Project Report

Machine Learning-Based Predictive Analytics for Aircraft Engine

Team ID: PNT2022TMID522013

Team Leader: Janani M

Team Member 1: Harishankar S M

Team Member 2: Surya M

Team Member 3: Sujitha A J

1. **INTRODUCTION**
   1.1 Project Overview
   1.2 Purpose
2. **LITERATURE SURVEY**
   2.1 Existing problem
   2.2 References
   2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   3.1 Empathy Map Canvas
   3.2 Ideation & Brainstorming
   3.3 Proposed Solution
   3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   4.1 Functional requirement
   4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   5.1 Data Flow Diagrams
   5.2 Solution & Technical Architecture
   5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   6.1 Sprint Planning & Estimation
   6.2 Sprint Delivery Schedule
   6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   7.1 Feature 1
   7.2 Feature 2
   7.3 Database Schema (if Applicable)
8. **TESTING**
   8.1 Test Cases
   8.2 User Acceptance Testing
9. **RESULTS**
   9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
   Source Code
   GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

Engine failure is highly risky and needs a lot of time for repair. Unexpected failure leads to loss of money and time. Predicting the failure prior will save time, effort, money and sometimes even lives. The failure can be detected by installing the sensors and keeping a track of the values. The failure detection and predictive maintenance can be for any device, out of which we will be dealing with the engine failure for a threshold number of days. The project aims to predict the failure of an engine by using Machine Learning to save loss of time & money thus improving productivity.

1.2 Purpose

In aviation, the use of maintenance data is highly critical in the analysis of reliability and maintenance costs. This is because predictive maintenance scheduling can be planned in line with estimates. The main target of predictive maintenance is to predict equipment failures and planning strategies for spare parts of the system components to analyse the reliability and maintainability of a complex repairable system. The results indicate that the proposed hybrid data preparation model significantly improves the accurate prediction of failure counts.

2. LITERATURE SURVEY

2.1 Existing problem

Accurate prediction of possible failures will increase the reliability of aircraft components and systems. The scheduling of maintenance operations helps determine the overall maintenance and overhaul costs of aircraft components. Maintenance costs constitute a significant portion of the total operating expenditure of aircraft systems.

2.2 Reference

S. No	Title	Author	Year of Publication
1	Machine Learning-Based Predictive Analytics for Aircraft Engine Conceptual Design	Michael T. Tong	2020
2	Failure Prediction of Aircraft Equipment Using Machine Learning with a Hybrid Data Preparation Method	Kadir Celikmih, Onur Inan , and Harun Uguz	2020
3	Data-Driven Prediction of Unscheduled Maintenance Replacements in a Fleet of Commercial Aircrafts	Gianluca Nicchiott and Julien Rüegg	2018
4	A rare failure detection model for aircraft predictive maintenance using a deep hybrid learning approach	Maren David Dangut Ian K. Jennions Steve King Zakwan Skaf	2022
5	Predictive Maintenance for Aircraft Engine Using Machine Learning: Trends and Challenges	F A Adryan1, K W Sastra1	2021
6	Analysis of Aeronautical Engines Based on Machine Learning	Enrique López Droguett	2021
7	Machine-Learning-Based Condition Assessment of Gas Turbines	Martí de Castro-Cros *, Manel Velasco and Cecilio Angulo	2021

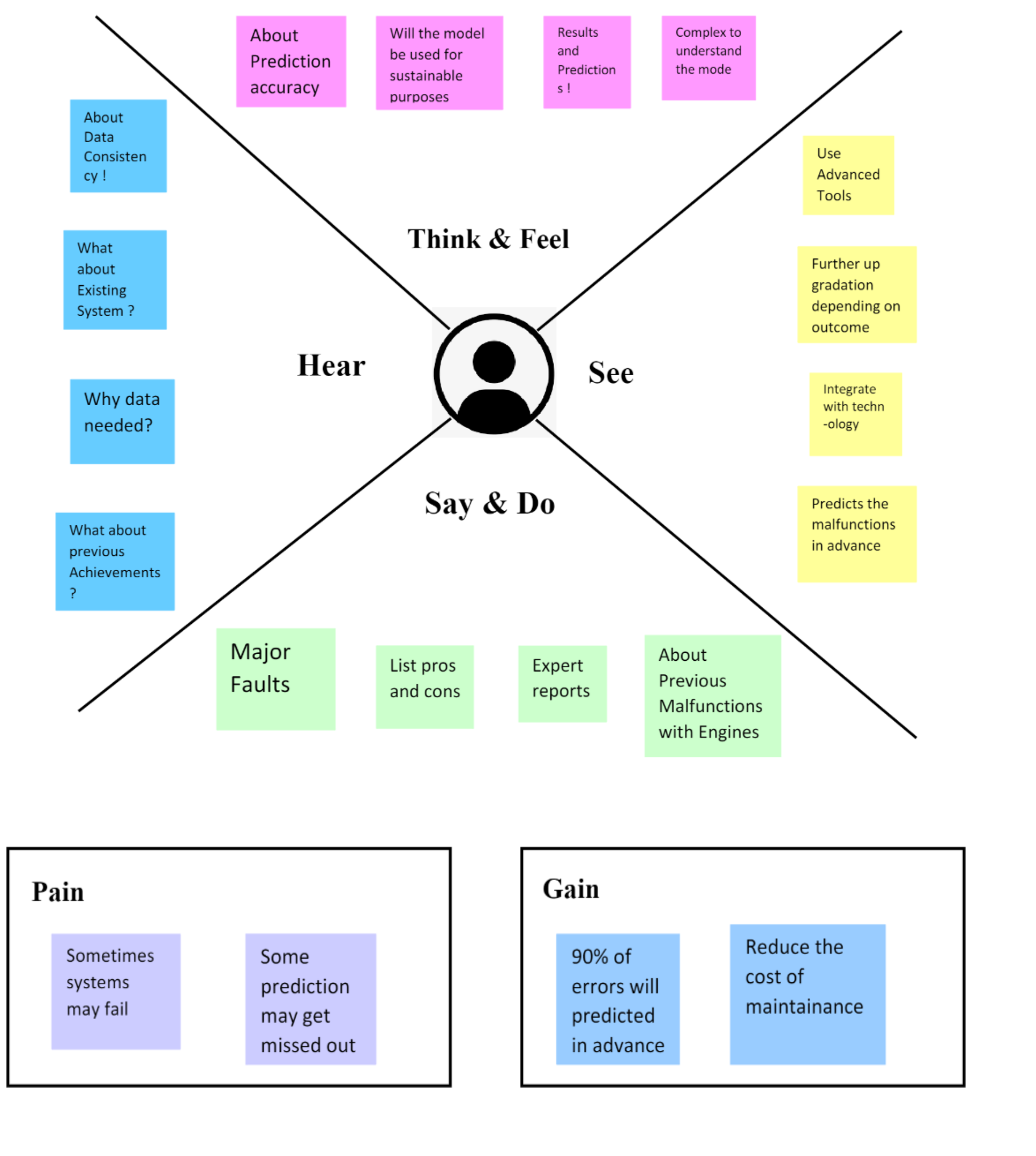
2.3 Problem Statement Definition

In aviation, the use of maintenance data is highly critical in the analysis of reliability and maintenance costs. This is because predictive maintenance scheduling can be planned in line with estimates. The main target of predictive maintenance is to predict equipment failures and planning strategies for spare parts of the system components to analyze the reliability and maintainability of a complex repairable system.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

The empathy map canvas for Machine Learning-Based Predictive Analytics for Aircraft Engine is shown:



The empathy map is plotted for knowing the accuracy for the predicted analysis it consists of four sections classified as think and feel, hear, see, say and do. Hear shows why data is needed previous achievements, What about existing system it, sees the advanced tools gradation depending on outcome integrate with technology predicts the malfunctions in advance it shows we should think about the prediction accuracy will the model used for sustainable purposes results and predictions the drawbacks are sometimes systems may fail some predictions may get missed out and the major positive is major errors will be predicted in advance and it reduces the cost of maintenance.

3.2 Ideation & Brainstorming

Machine Learning-Based Predictive Analytics for Aircraft Engine

Janani M (LEADER)

Hari Shankar S M

Surya M

Sujitha A J

Engine runtime Prediction

Now a day's industries are facing major problems with engines runtime prediction. so, we are going to make it easier with the help of machine learning techniques

Problem

How we are going to predict the working time of aircraft engine for a particular interval of time?

Brainstorm

Now we are going to share our ideas with a sticky note to address to have a solution on problem

Janani

Study Machine learning concepts	Getting solution about problem
Predict the engine condition using the ML methods	Test and Validate the model

Hari

Knowledge about fight data	Study about problem solution
Monitoring Weather Condition	Checking fight condition

Surya

check engine capacity	Study about the data
Gather the related models	Getting solution about problem

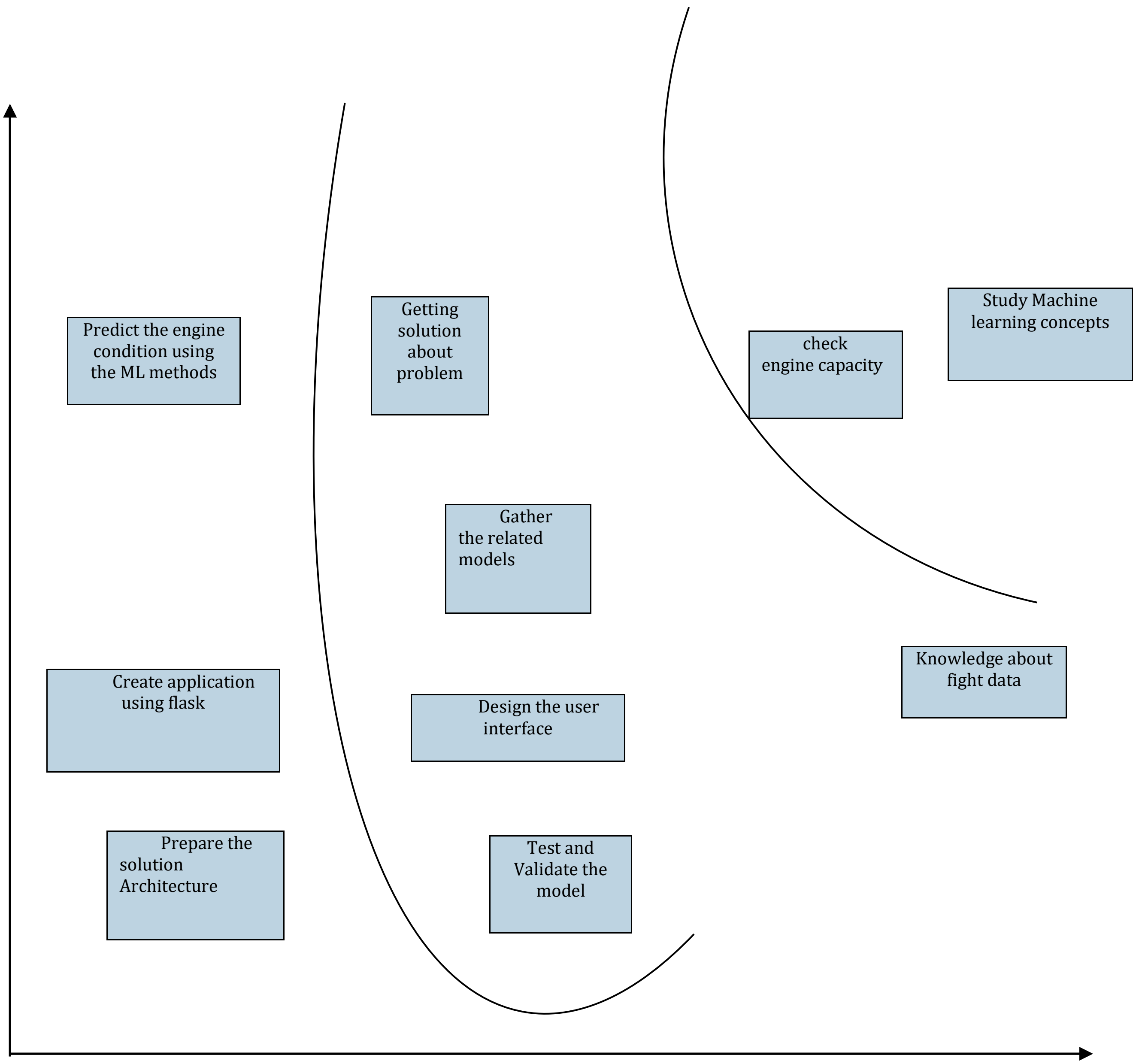
Sujitha

Prepare the solution Architecture	Learn ML learning methods
Design the user interface	Create application using flask

Flow Diagram

Designing of Prediction Model
Collecting data about engine
Learning ML Learning Techniques
Calculating the time
Prediction using ML techniques
Data classification techniques

Prioritization



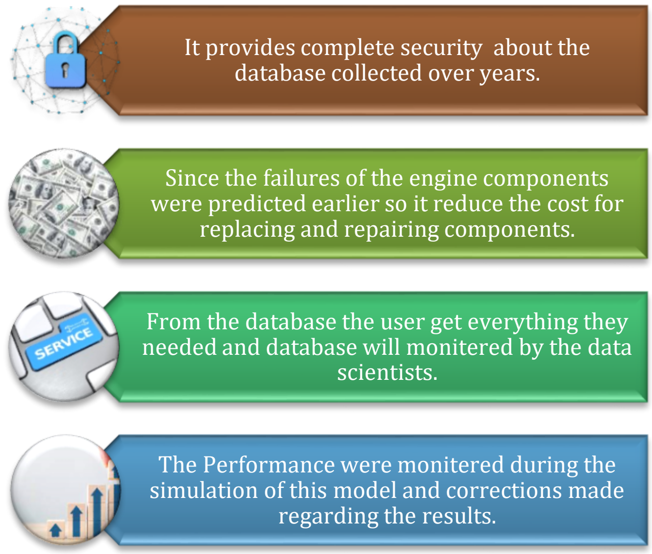
We are going to predict the working time of aircraft engine for a particular interval of time. Now a day's industries are facing major problems with engines runtime prediction. So, we are going to make it easier with the help of machine learning techniques. Some of the discussed points are Prepare the solution Architecture, Predict the engine condition using the ML methods, Monitoring Weather Condition, Study about problem solution, getting solution about problem, Study about problem solution, etc.…Then the data flow diagram is also shown here. Then the prioritization graph is plotted. Thus, the working time of the aircraft engine is studied.

3.3 Proposed Solution

S .No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• The unexpected failure of the aircraft engine components leads to increase the overall cost. • The limitless maintenance activities such as scheduled maintenance (Corrective maintenance, Preventive maintenance, Predictive maintenance) and unscheduled maintenance.
2.	Idea / Solution description	• To minimize the risk factors and improvement of aircraft engine, Engine To introduce machine learning models based on feature selection and data elimination to predict failure of the aircraft system. • To companies have generated and collected large amount of data over the years from various sources such as the database of currently development projects, previously completed development projects, and the designs that were not manufactured, are valuable for intelligence that can support new engine development. • To anticipating rare failure within a predetermined meaningful time frame.
3.	Novelty / Uniqueness	• Supervised machine-learning analytics for Aircraft engine were employed to find patterns in the database of 183 manufactured engines and engines that were studied previously in various NASA aeronautics projects. • It minimizes risk and improve in the technological field. • Based on the analytics airlines can know exactly what is happening, why it is happening, and what possible impact any event.
4.	Social Impact / Customer Satisfactio	• The database will help the aviation industry and it will need to update for every particular period of time interval (like 10 years…). • It will make the flight journey even safer.

<table>
<tr>
<td></td>
<td>n</td>
<td>

It reduces the manual checking of engine components.
Reduces the cost of repairing.

</td>
</tr>
<tr>
<td>5.</td>
<td>Business Model (Revenue Model)</td>
<td>


</td>
</tr>
<tr>
<td>6.</td>
<td>Scalability of the Solution</td>
<td>

The development used the database of 183 manufactured engines and engines that were studied previously in NASA aeronautics projects. The TSFC predictive analytics has an average accuracy of 98.3 percent, with 3.5 percent uncertainty. The engine core-size predictive analytics has an overall accuracy of 100 percent, with 4.3 percent uncertainty. Overall, both predictive analytics show remarkable prediction accuracy.
It would help to identify the best engine design expeditiously amongst several candidates.

<table>
<tr><th>Algorithms</th><th>Accuracy</th><th>Model Complexity</th><th>Treat with Uncertainties</th><th>Computational Time</th><th>Robustness</th><th>Observations</th></tr>
<tr><td>ANN</td><td>Reasonably high</td><td>Reasonable</td><td>Low</td><td>Reasonably low</td><td>Reasonably low</td><td>The performance can be improved with a large amount of accurate data</td></tr>
<tr><td>AE</td><td>High</td><td>Reasonably high</td><td>Reasonable</td><td>Reasonably low</td><td>Reasonable</td><td>Good generalisation properties</td></tr>
<tr><td>CNN</td><td>High</td><td>High</td><td>Reasonable</td><td>Reasonably high</td><td>Reasonable</td><td>-</td></tr>
<tr><td>ELM</td><td>Reasonably high</td><td>Reasonably low</td><td>Low</td><td>Low</td><td>Low</td><td>-</td></tr>
<tr><td>GAN</td><td>Low</td><td>High</td><td>Reasonably low</td><td>Reasonable</td><td>Reasonable</td><td>-</td></tr>
<tr><td>RNN</td><td>High</td><td>High</td><td>Reasonably low</td><td>Reasonable</td><td>Reasonably high</td><td>Large amount of and sequential data</td></tr>
<tr><td>Bayesian Models</td><td>High</td><td>Reasonable</td><td>Reasonably high</td><td>Reasonable</td><td>Reasonably high</td><td>-</td></tr>
<tr><td>Clustering</td><td>Reasonably high</td><td>Reasonable</td><td>Reasonable</td><td>Reasonably high</td><td>Reasonably high</td><td>The performance can be improved by having an accurate dataset</td></tr>
<tr><td>Decision Tree</td><td>High</td><td>Reasonable</td><td>Reasonable</td><td>Reasonably low</td><td>Reasonably high</td><td>-</td></tr>
<tr><td>Fuzzy Logic</td><td>High</td><td>Reasonably high</td><td>Reasonably high</td><td>Reasonably low</td><td>High</td><td>Generalisation properties with a reasonable amount of data; expertise in the field is needed</td></tr>
<tr><td>Genetic Programming</td><td>High</td><td>High</td><td>Reasonably low</td><td>High</td><td>High</td><td>-</td></tr>
<tr><td>L and NL Regression</td><td>Reasonable</td><td>Low</td><td>Low</td><td>Low</td><td>Reasonably low</td><td>Regulation can help to generalise the solution</td></tr>
<tr><td>PCA</td><td>Low</td><td>Low</td><td>Reasonably low</td><td>Low</td><td>Low</td><td>It works well only with linear data. Good performance in pre-processing data</td></tr>
<tr><td>SVM</td><td>Reasonably high</td><td>Reasonably low</td><td>Reasonable</td><td>Low</td><td>Reasonably low</td><td>The performance can be improved by using more sophisticated kernels</td></tr>
</table>
Table: Comparative study of the presented machine-learning-based models.
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td>

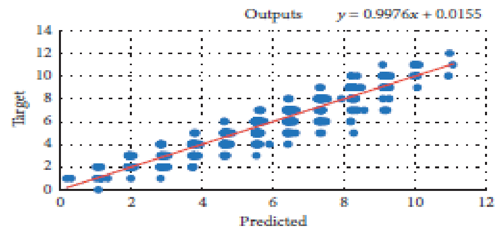


FIGURE 4: Correlation between predicted and target values of the dataset for LR.

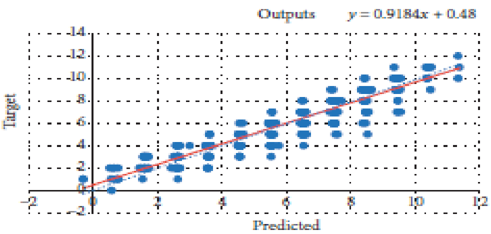


FIGURE 5: Correlation between predicted and target values of the dataset for SVR.

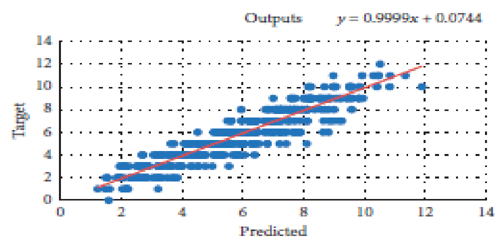


FIGURE 6: Correlation between predicted and target values of the dataset for MLP.

The results indicate that the proposed hybrid data preparation model significantly improves the accurate prediction of failure counts.
Comparing AE–CNN–BGRU with other similar deep learning methods, the proposed approach shows superior performance with 18% better precision, 5% in a recall, and 10% in g-mean. The results also indicate the model effectiveness in predicting component failure within a defined useful period that aids in minimising operational disruption.

</td>
</tr>
</table>

In this we are going to see about the Problem Statement, Idea / Solution description, Novelty / Uniqueness, Social Impact / Customer Satisfaction, Business Model, Scalability of the Solution. The unexpected failure of the aircraft engine components leads to increase the overall cost. To companies have generated and collected large amount of data over the years from various sources such as the database of currently development projects, previously completed development projects, and the designs that were not manufactured, are valuable for intelligence that can support new engine development. It minimizes risk and improve in the technological field. The database will help the aviation industry and it will need to update for every particular period of time interval. It will make the flight journey even safer. It provides complete security about the database collected over years. It would help to identify the best engine design expeditiously amongst several candidates

3.4 Problem Solution fit:

Project Title: Machine Learning-Based Predictive Analytics for Aircraft Engine

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID52013

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

We can get the customers by providing our model implementation that can be done to get a desired result, we can also use various methods by the data exploration and with continuous improvement.

Customers are businessmen, student, tourist, traveler and all the people traveling in flight.

6. CUSTOMER CONSTRAINTS CC

Customers require accurate and early predictions of the flight engine failure. And they also look for an alternate solution.

Constraints here could include physical movements, time, flight operations, militry operations, easing the noise, weather, reduced flows, length, size of aircraft and so on. There are also environmental requirements to consider.

5. AVAILABLE SOLUTIONS AS

Maintaining the structural failures where a broken connecting rod, crank valve, or camshaft is present account for 17% of engine failures occurs. Preventing the fuel problems.

The reliability analysis of aircraft engines is essential for ensuring the smooth functioning of each component of an aircraft engine

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

The engine failure occur due to the fuel problems like the contamination and exhaustion. We can prevent it by proper maintainence.

Engine failure occurs when a turbine engine unexpectedly stops producing power due to malfunction. This led to a lot of customer dissatisfaction.

9. PROBLEM ROOT CAUSE RC

The root cause of the problem is unforeseen and unpredictable engine failure that has increased the hazards of air travels.

Another root cause of the problem is unforeseen & unpredictable engine failure that cause cancellations and arrival, departure delays.

7. BEHAVIOUR BE

The purpose of this research is to develop methods that can be used to generate reliable and timely alerts.

We can encourage the customers to give the feedback and we should focus on the service quality and should increase the safety and security.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM	3. TRIGGERS TR To accurately predict the failure of an engine and track the flight. Mechanical failure by undertorqueing cylinder, Structural failures due to pilots ignorance and the fuel problems such as exhaustion and mismanagement. 4. EMOTIONS: BEFORE / AFTER EM The aircraft engine failure occurs; passengers often get annoyed and frustrated. They also might lose to reach on time to some important occasions. This happens when the customers are not satisfied with the services. At their dissatisfaction where they lose all their hopes in our services and start approaching others for a better solution.	10. YOUR SOLUTION SL By identifying the needs you can provide faster and effective support. We should improve our product and services and satisfy the customer needs.	8. CHANNELS of BEHAVIOUR CH ONLINE We can suggest the positive employee attitudes, behaviors, and prompt services recovery actions that generates more positive emotions. 8.2 OFFLINE We can make a effort to convert dissatisfied customers into loyal ones and we must make the customer feel goog about the experience they faced.	Identify strong TR & EM

The aircraft engine failure occurs and the passengers often get annoyed and frustrated. They also might lose to reach on time to some important occasions. This happens when the customers are not satisfied with the services. At their dissatisfaction where they lose all their hopes in our services and start approaching others for a better solution. So the solution for this can be achieved by improving our products and design of the project can be made with good value. To accurately predict the failure of an engine and track the flight Mechanical failure by under torqueing cylinder, Structural failures due to pilot’s ignorance and the fuel problems such as exhaustion and mismanagement. By identifying the needs you can provide faster and effective support. We should improve our product and services and satisfy the customer needs. We can suggest the positive employee attitudes, behaviours, and prompt services recovery actions that generates more positive emotions. We can make an effort to convert dissatisfied customers into loyal ones and we must make the customer feel good about the experience they faced.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

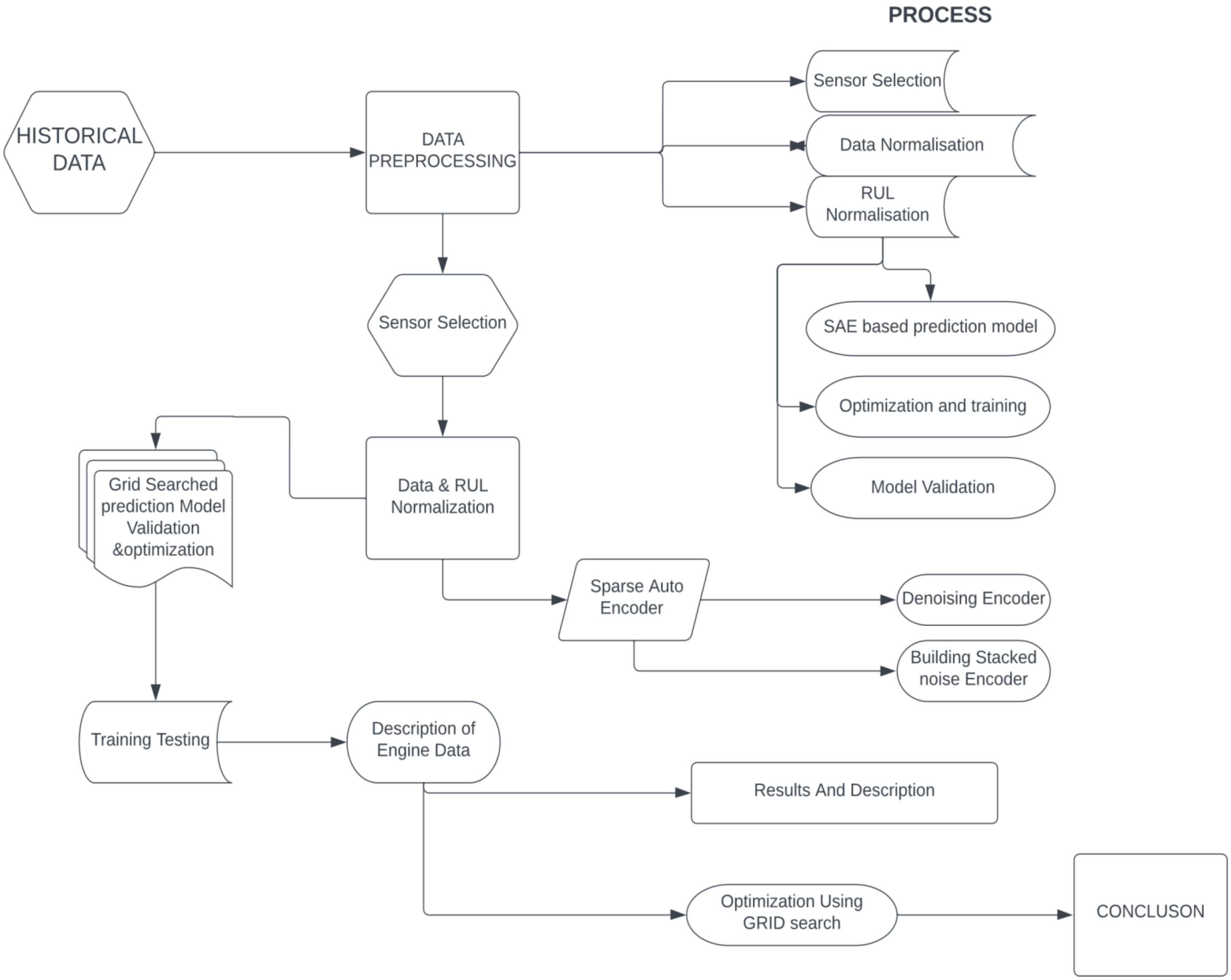
The input data from jet engine consists of three feature columns named Nozzle Area, Air fuel ration and Throttle position and one label column named Fault to train the model and to predict fault for streaming features. Trained model can predict whether the fault is detected or not based on the Input and time series values of features (Nozzle Area, Air fuel ration, and Throttle position).

4.2 Non - Functional requirement

Predicated throttle fault. The recurrent network consists of 22 hidden neurons and four-time delays for each of the inputs. The network is NARX (Nonlinear Autoregressive with External Inputs). The training algorithm utilized is Levenberg-Marquardt is minimizing the MSE. The NN false positives mostly happen at large throttle command changes, however, the overall prediction accuracy achieved is 96.4% over multiple simulation runs. The network is trained to identify the fault in damping of the throttle actuator. After the network is trained, the output is processed with a saturation and relay to achieve discrete outputs such that a logical indication is obtained indicating a fault.

5. PROJECT DESIGN

5.1 Data Flow Diagram

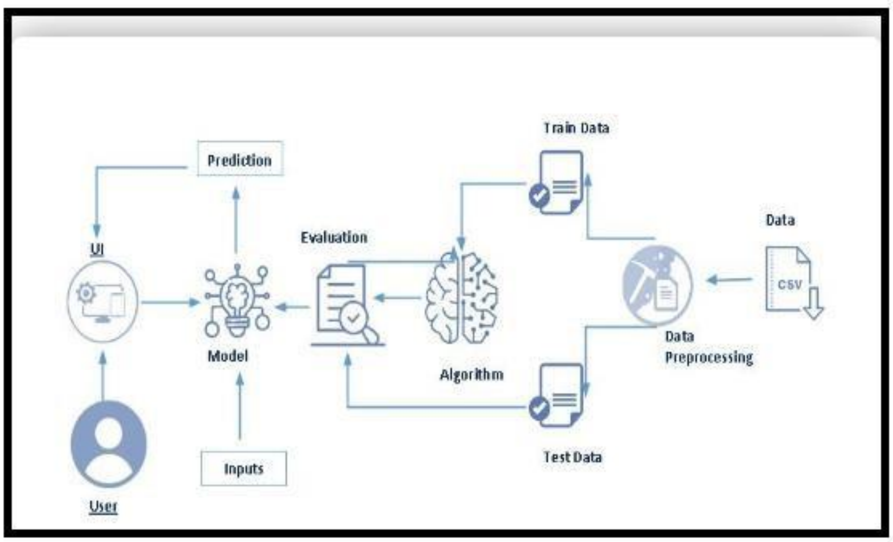


5.2 Solution & Technical Architecture

- The dataset containing historical sensor data related to an aircraft engine is loaded from a CSV file into Python using anyone of the machine learning libraries available (sklearn).
- This loaded data is then pre-processed and split into training and testing data in a specific proportion.
- A number of machine learning models are created using different algorithms to identify the best performing one among them.

- Since this is a classification problem, classification models like Logistic Regression, J48, Naïve Bayes, Random Forest, and Decision tree classifiers are used.
- The training data is then passed to all of the chosen models to train the model based on the historical sensor data to predict aircraft engine failures.
- All of the models are evaluated based on accuracy and a report is drawn.
- The model with the highest accuracy is chosen and implemented in the final product.
- A web application is created using Flask, a light-weight python web framework, and is then hosted on an IBM Cloud server.
- This server is made to host and run the trained machine learning model as well.
- Sensor data entered by users in the Flask web application are then sent to the trained machine learning model to predict the chances of engine failure.
- The classification model then classifies the engine as safe (1) or unsafe (0) based on the parameters the user enters.
- This solution can help reduce the risk of catastrophic mid-air engine failures, potential subsequent fatalities, and help improve the public's confidence in air travel.

Given below is the proposed solution's architecture diagram



6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Literature Survey &Information Gathering	Literature survey on theselected project & gathering information byreferring the, technical papers, research publications etc.	3 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the userPains & Gains, Prepare listof problem statements	23 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session andprioritize the top 3 ideas based on the feasibility & importance	23 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, businessmodel, social impact, scalability of solution, etc.	24 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solutionfit document.	29 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	19 SEPTEMBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions &experienceswith the application (entry to exit).	1 OCTOBER 2022
Functional Requirement	Prepare the functionalrequirement document.	2 OCTOBER 2022

6.2 Sprint Delivery Schedule

The main event during agile methodology is the sprint, the stage where ideas turn into innovation and valuable products come to life.

On one hand, agile sprints can be highly effective and collaborative. At the same time, they can be chaotic and inefficient if they lack proper planning and guidance. And for this reason, making a sprint schedule is one of the most important things you can do to ensure that your efforts are successful.

If you’re looking to schedule your next sprint, you’ve come to the right place. Keep reading to learn everything you need to know about sprint scheduling, including some tips to drive the best results.

3.Reports from JIRA

One part of ensuring the success and smooth operations of your projects in JIRA is reporting. It involves gaining the knowledge about the health, progress and overall status of your JIRA projects through Gadgets, report pages or even third-party applications. The goal of this guide is to provide an overview of the tools available to JIRA users today and how they can be used to fulfill the different types of reporting needs that users face today.

7. CODING & SOLUTIONING

```
import pandas as pd import NumPy as np

from sklearn.preprocessing import MinMaxScaler from sklearn.metrics import
confusion_matrix, accuracy_score

import matplotlib.pyplot as plt
plt.style.use('ggplot') %matplotlib inline

dataset_train=pd.read_csv("PM_train.txt",sep=' ',
header=None).drop([26,27],axis=1) col_names=['id', 'cycle',

'setting1','setting2','setting3','s1','s2','s3','s4','s5','s6','s7','s
8','s9','s10','s11','s12','s13','s14','s15','s16','s17','s18','s19','s

20','s21']

dataset_train.columns=col_names

print ('Shape of Train dataset: ',dataset_train.shape) dataset_train.head()

Shape of Train dataset:  (20631, 26)
```

```
   id  cycle  setting1  setting2  setting3     s1     s2      s3   s4  \
0   1      1   -0.0007   -0.0004     100.0  518.67  641.82  1589.70
 1400.60
1   1      2    0.0019   -0.0003     100.0  518.67  642.15  1591.82
 1403.14
2   1      3   -0.0043    0.0003     100.0  518.67  642.35  1587.99
 1404.20
3   1      4    0.0007    0.0000     100.0  518.67  642.35  1582.79
 1401.87
4   1      5   -0.0019   -0.0002     100.0  518.67  642.37  1582.85
 1406.22

     s5  ...     s12     s13      s14     s15   s16  s17   s18    s19  \
0  14.62  ...  521.66  2388.02  8138.62  8.4195  0.03  392  2388
 100.0
1  14.62  ...  522.28  2388.07  8131.49  8.4318  0.03  392  2388
 100.0
2  14.62  ...  522.42  2388.03  8133.23  8.4178  0.03  390  2388
 100.0
3  14.62  ...  522.86  2388.08  8133.83  8.3682  0.03  392  2388
 100.0
4  14.62  ...  522.19  2388.04  8133.80  8.4294  0.03  393  2388
 100.0

     s20     s21  0  39.06
23.4190
```

```
1   39.00  23.4236
2   38.95  23.3442
3   38.88  23.3739
4   38.90  23.4044

[5 rows x 26 columns]
```

```
dataset_test=pd.read_csv('PM_test.txt',sep='
',header=None).drop([26,27],axis=1)
dataset_test.columns=col_names # dataset test.head()
```

```
print('Shape of Test dataset:',dataset_train.shape) dataset_train.head()
```

```
Shape of Test dataset: (20631, 26)

   id  cycle  setting1  setting2  setting3    s1     s2      s3   s4  \
0   1      1   -0.0007   -0.0004     100.0  518.67  641.82  1589.70
1400.60
1   1      2    0.0019   -0.0003     100.0  518.67  642.15  1591.82
1403.14
2   1      3   -0.0043    0.0003     100.0  518.67  642.35  1587.99
1404.20
3   1      4    0.0007    0.0000     100.0  518.67  642.35  1582.79
1401.87
4   1      5   -0.0019   -0.0002     100.0  518.67  642.37  1582.85
1406.22

      s5  ...     s12      s13      s14     s15   s16  s17   s18    s19  \
0  14.62  ...  521.66  2388.02  8138.62  8.4195  0.03  392  2388
100.0
1  14.62  ...  522.28  2388.07  8131.49  8.4318  0.03  392  2388
```

100.0

2 14.62 ... 522.42 2388.03 8133.23 8.4178 0.03 390 2388
100.0

3 14.62 ... 522.86 2388.08 8133.83 8.3682 0.03 392 2388
100.0

4 14.62 ... 522.19 2388.04 8133.80 8.4294 0.03 393 2388
100.0

s20 s21 0 39.06

23.4190

1 39.00 23.4236
2 38.95 23.3442
3 38.88 23.3739
4 38.90 23.4044

[5 rows x 26 columns]

pm_truth=pd.read_csv('PM_truth.txt',sep='

',header=None).drop([1],axis=1)

pm_truth.columns=['more'] pm_truth['id']=pm_truth.index+1

pm_truth.head()

more id 0

112 1

1 98 2
2 69 3
3 82 4

4 91 5

rul=pd.DataFrame (dataset_test.groupby ('id') ['cycle'].max()).reset_index()

rul.columns=['id','max'] rul. head()

id max 0

1 31

1 2 49
2 3 126
3 4 106 4 5 98

pm_truth['rtf']=pm_truth['more']+rul['max'] pm_truth.head()

more id rtf 0 112

1 143

1 98 2 147
2 69 3 195
3 82 4 188 4 91 5 189

pm_truth.drop('more', axis=1, inplace=True)

dataset_test=dataset_test.merge(pm_truth,on=['id'],how='left')

dataset_test['ttf']=dataset_test['rtf'] - dataset_test['cycle'] dataset_test.drop('rtf', axis=1,

inplace=True) dataset_test.head()

id cycle setting1 setting2 setting3 s1 s2 s3 s4 \

0 1 1 0.0023 0.0003 100.0 518.67 643.02 1585.29
1398.21

1 1 2 -0.0027 -0.0003 100.0 518.67 641.71 1588.45
1395.42

2 1 3 0.0003 0.0001 100.0 518.67 642.46 1586.94
1401.34

3 1 4 0.0042 0.0000 100.0 518.67 642.44 1584.12

1406.42

4 1 5 0.0014 0.0000 100.0 518.67 642.51 1587.19 1401.92
s5 ... s13 s14 s15 s16 s17 s18 s19 s20

\

0 14.62 ... 2388.03 8125.55 8.4052 0.03 392 2388 100.0 38.86

1 14.62 ... 2388.06 8139.62 8.3803 0.03 393 2388 100.0 39.02

2 14.62 ... 2388.03 8130.10 8.4441 0.03 393 2388 100.0 39.08

3 14.62 ... 2388.05 8132.90 8.3917 0.03 391 2388 100.0 39.00 4 14.62 ... 2388.03 8129.54

8.4031 0.03 390 2388 100.0 38.99

s21 ttf 0 23.3735

142

1 23.3916 141
2 23.4166 140
3 23.3737 139
4 23.4130 138

[5 rows x 27 columns]

dataset_train['ttf']=dataset_train.groupby (['id']) ['cycle'].transform(max)-

dataset_train['cycle'] dataset_train.head()

id cycle setting1 setting2 setting3 s1 s2 s3 s4 \

0 1 1 -0.0007 -0.0004 100.0 518.67 641.82 1589.70
1400.60

1 1 2 0.0019 -0.0003 100.0 518.67 642.15 1591.82
1403.14

2 1 3 -0.0043 0.0003 100.0 518.67 642.35 1587.99

1404.20

3 1 4 0.0007 0.0000 100.0 518.67 642.35 1582.79
1401.87

4 1 5 -0.0019 -0.0002 100.0 518.67 642.37 1582.85
1406.22

s5 ... s13 s14 s15 s16 s17 s18 s19 s20

\

0 14.62 ... 2388.02 8138.62 8.4195 0.03 392 2388 100.0 39.06

1 14.62 ... 2388.07 8131.49 8.4318 0.03 392 2388 100.0 39.00

2 14.62 ... 2388.03 8133.23 8.4178 0.03 390 2388 100.0 38.95

3 14.62 ... 2388.08 8133.83 8.3682 0.03 392 2388 100.0 38.88 4 14.62 ... 2388.04 8133.80
8.4294 0.03 393 2388 100.0 38.90

s21 ttf 0 23.4190

191

1 23.4236 190
2 23.3442 189
3 23.3739 188
4 23.4044 187

[5 rows x 27 columns]

```
df_train=dataset_train.copy()

df_test=dataset_test.copy () period=30

df_train['label_bc']=df_train['ttf'].apply(lambda x: 1 if x <= period else 0)

df_test['label_bc']= df_test['ttf'].apply(lambda x: 1 if x <= period else 0) df_train.head()
```

```
   id  cycle  setting1  setting2  setting3     s1     s2      s3   s4  \
0   1      1   -0.0007   -0.0004     100.0  518.67  641.82  1589.70
1400.60
1   1      2    0.0019   -0.0003     100.0  518.67  642.15  1591.82
1403.14
2   1      3   -0.0043    0.0003     100.0  518.67  642.35  1587.99
1404.20
3   1      4    0.0007    0.0000     100.0  518.67  642.35  1582.79
1401.87
4   1      5   -0.0019   -0.0002     100.0  518.67  642.37  1582.85
1406.22

      s5  ...      s14     s15   s16  s17   s18    s19    s20      s21 ttf  \
0  14.62  ...  8138.62  8.4195  0.03  392  2388  100.0  39.06  23.4190
191
1  14.62  ...  8131.49  8.4318  0.03  392  2388  100.0  39.00  23.4236
190
2  14.62  ...  8133.23  8.4178  0.03  390  2388  100.0  38.95  23.3442
189
3  14.62  ...  8133.83  8.3682  0.03  392  2388  100.0  38.88  23.3739
188
4  14.62  ...  8133.80  8.4294  0.03  393  2388  100.0  38.90  23.4044
187

   label_bc  0
0
```

```
1                  0
2                  0
3                  0
4                  0
```

[5 rows x 28 columns]

x_train=df_train.iloc[:,:-1].values y_train=df_train.iloc[:,-1:].values

from sklearn.linear_model import LogisticRegression model =

LogisticRegression() model.fit(x_train,y_train)

C:\Users\Dell\anaconda3\lib\site-packages\sklearn\utils\ validation.py:993:

DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to

(n_samples,), for example using ravel().   y =

column_or_1d(y, warn=True)

C:\Users\Dell\anaconda3\lib\site-packages\sklearn\linear_model\ _logistic.py:814:

ConvergenceWarning: lbfgs failed to converge

(status=1):

STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in: https://scikit-learn.org/stable/modules/preprocessing.html Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logisticregression

n_iter_i = _check_optimize_result(LogisticRegression()

```
from sklearn.metrics import accuracy_score y_predlog=model.predict(x_train)
accuracy_score(y_predlog,y_train)

0.9999515292520964

from sklearn.metrics import confusion_matrix cm1 =
confusion_matrix(y_train,y_predlog) cm1

array([[17530,     1],
       [    0,  3100]], dtype=int64)

import joblib

joblib.dump(model,"engine_model.sav")

['engine_model.sav']

import pickle
file_name='engine_model.pkl' f =
open(file_name,'wb') pickle.
dump(model,f) f. close()
```

8. TESTING

8.1. User Acceptance Testing

The project has been tested extensively with a number of users. The users found the interface very easy to use. The Web pages were colourful and attractive. There were no unnecessary details in the web page. It was clean and simple that any new user could master it. The data input format was also simple. The user need not enter any unit. They could simply enter the value. The prediction time is fairly low at an average time of 1.5 seconds. This delay depends on the internet connectivity. The model has been hosted in IBM cloud.

Thus, with the API available, the model can be accessed remotely from any system provided IBM access key is given. The model predicts if the engine under consideration will fail in the immediate future or not. Although the prediction is not very accurate. Various inputs have been given by the users to test the consistency of the model. The model proved itself and all the users accepted the model as reliable and convenient.

9. RESULTS

9.1. Performance Metrics

The Support Vector Machine ML model that we have used here has better performance in speed and accuracy compared to other models. We have compared the performance metrics of 4 models and selected this as the best for the application. The model performed well for all the test cases. The API developed also performed good with no glitches or lag found during the testing phase.

10. ADVANTAGES & DISADVANTAGES

10.1. Advantages This system will definitely help in saving lives and preventing mid-air engine failures, by predicting them in advance. It will also result in an increased confidence in air travel, which is beneficial to both the air carriers as well as passengers. 10.2. Disadvantages The model may not always be accurate, and hence, could result in some false alarms. This may in turn lead to some unnecessary spending for service of an engine, but one can never put a price on human life. So, it is better to be more careful.

11. CONCLUSION

The Support Vector Machine ML model that has been used above performs well for our dataset. The model is fast and consumes less features. The API developed is also simple and userfriendly. By using this model, we could predict the Engine failure of the Aircraft provided the required input parameter. The model is not 100% accurate but it performs sufficiently. It can be concluded as the output cannot be predicted very accurately as there are several parameters that could affect the output and all those outputs cannot be taken in for training as it can result in a very complex and overtrained model. The features that have high weightage are considered in this model.

12. FUTURE SCOPE

The further works that can be done in this project is to include more features in model training to study the effect on the output. A long history of data (dataset of more than 3 years) can be used for training for increased accuracy. The application can be upgraded such that the input values from the sensors are directly fed to the model without the user entering it manually

14. GitHub & Project Demo Link:

GitHub: https://github.com/IBM-EPBL/IBM-Project-43871-1660720191

Project Demo Link: https://drive.google.com/file/d/1nV8O_lv4oq-N2wFa3iuC02oNm75omHxu/view?usp=sharing