A Project report on

Machine Learning Approach For Predictive Maintenance Aircraft Engine Using IBM Watson Studio

By

B.Pramod(19701A05A2)
M.Pramukha(19701A05A3)
S.Reshma(19701A05C0)
H.Saileela (19701A05C4)

Introduction:

1.1.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:

Predictive maintenance for aircraft can aid in determining the right moment to replace a part.This is critical because replacing too can lead to unexpected failures,flight delays,cancellations.Thus,predictive maintenance can help to resolve this problems.

2.Literature Survey:

2.1.Existing Problem:

The Infosys Travel & Hospitality practice helps airline enterprises leverage digital tools for safe and sustainable operations. We integrate diverse data streams using scalable data architecture to capitalize on data mining tools, predictive analytics, and machine learning-based rare event and Complex Event Processing (CEP) models. A digital ecosystem facilitates predictive maintenance to minimize grounding of aircraft for servicing

Our analytical solutions collate, correlate and analyze data from various aircraft systems, including Maintenance, Repair and Overhaul (MRO) history, engineering data, aircraft utilization records, flight parameters, crew logbooks, and weather reports. Our approach enables contextual diagnostics to predict failure and identify optimum maintenance intervals. The correlation of messages and patterns helps mitigate malfunctions, while revealing factors contributing to potential maintenance issues; for example, flight routes or flying practices causing safety issues, excessive wear and tear, or structural damage.

2.2.Proposed Solution:

Our solution for this problem is to create a dataset on the aircraft maintenance with id,settings , 21 sensors and trajectory fields in it.We load the dataset ,pre process it and calculate the time of failure based on the given dataset.Split the dataset into training and testing part.Then build the model by using appropriate algorithm.

There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms can be chosen according to the objective. As the dataset which we are using is a Classification dataset so you can use the following algorithms

- Logistic Regression
- Random Forest Regression / Classification
- Decision Tree Regression / Classification
- K-Nearest Neighbors
- Support Vector Machine

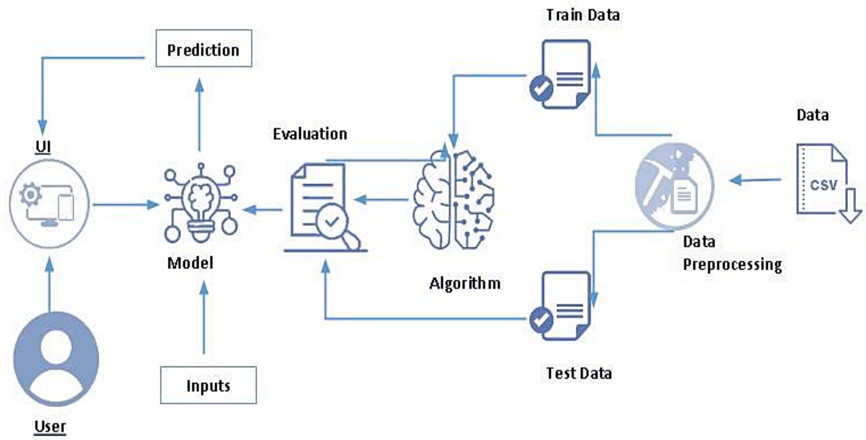
You will need to train the datasets to run smoothly and see an incremental improvement in the prediction rate.

After the model is built, we will be integrating it to a web application where a user can give the sensor data and get to know if the engine fails in the next 30 days or we can even generate the random values and know the prediction for those random data.

Then run the application.

3.Theoritical analysis:

3.1:Block diagram:



In this the user gives input to the UI(application).Model take it and evaluate with the help of train data ,test data and algorithm.Evaluate it and give predictions based on the inputs given by the user.Based on the predictions the user will evaluate whether there is a need for providing maintenance to the aircraft or not.

3.2:Hardware or Software Designing:

software requirements:

- Identify the System Boundary: Develop a clear definition of the boundary between the system and its environment. This provides a solid understanding of what lies within the system to be built and what lies within a larger environment. This is done by identifying a set of variables in the environment the system will monitor and control.
- Define the system boundary early in the requirements engineering process by identifying a preliminary set of monitored and controlled variables.
- Choose environmental variables that exist in the environment independently of the system to be developed.
- Choose controlled variables that are under the direct control of the system being specified.

- Choose monitored variables that are being directly sensed by the system being specified.
- Ensure the monitored and controlled variables are as abstract as possible and do not include implementation details.
- Avoid incorporating details of the operator interface in the monitored and controlled variables. Instead, define monitored or controlled variables that describe the information to be conveyed independent of its presentation format.
- Completely define all physical interfaces to the system, including definitions for all discrete inputs, all messages, all fields in a message, and all protocols followed.
- Develop the operational concepts.
- Identify the Environmental Assumptions
- Develop the functional architecture.
- Revise the architecture to meet implementation constraints.
- Identify the system modes.
- Develop the detailed behavior and performance requirements.
- DEFINE THE SOFTWARE REQUIREMENTS.
- Define the Software Requirements: With careful structuring, the software requirements and their architecture can map directly to the system requirements and their architecture. This recommended practice describes how to define the software requirements as a straightforward extension of the system requirements.
- For each input the software must read, provide a description of anything the software developer must know to access and correctly interpret the input. This may include an input description, the data format, the range of values it may assume, its location, and any protocols to follow when accessing it.
- For each input the software must read, provide a specification of its accuracy, where accuracy refers to the amount that its value may deviate from its ideal value.
- For each input the software must read, provide a specification of its latency, where

latency refers to the maximum time that its value may lag behind the true value of the monitored variable or variables it represents.

- For each monitored variable, specify how to recreate an image of the monitored variable in software from the input variables.
- For each monitored variable, specify how to recreate its status attribute from the input variables.
- If design choices must be made when recreating a monitored variable in software that may affect the externally visible system behavior or system safety, flag those decisions as derived software requirements to be reviewed by the safety assessment process.
- For each output the software must set, provide a description of anything the software developer must know to access and correctly set its value. This may include an output description, the data format, the range of values it may assume, its location, and any protocols to follow when accessing it.
- For each output variable the software must set, provide a specification of its latency, where latency refers to the maximum allowed time from when the output variable is set until it changes the controlled variable value.
- For each output the software must set, provide a specification of its accuracy, where accuracy refers to the amount the affected control variable can diverge from its ideal value.
- For each controlled variable, specify how to set the output variables values based on the value of the controlled variable image in software.
- For each controlled variable, confirm that the latency and accuracy specified in the system requirements can be met given the latency and accuracy of the input and output variables and the computation time of the software.

Hardware requirements:

The on-site technician needs an iOS or Android mobile device with a broadband connection for a **remote video inspection**. CloudVisit Aircraft Maintenance Software is able to connect with third-party devices such as a borescope, allowing you to inspect inaccessible spaces such as an engine cylinder and enhance aircraft maintenance, inspection, and aviation safety.

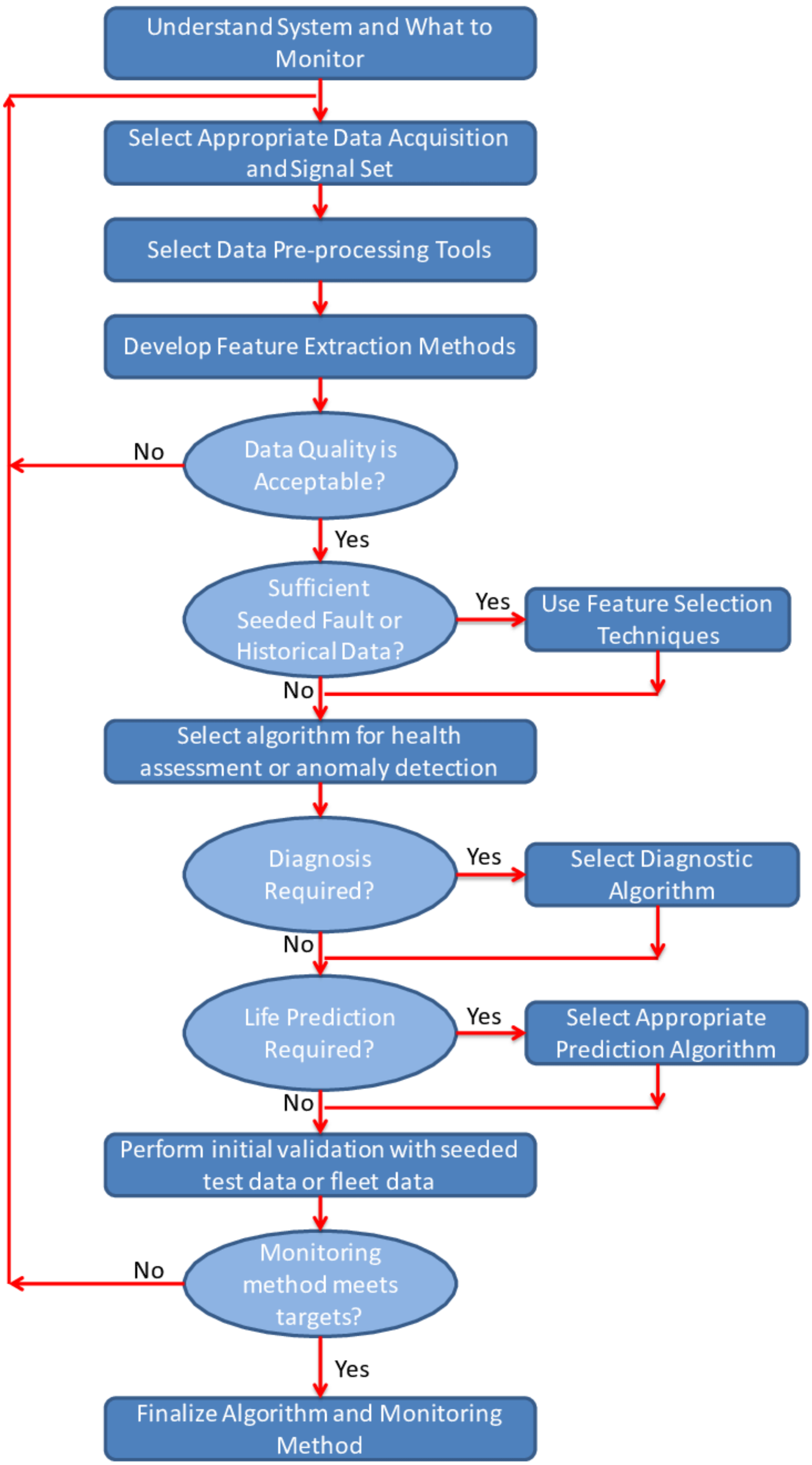
4.Experimental Investigation:

Operations research methods are widely used for different planning and scheduling problems in the airline industry. These problems can be divided into four major classes: Flight scheduling, fleet assignment, aircraft maintenance routing and crew scheduling (Liang & Chaovalitwongse, 2009). In general terms, the flight scheduling problem deals with the scheduling of flights so that the market demand is met. The fleet assignment problem sorts out the assignments of plane types to fleets for predetermined flights with the aim of maximizing the total profit. While the crew scheduling problem tries to handle the assignments of crew members to each aircraft, the aircraft maintenance routing problem (AMRP), which is the main focus of this study, deals with arranging routes for the aircrafts so that the maintenance regulation constraints are not violated. Although all of these problems have been widely studied for the last few decades, the challenge remains due to high complexity of airline networks and increasing size of the industry.

There are established thresholds for the number of consecutive flight days and the number of operating hours for an aircraft beyond which a predetermined maintenance check must take place. The maintenance checks have different frequencies and duration s depending on their type (Clarke, Johnson, Nemhauser, & Zhu, 1997). A *type A* check is repeated every 65–125 hours of flying or every week and it involves visual inspection of major systems for about eight hours. A *type B* check is repeated every 300–600 hours of flying and lasts around 1–3 days. *Type C* and *type D checks* are repeated once in every one to four years and they can be only completed at specialized hangars in about one month (Sriram & Haghani, 2003). AMRP addresses short-term maintenance requirements with shorter maintenance frequencies. The rationale behind this is that longer checks which are less frequent directly affect the fleet capacity; hence, such maintenance checks must be considered while solving the fleet assignment problem.

Among the many studies that deal with AMRP, most of the early ones address the problem at a more tactical level, ignoring the operational level constraints and dynamic issues. In these studies, the aim is to find a unique rotation of flights (starting and ending at the same location) that will be repeated by each aircraft in the fleet with a certain lag. In practice, using a single rotation for the entire fleet may not be applicable due to stochasticity and operational considerations in the airline industry. Thus, AMRP has also been addressed at a more operational level to assign *maintenance feasible* flight sequences to each individual aircraft (identified by its tail number) of a given fleet by considering the current states of the aircrafts while covering all the flights in the flight schedule over a short-term planning horizon. A route is maintenance feasible when it contains no maintenance-free segment of flights whose accumulated duration is larger than the remaining time of the corresponding aircraft. The remaining time of an aircraft is defined as the difference between legal flying hour limit, which is the amount of time allowed between consecutive maintenance operations, and the accumulated flight duration since its last maintenance operation. In this study, our aim is to develop a fast and responsive methodology to solve the operational aircraft maintenance routing problem (OAMRP). For this purpose, a new ILP formulation is proposed for OAMPR. After attempting to solve this new model by exact methods, a heuristic method based on compressed annealing (CA) is proposed. The performance of the CA heuristic is validated with respect to exact solutions on a relatively small real-life flight network, then its applicability to larger networks is shown by using examples from the literature. Furthermore, the model and numerical examples are extended to incorporate capacity considerations for the maintenance facilities. Lastly, a rolling horizon based procedure is proposed to update the existing routes when some of the maintenance decisions are already fixed.

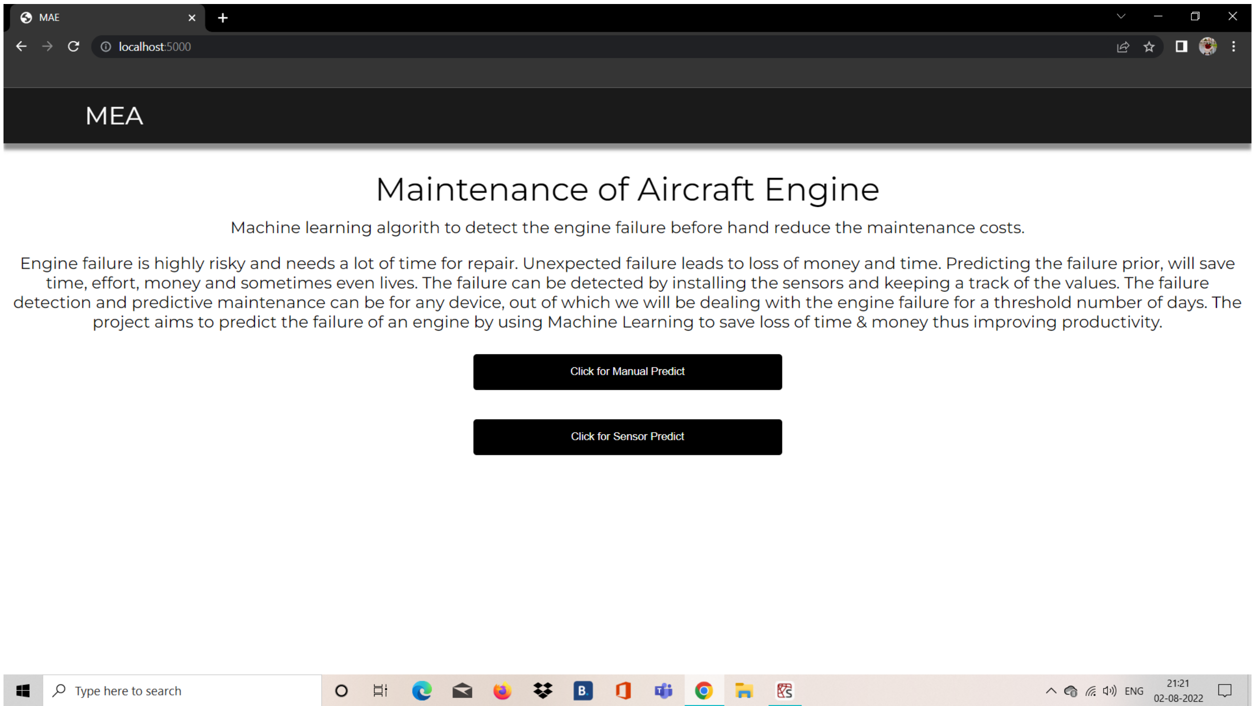
5.Flowchart:


Understand System and What to Monitor
Select Appropriate Data Acquisition and Signal Set
Select Data Pre-processing Tools
Develop Feature Extraction Methods
Data Quality is Acceptable?
No
Yes
Sufficient Seeded Fault or Historical Data?
Yes
Use Feature Selection Techniques
No
Select algorithm for health assessment or anomaly detection
Diagnosis Required?
Yes
Select Diagnostic Algorithm
No
Life Prediction Required?
Yes
Select Appropriate Prediction Algorithm
No
Perform initial validation with seeded test data or fleet data
Monitoring method meets targets?
No
Yes
Finalize Algorithm and Monitoring Method

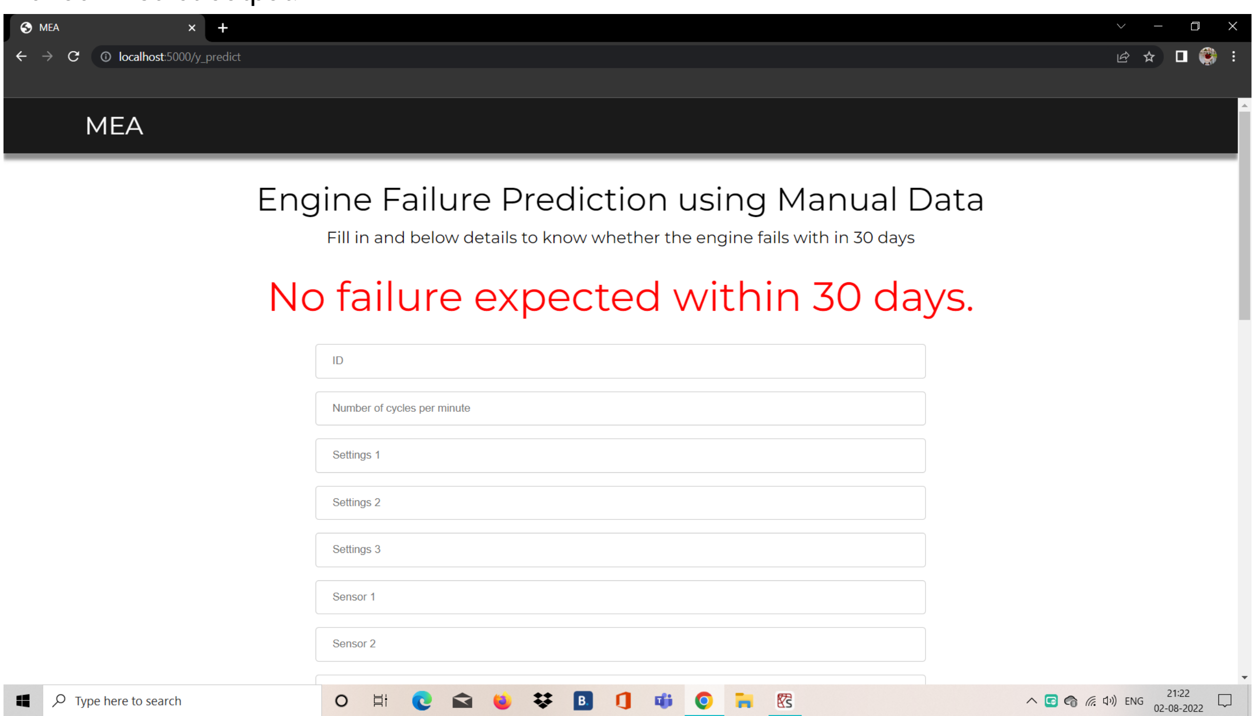
6.Result:

As a result you will get whether the maintenance is needed for the aircraft or not .This will helps to reduce the loss of money and time.It even prevents the health problems caused or damaged caused during aircraft avitation.

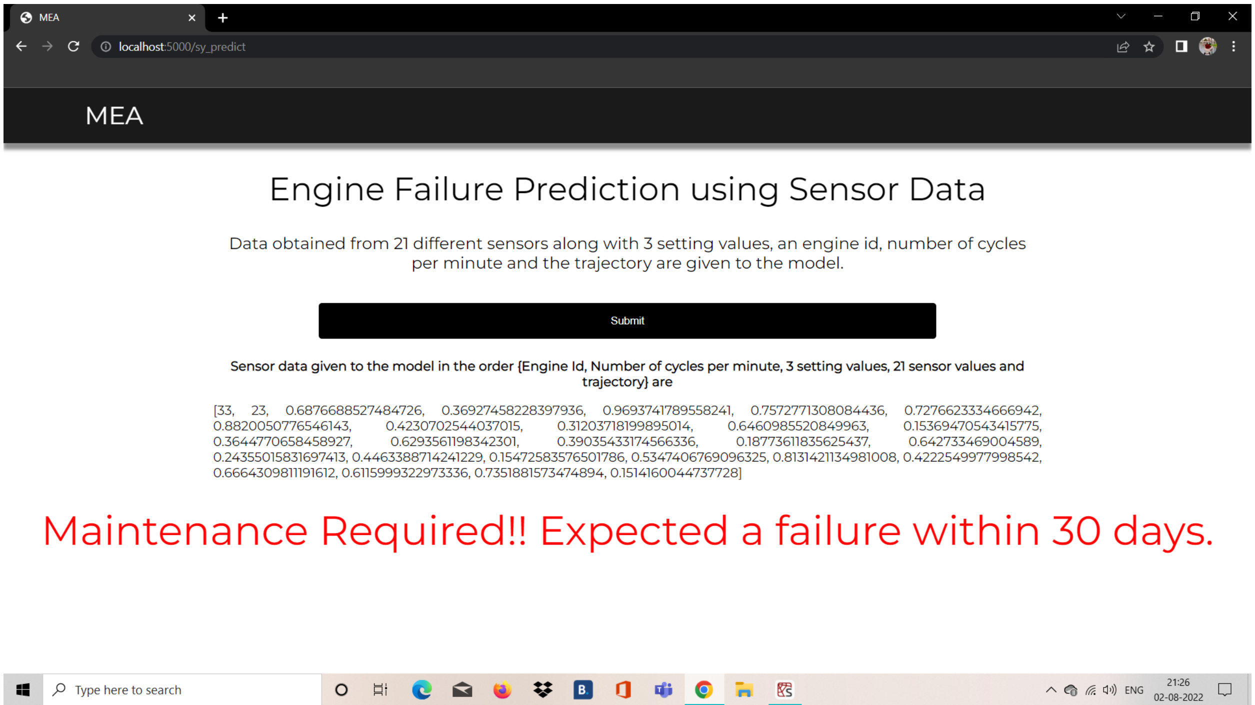
Home Page Output:



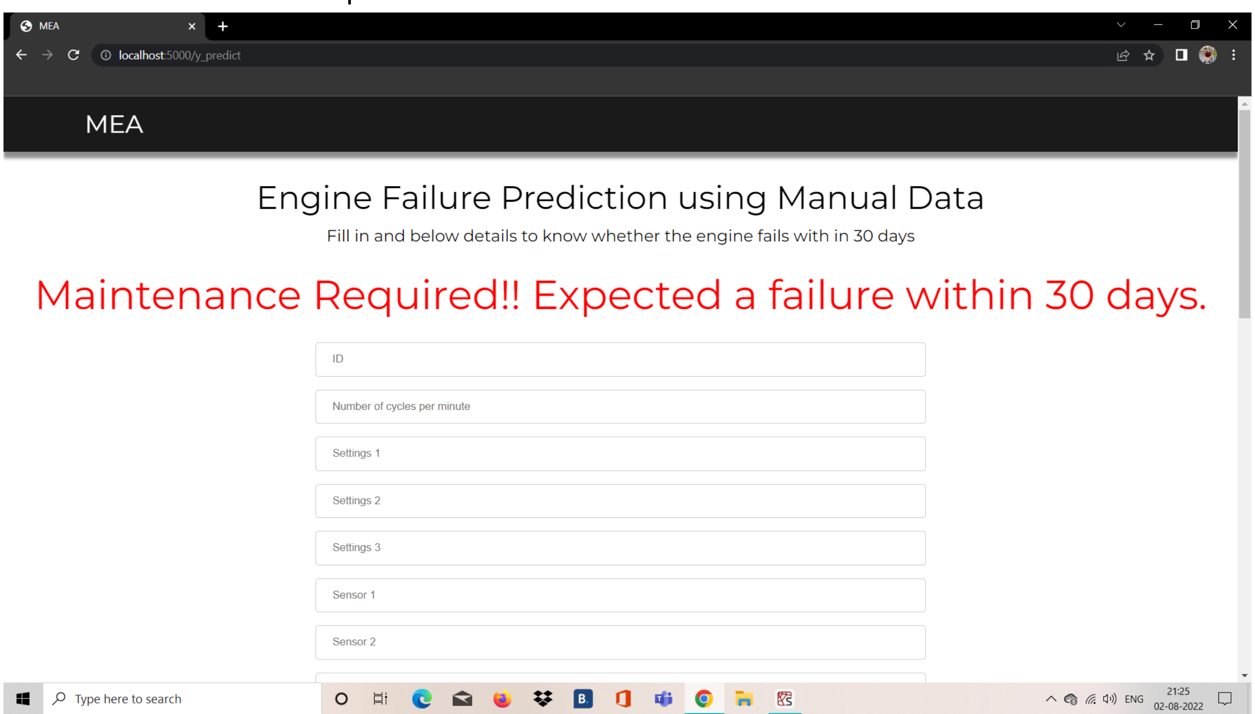
Manual Predict output:



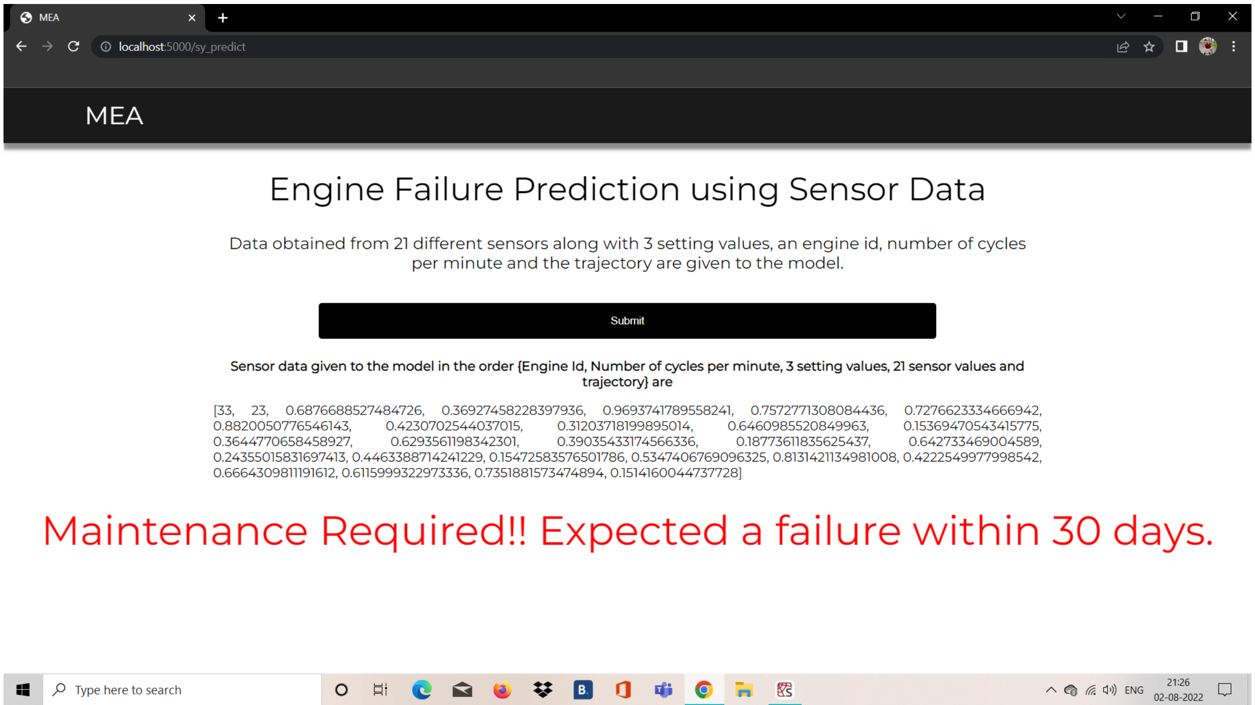
Sensor Predict Output:



IBM manual Predict Output:



IBM sensor Predict output:



7.Advantages and Disadvantages:

Pros:

Exciting and Challenging in a fun way

Working on the airport and watching it’s mundanities is quite relishing.

A lot of respect comes from friends and relatives.

You are responsible for the passengers’ lives so it gives a sense of satisfaction.

Can be high paying if you manage to secure a job in a good reputed airline.

Cons:

Rotating shifts may lead to messy sleep schedule and daily routine.

Your employer may transfer you to any city as per the requirements. (Although it is infrequent but still a possibility)

Cannot become an AME unless your employer trains and authorizes you.

Grounded flights and Staff shortages call for extra duty hours. So spending time with family becomes difficult.

Very less or no pay in the initial years of gaining experience.

8.Applications:

- ENGINE MAINTENANCE :

Over its service lifetime, the majority of an aircraft's maintenance exposure arises from three main areas: airframe, engine and components. Making up a significant contribution of about 30-40% of the total maintenance expenses, expenditures arising from the engine exhibit an important impact on the market value of the whole aircraft at any given time .

- COST-BENEFIT ANALYSIS:

In general, the Shop Visit Rate (SVR) of an engine may be broken into the scheduled removal rate (e.g. resulting from expiry of Life-Limited Parts (LLPs), performance deterioration and service bulletin compliance) and unscheduled removal rate. The latter measures the number of times unexpected engine anomalies or failures require engine removal for repair or refurbishment before normal maintenance intervals are reached (Ackert, 2012). This causes a shop maintenance event with associated Shop Visit Costs SVCs and the necessity of installing an airworthy (new or repaired) engine.

- POTENTIAL ANALYSIS FOR REDUCING UNSCHEDULED ENGINE MAINTENANCE COSTS:

With the aim of assessing cost reduction potentials regarding UER based on failure prediction .

9.Conclusion:

The main objective of predictive maintenance is to predict when equipment failures can occur. Then prevent that failure by taking relevant actions. Predictive Maintenance System (PMS) monitors future failures and will schedule maintenance in advance.

10.Future Scope:

Predictive maintenance is one of the most important topics in IT right now, not only in the aviation industry, but also in many other sectors. Here we can see the potential big data has to make maintenance processes more efficient and significantly reduce operating costs. Despite constant improvement in IT systems, the field remains complex and calls for a targeted approach to providing advice. First-hand knowledge and taking all perspectives into account are absolutely essential.

11.Bibilography:

1. Tong, M.T., "Using Machine Learning To Predict Core Sizes of High-Efficiency Turbofan Engines," GT2019-91432, ASME Turbo-Expo 2019, June 17-21, 2019.

2. Daly, M., "Jane's Aero-Engine," 2017-2018.

3. Meier, N., "Civil turbojet/turbofan specifications." http://www.jet-engine.net/civtfspec.html. Accessed August, 2018.

4. GE Aviation. https://www.geaviation.com/commercial

5. Pratt and Whitney. https://www.pw.utc.com/products-and-services/products/commercial-engines

6. Rolls Royce. https://www.rolls-royce.com/products-and-services/civil-aerospace

7. CFM International. https://www.cfmaeroengines.com/

8.M. Luo, Z. Xu, H.L. Chan, and M. Alavi. Online predictive maintenance approach for

semiconductor equipment. In Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE, pages 3662–3667. IEEE, 2013.
9. Khelif. (2014). Unsupervised Kernel Regression Modeling Approach for RUL Prediction.
10.Saxena A., Goebel K. Turbofan engine degradation simulation data set. NASA Ames Prognostics Data Repository (https://tiarcnasagov/tech/dash/groups/pcoe/ prognostic data-repository/), NASA Ames Research Center, Moffett Field, CA2008;
11.Future Scope:https://www.lufthansa-industry-solutions.com/de-en/solutions-products/aviation/advancing-the-future-of-predictive-maintenance.
12.Reasearch Gate:https://www.researchgate.net/figure/Prognostic-and-Health-Management-Development-Flow-Chart_fig1_271270784.
13.Hinton, G.E., Krizhevsky, A., Srivastava, N., Sutskever, I., & Salakhutdinov, R., "Dropout: A simple Way to Prevent Neural Networks from Overfitting." Journal of Machine Learning Research, 15, 1929-1958. June, 2014.
14. Kingma, D. P. and Ba, J., "Adam: A Method for Stochastic Optimization," International Conference on Learning Representations, May 2015.
15. Google, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Retrieved on February 20, 2019 from: https://www.tensorflow.org/

Appendix:

Source code:

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import joblib
import random
#import requests
# NOTE: you must manually set API_KEY below using information retrieved from your
IBM Cloud account.
'''API_KEY = "yRrxWYCduJe9oe664g4jPMnAK7oEXdcycNCMXbaFuYov"
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__)
model = joblib.load("engine_model.sav")
app = Flask(__name__)
@app.route('/')
def home():
   return render_template('base.html')
```

```
@app.route('/m_predict',methods=['POST'])
def mpred():
    return render_template('Manual_predict.html')
@app.route('/s_predict',methods=['POST'])
def spred():
    return render_template('Sensor_predict.html')
@app.route('/y_predict',methods=['POST'])
def y_predict():
    x_test = [[int(x) for x in request.form.values()]]
    print(x_test)
    a = model.predict(x_test)
    pred = a[0]
    if(pred == 0):
        pred = "No failure expected within 30 days."
    else:
        pred = "Maintenance Required!! Expected a failure within 30 days."

    return render_template('Manual_predict.html', prediction_text=pred)
@app.route('/sy_predict',methods=['POST'])
def sy_predict():
    inp1=[]
    inp1.append(random.randint(0,100)) #id
    inp1.append(random.randint(0,365)) #cycle
    for i in range(0,25):
        inp1.append(random.uniform(0,1))
        #inp1.append(random.randint(0,365)) #ttf
    pred=model.predict([inp1])
    if(pred == 0):
        pred = "No failure expected within 30 days."
    else:
        pred = "Maintenance Required!! Expected a failure within 30 days."
    return render_template('Sensor_predict.html', prediction_text=pred,data=inp1)

if __name__ == '__main__':
    app.run(debug=False)
```