Externship Program on Applied Data Science

Project Report

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

Guide: Shivani Kapoor

Group Members:

Tenzin Tsundue

Tenzin Gendun

Aman Kumar Jha

June 2022

INDEX

1 Introduction

1.1 Overiew

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.

1.2 Purpose

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

2 Literature Survey

2.1 Existing Problem

Every tools, machinery, and other durable equipment has to be checked for maintenance after a certain period of time. Though all those hardware has defined useful life, due to unfortunate damages their remaining useful life (RUL) reduces signficantly leading to undesirable incidents. Nevertheless with regular check ups and proper maintenance we can prevent those incidents and make sure those hardwares last as long it supposed to.

And aircraft is one of the machinery which has really high cost maintenance as according to 'Airline maintenance cost executive commentary FY2019 data Public Version' cost of aircraft maintenance is estimated to be 10.3% of the total airline operating costs, with approximately 3.3 million dollars spent on maintenance per aircraft in 2019. With such high maintenance cost, engineers are doing thier work deligently and numerous research as also been done on this issue.

Individual data science enthusiasts have published thier works on popular platform such as towardsdatascience[2] and kaggle[3].

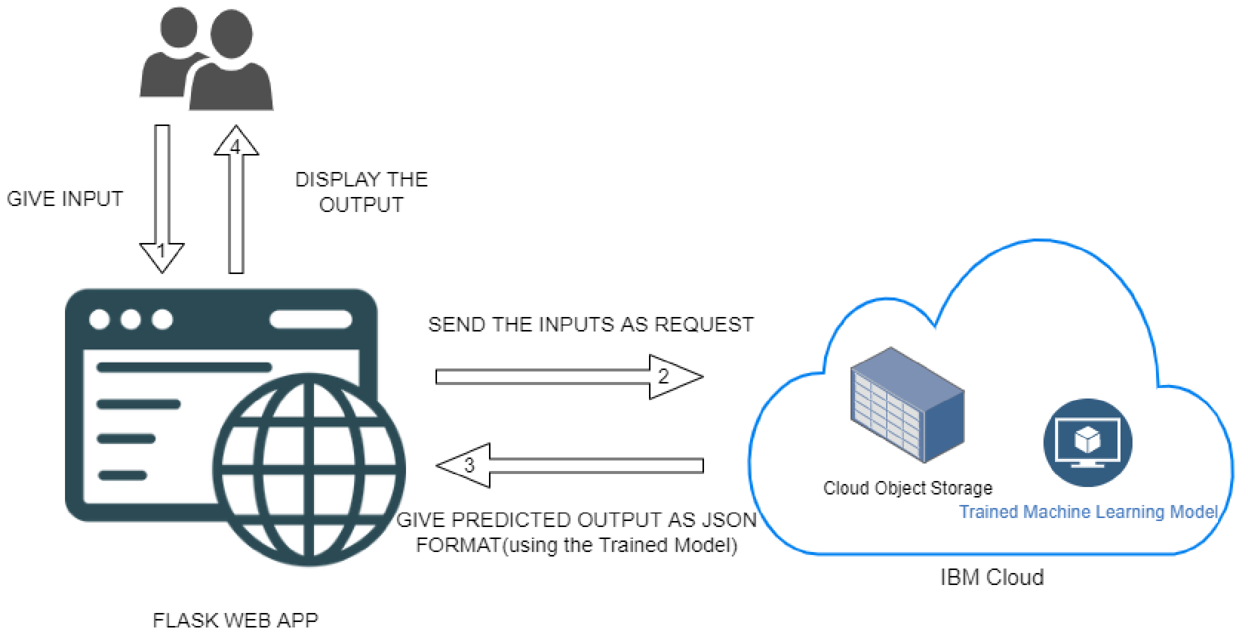
Published papers on this issue includes [4] where Sourajit et al used ensemble trees algorithm with IIoT for turbofan engines predictive maintenance.

2.2 Proposed Solution

To solve this problem, we used the available datasets of an aircraft engine with total of 21 sensors throughout the multiple cycles. We used machine learning algorithm called Logistic Regression to classify whether the engine needs a maintenance or not with 30days as threshold period and then displaying suggestive message for both the cases.
Then a beautiful interface is made in flask application and sending input as request to ibm cloud with trained model. The predicted output is then send back through an api.

3 Theoritical Analysis

3.1 Block diagram



3.2 Hardware / Software designing

Hardware requirements includes

i)Processor generation 8 or above

ii)4 cores or more and 8 threads or more

iii)Compatible with 4GB but should go for 8 and above

iv)Operating system: windows 10, linuxs

Software requirements includes

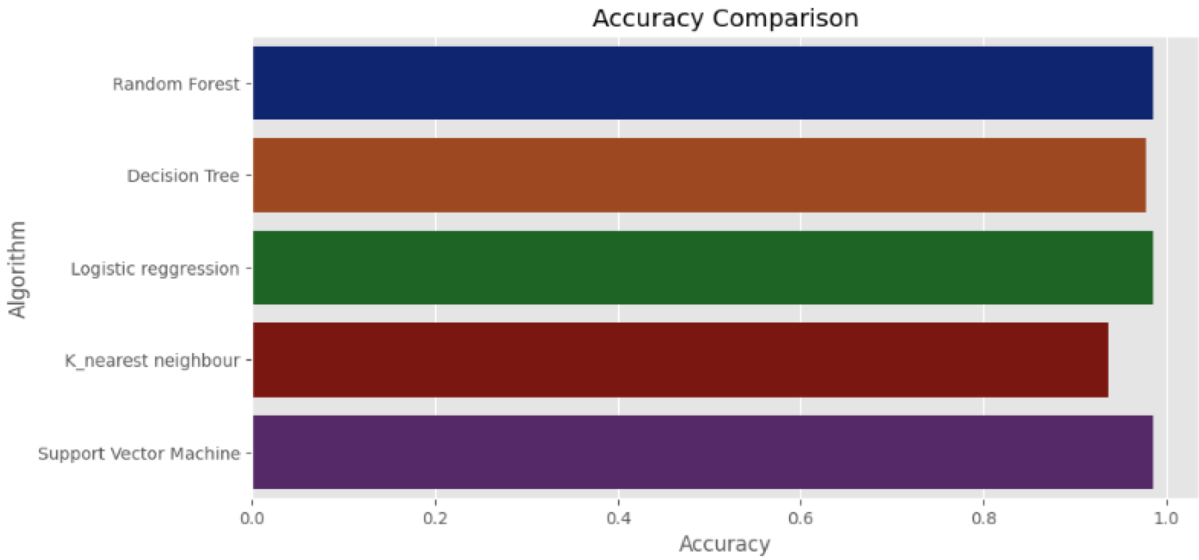
i)Python

ii)Flask with its modules(render_template, request)

iii)IBM Cloud account

iv)Machine Learning modules i.e sklearn, numpy, pandas, etc...

4 Experimental Investigations



ML Algorithms	Accuracy Performance		Time Taken	
	Training Data Set	Test Data Set	Training Data Set	Testing Data Set
Random Forest	0.9973	0.9861	478 ms	21.8 ms
Decision Tree	1	0.9777	272 ms	4.51 ms
Logistic Regression	0.9602	0.9862	825 ms	5.7 ms
K nearest neighbour	0.9956	0.9362	5.51 ms	6.34 s
SVM	0.9622	0.9856	1min 5s	963 ms

Training Data Set	
x_train shape	(20631,26)
y_train shape	(20631,1)
x_test shape	(13096,26)
y_test shape	(13096,1)

Table: Performance and Time Taken of various Machine Learning Algorithms used of Training and Test Data Set.

Points to be noted:

i)Accuracy are given in range of [0,1]. One can multiply it with 100 to get in % percentage.

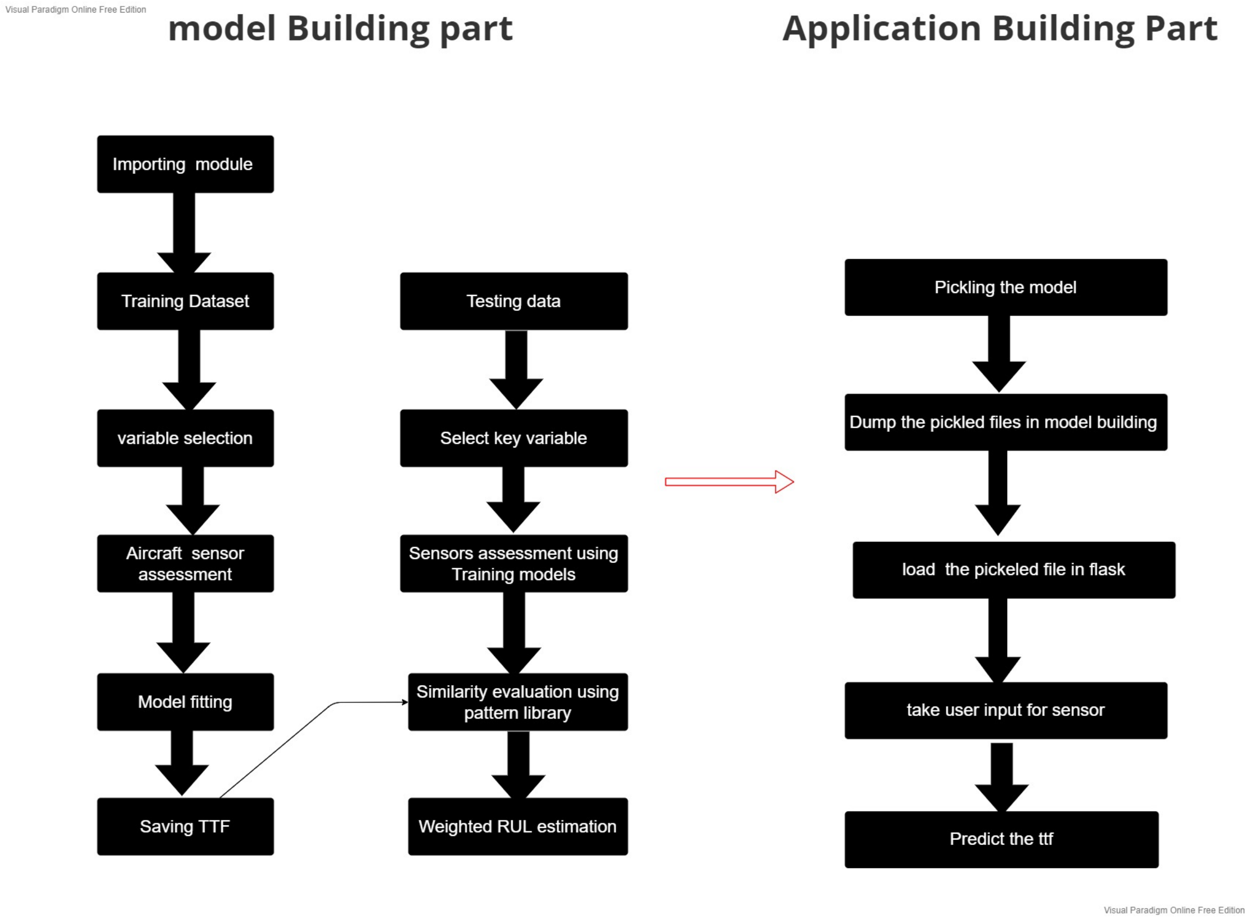
ii)Time taken is calculated in google colab using '%%time' command.

iii)ms = milli seconds , s = seconds, min = minute

From the table, we can noticed which algorithm scored high in accuracy and took less time in training and testing the data set.

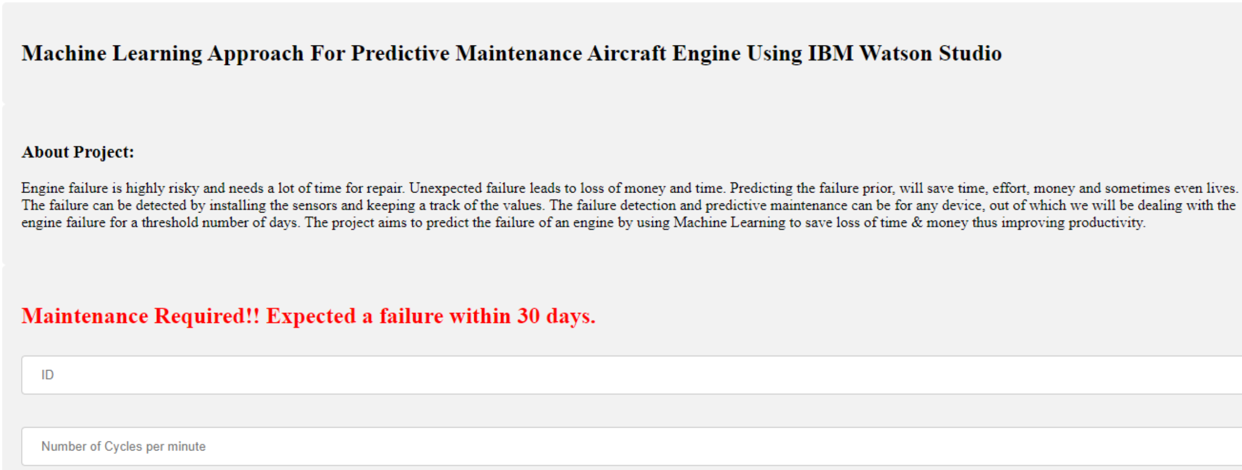
We chose Logistic Regression algorithm due to the fact, it has highest test accuracy as well as it taking really less time on both training and testing set.

5 Flowchart



6 Result

We were able to achieve 98.62% accuracy on 13096 test instances after training with 20631 training data instances using Logistic Regression.



7 Advantages & Disadvantages

7.1 Advantages

i)The model had high accuracy on test data set, so it will perform well on unseen datasets.

ii)The project has an attractive interface for manual prediction which make it easy to use for non-technical people.

iii)Code modification can be done easy from ibm jupyter notebook if any changes has to be made directly.

7.2 Disadvantages

i)The trained Machine Learning model is stored in ibm cloud connnected through account on limited free period.

ii)The services being used on ibm cloud has restrictions on resources provided such as storage, number of request, capacity..

iii) The logistic model had high accuracy on both training and test dataset. Though the performance of the model is beyond outstanding there could be a case that the dataset was too simple and wasnt drawn from wide area of range.

8 Applications

This project can used for real time manual prediction of aircraft engine with proper data input and can be a reference for any hardware maintenance issue research purposes.

9 Conclusion

We would like to humbly concluded that we made an complete web application regarding predicting maintenance issue on Aircraft Engine collated with IBM Watson Studio.

10 Future scope

For future, we could make the User Interface(UI) more interactive including dashboards and details happening in behind the scene.

11 Biblography

[1] https://smartinternz.com/externship
[2] https://towardsdatascience.com/predictive-maintenance-of-turbofan-engines-ec54a083127
[3] https://www.kaggle.com/datasets/maternusherold/pred-maintanance-data
[4] https://dl.acm.org/doi/abs/10.1145/3297280.3297363

12 Screeenshots of project

app_ibm.py > predict

```python
from flask import Flask, render_template, request
import requests


# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "dtEJtUBwh-4FQHs46Ph-mq8d9VTPBNvgIfLIAMPDUMe8"
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 hello_world():
    return render_template('index.html')

@app.route("/predict", methods= ['POST'])
def predict():
    id = request.form['id']
    cycle = request.form['cycle']
    setting1 = request.form['setting1']
    setting2 = request.form['setting2']
    setting3 = request.form['setting3']

    s1 = request.form['s1']
    s2 = request.form['s2']
    s3 = request.form['s3']
```

```
model_details
```


```
{'entity': {'hybrid_pipeline_software_specs': [],
  'label_column': 'l0',
  'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',
   'name': 'runtime-22.1-py3.9'},
  'training_data_references': [{'connection': {'access_key_id': 'not_applicable',
     'endpoint_url': 'not_applicable',
     'secret_access_key': 'not_applicable'},
    'id': '1',
    'location': {},
    'schema': {'fields': [{'name': 'f0', 'type': 'float'},
      {'name': 'f1', 'type': 'float'},
      {'name': 'f2', 'type': 'float'},
      {'name': 'f3', 'type': 'float'},
      {'name': 'f4', 'type': 'float'},
      {'name': 'f5', 'type': 'float'},
      {'name': 'f6', 'type': 'float'},
      {'name': 'f7', 'type': 'float'},
      {'name': 'f8', 'type': 'float'},
      {'name': 'f9', 'type': 'float'},
      {'name': 'f10', 'type': 'float'},
      {'name': 'f11', 'type': 'float'},
      {'name': 'f12', 'type': 'float'},
      {'name': 'f13', 'type': 'float'},
      {'name': 'f14', 'type': 'float'},
      {'name': 'f15', 'type': 'float'},
      {'name': 'f16', 'type': 'float'},
      {'name': 'f17', 'type': 'float'},
      {'name': 'f18', 'type': 'float'},
      {'name': 'f19', 'type': 'float'},
      {'name': 'f20', 'type': 'float'},
      {'name': 'f21', 'type': 'float'},
      {'name': 'f22', 'type': 'float'},
      {'name': 'f23', 'type': 'float'},
      {'name': 'f24', 'type': 'float'},
      {'name': 'f25', 'type': 'float'}],
```