# VARUVAN VADIVELAN INSTITUTE OF TECHNOLOGY

## DEVELOPING A VEHICLE PERFORMANCE ANALYZER MODEL USING MACHINE LEARNING

Bonafide record of work done by

**VIGNESH (612819106049)**

**DINESH (612819106007)**

**PARTHIBAN(612819106028)**

**JEISHEELAN(612819106301)**

Project report submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING – ECE

of Anna University

**NOVEMBER 2022**

**CONTENTS 1. INTRODUCTION**

## CHAPTER 1 INTRODUCTION

Vehicle performance analysis is the study of performance of a vehicle. The performance of any vehicle depends upon all the forces and moments that act upon it. These forces and moments, for the most part are caused by interaction of the vehicle with the surrounding medium(s) such as air or water (e.g. fluid static and dynamic forces), gravitational attraction (gravity forces), Earth's surface (support, ground, or landing gear forces), and on-board energy consuming devices such as rocket, turbojet, piston engine and propellers (propulsion forces). Consequently, in order to fully understand the performance problem, it is necessary to study and in some way characterize these interacting forces. Defects caused by these forces can be identified and analysed and the performance can be improved.

### PROJECT OVERVIEW

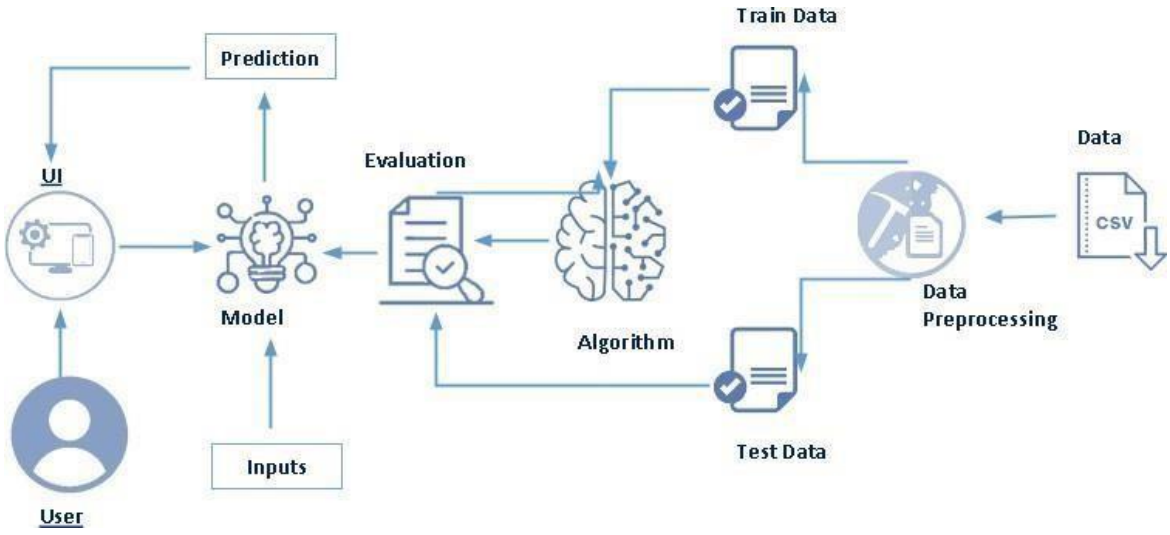



**Figure 1.1. Technical Architecture**

Vehicle performance can be predicted using a machine learning algorithm. Rows of feature vectors, such as aerodynamics, rigidness of the vehicle body, fuel consumption, and sensor detection provide the input to our algorithm. The decision tree classifier is then used to determine the performance of the vehicle. When there is a defect found in any of the given inputs, a vehicle is deemed to be not upto its performance level .

## PURPOSE

The main goal of this project is to predict the vehicle's performance using machine learning algorithms. Vehicle performace analysing is one of the difficulties in the industrial environment because there are many unpredictabilities. One such condition is sensor defects, which can result in unwanted sensory alarms in the vehicle. Defects in aerodynamics can be brought by a damage on the body of the vehicle. Hence Predicting defects can

improve vehicle performance and customer satisfaction, which will result in a positive impact on the economy.

## CHAPTER 2
# LITERATURE SURVEY

| LITERATURE SURVEY: | | | | |
|---|---|---|---|---|
| **TITLE** | **AUTHOR** | **ALGORITHM** | **ADVANTAGES** | **DISADVANTAGES** |
| **VEHICLE PERFORMANCE ANALYZER BASED ON DEEPLEARNING AND LEVENBERG-MARQUART ALGORITHM** | Daniel A.Roberts and Sho Yaida,Nikhil Buduma,Nithi Buduma, Joe papaand Nicholas Locascio. | Deep Learning | Integrating a GPS tracking device is beneficial to reduce the overhead cost as it prevents the consumption of fuel, unnecessary overtimeexpenses, and unauthorized vehicle usage. The tracking system helps you to monitor the activitiesof drivers more efficiently. | GPS receivers rely on signals from at least four satellites. If they only connect with three, the positioning is not entirely accurate. When obstacles such as walls, buildings, skyscrapers, and trees obstruct a signal, problemscan arise. |

| | | | | |
|---|---|---|---|---|
| **VEHICLE PERFORMANCE ANALYSIS USING MACHINE LEARNING ALGORITHM (XGBOOST)** | Oliver Theobald,AndriyBurkov,Chip Huyen,Robert Munro | Machine Learning | Different ML algorithms to predict ifa Vehicle performancewill be good or bad. So, it will not be aiming to get the highest accuracy possible, because it would be quite easy byadding a series of features that will bias the model in terms of predictive power. So, this information was looked at as part of theExploratory Data Analysis (EDA). | The authors compare various machine learning algorithms topredict vehicle performance, but failed to consider simple neural networks and decision tree classifiers. Sosimple machine learning algorithms like decision tree and simple neural networks to be implemented to predict vehicle performance, andinvestigate if we can predict vehicle performance with fewer feature-set accurately. |

| | | | | |
|---|---|---|---|---|
| **BIG DATA IN VEHICLE PERFORMANCE ANALYZER PREDICTION FOR MAINTAINING DATABASE OF THE ENGINE'S PERFORMANCE OF VEHICLES.** | Nathan Marz, James Warren,V.K.Jain,Tony Guida | Big Data | The data source and pre-processing steps including the data merging and cleansing will be introduced. As each dataset is untidywith messy redundant records and missing values. Using forward type vehicle simulation,It aims at comparing the potential and limitations of front wheel drive and rear wheel drive electric motorplacements for regenerative braking under extreme driving situations. | Data is thoroughly examined for integrity criteria as well. Since expected model is to work with all the forms like offline, near line and online data, the irrelevant and unnecessary parameters that could overburden the dataset is reduced. Dropped the null values and assigned zero to Not aNumber (NaN) values as one of the data cleansing activities. The data types of time factors such as scheduled time, motion time etc., are found to be in float point and proper conversion of input time to standard date time format. Finally, the data is analysed for distribution, converting and pre-processing. Then different datasets such the horse power, engine size, mileage and etc., are integrated and normalized to identify thecorrelating factors that affect the vehicle performance. |

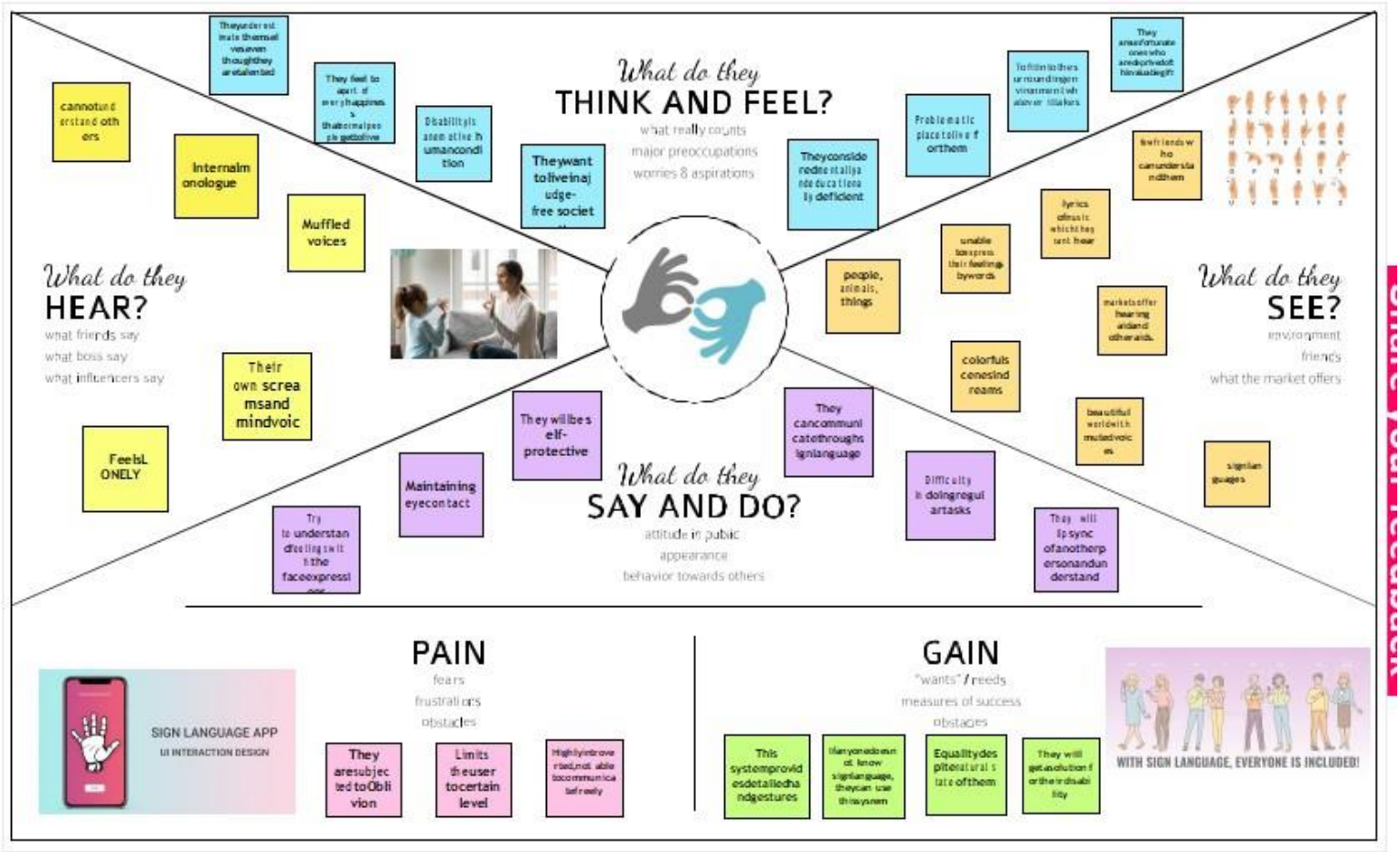
| | | | | |
|---|---|---|---|---|
| **PREDICTION OF THE VEHICLE'S PERFORMANCE USING DATA MINING.** | Shantanu Pardhi,Ajinkya Deshmukh, Hugo<br><br>Ajrouche | Data Mining | First, the considered dynamic/data-driven modelling approach for the complete traction chain with attention to theeffects of detailed vehicledynamics has been implemented in MATLAB Simulink. Simple parallel regenerative braking technique and recuperation favouring brake distribution strategies have been employed on a performance electric car considering front and rearwheel propulsion cases. Powertrain behaviour in adynamic driving scenario has been investigated to understand how the two cases with the corresponding recuperation favouring braking strategies performunder elevated transient vehicle dynamics. Finally,the impact of normal load transfer, tyre slip and wheel adhesion limits on regenerative braking has been quantitatively compared for the complete range of brake pedal demands using high-speed braking tests while avoiding wheellock-up. | Data collected are vehicles performance is stored in a warehouse willbe updated only at certain points in time. In this way,there is a trade-off exists between the correctness ofdata and the substantial effort required to bring thedata into the warehouse. Data warehouses of vehicle performance also provide a great deal of opportunities for performing data mining tasks such as classificationand summarization. Updates are collected andapplied to the data warehouse periodically. Then, all patterns derivedfrom the warehouse by some data mining algorithm have to be updated as well. |

# CHAPTER 3 IDEATION & PROPOSED SOLUTION

## EMPATHY MAP CANVAS

**Figure 3.1. Empathy Map**

## IDEATION & BRAINSTORMING



## Step 1 - Team Gathering, Collaboration and Selecting the Problem Statement

## Step 2 - Brainstorm, Idea Listing and Grouping and Idea Prioritization

Template

# Vehicle Performance Analysis.

In this,we are going to see about how can we analyse the performance of vehicles and help our customers to maintain their vehicles for a long time.This is where our project comes.

**10 minutes** to prepare
1 hour to collaborate
**2-8 people** recommended

### Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering
*our team include 4 members.*

Set the goal
*analyse the performance of the vehicles.*

Learn how to use the facilitation tools
*Google collab.*

Open article →

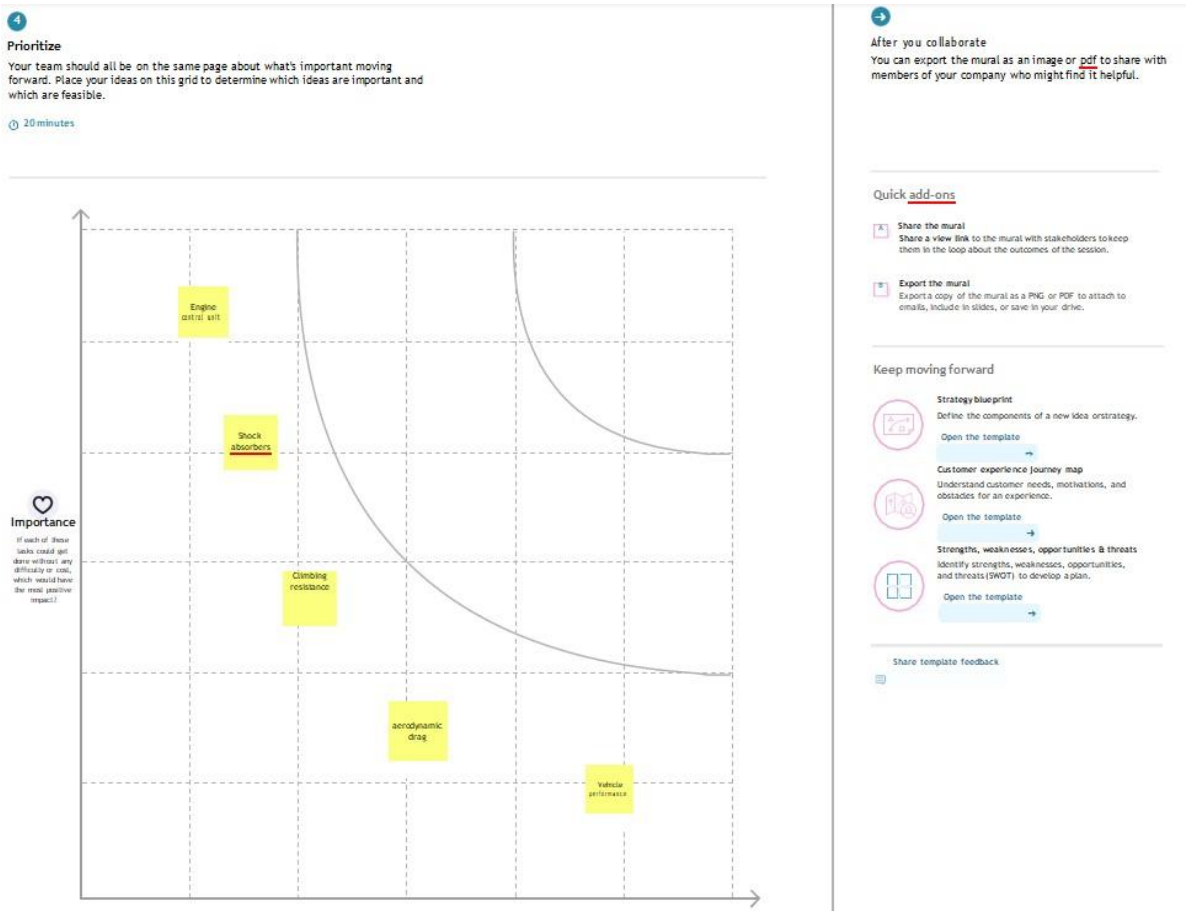
### 1 Define your problem statement

vehicle performance analysis, the automation in automobile by analyzing its performance first and help our customers take care of their vehicles.

5 minutes

PROBLEM

Is it possible to analyse the performance of vehicles effectively at any situation?




## PROPOSED SOLUTION

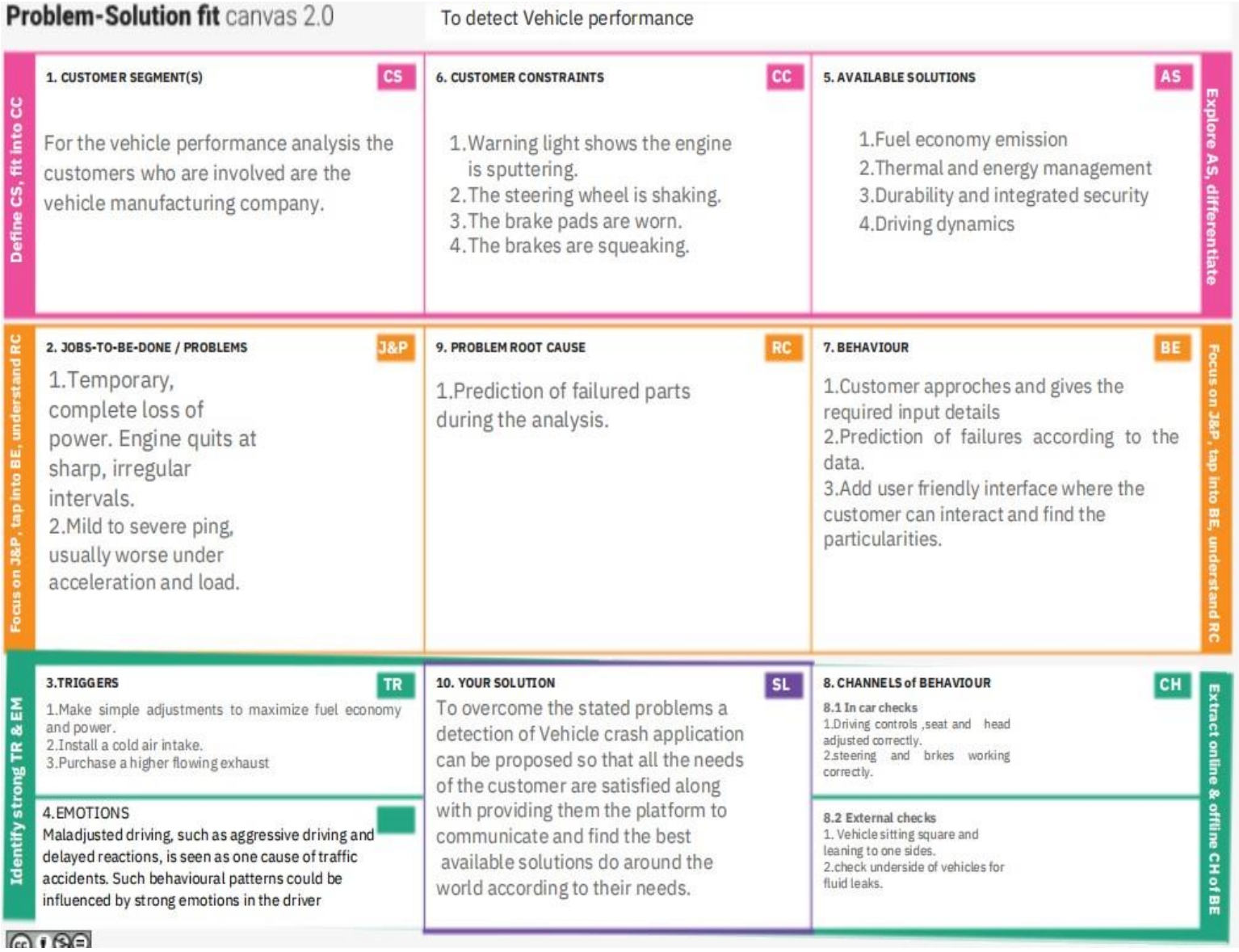
Project team shall fill the following information in proposed solution template.

| S.No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | vehicle performance analysis, the automation in automobile by analyzing its performance first and help our customers take care of their vehicles. |

| | | |
|---|---|---|
| 2. | Idea / Solution description | In the past years, several machine learning algorithms have been proposed to predict vehicle's performance. Most studies predict vehicle's performance using (i) binary classifiers (delayed/not delayedflight), (ii) multi-class classifiers (multiple delay classes), or (iii) regression (estimating the delay value). |
| 3. | Novelty / Uniqueness | In this project we use anaconda navigator , Scikit-learn ,NumPy, Pandas, flask ,Matplotlib. These makes project as more unique than other ways in vehicle performance prediction |
| 4. | Social Impact / Customer Satisfaction | This has a major impact on the drivers' experience and social welfare. Except from the direct impact on passengers, there are also impacts on roadways, in terms of fines and operational costs as well as the environment, in terms of increased fuelconsumption or emissions of an inefficient system. Accordingly, Improving the understanding and prediction of performance is in thebest interest of many stakeholders in air transportation, including navigation service providers and network managers, as well as passengers. |
| 5. | Business Model (Revenue Model) | The application of machine learning to business processes has led to higher levels of acceleration, growth, and adaptability than ever before. Revenue model should look to |

| | | |
|---|---|---|
| | | incorporate machine learning and to secure better market position and competitive differentiation. |
| 6. | Scalability of the Solution | Machine learning scalability is scaling ML models to handle massive data sets and perform many computations in a cost-effective and time-saving way of vehicle's performance prediction |

## PROBLEM SOLUTION FIT




# CHAPTER 4

# REQUIREMENT ANALYSIS

### FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration through Form Registration<br>through Gmail |
| FR-2 | User Confirmation | Confirmation via Email<br>Confirmation via OTP |
| FR-3 | Display of services | It display various services that can be opted for particular vehicle. |
| FR-4 | Safety Measures | It tracks the vehicle location to avoid theft. |
| FR-5 | Performance Analysis | It analysis the performance of the vehicle before and after service provided. |
| FR-6 | Bill payment mode | It provides various payment options. |

## Non-functional Requirements:

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

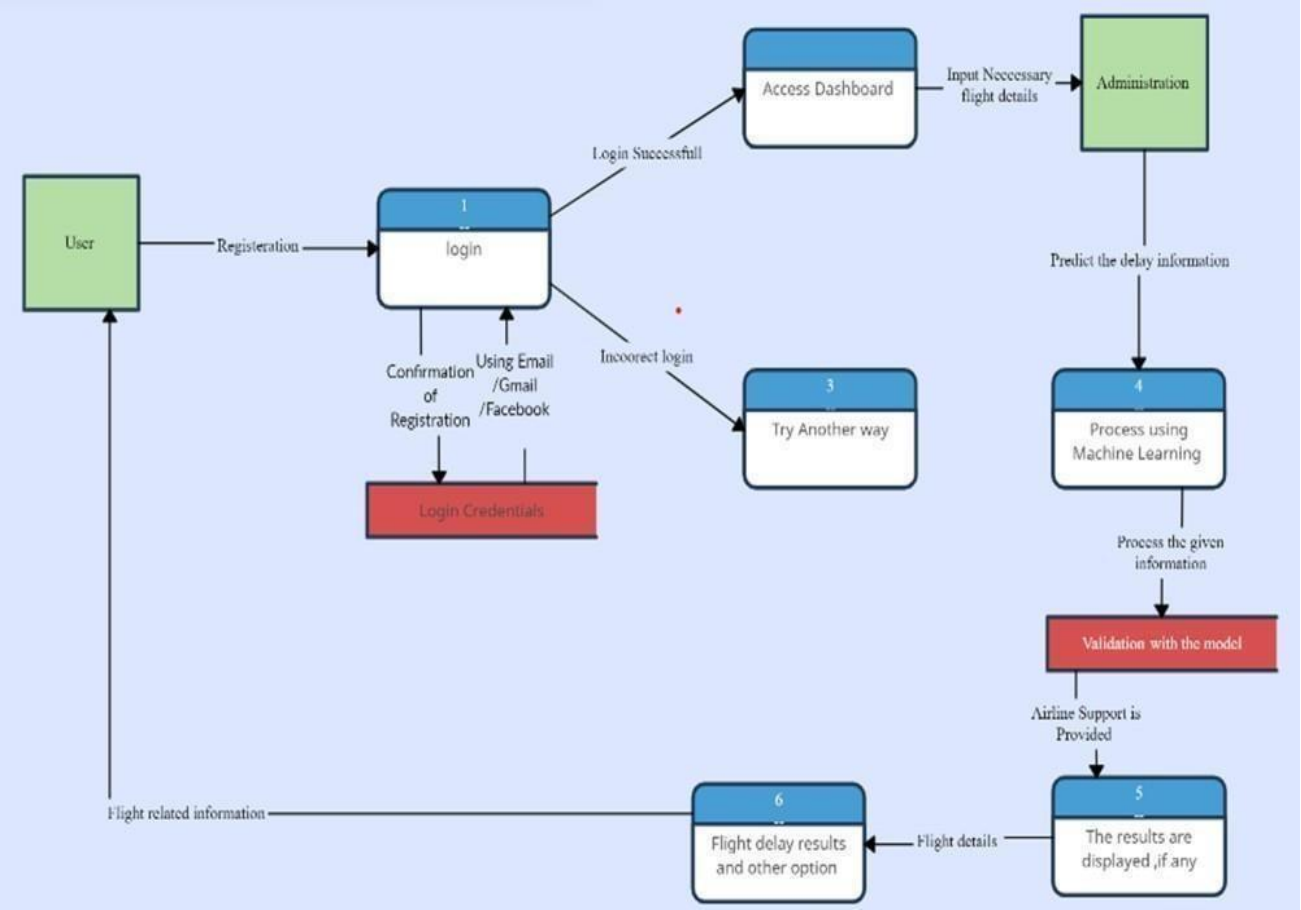
| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | The UX/UI will be user-friendly and will be highly responsive . |
| NFR-2 | **Security** | It provides a facility to track vehicles location that satisfies security. |
| NFR-3 | **Reliability** | The database of vehicles can be maintained for long time ensuring reliability. |
| NFR-4 | **Performance** | The application is programmed to efficiently work under circumstances like network congestion etc |
| NFR-5 | **Availability** | The application available for everyone at ease. The UX/UI experience should be understandable. |
| NFR-6 | **Scalability** | The application is designed in such a way that it can handle ,if any high traffic occurs. |

# CHAPTER 5 PROJECT DESIGN

### 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 stored.

**Figure 5.1. Data flow diagram**




## SOLUTION & TECHNICAL ARCHITECTURE

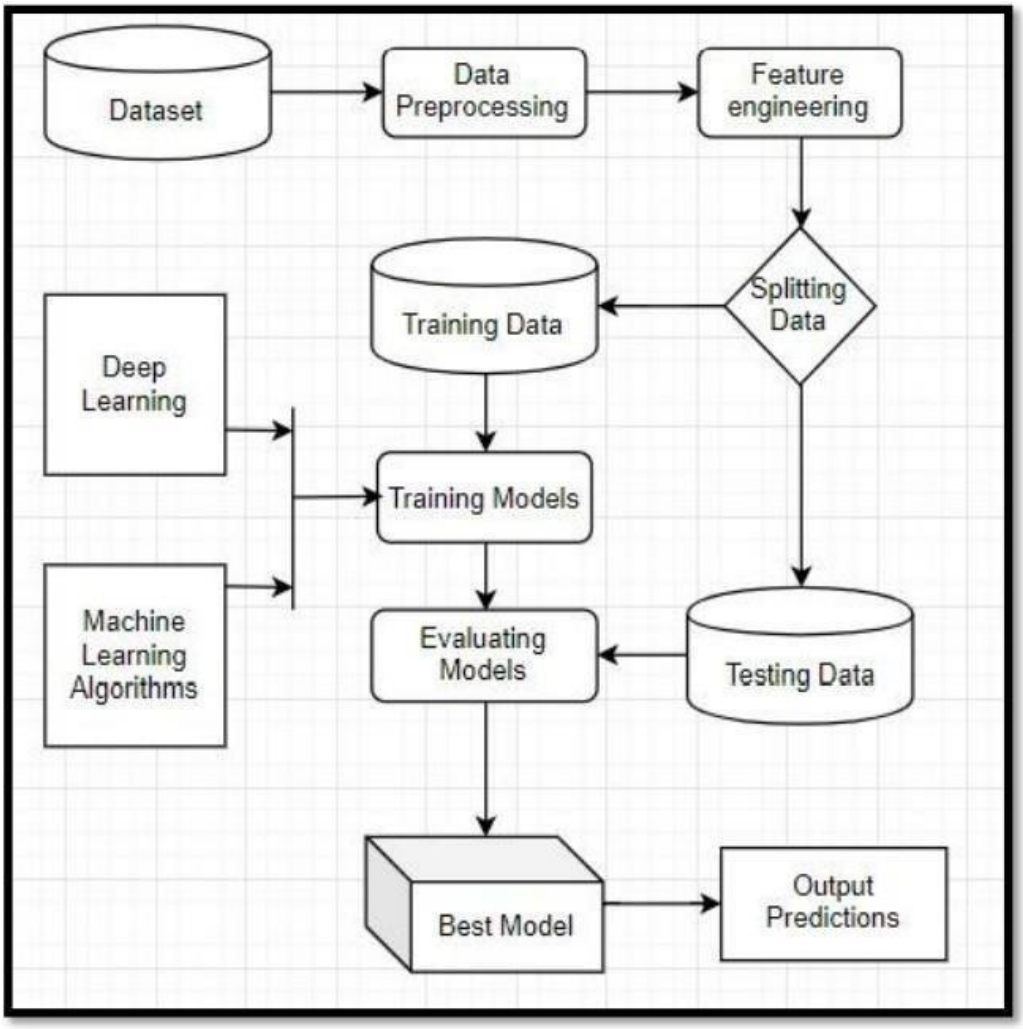



**Figure 5.2. Solution Architecture**

| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | How user interacts with application e.g. Web UI, Mobile App, Chatbot etc. | Python-Flask |
| 2. | Application Logic-1 | Logic for a process in the application | Python |
| 3. | Application Logic-2 | Logic for a process in the application | IBM Watson STT service |
| 4. | Application Logic-3 | Logic for a process in the application | IBM Watson Assistant |
| 5. | Database | Data Type, Configurations etc. | MySQL |

| | | | |
|---|---|---|---|
| 6. | Cloud Database | Database Service on Cloud | IBM DB2, |

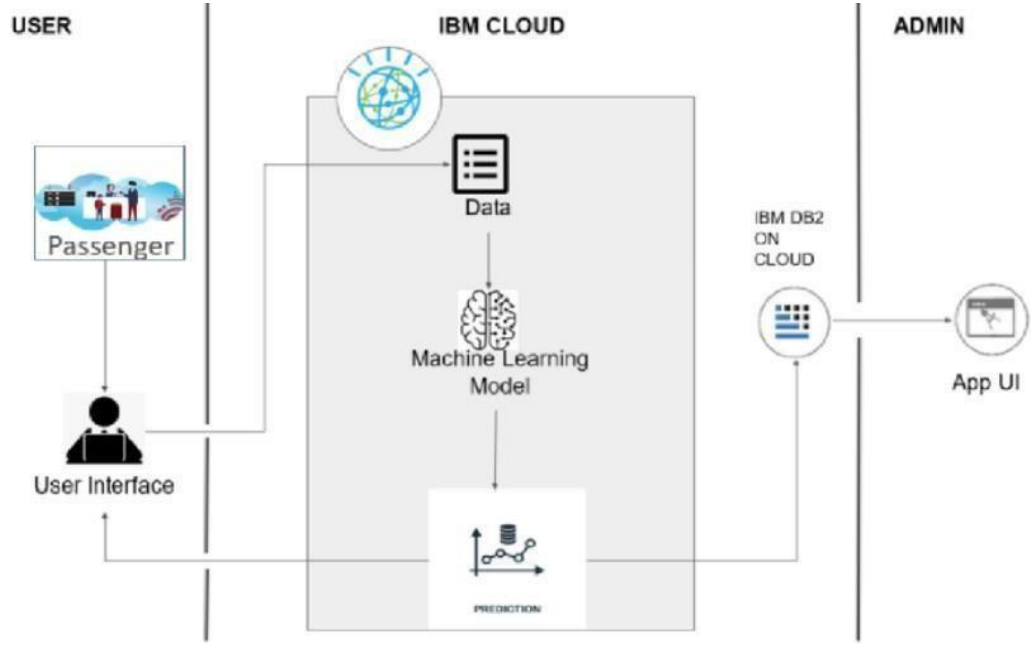



**Figure 5.3. Technology Stack**

**Components & Technologies**

Table-1 : Components & Technologies:

| | | | |
|---|---|---|---|
| 7. | File Storage | File storage requirements | IBM Block Storage |
| 8. | External API-1 | Purpose of External API used in the application | IBM Weather API |
| 9. | External API-2 | Purpose of External API used in the application | Flight Confirmation API |
| 10. | Machine Learning Model | Purpose of Machine Learning Model | Evaluation and Prediction Model |

| | | | |
|---|---|---|---|
| 11. | Infrastructure (Server / Cloud) | Application Deployment | IBM Cloud |

Table-2: Application Characteristics:

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | List the open-source frameworks used | Python-Flask |
| 2. | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | Encryptions, IAM Con |
| 3. | Scalable Architecture | Justify the scalability of architecture (3 – tier, Python Micro-services) | |
| 4. | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | IBM Cloud |
| 5. | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc. | Python |

# CHAPTER 6

# PROJECT PLANNING & SCHEDULING

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

| 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. | 2 | high | 3 |
| Sprint -1 | | USN-2 | As a user, I will receive a confirmation email once I have registered for the application | 1 | high | 1 |
| Sprint -2 | | USN-3 | As a user, I can register for the application through Facebook | 2 | low | 1 |
| Sprint -1 | | USN-4 | As a user, I can register for the application through google account | 2 | high | 1 |
| Sprint -1 | LOGIN | USN-5 | As a user, I can log into the application by entering email & password | 1 | high | 1 |
| | | | | | | |
| Sprint-1 | DASHBOARD | USN-6 | As a customer I can check with all the Flight details available on the website and choose the correct prediction | 3 | high | 3 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Sprint-4 | Customer support | USN-7 | As a user I want to contact with the customer support when there is any query with the the application | 2 | low | 2 |
| Sprint-1 | User details display | USN-8 | As a customer I should be able to see all my given details filled with the registration process | 2 | high | 2 |
| Sprint-2 | algorithm | USN-9 | As a customer, I should be able to get the accurate prediction of the flight details available on the website | 5 | high | 4 |
| Sprint-2 | | USN-10 | As a customer I should be updated with various best available. | 2 | medium | 2 |
| Sprint-3 | IBM watson for storage and organization | USN-11 | As a customer I should be able to give all the necessary details and predict the flight arrival and provide best user experience. | 3 | high | 2 |
| Sprint-4 | Flight management | USN-12 | As a customer I can manage the details of the flight arrival timings and schedule changes. | 5 | high | 4 |

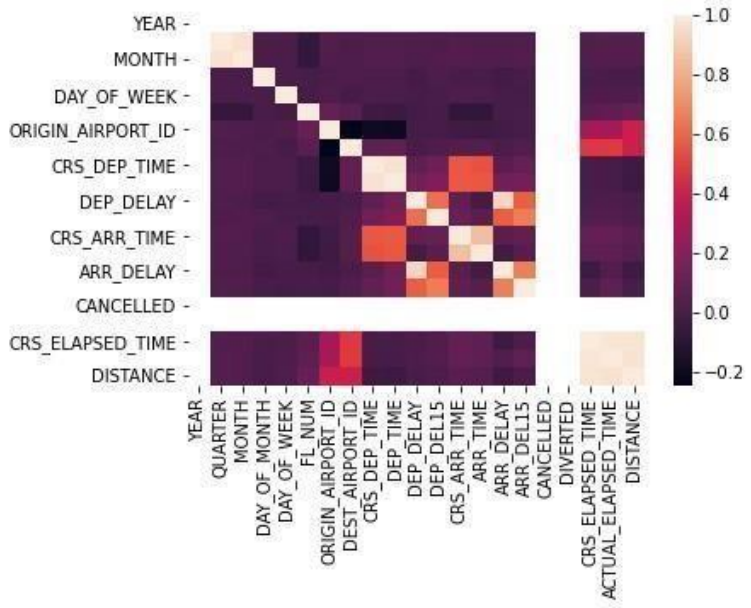
Project Tracker, Velocity & Burndown Chart: (4 Marks)

| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
|---|---|---|---|---|---|---|
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | | |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | | |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | | |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | | |
| | | | | | | |

# CHAPTER 7 CODING AND SOLUTIONING

## FEATURE 1 - CORRELATION BETWEEN THE VARIABLES IN THE DATASET

```
In [19]: sns.heatmap(data.corr())
Out[19]: <AxesSubplot:>
```




This will help us to find out the correlation between the variables in the dataset which would help us to find out the columns that are unnecessary and hence to be dropped.

## FEATURE 2 - ONE HOT ENCODING

```
In [39]: data=pd.get_dummies(data,columns=['ORIGIN','DEST'])

In [40]: data['ARR_DEL15'].value_counts()
Out[40]: 0.0    9668
         1.0    1375
         Name: ARR_DEL15, dtype: int64

In [41]: data.tail()
```

Out[41]:

| | FL_NUM | MONTH | DAY_OF_MONTH | DAY_OF_WEEK | CRS_ARR_TIME | DEP_DEL15 | ARR_DEL15 | ORIGIN_0 | ORIGIN_1 | ORIGIN_2 | ORIGIN_3 | ORIGIN_4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **11226** | 1715 | 12 | 30 | 5 | 12 | 0.0 | 0.0 | 0 | 1 | 0 | 0 | 0 |
| **11227** | 1770 | 12 | 30 | 5 | 20 | 1.0 | 0.0 | 0 | 0 | 0 | 0 | 1 |
| **11228** | 1823 | 12 | 30 | 5 | 22 | 0.0 | 0.0 | 0 | 1 | 0 | 0 | 0 |
| **11229** | 1901 | 12 | 30 | 5 | 18 | 0.0 | 0.0 | 1 | 0 | 0 | 0 | 0 |
| **11230** | 2005 | 12 | 30 | 5 | 9 | 0.0 | 0.0 | 1 | 0 | 0 | 0 | 0 |

The cities in both Origin and Destination are one-hot encoded using the above code.

## FEATURE 3 - SAVING THE MODEL WEIGHTS FOR DEPLOYMENT

SAVING THE MODEL

```
In [63]: pickle.dump(classifier,open('flight_new.pk1','wb'))

In [64]: from sklearn.metrics import confusion_matrix
         confusion_matrix(predicted, y_test)

Out[64]: array([[1825,  129],
                [ 138,  117]], dtype=int64)

In [66]: from sklearn.metrics import classification_report
         print(classification_report(predicted, y_test, labels=[1, 2]))
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1 | 0.48 | 0.46 | 0.47 | 255 |
| 2 | 0.00 | 0.00 | 0.00 | 0 |
| micro avg | 0.48 | 0.46 | 0.47 | 255 |
| macro avg | 0.24 | 0.23 | 0.23 | 255 |
| weighted avg | 0.48 | 0.46 | 0.47 | 255 |

The above code will save the model weights for further deployment in IBM Cloud and also measure the performance metrics.

**FEATURE 4 - FLASK INTERFACE - UI**

```
from flask import Flask, request, render_template
import numpy as np import pandas as pd import
pickle import os model =
pickle.load(open('flight_new.pk1','rb')) app =
Flask(_name_)
@app.route('/') def
home():
    return render_template("mainpage.html")

@app.route('/prediction',methods=['GET','POST'])
def predict():
    name = request.form['fname'] month =
    request.form['month']  dayofmonth  =
    request.form['daymonth'] dayofweek =
    request.form['dayweek']  origin  =
    request.form['origin']
    if(origin == "msp"): origin1, origin2, origin3, origin4,
        origin5 = 0,0,0,0,1
    if(origin == "dtw"): origin1, origin2, origin3, origin4,
        origin5 = 1,0,0,0,0
    if(origin == "jfk"): origin1, origin2, origin3, origin4,
        origin5 = 0,0,1,0,0,
    if(origin == "sea"): origin1, origin2, origin3, origin4,
        origin5 = 0,1,0,0,0
    if(origin == "atl"):
```

```
    origin1, origin2, origin3, origin4, origin5 = 0,0,0,1,0
  destination = request.form['destination'] if(destination == "msp"):
  destination1,destination2,destination3,destination4,destination5 = 0,0,0,0,1
  if(destination == "dtw"):
    destination1,destination2,destination3,destination4,destination5 = 1,0,0,0,0
  if(destination == "jfk"):
    destination1,destination2,destination3,destination4,destination5 = 0,0,1,0,0
  if(destination == "sea"):
    destination1,destination2,destination3,destination4,destination5 = 0,1,0,0,0
  if(destination == "atl"):
    destination1,destination2,destination3,destination4,destination5 = 0,0,0,1,0
  dept = request.form['sdeparttime']
  arrtime = request.form['sarrivaltime']
  actdept =
  request.form['adeparttime'] dept15 =
  int(dept)-int(actdept) y_pred =
  model.predict(total) print(y_pred)
  if(y_pred == [0.]):
    ans = "The Flight will be on time"
  else:
    ans = "The Flight will be delayed"
  return render_template("index.html",data = ans) app.run(debug=True)
```

**Explanation:**

The above code will be able to get the details of the flight from the user in the respective text fields created using the HTML, scale the inputs and give the inputs to the model which has been developed already. The predictions are shown in another HTML page.

**FEATURE 5 - HTML PAGES FOR FRONTEND DESIGN**

```
<html>
.image{ background-image:
  url("vec.jpg"); background-
  repeat: no-repeat; background-
  size: cover; height:100% ;
  width:100% ; color: azure;
  padding-bottom: 4%;

}
.text{ font-family: Georgia, 'Times New Roman', Times,
  serif; font-weight: 800; color: azure;
}

.form{ margin: 0 auto; border: 1px solid
  black; box-shadow: white 2px 2px
  7px 4px; width: 50%;
```

```
}

.input{ text-align:
    center;
    outline:corner;
}
.input:hover{ box-shadow: black 4px
    2px 1px 1px;
}
.select{ width:
    40%;
}
</html>
```

# CHAPTER 8
# RESULTS

## PERFORMANCE METRICS

### Training Accuracy

**MODEL EVALUATION**

```
acc=accuracy_score(predicted,y_test)
```

```
acc
```

```
0.8791308284291535
```

### Confusion Matrix

```
from sklearn.metrics import confusion_matrix
confusion_matrix(predicted, y_test)
```

```
array([[1825,  129],
       [ 138,  117]], dtype=int64)
```

### Classification Model

```
from sklearn.metrics import classification_report
print(classification_report(predicted, y_test, labels=[1, 2, 3]))
```

```
              precision    recall  f1-score   support

           1       0.48      0.46      0.47       255
           2       0.00      0.00      0.00         0
           3       0.00      0.00      0.00         0

   micro avg       0.48      0.46      0.47       255
   macro avg       0.16      0.15      0.16       255
weighted avg       0.48      0.46      0.47       255
```

# CHAPTER 8 ADVANTAGES AND DISADVANTAGES

**Advantages**

- Customers are happy
- The performance of the vehicle is enhanced
- Prior information will be sent if there is a defect in the performance
- The current performance status of the vehicle can be tracked

**Disadvantages**

- Wrong prediction due to wrong input data
- If the prediction is wrong, then there will be issues in the vehicle's performance

# CHAPTER 9 CONCLUSION

In this project, we use fuel usage, aerodynamics, and sensor systems as factors to predict the vehicle performance. In the end, our model correctly predicts the defects and correctly shows the analysis. As a result, there can be additional features related to the causes of defects in the performance of the vehicle that are not yet discovered using our existing data sources.

# CHAPTER 10 APPENDIX

## Source codes

### Exploratory Data Analysis

```
#!/usr/bin/env python
# coding: utf-8

# **Importing all the libraries** #

In[1]:

import numpy as np import pandas as pd import
matplotlib.pyplot as plt import seaborn as sns
import pickle
get_ipython().run_line_magic('matplotlib', 'inline')
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import
train_test_split from sklearn.preprocessing import
StandardScaler from sklearn.tree import
DecisionTreeClassifier from sklearn.metrics import
accuracy_score import sklearn.metrics as metrics

# **Importing the dataset**

# In[2]:
import numpy as np
from flask import Flask,request,jsonify,render_template
import request import json
API_KEY="HEiDudCYxnCSUXVZnh0YbyeSb7v9r9qqECxOQ9ASyXKI"
token_response=request.post("https://us-
south.ml.cloud.ibm.com/identity/token",data={"apikey":API_KEY,"grant_type":"client_credentials
"} mltoken=token_response.json()["access_token"] print("mltoken",mltoken)
header={'content-type':'application/json','Authorization':'Bearer'+ mltoken}
import pickle
app=Flask(_name_)
model=pickle.load(open('regression.pkl','rb'))

@app.predict('/') def
intro():
    return render_template('index.html')
@app.route('/y_predict', methods =["POST"]) def
y_predict():
```

```
    cyl=request.form["input1"]
    dis=request.form["input2"]
    hp=request.form["input3"]
    w=request.form["input4"]
    a=request.form["input5"]
    my=request.form["input6"]
    ori=request.form["input7"]
    total=[[int(cyl),int(dis),int(hp),int(w),int(a),int(my),int(ori)]]


    p=model.predict(total)
    p=p[0]
    return render_template('index.html',prediction_text=p,label="The performance of the car is
".format(p)) if(_name=='main_')
:
    app.run(debug = True,port=9000)
```

# CHAPTER 11 APPENDIX

## SOURCE CODE

```
import numpy as np
from flask import Flask,request,jsonify,render_template
import request import json
API_KEY="HEiDudCYxnCSUXVZnh0YbyeSb7v9r9qqECxOQ9ASyXKl"
token_response=request.post("https://us-
south.ml.cloud.ibm.com/identity/token",data={"apikey":API_KEY,"grant_type":"client_credentials"}
mltoken=token_response.json()["access_token"]
print("mltoken",mltoken)
header={'content-type':'application/json','Authorization':'Bearer'+          mltoken}
import pickle
app=Flask(_name_)
model=pickle.load(open('regression.pkl','rb'))

@app.predict('/') def
intro():
return render_template('index.html')

@app.route('/y_predict', methods =["POST"]) def
y_predict():
cyl=request.form["input1"]
dis=request.form["input2"]
hp=request.form["input3"]
w=request.form["input4"] a=request.form["input5"]
my=request.form["input6"]
ori=request.form["input7"]
total=[[int(cyl),int(dis),int(hp),int(w),int(a),int(my),int(ori)]]
p=model.predict(total)    p=p[0]    return    render_template('index.html',prediction_text=p,label="The
performance of the car is ".format(p)) if(_name=='main_') :
app.run(debug = True,port=9000)
```

**Train the ML Model**

```
# # SPRINT-2

# **TRAIN-TEST-SPLIT**

# In[45]:

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=

0) # In[46]: x_test.shape # In[47]:

x_train.shape

# In[48]:

y_test.shape #

In[49]:

y_train.shape

# **Scaling** # In[50]:

sc = StandardScaler()

# In[51]:

x_train=sc.fit_transform(x_train)
# In[52]:

x_test=sc.fit_transform(x_test)

# **Model Building**

# In[53]:

classifier = DecisionTreeClassifier(random_state=0)

# In[54]:

classifier.fit(x_train,y_train)

# In[55]:
```

```
predicted =
classifier.predict(x_test) # In[56]: predicted #
In[57]: y_test
# **MODEL EVALUATION**
# In[58]:
acc=accuracy_score(predicted,y_test
) # In[59]: acc
# In[ ]:
# In[60]:
data[data['ARR_DEL15']>0].iloc[33].values
# In[61]:
sample=[[1.187e+03, 1.000e+00, 1.500e+01, 5.000e+00, 1.900e+01, 1.000e+00,
0.000e+00, 0.000e+00, 0.000e+00, 1.000e+00, 0.000e+00, 0.000e+00,
0.000e+00, 0.000e+00, 0.000e+00, 1.000e+00]]
# In[62]:
classifier.predict(sample)
# **SAVING THE MODEL**
# In[63]:
pickle.dump(classifier,open('car_new.pk1','wb'))#
In[64]:
from sklearn.metrics import confusion_matrix confusion_matrix(predicted,
y_test)
# In[66]:
from sklearn.metrics import classification_report print(classification_report(predicted,
y_test, labels=[1, 2]))
```

# In[ ]:

**Mainpage – HTML Code**

```
<html>
<div align="center" class="logbg">
<head>
<meta charset="UTF-8">
<center>
<table>
<tr>
<td><h1><br>Prediction of Flight Delay<br><br></h1></td>
</tr>
</table>
</center>
</head>
<body background='C:\Users\Public\project\templates\flight_4.jpg'>
<form action="http://localhost:5000/prediction" method="POST" >
<center>
<table>
<tr>
<td>Enter the flight number:</td>
<td><input type="number" name="fname"><br></td>
</tr>
<tr>
<td>Month:</td>
<td><input type="number" name="month"><br></td>
</tr>
<tr>
<td>Day of Month:</td>
<td><input type="number" name="daymonth"><br></td>
</tr>
<tr>
<td>Day of Week:</td>
<td><input type="number" name="dayweek"><br></td>
</tr>
<tr>
<td>Origin:</td>
<td><select name="origin">
<option value="atl">ATL</option>
<option value="dtw">DTW</option>
<option value="sea">SEA</option>
<option value="msp">MSP</option>
<option value="jfk">JFK</option>
</select></td>
<tr>
<tr>
```

```
<td>Destination:</td>
<td><select name="destination">
<option value="atl">ATL</option>
<option value="dtw">DTW</option>
<option value="sea">SEA</option>
<option value="msp">MSP</option>
<option value="jfk">JFK</option>
</select></td>
<tr>
<tr>
<td>Scheduled Departure Time:</td>
<td><input type="number" name="sdeparttime"><br></td>
</tr>
<tr>
<td>Scheduled Arrival Time:</td>
<td><input type="number" name="sarrivaltime"><br></td>
</tr>
<tr>
<td>Actual Departure Time:</td>
<td><input type="number" name="adeparttime"><br></td>
</tr>
<tr>
<td><br><input type="submit" class="btn" value="SUBMIT"></br>
</tr>
</table>
</center>
</form>
</body>
</div>
</html>
```

**Prediction Page - HTML Code**

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
    <link rel="stylesheet" href="{{url_for('static',filename='css/style.css'}}">
    <title>Project</title>
</head>
<body>
```

```
<section class="image ">
<form action="{{url_for('y_predict')}}"method="post">
   <h1 class="text text-center"><i><b>Vehicle Performance Analyzer</b></i></h1>
   <form class="container form p-4 mt-5">
      <div class="container text-center">
         <div class="row">
            <div class="col">
               <label class="form-label me-4" for="input1">No.of.cylinders</label>
            </div>
            <div class="col">
               <input type="text" class="input" id="input1" placeholder="nos">
            </div>
         </div>
         <div class="row mt-3">
            <div class="col">
               <label class="form-label me-5" for="input2">Displacement</label>
            </div>
            <div class="col">
               <input type="text" class="input" id="input2" placeholder="mts">
            </div>
         </div>

         <div class="row mt-3">
            <div class="col">
               <label class="form-label me-4" for="input3">Horsepower</label>
            </div>
            <div class="col">
               <input type="text" class="input" id="input3" placeholder="HP">
            </div>
         </div>


                        <div class="row mt-3">
            <div class="col">
               <label class="form-label me-4" for="input4">Weight</label>
            </div>
            <div class="col">
               <input type="text" class="input" id="input4" placeholder="Kgs">
            </div>
         </div>

         <div class="row mt-3">
             <div class="col">
               <label class="form-label me-4" for="input5">Accelaration</label>
            </div>
            <div class="col">
```

```
                <input type="text" class="input" id="input5" placeholder="m/s">
              </div>
            </div>

            <div class="row mt-3 ">
              <div class="col">
                <label class="form-label me-5" for="input6">Model year</label>
              </div>
              <div class="col ms-4">
                <input type="text" class="input" id="input6" placeholder="year">
              </div>
            </div>

            <div class="row mt-3">
               <div class="col">
                <label class="form-label me-4" for="input7">Origin</label>
              </div>
              <div class="col">
                <input type="text" class="input" id="input7" placeholder="origin">
              </div>
            </div>
            <button type="submit" class="btn">login</button>
                            {{prediction_text}}

          </div>
        </form>
    </section>


    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
integrity="sha384-
oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js" integrity="sha384-
IDwe1+LCz02ROU9k972gdyvl+AESN10+x7tBKgc9I5HFtuNz0wWnPclzo6p9vxnk"
crossorigin="anonymous"></script>
</body>
</html>
```

### 13.1.4. Flask Application

```
from flask import Flask, request, render_template
import numpy as np import pandas as pd import
pickle import os model =
pickle.load(open('flight_new.pk1','rb')) app =
Flask(_name_)
```

```
@app.route('/') def home(): return
render_template("mainpage.html") import
numpy as np from flask import
Flask,request,jsonify,render_templat
    e import request import json
    API_KEY="HEiDudCYxnCSUXVZnh0YbyeSb7
v9r9qqECxOQ9ASyXKI"
    token_response=request.post("https://us-
south.ml.cloud.ibm.com/identity/token",data={"api
key":API_KEY,"grant_type":"client_credentials"}
mltoken=token_response.json()["access_token"

]
    print("mltoken",mltoken) header={'content-
type':'application/json','Authorization':'Bearer'+

mltoken}
    import pickle
    app=Flask(_name_)
    model=pickle.load(open('regression.pkl','rb'))


    @app.predict('/')
    def intro():
        return render_template('index.html')
```

```
@app.route('/y_predict', methods =["POST"]) def
y_predict():
    cyl=request.form["input1"]
    dis=request.form["input2"]
    hp=request.form["input3"]
    w=request.form["input4"]
    a=request.form["input5"]
    my=request.form["input6"]
    ori=request.form["input7"]

total=[[int(cyl),int(dis),int(hp),int(w),int(a),int(my),in
t(ori)]]

    p=model.predict(total) p=p[0] return
render_template('index.html',prediction_text=p,la
b el="The performance of the car is ".format(p))
if(_name=='main_') :

    app.run(debug = True,port=9000)
app.run(debug=True)
```

**GITHUB LINK**

**Github link**

https://github.com/IBM-EPBL/IBM-Project-53421-1661401778