.03	392
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	522.28	2388.07	8131.49	8.4318	0.03	392
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	522.42	2388.03	8133.23	8.4178	0.03	390
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	522.86	2388.08	8133.83	8.3682	0.03	392
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	...	522.19	2388.04	8133.80	8.4294	0.03	393
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
20626	100	196	-0.0004	-0.0003	100.0	518.67	643.49	1597.98	1428.63	14.62	...	519.49	2388.26	8137.60	8.4956	0.03	397
20627	100	197	-0.0016	-0.0005	100.0	518.67	643.54	1604.50	1433.58	14.62	...	519.68	2388.22	8136.50	8.5139	0.03	395
20628	100	198	0.0004	0.0000	100.0	518.67	643.42	1602.46	1428.18	14.62	...	520.01	2388.24	8141.05	8.5646	0.03	398
20629	100	199	-0.0011	0.0003	100.0	518.67	643.23	1605.26	1426.53	14.62	...	519.67	2388.23	8139.29	8.5389	0.03	395
20630	100	200	-0.0032	-0.0005	100.0	518.67	643.85	1600.38	1432.14	14.62	...	519.30	2388.26	8137.33	8.5036	0.03	396

20631 rows × 26 columns

Input schema

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20631 entries, 0 to 20630
Data columns (total 27 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   id        20631 non-null  int64
 1   cycle     20631 non-null  int64
 2   op1       20631 non-null  float64
 3   op2       20631 non-null  float64
 4   op3       20631 non-null  float64
 5   sensor1   20631 non-null  float64
 6   sensor2   20631 non-null  float64
 7   sensor3   20631 non-null  float64
 8   sensor4   20631 non-null  float64
 9   sensor5   20631 non-null  float64
 10  sensor6   20631 non-null  float64
 11  sensor7   20631 non-null  float64
 12  sensor8   20631 non-null  float64
 13  sensor9   20631 non-null  float64
 14  sensor10  20631 non-null  float64
 15  sensor11  20631 non-null  float64
 16  sensor12  20631 non-null  float64
 17  sensor13  20631 non-null  float64
 18  sensor14  20631 non-null  float64
 19  sensor15  20631 non-null  float64
 20  sensor16  20631 non-null  float64
 21  sensor17  20631 non-null  int64
 22  sensor18  20631 non-null  int64
 23  sensor19  20631 non-null  float64
 24  sensor20  20631 non-null  float64
 25  sensor21  20631 non-null  float64
 26  RUL       20631 non-null  int64
dtypes: float64(22), int64(5)
memory usage: 4.4 MB
```

Predicting

- The system displays RUL according to user's Input.
- The system prevents the set of inputs required from the user.
- The user gives required information.
- The system should able to predict the RUL According to the user input given.

Logging

✓ We have chosen File logging.

✓ System logs each and every system flow.

✓ Each and every user's input information is logged.

Database

The system stores each and every data given by the user or received on request to the database. We have used Cassandra.

Deployment

1.Heroku

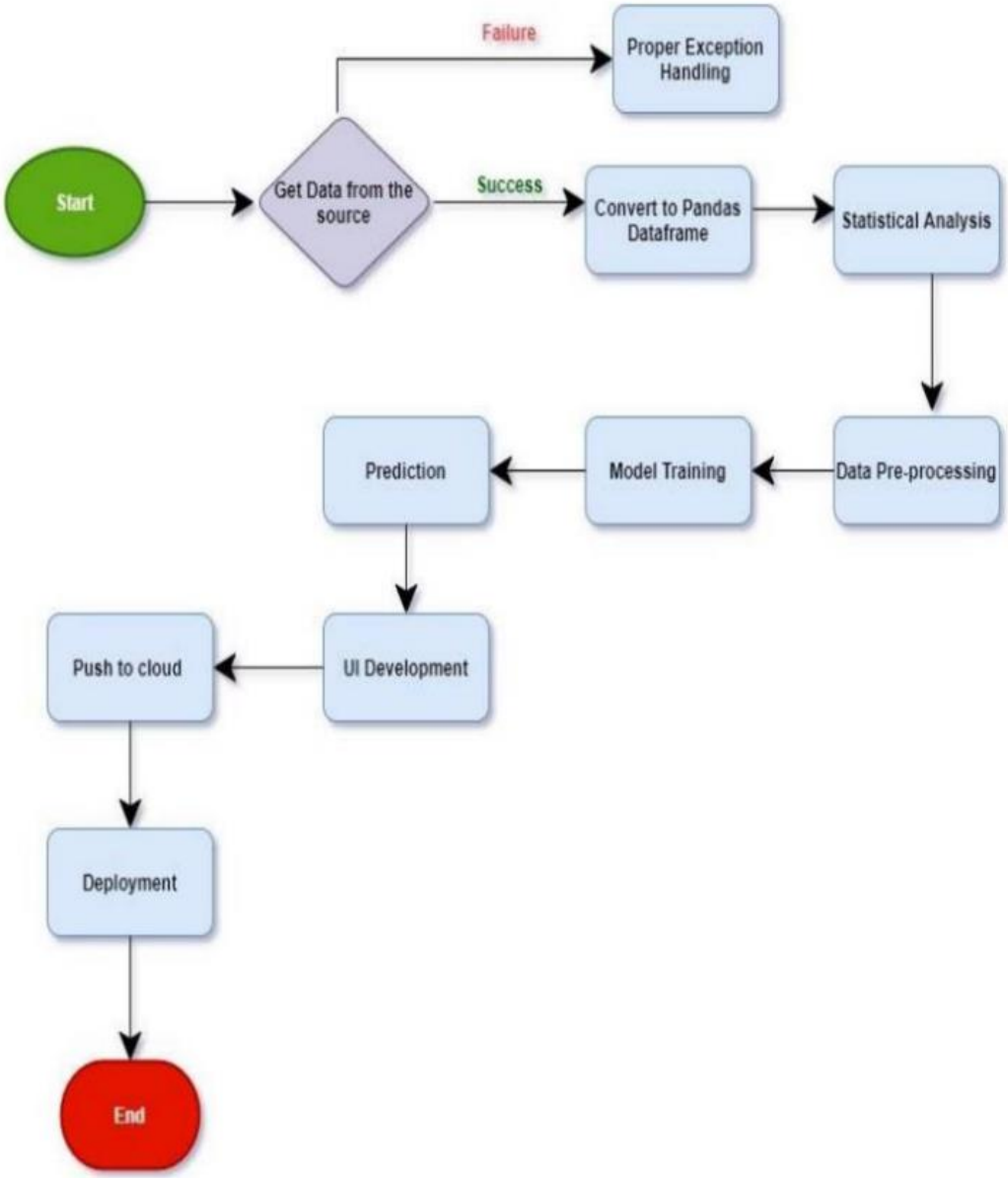
3.Technology Stack

* Python
* Stream lit
* Python Libraries
* Machine Learning algorithms

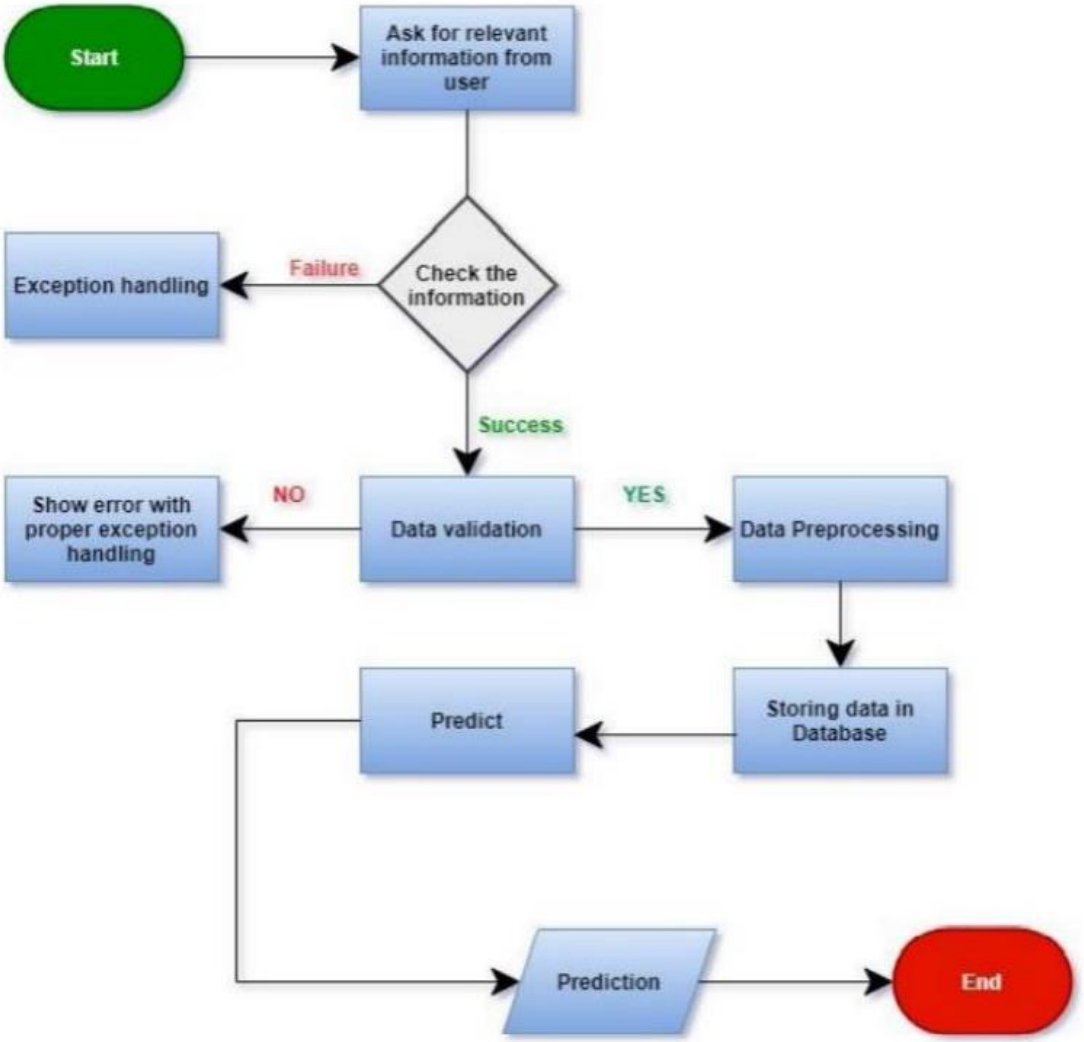
4. Proposed Solution

This system requires some sensor column values.

5.Model Training / Validation Workflow



6. User I/O Workflow



7.Test Cases.

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether user is able to edit all input fields	1. Application is accessible 2. User is logged in to the application	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. User is logged in to the application	User should get Submit button to submit the inputs
Verify whether user is presented with recommended results on clicking submit	1. Application is accessible 2. User is logged in to the application	User should be presented with recommended results on clicking submit
Verify whether the recommended results are in accordance to the selections user made	1. Application is accessible 2. User is logged in to the application	The recommended results should be in accordance to the selections user made