MACHINE LEARNING BASED VEHICLE PERFORMANCE ANALYZER

Bonafide record of work done by

VIGNESWARAN R.R - 811519104118

WASIM AHMED K - 811519104122

MUTHUKUMARAN S - 811519104077

YADHEENDRAN K - 811519104123

INDUSTRY MENTOR : NIDHI

FACULTY MENTOR : P.SIVAMALAR

BACHELOR OF ENGINEERING

BRANCH: COMPUTER SCIENCE AND ENGINEERING

NOVEMBER 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(AUTONOMOUS INSTITUTION)

SAMAYAPURAM, TRICHY – 621 112

CHAPTER 1
INTRODUCTION

1.1 Project Overview

Predicting a vehicle's performance level is a significant and intriguing challenge. The current study's primary objective is to improve specific vehicle behavior by forecasting automobile performance. The system's fuel consumption and efficiency can both be greatly improved by this. Analyses of the vehicle's performance in relation to, among other things, the kind of engine, number of cylinders, fuel type, and horsepower. These variables can be used to predict the automobile's health. Getting, examining, deciphering, and recording wellbeing information in light of the three components is a consistent cycle. Both prediction engines and engine management systems are heavily dependent on combinable performance metrics like mileage, dependability, adaptability, and cost. Analyzing the components using a variety of well-known machine learning techniques, such as linear regression, decision trees, and random forests, is essential to improving the vehicle's performance efficiency. The power, lifespan, and range of automotive traction batteries are currently the "hot subjects" in automotive engineering. Here, mileage performance is also taken into account. Using a variety of methods and neural networks, we will create models to solve this problem. After that, we'll compare which algorithm correctly predicts car mileage.

1.2 Purpose

The classification of driver usage patterns and the application of both supervised and unsupervised Machine Learning techniques to data from automotive engine sensor systems via a distributed online sensing platform. Various fields, including fleet management, the insurance market, fuel consumption optimization, and CO_2 emission reduction, can benefit from these platforms. As a result, the project's primary objective is to use a variety of machine learning algorithms to predict how well the car will perform in order to enhance specific vehicle behaviors.

CHAPTER 2
LITERATURE SURVEY

2.1 Existing Problem

The development of new technologies has led to an increase in the processing potential of car sensing data in recent years. For instance, analyzing how drivers act while seated at the wheel necessitates having this kind of data. Utilization patterns based on engine sensor data have received very little attention, so their full potential has not been realized by taking into account all engine sensors. The application of both supervised and unsupervised machine learning techniques to automotive engine sensor data in an effort to close this gap and perform classification via a distributed online sensing platform aims to do so. Various fields, including fleet management, the insurance market, fuel consumption optimization, and CO2 emission reduction, can benefit from these platforms.

2.2 Problem Definition

Consequently, by examining the existing issue and learning from the various papers in the literature review. The following is a way to frame the problem's definition:

"Using a variety of machine learning algorithms, to predict the performance of the car and improve certain behaviors of the vehicle".

2.3 References

2.3.1 ML Based Real-Time Vehicle Data Analysis for Safe Driving Modeling:

A machine learning approach is used to analyze and predict vehicle performance in real time in the paper "Machine Learning Based Real-Time Vehicle Data Analysis for Safe Driving Modeling. "The OBD-II scanner's data analysis and, ultimately, the provision of driver safety solutions are the primary goals. The vehicle's meta features are analyzed in the cloud and shared with the relevant parties. An OBD-II scanner and a miniature dash cam make up the proposed system, which continuously transmits data to a cloud server for analysis.

2.3.2 Multivariate LinearRegression Model:

When we want to predict a variable's value based on the values of two or more variables, we use this method. The variable we need to foresee is known as the Reliant Variable, while those used to work out the reliant variable are named as Free Factors.

The model is trained using parameters like fuel economy, average speed, maximum speed, fourth section speed, interval driving distance, driving time in the green zone, traveling time, emergency accelerated value, emergency decelerated value, fourth rpm time value, and fifth rpm time value.

They hypothesize an Economic Driving Index (ECN_DRVG_INDX) and a Safe Driving Index (SFTY_DRVG_INDX) outcome from the obtained real-time data using the Min-Max normalization technique. The results have demonstrated that they approximate 80 percent fit the given characteristics.

2.3.2 Machine Learning Approach Based on Automotive Engine Data Clustering for Driver Usage Profiling Classification:

The paper "A Machine Learning Approach Based on Automotive Engine Data Clustering for Driver Usage Profiling Classification" proposes using supervised and unsupervised machine learning techniques on automotive engine sensor data to identify drivers' usage patterns and classify them using a distributed online sensing platform. This platform can be useful in a variety of fields, including fleet management, the insurance market, fuel consumption optimization, CO_2 emission reduction, and others.

We employ the following Machine Learning models for clustering and class labels because automotive engine data lacks a class label:

K means:

An unsupervised learning algorithm called K-Means Clustering divides the unlabeled dataset into various clusters. If K=2, then there will be two clusters, if K=3, then there will be three clusters, and so on. K is the number of predefined clusters that must be created during the process. Since each cluster is linked to a centroid, it is an algorithm based on centroids. The primary point of this calculation is to limit the amount of distances between the data of interest and their comparing bunches.

Expectation-Maximization:

An approach for performing maximum likelihood estimation in the presence of latent variables is the expectation-maximization algorithm. It accomplishes this by first estimating the latent variable values and then optimizing the model before repeating these two steps until convergence. It is primarily utilized for density estimation with missing data, such as clustering algorithms like the Gaussian Mixture Model, as it is an efficient and universal approach.

Hierarchical Clustering:

Another unsupervised machine learning algorithm for grouping unlabeled datasets into a cluster is hierarchical clustering. The dendrogram is the tree-shaped structure that we construct in this algorithm for the hierarchy of clusters.

Machine learning algorithms for Classification:

Decision Tree:

Other learning algorithms that divide the input space into regions and have distinct parameters for each region include the decision tree and its variants. They are categorized as a non-parametric supervised learning technique that is frequently utilized in decision-making, decision representation, classification, and regression. A decision tree's structure is similar to a flowchart: each internal node represents a "test" on an attribute, each branch the test's result, and each leaf node a class label. In addition, the paths that lead from the root to the leaf represent classification rules.

KNN:

The K-NN algorithm stores all of the data that is available and uses similarity to classify a new data point. This indicates that the K- NN algorithm can easily classify new data into a well-suited category when it appears.

Multilayer Perceptron:

A fully connected type of feed forward artificial neural network known as a multilayer perceptron. It uses proximity to classify or predict a data point's grouping by using proximity. It can be used for either classification or regression problems, but most of the time it is used as a classification algorithm because it assumes that similar points can be found close to each other.

Naive Bayes

A collection of supervised learning algorithms known as "naive Bayes methods" are based on applying Bayes' theorem with the "naive" assumption that every pair of features has conditional independence based on the value of the class variable. It is not a single algorithm; rather, it is a family of algorithms whose fundamental principle is that each pair of features being classified is distinct from each other.

Random Forest

Random forests, also known as random decision forests, are a type of ensemble learning for classification, regression, and other tasks. It works by building a lot of decision trees during training. The class that was chosen by the majority of the trees is what the random forest produces for classification tasks. The individual trees' mean or average prediction is returned for regression tasks. In general, random decision forests outperform decision trees, but their accuracy is lower than that of gradient-boosted trees. Random decision forests correct for decision trees' tendency to overfit their training set. Notwithstanding, information qualities can influence their presentation.

Support Vector Mechanism:

Support Vector Machine, also known as SVM, is one of the most widely used supervised learning algorithms. It can be used to solve regression and classification problems. The SVM algorithm's objective is to find the most effective line or decision boundary for classifying n-dimensional space, allowing us to quickly place a new data point in the appropriate category in the future. A hyperplane is the name given to this best decision boundary. The extreme points and vectors that aid in the creation of the hyperplane are selected by SVM. The algorithm is referred to as a Support Vector Machine because these extreme cases are known as support vectors.

2.3.3 Driving Behavior Analysis Using Machine and Deep LearningMethods for Continuous Streams of Vehicular Data:

Nikolaos Peppes, Theodoros Alexakis, Evgenia Adamopoulou, and Konstantinos Demestichas wrote a paper titled "Driving Behavior Analysis Using Machine and Deep Learning Methods for Continuous Streams of Vehicular Data" that aims to combine well-known machine and deep learning algorithms with open-source tools to collect, store, process, analyze, and correlate various vehicle data flows.

Machine Learning Algorithms for Classification:

Support Vector Mechanisms (SVM):

A supervised machine learning algorithm used for both classification and regression is support vector machines. In an N-dimensional space, SVM uses the hyperplane to classify data points. In support vector classification, the separation function is a linear combination of support vector-linked kernels.

Decision Tree-Based Algorithms:

Other learning algorithms that divide the input space into regions and have distinct parameters for each region include the decision tree and its variants. They are categorized as a non parametric supervised learning method that is frequently utilized in decision-making, decision representation, classification, and regression. A decision tree has a structure that is similar to a flowchart: each internal node represents a "test" on an attribute, each branch represents the test's outcome, and each leaf node represents a class label. In addition, the paths that lead from the root to the leaf represent classification rules. The decision tree (DT), extra trees (ExT), and random forest were the three decision tree-based models evaluated in relation to various learning strategies.

Random Forest

Random forests, also known as random decision forests, are a type of ensemble learning for classification, regression, and other tasks. It works by building a lot of decision trees during training. The class that was chosen by the majority of the trees is what the random forest produces for classification tasks. The individual trees' mean or average prediction is returned for regression tasks. Random decision forests eliminate the tendency of decision trees to overfit their training set. Though their accuracy is lower than that of gradient-boosted trees, random forests generally perform better than decision trees. Notwithstanding, information qualities can influence their presentation.

Deep Learning Model:

RNN-based algorithms:

Due to their robustness and ability to handle nonlinear data despite typically structured, single or advanced structured, multiple hidden layers, RNN-based models are now widely used. There are three layers in RNN: layers for input, hidden, and output. The number of layers and, as a result, the number of computational resources will increase as the complexity of the problem increases. For the purpose of predicting the Driving Behavioral Analysis in this case, both of the aforementioned structures of the RNN-based models were used.

Multilayer Perceptron:

A fully connected type of feed forward artificial neural network known as a multilayer perceptron. It uses proximity to classify or predict a data point's grouping by using proximity. It can be used for either classification or regression problems, but most of the time it is used as a classification algorithm because it assumes that similar points can be found close to each other.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map

The empathy map's primary goal is to improve communication between developers and users. The empathy map for the vehicle performance analyzer that is based on machine learning is shown in Figure 3.1.

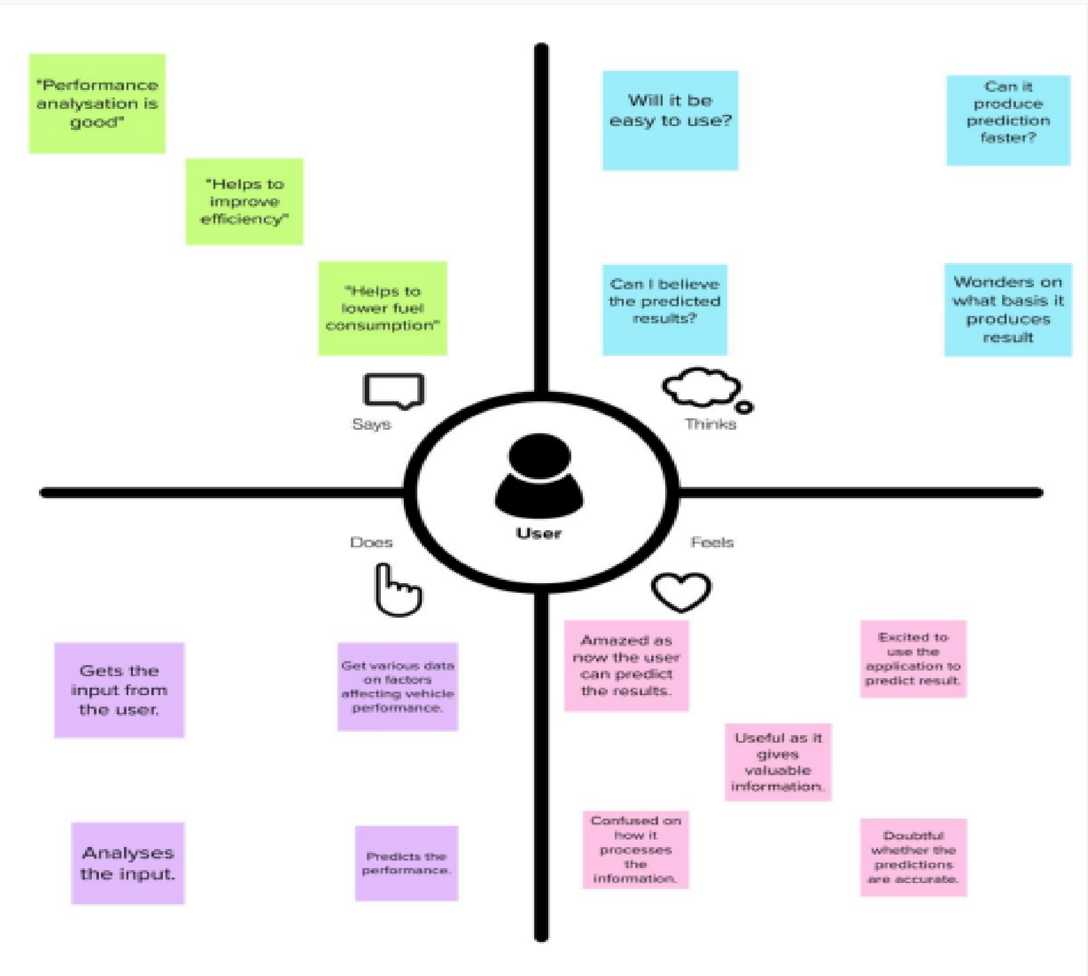
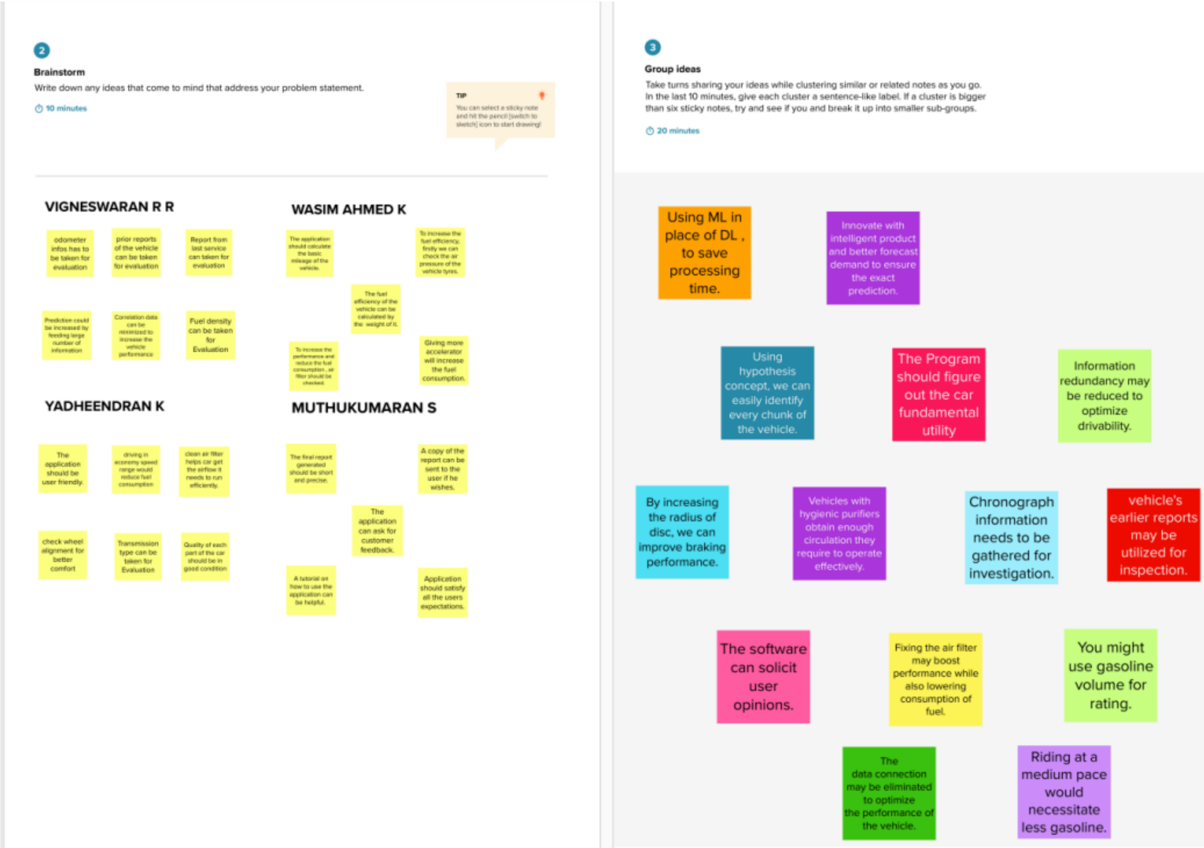


Figure 3.1 - Empathy Map

3.2 Ideation and Brainstorming

This is frequently the most exciting phase of a project because the goal of ideation and brainstorming is to generate a large number of ideas that the team can then filter and narrow down to inspire new and better design solutions and products. Figure 3.2 depicts the stages of ideation and brainstorming for the vehicle performance analyzer based on machine learning.



Step-3: Idea Prioritization

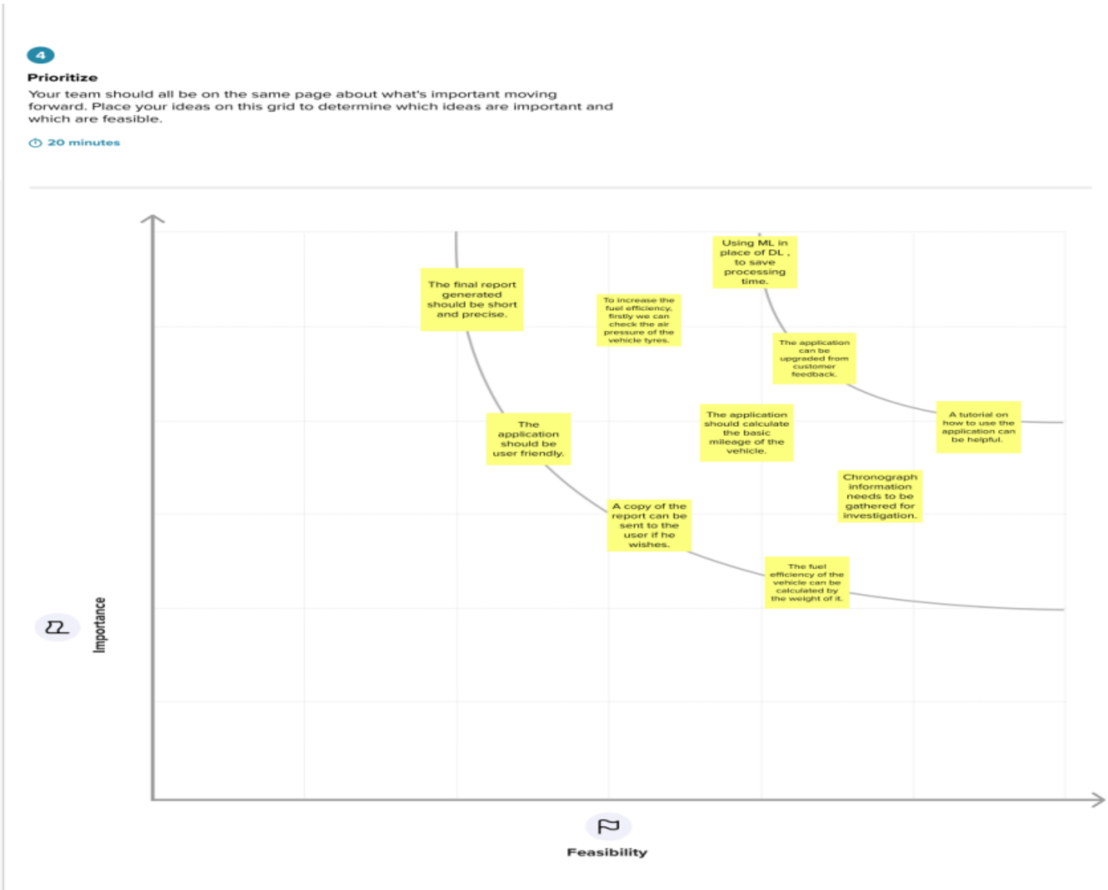


Figure 3.2 – Ideation & Brainstorming

3.3 Proposed Solution

S.No	Parameter	Description
1	**Problem Statement (Problem to be solved)**	Predicting the performance level of cars is an important and interesting problem. The main goal is to predict the car's performance to improve certain vehicle behaviors. This can significantly help to improve the system's fuel consumption and increase efficiency. The performance analysis of the car is based on the engine type, no of engine cylinders, fuel type, horsepower, etc. These are the factors on which the health of the car can be predicted. It is an ongoing process of obtaining, researching, analyzing, and recording health based on the above three factors. The performance objectives like mileage, dependability, flexibility, and cost can be grouped together to play a vital role in the prediction engine and engine management system. This approach is a very important step toward understanding the vehicle's performance.
2	**Idea / Solution description**	There are several suggestions for improving vehicle performance. Analyzing these many elements and attributes offers a broad and refined answer for improving the vehicle's performance. We updated several parts and increased some attributes to give higher performance and mileage strength, efficiency, and comfort.

3	**Novelty / Uniqueness**	Unlike other Vehicle Performance Analyzers on the market, our analyzer concentrates on refining the vehicle's performance, bringing out its full potential, and improving on any areas that may be improved.
4	**Social Impact / Customer Satisfaction**	Analyzing a vehicle's performance may be beneficial in a variety of ways. One of the most significant advantages is that fuel consumption (petrol/diesel, etc.) may be drastically decreased, lowering fuel costs as well as pollutants from the engine (exhaust gases).
5	**Business Model (Revenue Model)**	This proposal intends to increase vehicle performance and, more importantly, minimize emissions. The primary business model seeks a low profit while providing maximum performance.
6	**Scalability of the Solution**	The key advantages of this project are that it can be housed on larger cloud platforms like as IBM Watson, etc., and that it can be accessible from anywhere in the world.

3.4 Problem Solution Fit

The problem solution fit is the solution one has found to address the problem of the customer. Figure 3.4 depicts the solution fit for the machine learning based vehicle performance analyzer.

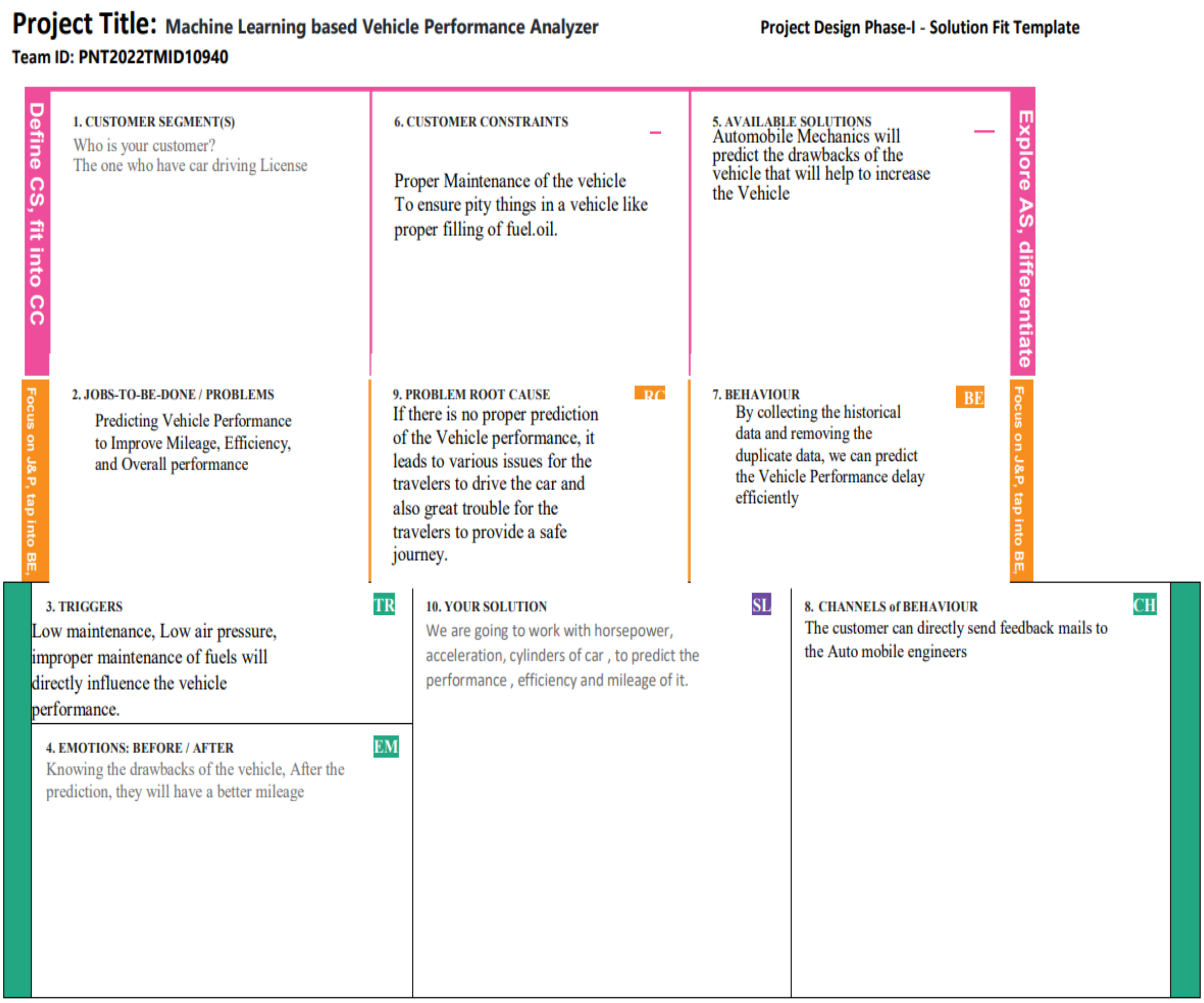


Figure 3.4 – Problem SolutionFit

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional Requirements

Table 4.1 are the functional requirements of the proposed solution.

FR No.	FunctionalRequirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Enter the input	Get input through the form
FR-2	User Essential	Predict the performance of the vehicle
FR-3	Data preprocessing	Sample dataset for training purpose
FR-4	User input Evaluation	Evaluating the given user values
FR-5	Prediction	Fuel consumption and efficiency of the vehicle

Table 4.1 – Functional Requirements

4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Based on the results, the analyzer enables the user to optimize performance. It is simple to use and only requires the necessary info.
NFR-2	**Security**	Due to excellent performance, a low frequency of problems, and affordable repair costs, the dependability rating is good.
NFR-3	**Reliability**	The dependability rating is high owing to the best performance, low frequency of issue recurrence, and cheap repair cost.
NFR-4	**Performance**	The vehicle's quality and infrastructure have been improved to deliver greater performance such as good mileage, easy travel thanks to good suspension, and improved engine performance.
NFR-5	**Availability**	Researchers acquire the necessary data, which may then be utilized to deliver improved outcomes.
NFR-6	**Scalability**	Our project is more scalable since our model analyses all information and gives a more precise answer. We could attain optimum performance with little changes to the car.

Table 4.2 – Non-Functional Requirements

CHAPTER 5
PROJECT DESIGN

5.1 Dataflow Diagram

An Information Stream Graph (DFD) is a customary visual portrayal of how data streams inside a framework. A flawless and clear DFD can subsequently portray the perfect proportion of the framework prerequisites graphically. Not only does it demonstrate how data enters and exits the system, but it also demonstrates what alters the data and where it is stored. The DFD for the given project is depicted in Figure 5.1.

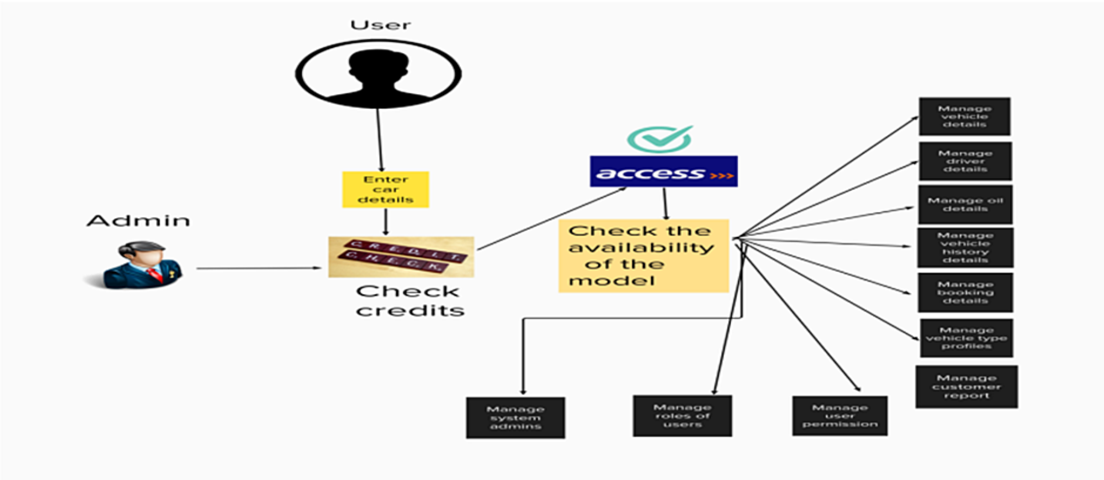


Figure 5.1 – Dataflow Diagram

5.2 Technical Architecture

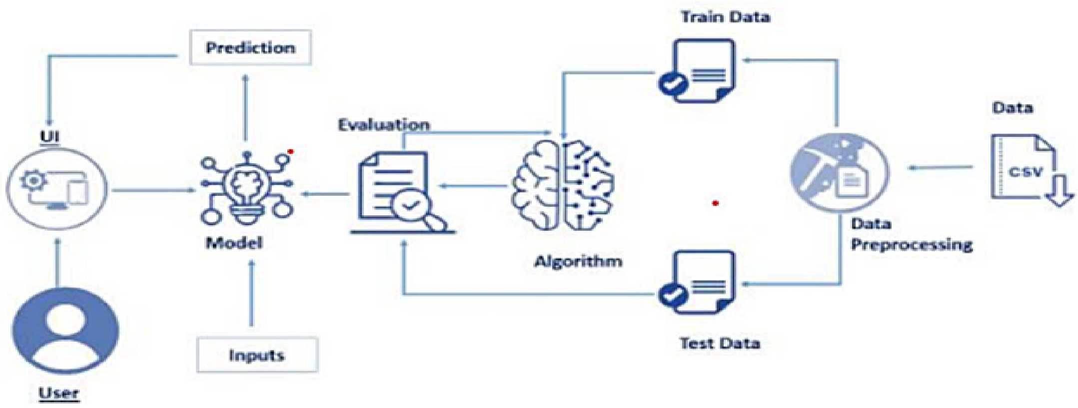


Figure 5.2 TechnicalArchitecture

5.2.1 Component and Technologies

S.No	Component	Description	Technology
1.	User Interface	The user has a better experience and may visit the website easily with the aid of web UI.	HTML, CSS
2.	Application Logic-1	Customer can give their vehicle details.	IBM Watson STT service
3.	Application Logic-2	The performance of their car and the vehicle itself may be checked by the customer following a service.	IBM Watson Assistant
4.	Cloud Database	Cloud database service	IBM DB2, IBM Clouding, etc.
5.	File Storage	Storage needs for files	Cloud Object Storage
6.	Machine Learning Model	To create a model for prediction	Random Forest Regressor
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration:	IBM Cloud Services

Table 5.2.1 – Components and Technologies

5.2.2 Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	List the open-source frameworks used in FLASK	The technology of Open source framework PYTHON
2	Security Implementations	By improving your car’s performance your car will live for long	-
3	Scalable Architecture	Justify the architecture's capacity to scale (3 – tier, Micro- services) Because application servers may be installed on several computers, scalability is improved. The database only needs connections from a small number of application servers, so it doesn't have to establish lengthier connections with every client.	Presentation Layer – FLASK (HTML, CSS) Application Layer – Flask (Python) Data Layer – IBM DB2
4	Availability	Justify the availability of applications (e.g., use of load balancers, distributed servers, etc.)	-
5	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc.	-

Table 5.2.2 – Application Characteristics

5.3 User Stories

UserType	**Functional Requirements**	**User Story Number**	**User Story/Task**	**Acceptance Criteria**	**Priority**	**Release**
Customer	Access the Webpage	USN -1	Anyone can access the webpage to check the specifications of the vehicle	I can access my webpage online at any time	High	Sprint-1
Admin	Performance of the Vehicle	USN - 2	As per the usage of the user, the performance of the vehicle should be predictable.	Prediction can be donein an easy way.	High	Sprint-2
Admin	The connection between a webpage and the machine learning model	USN - 3	Connection should be made between the webpage and the machine learning model for user input prediction.	The webpage can be connected to a machine learning model.	High	Sprint-3
Customer	Accuracy to check the performance and health of the car	USN - 4	By using our prediction, it helps to check the health of the car.	The efficiency ofthe car can be predicted.	High	Sprint-4

Table 5.3 – User Stories

CHAPTER 6
PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirements	User Story Number	User Story	Story Points	Priority	Team Members
Sprint-3	Checking	USN - 3	As a user, I can enter the details of the car which will be checked with the credit details	20	High	K. Wasim Ahmed
Sprint 3	Confirmation	USN - 3	As a user, I will receive confirmation for access to the availability	10	Low	K. Yadheendran
Sprint 4	Managing	USN - 4	As a user, I receive the report of managing driver details, booking details, system admin,and so on	10	Low	K. Wasim Ahmed
Sprint 1	Dataset	USN - 1	Provide a dataset to predict the vehicle performance	20	High	R R. Vigneswaran
Sprint-1	Data Cleaning	USN - 1	Remove the null values and outliers from the data.	20	High	R R. Vigneswaran
Sprint-2	Predicting	USN - 2	By using Machine learning models, Predictions are made by using the dataset provided	20	High	R R. Vigneswaran
Sprint-4	Result	USN - 4	Whether the vehicle has highefficiency or not is determined.	15	Medium	S. Muthukumaran

Table 6.1 – Sprint Planning

6.2 Sprint Delivery Schedule

Sprint	Story Points	Duration (days)	Sprint Start Date	Story Points Completed	Sprint Release Date
Sprint 1	30	6	24 Oct 2022	30	29 Oct 2022
Sprint 2	20	6	31 Oct 2022	20	05 Nov 2022
Sprint 3	20	6	07 Nov 2022	20	12 Nov 2022
Sprint 4	30	6	14 Nov 2022	30	19 Nov 2022

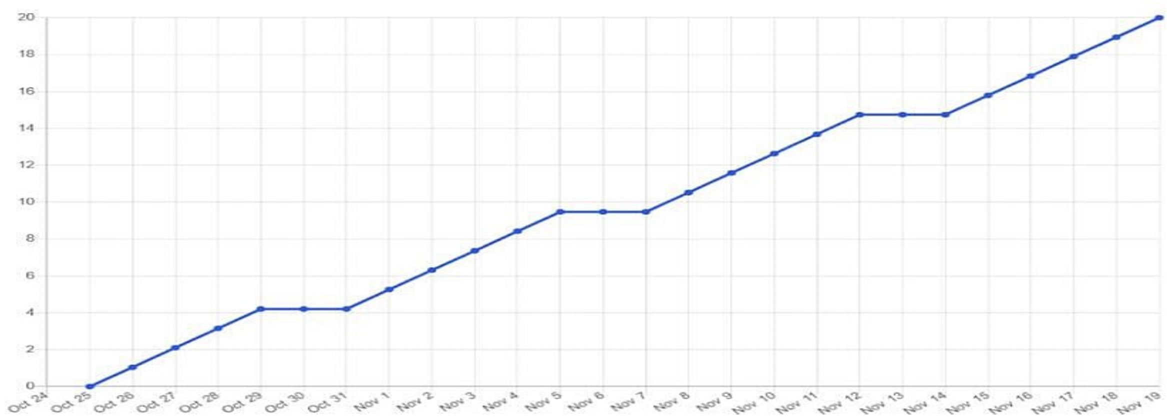
Table 6.2 – Sprint Delivery Schedule

6.3 Reports for JIRA

Velocity: Imagine we have a 10-day sprint duration, and the velocity of team is 20 (points persprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart: A burndown outline is a graphical portrayal of work passed on to do versus time. Agile software development techniques like Scrum frequently employ it. Burn down charts, on the other hand, can be used for any project that has measurable progress over time.



CHAPTER 7
CODING AND SOLUTION

7.1 Feature 1

FR No.	Feature	Description
FR-1	Enter the input	Get input through the form
FR-2	User Essential	Predict the performance of the vehicle
FR-3	Data preprocessing	Sample dataset for training purpose
FR-4	User input Evaluation	Evaluating the given user values
FR-5	Prediction	Fuel consumption and efficiency of the vehicle

Table 7.1 – Description for Feature 1

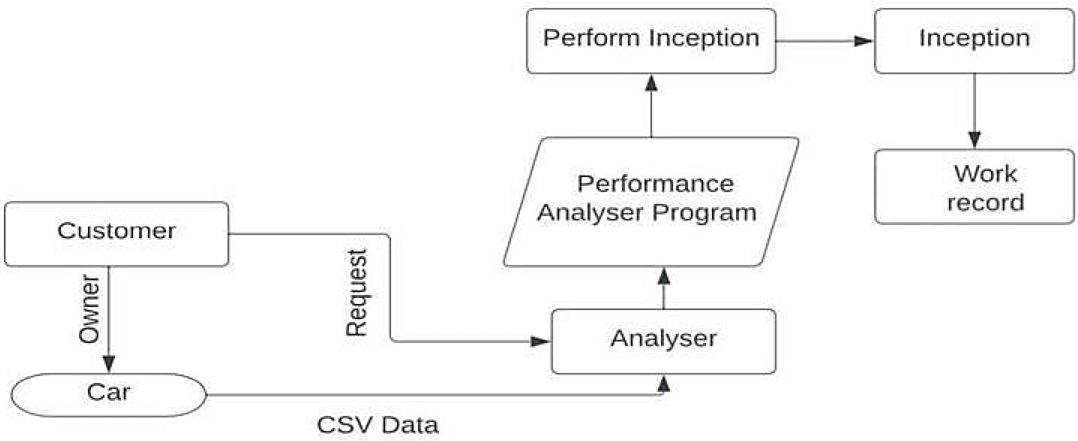


Figure 7.1 – Dataflow Diagram for Feature 1

The order happens is in the "apps.py" record where assuming the result is lesser than 9 the vehicle is said to have the most exceedingly awful presentation. The output is considered to be of low performance if it ranges between 9 and 17.5.It is performing medium if it is between 17.5 and 29.A vehicle is said to have high performance if its output is between 29 and 46, and if it is above 46, it has very high performance.

CHAPTER 8
TESTING

8.1 Test Cases

Date	3-Nov-22
Team ID	PNT2022TMID10940
Project Name	Project - Machine Learning bas
Maximum Marks	4marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N	BUG ID	Executed By
HomePage_TC_001	Functional	Home Page	Verify user is able to enter the data into the text field in the webpage and click the		1.Enter URL 2.Enter the values	[8,307.0,130,3504,12.0,70,1,"audi"]	Page Refresh	Working as expected	Pass				Wasim Ahmed K
HomePage_TC_002	Functional	Home Page	Verify if the user is able to view the output after the submit button has been		1.Click the submit button		Vehicle Performance with mileage 21.1	Working as expected	Pass				Wasim Ahmed K

Figure 8.1 – Test Cases Run

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] 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	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	1	0	0	2
Duplicate	1	0	0	0	1
External	1	0	0	0	1
Fixed	1	1	1	1	4
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	4	2	1	1	13

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	4	0	0	4
Client Application	4	0	0	4
Security	1	0	0	1

Figure 8.2 – UserAcceptance Testing

CHAPTER 9

RESULTS

9.1 Performance Metrics

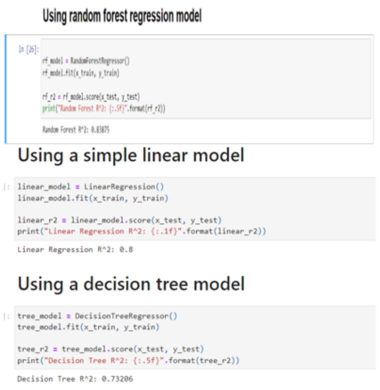
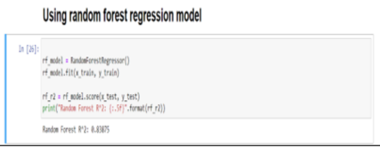
S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** R2 score -	
2.	Accuracy	Training Accuracy - 0.83855	

Figure 9.1 – Performance Metrics

CHAPTER 10
PROS AND CONS

10.1 Pros

i. The model's use of the Random Forest Algorithm facilitates both classification and regression tasks.
ii. A random forest makes accurate and simple-to-understand predictions.
iii. It is simple to handle large datasets.
iv. The Random Forest Algorithm predicts outcomes with greater precision.

10.2 Cons

v. Including a large number of trees in the model can make the random forest algorithm too slow and ineffective for real-time predictions, which is the main drawback.
vi. The arbitrary timberland calculation is very delayed to make forecasts whenever it is prepared.

CHAPTER 11

CONCLUSION

The ability to estimate a car's performance level presents a big and fascinating challenge. Forecasting vehicle performance in order to improve particular vehicle behavior was our main goal. performance evaluation of the car considering its horsepower, cylinder count, fuel type, and engine type, among other things. Based on factors, like horsepower, cylinder count, fuel type, and engine type, the health of the car is forecasted. We analyzed the components using a number of well-known machine learning approaches, like linear regression, decision trees, and random forests, in order to optimize the performance efficiency of the vehicle. The power, longevity, and range of automobile traction batteries are now the "hot topics" in automotive engineering. In this case, we additionally consider mileage performance. To answer this problem, we have built the models using a variety of methods and neural networks. We've then compared which algorithm is most accurate in forecasting car performance (Mileage). A front-end webpage was designed to help give the user an attractive front while they input the values required by the developed machine learning model. The IBM cloud platform was used to develop the model.

CHAPTER 12

FUTURE WORKS

Because the dataset utilized for this model is an old vehicle dataset, the model's accuracy might suffer if the information on automobiles published recently were provided as input. As a result, in the future, we recommend using the most recent dataset set incorporating vehicle information to assist in training the model. We also intend to test additional classification algorithms, such as SVM and Decision Tree, in place of Random Forest to see if there is any improvement in accuracy. Finally, we recommend that the machine learning model be scaled to examine the performance of a broader range of vehicles.

CHAPTER 13
APPENDIX

13.1 Source Code
13.1.1 Sourcing_end_point.py

```
import pip._vendor.requests
from pip._vendor import requests
import json
# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "McA0cABIxbmWF-itHuc3Tat6XJ0FtvRJHwgQjLDcZI5R"
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}

# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"field":
[["cylinders","displacement","horsepower","weight","acceleration","model year","origin","make"]],
"values": [[8,307,130,3504,12,70,1,4]]}]}

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/17716fa2-6c0e-4dd3-b8f3-6dbfd44a61df/predictions?version=2022-11-18', json=payload_scoring,
 headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
predictions=response_scoring.json()
print(predictions['predictions'][0]['values'][0][0])
```

13.1.2 app.py

```
import requests
from  flask import *
import pandas as pd
# NOTE: you must manually set API_KEY below using information retrieved from
your IBM Cloud account.
API_KEY = "McA0cABIxbmWF-itHuc3Tat6XJ0FtvRJHwgQjLDcZI5R"
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__,template_folder="template")

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

@app.route('/y_predict', methods=['POST'])

def y_predict():
   data = pd.read_csv('Temp_file.csv')
  cB = request.form["car name"]
  cB=cB.split(' ')[0]
   for i in range(len(data["Brand"])):
     if cB == data["Brand"].iloc[i]:
      cB=data["Encoded"].iloc[i]
   cy = request.form["cylinder"]
  disp = request.form["disp"]
  hP = request.form["hP"]
   weight = request.form["w"]
   Acc = request.form["Acc"]
  mY = request.form["Model"]
   origin = request.form["orgin"]

   t = [[int(cy),int(disp),int(hP),int(weight),int(Acc),int(mY),int(origin),int(cB)]]

   # NOTE: manually define and pass the array(s) of values to be scored in the next line
  payload_scoring = {"input_data": [{"field": ["cylinders" , "displacement"
,"horsepower","weight" , "acceleration" ,"model year" ,"orgin","make"], "values": t}]}

  response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/17716fa2-6c0e-4dd3-b8f3-
6dbfd44a61df/predictions?version=2022-11-18', json=payload_scoring,
   headers={'Authorization': 'Bearer ' + mltoken})
   print("Scoring response")
   prediction = response_scoring.json()

   print(prediction)
   out = prediction['predictions'][0]['values'][0][0]

   op = "Your car mileage is " + str(round(out,2))
   if out>30:
     op = op + "! It is astoundingly healthy!"
```

```
    elif out>20:
      op = op + "! It seems healthy!"
    elif out>15:
      op =  op + "! Not bad!"
     else:
      op = op + "! It needs proper maintenance"

     return render_template('index.html' , prediction_text=op)

  if( __name__ == "__main__"):

app.run()
```

13.1.3 index.html

```
<!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">
  <title>Car Performance Analyzer</title>
  <style>



input {
  caret-color: red;
}

body {
  margin: 0;
  width: 100vw;
  height: 150vh;
  background-image: linear-gradient(to bottom right, lightblue, gray);
  display: flex;
  align-items: center;
  text-align: center;
  justify-content: center;

  font-family: poppins;
}

.container {
  position: relative;
  width: 700px;
  height: 850px;
  border-radius: 20px;
  padding: 40px;
  box-sizing: border-box;
  background: #ecf0f3;
  box-shadow: 5px 5px 10px #cbced1, -5px -5px 10px white;
}


.brand-title {
  margin-top: 10px;
  font-weight: 900;
  font-size: 1.8rem;
  color: #1DA1F2;
  letter-spacing: 1px;
}

.inputs {
  text-align: left;
  margin-top: 30px;
}
```

```
label, input, button {
  display: block;
  width: 100%;
  padding: 0;
  border: none;
  outline: none;
  box-sizing: border-box;
}

label {
  margin-bottom: 4px;
}

label:nth-of-type(2) {
  margin-top: 12px;
}

input::placeholder {
  color: gray;
}

input {
  background: #ecf0f3;
  padding: 10px;
  padding-left: 20px;
  height: 50px;
  font-size: 14px;
  border-radius: 50px;
  box-shadow: inset 6px 6px6px #cbced1, inset -6px -6px 6px white;
}

button {
  color: white;
  margin-top: 20px;
  background: #1DA1F2;
  height: 40px;
  border-radius: 20px;
  cursor: pointer;
  font-weight: 900;
  box-shadow: 6px 6px6px #cbced1, -6px -6px 6px white;
  transition: 0.5s;
}

button:hover {
  box-shadow: none;
}
#cylinder{
  background: #ecf0f3;
  padding: 10px;
  padding-left: 20px;
  height: 40px;
  font-size: 14px;
  border-radius: 50px;
  box-shadow: inset 6px 6px6px #cbced1, inset -6px -6px 6px white;
}
#origin{
  background: #ecf0f3;
  padding: 10px;
  padding-left: 20px;
  height: 40px;
  font-size: 14px;
  border-radius: 50px;
  box-shadow: inset 6px 6px6px #cbced1, inset -6px -6px 6px white;
}

a {
  position: absolute;
  font-size: 8px;
  bottom: 4px;
  right: 4px;
  text-decoration: none;
  color: black;
  background: yellow;
  border-radius: 10px;
  padding: 2px;
}
```

```
    h1 {
      position: absolute;
      top: 0;
      left: 0;
    }

    #predicted_value {
      color: green;
      padding-top: 15px;
      font-size: larger;
      font-weight: bolder;
    }

      </style>
    </head>
    <body>
      <form action="/y_predict" method="post">
        <div class="container">

          <div class="brand-title">CAR PERFORMANCE ANALYZER</div>
          <div class="inputs">
           <label>CAR NAME</label>
           <input type="text" name="car name" placeholder="Enter car name with brand" required=""/>
           <label>DISPLACEMENT</label>
           <input type="number" name="disp" placeholder="Example: 220" required=""/>
           <label>SELECT NO. OF CYLINDER</label>
           <select name="cylinder" id="cylinder">
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
            <option value="6">6</option>
            <option value="7">7</option>
            <option value="8">8</option>
          </select>

           <label>HORSE POWER</label>
           <input type="number" name="hP" placeholder="Example: 200" required/>
           <label>WEIGHT</label>
           <input type="number" name="w" placeholder="in kg" required/>
           <label>ENTER ACCELERATION</label>
           <input type="number" name="Acc" placeholder="Example: 11.2" required=""/>
           <label>ENTER MODEL YEAR</label>
           <input type="text" name="Model" placeholder="YY" required=""/>
           <label>SELECT ORGIN</label>
           <select name="orgin" id="origin" required="">
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
          </select>

           <button type="submit">Submit</button>


         </div>
      </form>
      <p id="predicted_value">
       {{prediction_text}}
      </p>
    </body>
</html>
```

13.2 GitHub & Project Demo Link

https://github.com/wasimahmed21/SI-GuidedProject-9050-1663041940