Machine Learning Based Predictive Analytics for Aircraft Engine

CONTENTS

Machine Learning-Based Predictive Analytics for Aircraft Engine

Team ID	**PNT2022TMID00757**
Project Name	Machine Learning-Based Predictive Analytics for Aircraft Engine
Team Members	**Vajram suman (Leader)** **Rakesh.R** **Rahul.C** **Yogesh.K**

1. INTRODUCTION

1.1 Project overview

Machine learning is a branch of artificial intelligence that uses statistical technique and mathematical algorithms to enable a machine to learn from data, to analyze data patterns, and to make decisions with minimal human intervention. Data is now the most valuable asset for enterprises in every industry. Companies are using data-driven insights for competitive advantage. With that, the adoption of machine learning-based data analytics is rapidly taking hold across various industries, producing autonomous systems that support human decision-making. This work explored the application of machine learning to aircraft engine performance prediction Supervised machine-learnin algorithms for regression and classification were employed to study patterns in an existing, open-source database of production and research turbofan engines, and resulting in predictive analytics for use in predicting performance of new turbofan designs.

1.2 Purpose

Predictive analytics help us to understand possible future occurrences by analyzing the past. Predictive modeling solutions are a form of data-mining technology that works by analyzing historical and current data and generating a model to help predict future outcomes. Machine learning, on the other hand, is a subfield of computer science that, as per Arthur Samuel's definition from 1959. gives 'computers the ability to learn without being explicitly programmed'.

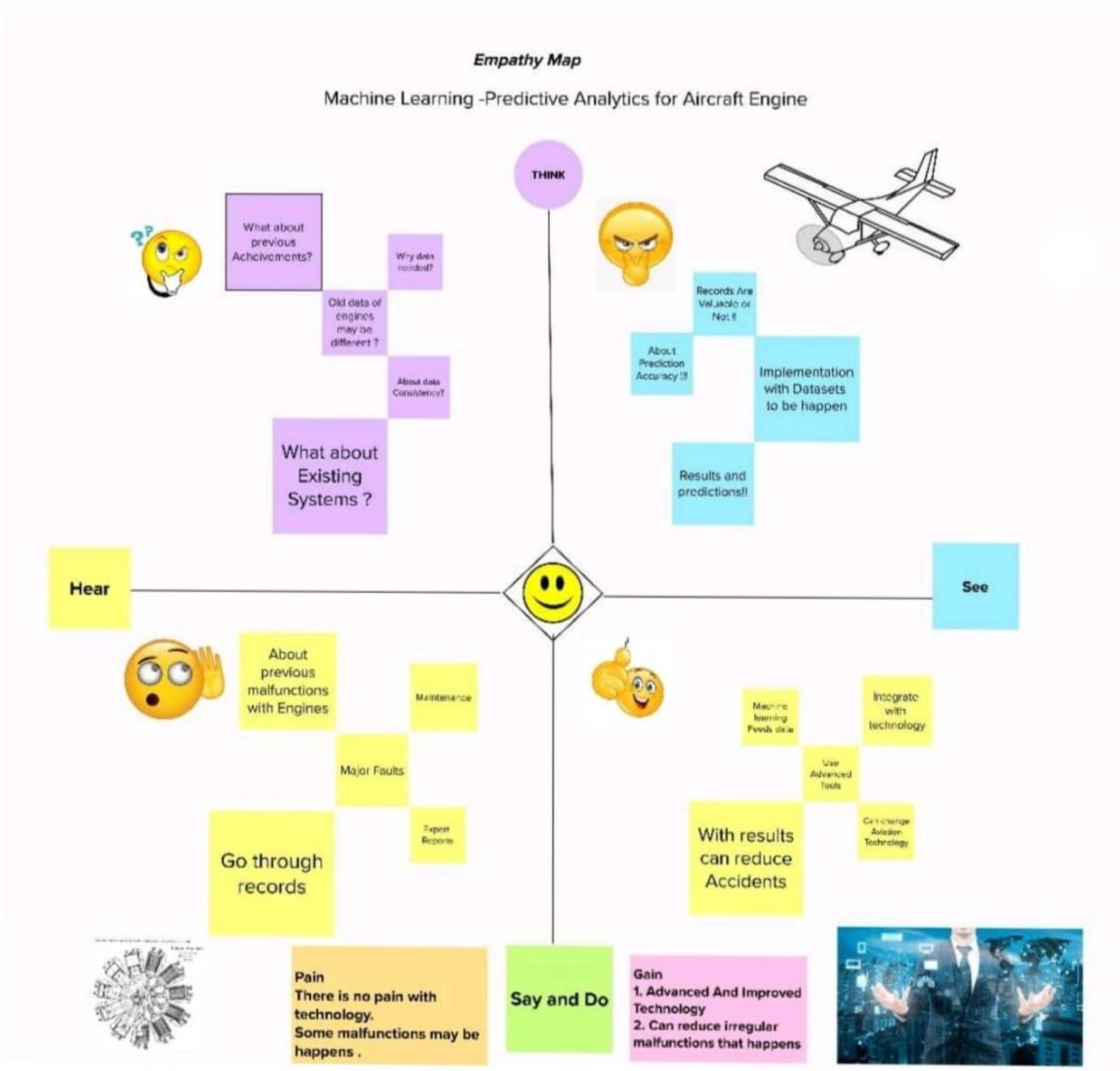
2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

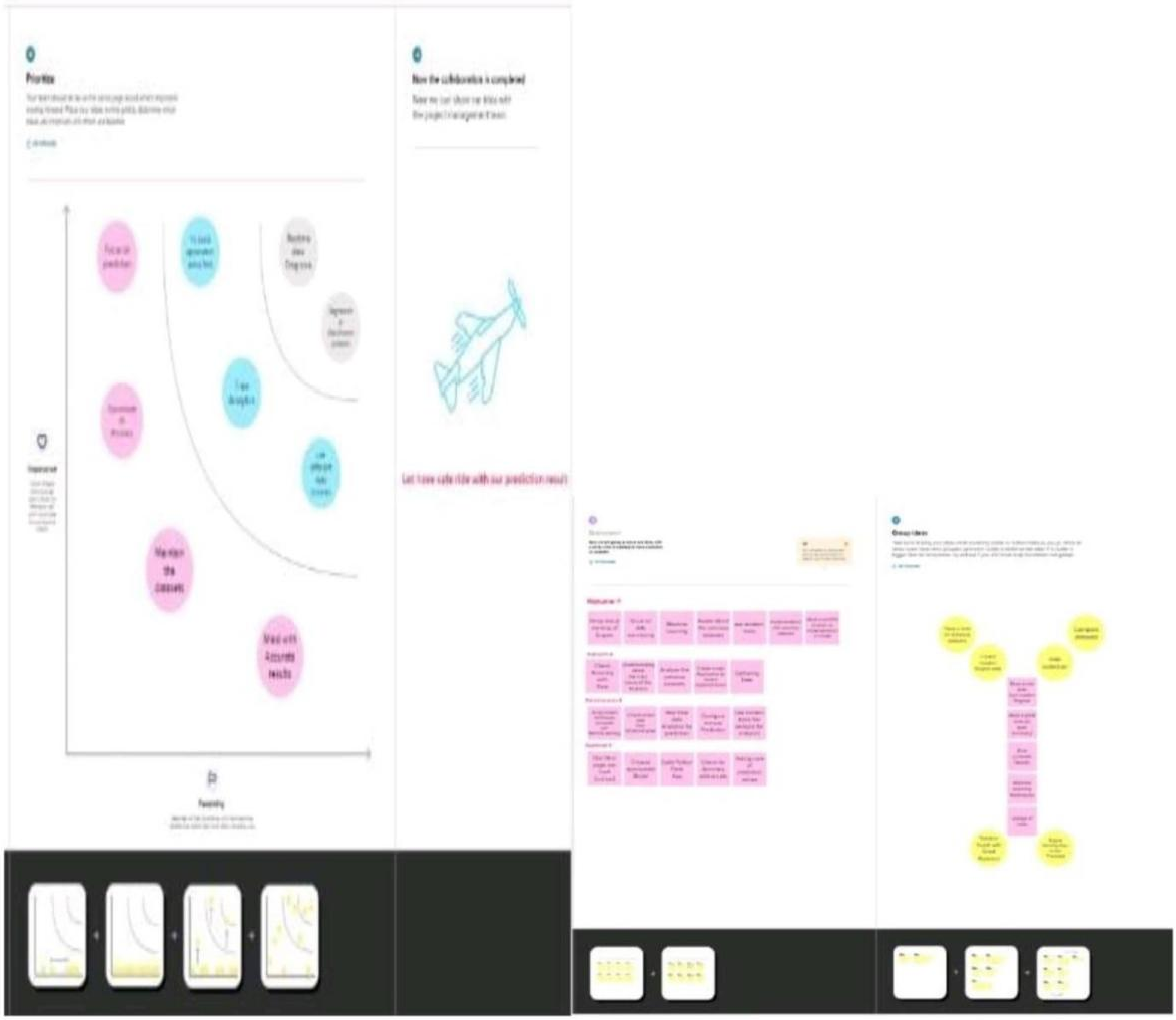
The majority of the returns we receive from the field are found not to be issues with the turbocharger itself, but in most cases they are problems with the system's installation, inadequate pre-lubrication, or other operational issues.Typically a mechanic must inspect and diagnose operational issues that may include an inability for the aircraft to reach altitude; pressurization issues; the system's inability to reach the maximum-rated manifold pressure; a surging or dropping off of manifold pressure when climbing or descending; and/or oil leaks from the compressor or turbine side of the turbocharger.

https://www.geaviation.com/commercial

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING



Prioritize
Now the collaboration is completed
Maintain the datasets
Meet with Accurate results
Feasibility
Let have safe ride with our prediction result
Group ideas

3.3 Proposed Solution

To predict the failure of an engine by using Machine Learning to save loss of time & money thus improving productivity. Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. The failure can be predicted by installing the sensors and keeping a track of the values.

3.4 Proposed Solution Fit

To predict the failure of an engine by using Machine Learning to save loss of time & money thus improving productivity.

- Novelty / Uniqueness : Gas-turbine engines are critical to the operation of most industrial plants, aircraft, and heavy vehicles such as military armor and transport ships, and their associated maintenance costs can be high.
- Social Impact / Customer : Satisfaction U nhappy or disengaged customers naturally mean fewer passengers and less revenue. It's important that customers have an excellent experience every time they travel.
- Business Model (Revenue Model) : While safety and performance are the primary goals of aircr aft maintenance.
- Scalability of the Solution : The Scalability calculated by machine learning methods.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

- FR-1 User Registration ,Registration through Form Registration through Gmail Registration through LinkedIN
- FR-2 User Confirmation ,via Email Confirmation via OTP
- FR-3 Tracking Expense Helpful insights about money management
- FR-4 Alert Message Give alert mail if the amount exceeds the budget limit ● FR-5 Category This application shall allow users to add categories of their expenses

4.2 Non Functional requirement

Following are the non-functional requirements of the proposed solution.

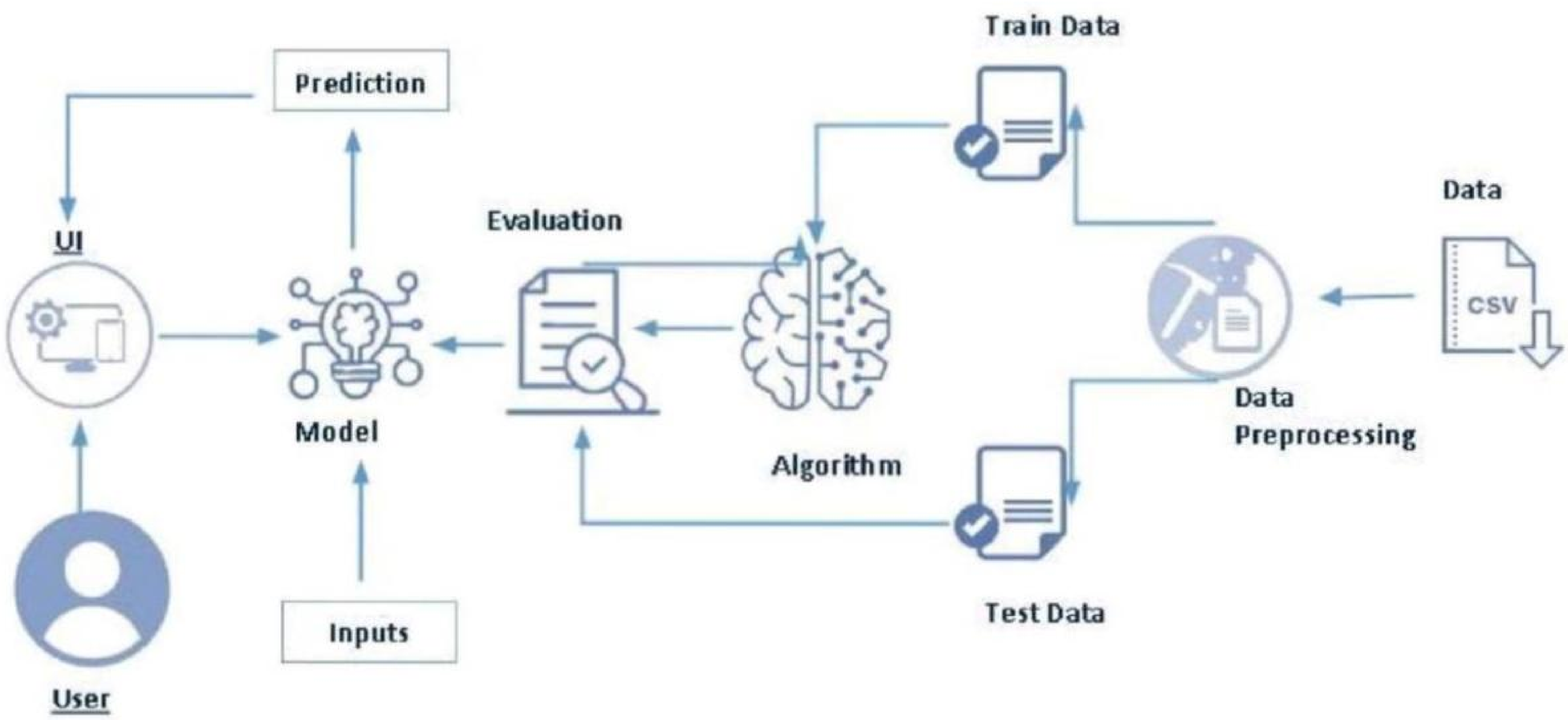
- NFR-1 Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process.
- NFR-2 Security consists of the platforms which protect your organization's users, endpoints and their online activity to more efficiently correlate threats. As users are increasingly logging in to networks via their personal devices, securing these is just as important as securing company owned devices.
- NFR-3 Reliability requirements are typically part of a technical specifications document. They can be requirements that a company sets for its product and its own engineers or what it reports as its reliability to its customers. They can also be requirements set for suppliers or subcontractors.
- NFR-4 Performance requirements define how well the software system accomplishes certain functions under specific conditions. Examples include the software's speed of response, throughput, execution time and storage capacity. The service levels comprising performance requirements are often based on supporting end-user tasks.
- NFR-5 Availability describes how likely the system is accessible to a user at a given point in time. While it can be expressed as an expected percentage of successful requests, you may also define it as a percentage of time the system is accessible for operation during some time period.
- NFR-6 Scalability Scalability requirements are, in essence, a reflection of the organization's ambition to grow and the need for a solution to support the growth with minimal changes and disruption to everyday activities.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is store.

5.2 Solution and Technical Architecture



5.3 User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application	I can access my account / dashboard	High	Sprint 1

			by entering my email, password			
			and confirming my password.			
		USN-2	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Medium	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can register by entering the details	Low	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can access my dashboard	High	Sprint 1
	Facebook login	USN-5	As a user, I can log into the application by using facebook	I can access my dashboard	Medium	Sprint 1

	Gmail login	USN-6	As a user, I can log into the application by using gmail	I can access my account / dashboard	Low	Sprint 1
	Analyze or detect problems	USN-7	As a user, I can able to analyze the problem in aircraft engine.	I can analyze the problem	High	Sprint 1
	Identify the fault engine	USN-8	As a user, I can identify the engine that is get fault	I can access the engine data	Medium	Sprint 1
	Solution	USN-9	As a user, I can view the solution for mina d and major problems	I can receive alert email	High	Sprint 1
	solution	USN-10	As a user I can find the solution and suggestion for maintain the engine	I can track expense	High	Sprint 1

6. PROJECT PLANNING & SCHEDULING

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	4
Sprint-1	Facebook Registration	USN-2	As a user, I can register for thapplicationt hrough Facebook	4	Medium	4
Sprint-1	Gmail	USN-3	As a user, I	3	Low	4

	registration		can register for the applicationthrough Gmail			
Sprint-2	login	USN-4	As a user, I can log into the application by entering email & password	5	High	4
Sprint-2	Facebook	USN-5	As a user, I can log in into this application through Facebook	4	Medium	4
Sprint-2	Email	USN-6	As a user, I can log in into this application by entering my Google Account	3	Low	4
Sprint-3	Analyzing / Detecting Problems	USN-7	As a user, I can able analyze the defects in Aircraft Engine	5	High	4
Sprint-3	Analyzing / Detecting Problems	USN-8	As a user, I can able to view the repeated problems occurs in Aircraft Engine	4	Medium	4
Sprint-3	Analyzing / Detecting Problems	USN-9	As a user, I can able to find the defects occurs in Aircraft	4	Low	4

			Engine			
Sprint-4	Solution	USN-10	As a user, I can view the solution for minor problems of the Aircraft Engine	3	Medium	4
Sprint-4	Solution	USN-11	As a user, I can view the solution for major problems of the Aircraft Engine	5	High	4
Sprint-4	Solution	USN-12	As a user, I can find the solution and suggestion to maintain for regular services	4	Low	4

Project Tracker, Velocity & Burndown Chart: (4 Marks):

Sprint	Total Story	Duration	Sprint St...

alerts to know about the engine performance using the sensor values. Alerts if any performance fault is found in data.It helps to maintain the engine in a proper state.

Codes:

deploy. Py:

```
from flask import Flask, render_template, request
import numpy as np


import requests

# NOTE: you must manually set API_KEY below using information retrieved from
your IBM Cloud account.
API_KEY = "<Your API>"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}

app = Flask(__name__)


@app.route('/')
def home():
    return render_template('home.html')


@app.route('/login')
def home():
    return render_template('login.html')


@app.route('/register')
def home():
    return render_template('register.html')


@app.route('/index')
```

```
@app.route('/result', methods= ['POST'])
def result():
    try:
        if request.method == 'POST':
            l=[]
            l.append(float(request.form['id']))
            l.append(float(request.form['cycle']))
            l.append(float(request.form['set1']))
            l.append(float(request.form['set2']))
            l.append(float(request.form['set3']))
            l.append(float(request.form['s1']))
            l.append(float(request.form['s2']))
            l.append(float(request.form['s3']))
            l.append(float(request.form['s4']))
            l.append(float(request.form['s5']))
            l.append(float(request.form['s6']))
            l.append(float(request.form['s7']))
            l.append(float(request.form['s8']))
            l.append(float(request.form['s9']))
            l.append(float(request.form['s10']))
            l.append(float(request.form['s11']))
            l.append(float(request.form['s12']))
            l.append(float(request.form['s13']))
            l.append(float(request.form['s14']))
            l.append(float(request.form['s15']))
            l.append(float(request.form['s16']))
            l.append(float(request.form['s17']))
            l.append(float(request.form['s18']))
            l.append(float(request.form['s19']))
            l.append(float(request.form['s20']))
            l.append(float(request.form['s21']))
            l.append(float(request.form['s22']))
            print(l)
            # NOTE: manually define and pass the array(s) of values to be scored
in the next line
            payload_scoring = {"input_data": [{"fields":
['f0','f1','f2','f3','f4','f5','f6','f7','f8','f9','f10','f11','f12','f13','f14'
,'f15','f16','f17','f18','f19','f20','f21','f22','f23','f24','f25','f26'],
"values": [l]}]}

response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/c287130b-1697-49d5-86e7-
302bd8fccdcd/predictions?version=2022-11-14', json=payload_scoring,
```

```python
            print("Scoring response")
            print(response_scoring.json())
            pred = response_scoring.json()
            output = pred['predictions'][0]['values'][0][0]
            print(output)

            if output >=1 and output <=2 :
                return render_template('result.html',data="normal")
            elif output >2:
                return render_template('result.html',data="excess")
            else :
                return render_template('result.html',data="low")
    except:
        return render_template('result.html',data="error")
if __name__=="__main__":
    app.run(debug=True)
```

8. TESTING:

8.1 TESTING:

- Login Page
- Prediction Page
- Result Page

8.2User Acceptance Testing:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Product Name] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severit y 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	8	15
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	9	2	4	11	20
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	0	1	8
Totals	22	14	11	22	5 1

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	F ai l	Pa ss
Login	7	0	0	7
Prediction	27	0	0	27
Result	4	0	0	4

9. RESULTS

9.1 Homepage


Enter Engine Parameters
Engine ID
Cycle
Setting 1
Setting 2
Setting 3
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14-A
S14-B
S15
S16
S17
S18
S19
S20

RAKESH AND TEAM

PREDICTIVE ANALYSIS FOR AIRCRAFT ENGINE USING MACHINE LEARNING

THE ENGINE REQUIRES IMMEDIATE SERVICE

ANOMALIES FOUND IN THE GIVEN DATA - ENGINE MAY ENCOUNTER ISSUES WITHIN 30 DAYS

GO BACK

10.2 DISADVANTAGES:

Although machine learning is considered to be more accurate it is highly vulnerable. For example, a set of programs provided to the machine may be biased or consist of errors. The same program is used to make another forecast or prediction then there will be a chain of errors that could be formed which may, although recognized but take some time to find out the source of the error.

The more data a machine gets the more accurate and efficient it becomes thus more data is required to input to the machine for better forecasting or decision making. But it may sometimes not be possible. Also, the data must be unbiased and of good quality. Data requirements are problematic sometimes.

As we have already seen that a little manipulation or biased data could lead to a long drawn error chain and therefore there are chances of the inaccuracy of interpretation also. Sometimes data without any error could also be interpreted inaccurately by the machine as the data provided previously may not fulfill all the basics of the machine.

11 CONCLUSION:

Overall, the results show that by bringing together sufficient (big) high quality data, robust machine learning algorithms, and data science, machine learning-based predictive analytics can be an effective tool for engine design-space exploration during the conceptual design phase. It would help to identify the best engine design expeditiously amongst several candidates. The promising results of the predictive analytics show that machine-learning techniques merit further exploration for application in aircraft engine conceptual design.

To further improve the accuracy (and reduce the uncertainty) of TSFC prediction, the database needs to be expanded. However, the limitation of publicly available engine data is a challenge to overcome.

12. FUTURE SCOPE:

- Early predictions avoid the accident and other problems.
- The process maintenance become easier..
- Predicting future also saves the money and the resources.
- Controls the machine and its performance.
- Train model in various machines can useful for the performance and maintenance.
- Machine learning algorithms can used for the models and the models monitor the performances.
- The algorithms can be update in high performance like the solution it will find itse

GIT HUB AND DEMO LiNK

https://github.com/IBM-EPBL/IBM-Project-5921-1658819999.git