Team ID	PNT2022TMID06676
Team Members	Allvin Lawrence D
	Ashvanthi S
	Arunmozhivarman
	Roopan

Machine Learning Based Vehicle Performance analyzer

Documentation

Table of Contents

1. INTRODUCTION (yet to change)

1.1 Project Overview

The automotive industry is extremely competitive. With increasing fuel prices and picky consumers. Automobile makers are constantly optimizing their processes to increase fuel efficiency. So, we can help the predicting processor done easier by developing the application.

1.2 Purpose

The purpose of this project is to give the customer a portal to predict the performance of the vehicle (miles per gallon). Now a days fuel prices are increasing and automobile industries tryto optimize the vehicle for running them using less fuel. This application helps them to predict the performance of the vehicle.

2. LITERATURE SURVEY

2.1 Existing Problem

It is hard to predict the performance of the vehicle. It takes us a lot of time and hard work to predict the performance. if the vehicle designing engineer able to predict the performance of vehicle with less amount of the time, It make the Engineer design and testing process easier for them.

2.2 References

 https://www.researchgate.net/publication/273951090_Simulation_for_prediction_of_ve hicle_efficiency_performance_range_and_lifetime_A_review_of_current_techniques_and _their_applicabilit   y_to_current_and_future_testing_standards

 https://www.etssolution-asia.com/blog/vehicle-performance-engineering

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas
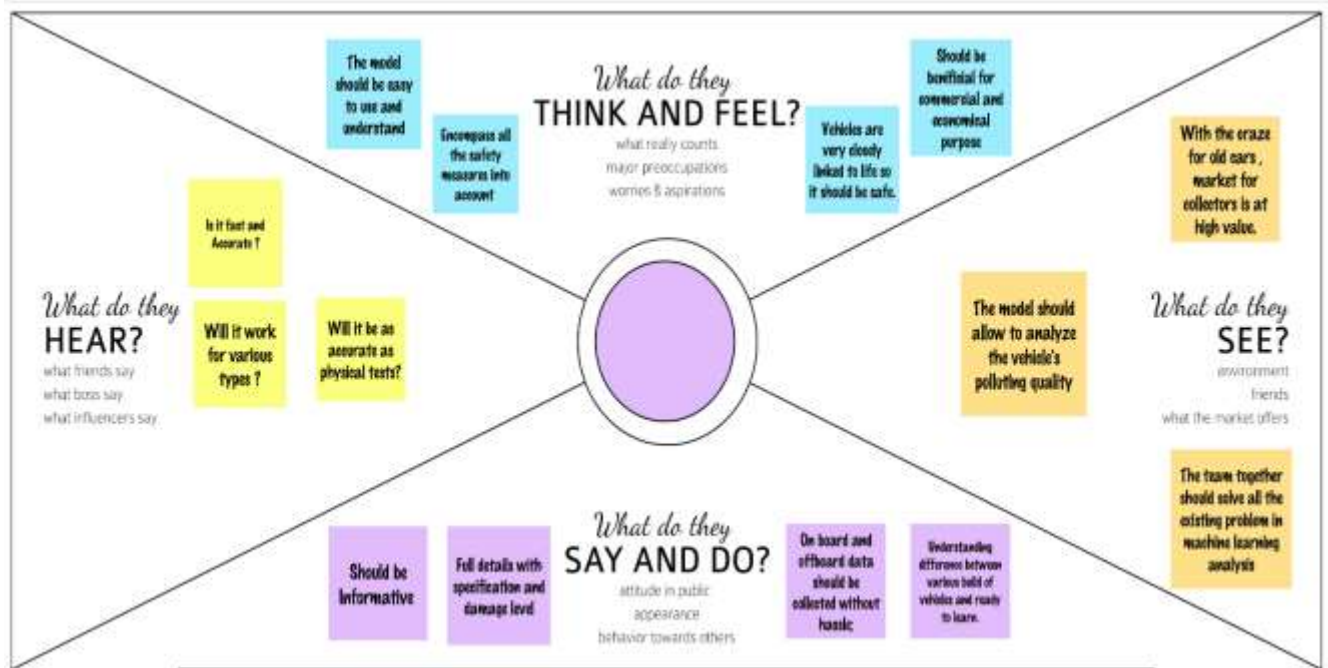
3.2) Ideation & Brainstorming

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

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

Share template feedback

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

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕐 5 minutes

> **PROBLEM**
> A detailed report of the vehicle health status and finding out of problem in the vehicle through our machine learning algorithm

Key rules of brainstorming
To run a smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

❷ Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕐 10 minutes

Tip

❸ Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

🕐 30 minutes

Major services

Performance analyser

Minor services

Prediction using algorithm

④

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⏱ 20 minutes

♡

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

7
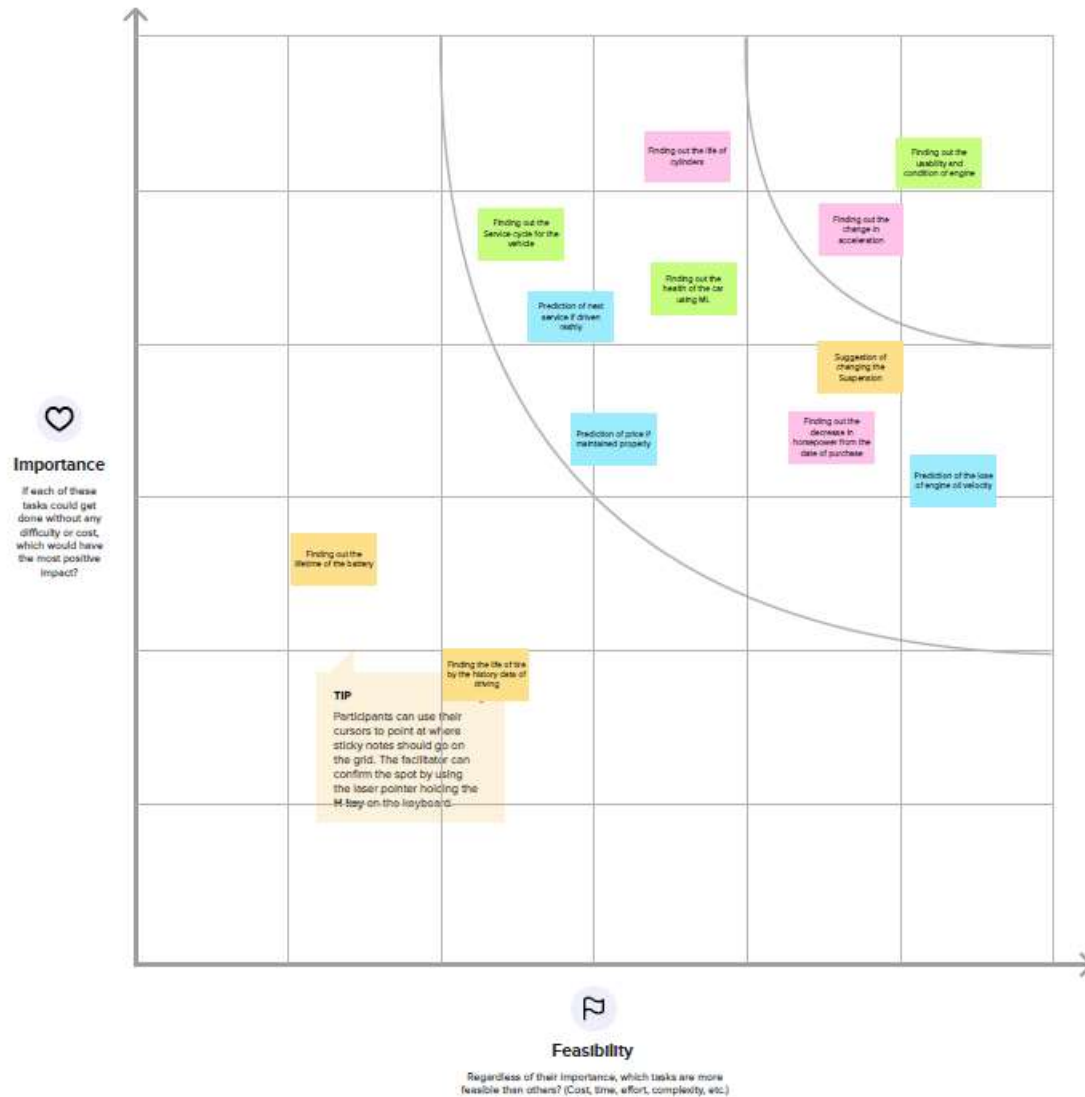
3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To analyse a vehicle's performance with respect to the various data collected from the vehicle and using Machine learning algorithm to find the difference between the ideal/new vehicle of the same kind. The application allows the user (either the everyday customer or workshop owners) to analyse a vehicle using the onboard data such as total drive mile and etc., to analyse the performance of the vehicle in order to enhance or repair purposes without having to spend hours on testing or taking the vehicle apart. The algorithm will allow to the user to know which aspect of the vehicle needs a check-up and should be taken care of most.
2.	Idea / Solution description	Updating a details dataset of cars/vehicle of any desire with ideal specification sheet in format of .excel or .csv file. Train the algorithm to receive the data from the user onboard data and compare to analyse performance for the said vehicle.
3.	Novelty / Uniqueness	This system allows a wide range of vehicles to be set up for analysis. The ML allows for faster and swappable data set to be trained for various different vehicles and expected performance in various different regions. This allows varied performance analysis depending on the vehicle's region of use. For example, the battery life of a car is dependent on the temperature of the surroundings; suspension depends on the type of terrain the vehicle most frequently visits or raids on.

4.	**Social Impact / Customer Satisfaction**	This method is highly feasible and does not require any heavy new equipment to be carried over. Being extremely cost effective allows it to be consumer pleasing and can be modified with various datasets allowing It to be used in various regions around the world. With recent e-vehicle trend and even gas cars come to the market with digital control and computer data allows getting onboard data a lot easier without any extra gadget to be placed in the vehicle. The model can also be specifically trained for other features and heavy vehicles if needed. For eg: Military vehicles.
5.	**Business Model (Revenue Model)**	Can be a monthly subscription feature for using the cloud controlled ML software / or paid application with dedicated UI and cloud (similar to adobe applications).
6.	**Scalability of the Solution**	Complexity of the model can be increased by adding various different types of vehicles and considering age and wear of the vehicle as considerable parameter. Data from recent repairs can also be added to account for increase or decrease in performance. This application can also be paired with insurance claiming software for faster and efficient account claiming.

3.4 Proposed Solution fit

1. CUSTOMER SEGMENT(S) `CS`

- The General Car driving consumer.
- Workshop and mechanic owners
- Refurbishing and car-enthusiasts
- Military personnel (for army vehicles

5. CUSTOMER CONSTRAINTS `CC`

- Lack of motor knowledge
- Trust and working constraints
- Time and money
- Device availability.
- Difference in usage criteria can change the data.

7. AVAILABLE SOLUTIONS `AS`

- Body workshops and repair service centres of specific
- Free lance mechanics and automated repair stations
- Cons: expensive and is only available in certain locations makes it impossible reach.

2. JOBS-TO-BE-DONE / PROBLEMS `J&P`

- The need to analyse performance drops in vehicles to avoid expensive repairs.
- Driving to the workshop if living in outskirts
- Workshop owners or employees need to take apart or use extensive machinery to check for performance issues.

6. PROBLEM ROOT CAUSE `RC`

- People need to check their car performance so they don't break down at awkward times leading to risky situation.
- Estimation of price and repurchase value of the car.
- Car evaluation requires higher machinery to do so in recent times.
- High cost and time consuming.

8. BEHAVIOUR `BE`

- Vehicular analysis is often overlooked. Ignoring blatant signs of repair etc.
- Spend extra pounds for small fix.

3. TRIGGERS `TR`

- Long Drive Hours to reach a body shop
- Paying extensive amounts for a checkup.
- Extensive machinery needed.
- Waiting time for parts for the specified vehicle
- Easy to learn and use also used for comparison.

4. EMOTIONS: BEFORE / AFTER `EM`

BEFORE:

- Frustration for not knowing the vehicle better and feeling tired for driving long hours. Completely blind-sighted on the performance drop in vehicle.
- 2nd hand purchase of vehicles can lead to purchasing a non performing vehicle making the consumer cheated.

AFTER:
Enables the people to check their vehicle at home at their leisure. Trusting and feeling easy to do check on vehicle and time saving and cost effective.

7. YOUR SOLUTION `SL`

- A Machine learning cloud application that allows the user either a common consumer or bodyshop owner to compare their original or last review performance to their current vehicular performance and find out which part of the vehicle needs repair and how long till the next repair cycle etc
- The solution can also be fine tuned to suit specific companies of specific vehicle type by changing the data set that will used for the algorithm to process information.
- Using onboard data such as driving pattern, location, hours on the road terrain, repair history etc we can give out a clean analysis
- This can also benefit car enthusiast.

10.CHANNELS OF BEHAVIOUR `CH`

ONLINE

The data from their onboard system is taken and fed to the cloud , from where the algorithm kicks in to find the ideal case and give out a performance list.,

OFFLINE

Addition of a onboard system might be required to record and now the consumers can analyse and know more about their vehicle health and plan ahead.

4.REQUIREMENT ANALYSIS

4.1Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form. Registration through any cloud interface. Registration through Gmail.
FR-2	User Confirmation	Confirmation via Email. Confirmation via OTP.
FR-3	Reset Password	Reset password through mail or phone number
FR-4	Enter DATA from onboard	Add file type or name of system Through a form to enter various details of vehicle
FR-5	User result	Result of the analysis should be displayed or mailed to the user.
FR-6	User Feedback	User should be able to reflect on the results and enter a suggestion or feedback with use of the application

4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The Application allows the common consumer and professional workshop owners to understand the difference in performance and with respect to which aspect of the vehicle.
NFR-2	Security	The user's credential is crypted and only used for analyzing purposes.
NFR-3	Reliability	As the model is trained with various data sets, rate of error is extremely low
NFR-4	Performance	Faster and efficient (cost wise) compared to picking apart the vehicle parts. Performance can also be limited of the type of vehicle data set is limited or the data is allowed to train for longer time.
NFR-5	Availability	Can be made easily available as cloud application.
NFR-6	Scalability	Better scalability and easy to do so.

5.PROJECT DESIGN

5.1 Data Flow Diagrams
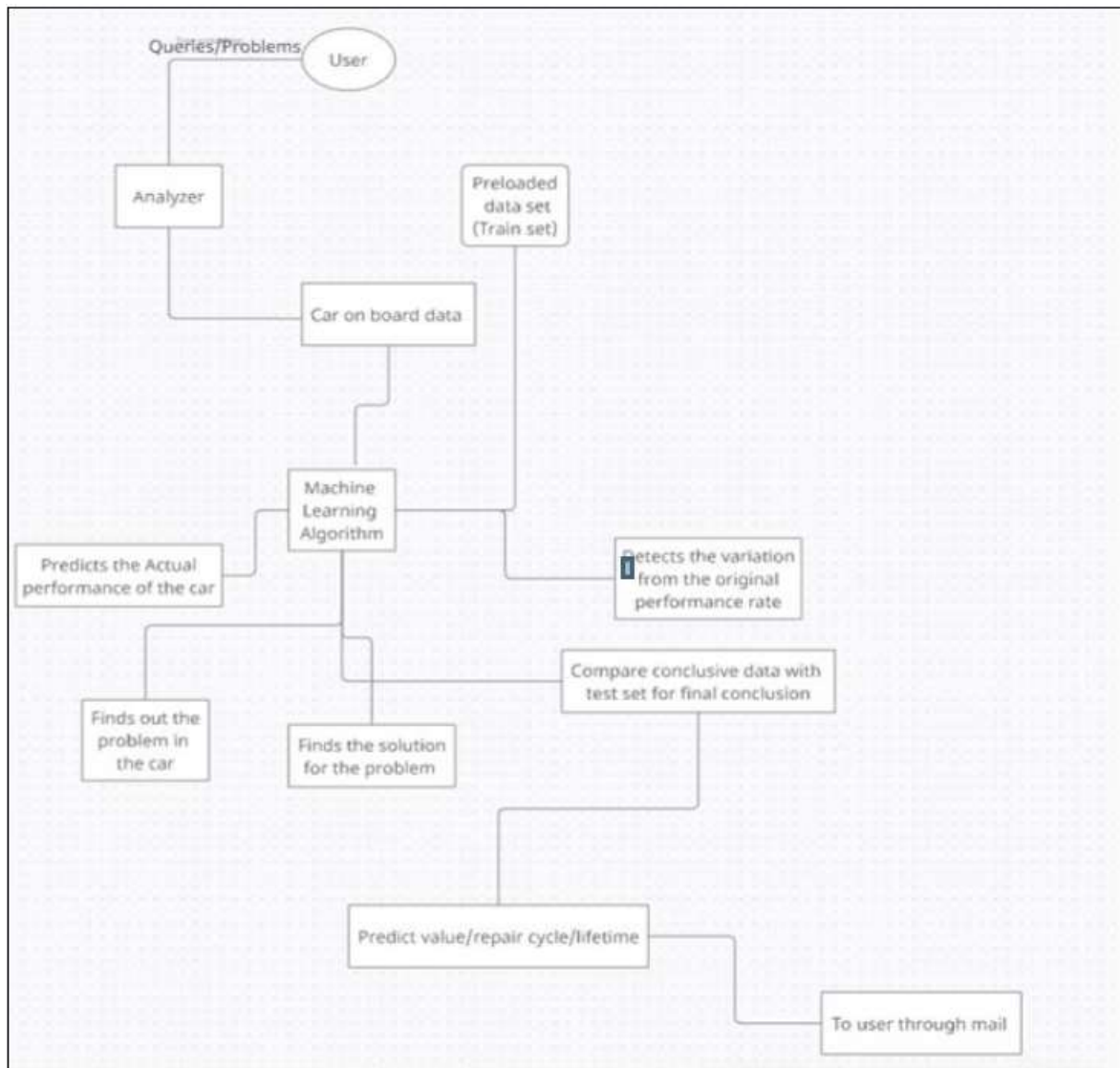
5.2 Solution & Technical Architecture
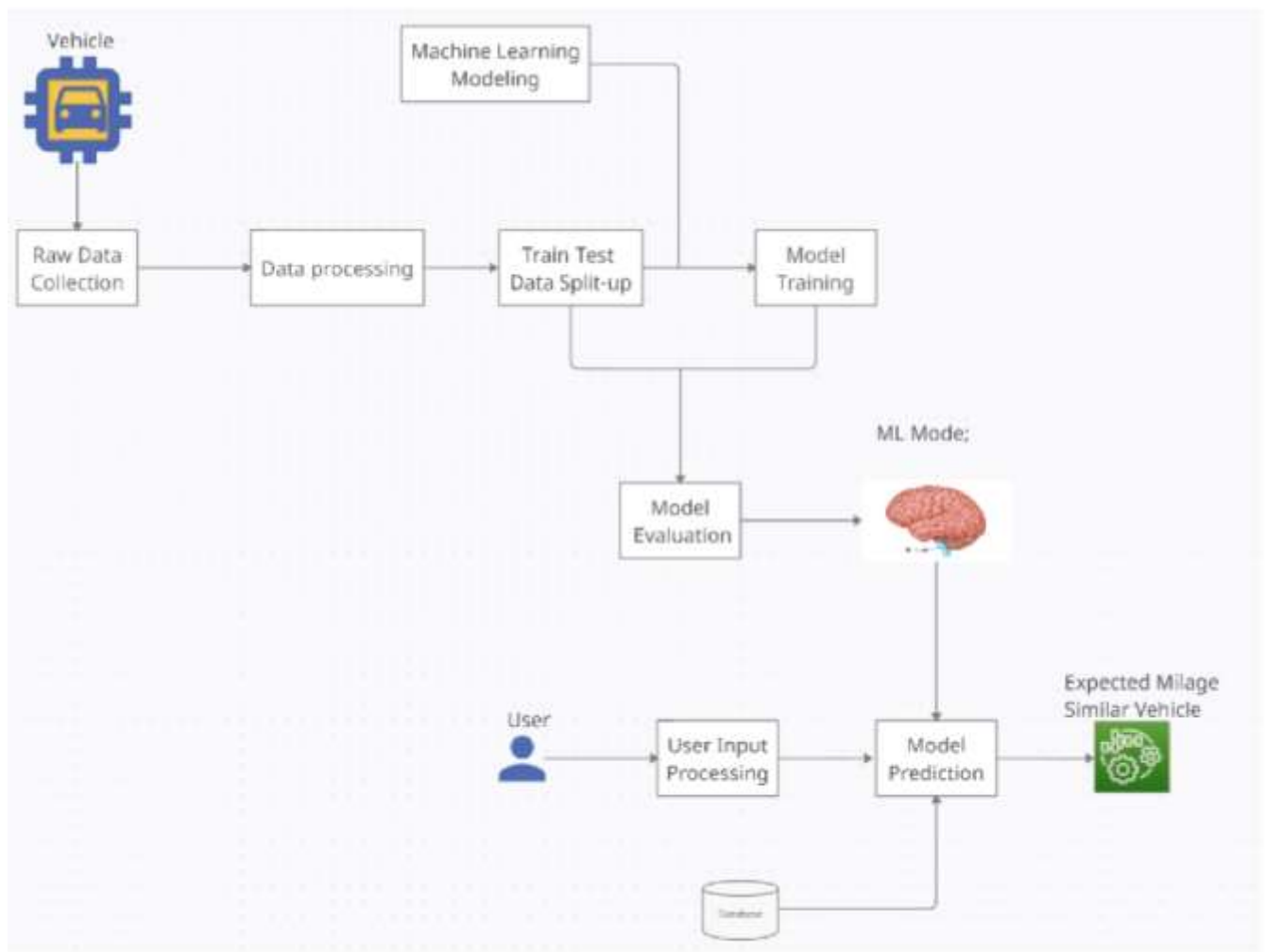
5.3 User Stories (yet to change)

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Visiting Webpage	USN-1	As a user, I can able to view the website using the good domain name	I can access the website	High	Sprint-4
	Design	USN-2	As a user, I can able to Enter the data	I can submit the data to the server to preict	High	Sprint-1

1

	Result	USN-3	As a user I can get the predicted performance	I get the MPG value	High	Sprint 3
	Design	USN-4	Good experience and less time consuming	I get user friendly UI	Low	Sprint 4
	Result	USN-5	Website is fast	I get result faster	Low	Sprint 4
	Result	USN-6	As a user I expect the prediction is highly accurate	High value	Medium	Sprint 3

6.PROJECT PLANNING &SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Download the dataset.	20	Medium	1
Sprint-1	Data Pre-processing	USN-2	Import libraries and read the dataset	4	Medium	1
Sprint-1		USN-3	Handle the missing value and label the encoding	4	Medium	1
Sprint-2	Model Training and Checking for Effeicient Method	USN-4	Split the dataset into train and test data	2	High	1
Sprint-2		USN-5	Training the model using random for aggression	2	High	1
Sprint-3	Model Building and developing in IBM cloud	USN-6	Train the datasets to run smoothly and see an incremental improvement in the prediction rate for the available Machine Learning algorithms.	5	Low	1
Sprint-3		USN-7	Build The Model With The Decision Tree Algorithm	6	Low	2
Sprint-4	Application Building	USN-8	Build Python Code	5	Medium	1,2,3,4
Sprint-4		USN-9	Output	5	Medium	1,2,3,4

6.2 Sprint Delivery Schedule

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	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	06 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	14 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	20 Nov 2022

7.CODING & SOLUTION
7.1 GUI.

Created A GUI based interface for the easy utilization for the customer using HTML,CSS and Python

Output:

7.2 MODEL

Implentation of the model integrated with HTML and CSS in Appendex

8 TESTING.

Input dataset:

```
array([[-0.76894131, -0.6338073 , -0.6601792 , -0.55841396,  0.37006707,
        -0.38798451, -0.75260027],
       [ 1.64342359,  2.07850256,  3.38793037,  1.59935958, -2.34958594,
        -0.64240058, -0.75260027],
       [-0.76894131, -0.50697267, -0.55222961, -0.74265735, -1.53369004,
         1.64734406, -0.75260027],
       [ 0.43724114,  0.37111326, -0.25536825,  0.40177755,  0.17580614,
        -0.13356844, -0.75260027],
       [ 1.64342359,  1.15163409,  0.68919065,  1.08206084, -0.05730697,
         0.88409585, -0.75260027],
       [-0.76894131, -0.86796355, -1.06499016, -1.03791919,  0.25351051,
         1.39292799, -0.75260027],
       [-0.76894131, -0.87772006, -0.714154  , -0.73675211, -0.40697665,
        -0.38798451,  0.47613487],
       [-0.76894131, -0.99479818, -1.01101537, -1.28593914,  0.68088456,
         1.13851192,  0.47613487],
       [-0.76894131, -0.87772006, -1.01101537, -1.10878203,  0.33121488,
         0.37526371,  1.70487001],
       [ 0.43724114,  0.1272005 , -0.52524222, -0.39779152,  0.17580614,
        -1.4056488 , -0.75260027],
       [-0.76894131, -1.08260677, -1.41582632, -1.50561395,  0.37006707,
        -0.38798451,  1.70487001],
       [ 1.64342359,  1.59067705,  2.03856051,  1.71982642, -1.33942911,
         0.12084763, -0.75260027],
       [ 0.43724114,  0.61502602, -0.714154  ,  0.76790224,  2.11841544,
         0.12084763, -0.75260027],
       [-0.76894131, -0.41916407, -0.22838085, -0.30566982, -0.64008976,
         1.64734406,  1.70487001],
       [-0.76894131, -0.64356381,  0.2304049 , -0.81470124, -1.18402036,
        -1.4056488 ,  0.47613487],
       [ 1.64342359,  1.27846872,  1.2289386 ,  0.6049177 , -1.76680315,
        -1.4056488 , -0.75260027],
       [-0.76894131, -0.87772006, -0.714154  , -0.86902942,  0.09810177,
         1.13851192,  0.47613487],
       [ 1.64342359,  1.59067705,  0.55425367,  1.15292368,  0.71973674,
         0.88409585, -0.75260027],
       [-0.76894131, -0.87772006, -0.44428003, -0.93753017, -0.40697665,
        -1.15123272,  1.70487001],
       [-0.76894131, -0.87772006, -0.714154  , -0.86666732, -0.56238539,
         0.37526371,  0.47613487],
       [ 1.64342359,  1.27846872,  0.82412764,  1.07025036, -0.13501135,
         0.88409585, -0.75260027],
       [-0.76894131, -0.72161589, -0.25536825, -0.76273515, -0.01845479,
        -0.89681665,  1.70487001],
       [-0.76894131, -1.13138933, -1.06499016, -1.35916408,  1.34137172,
        -1.15123272,  1.70487001],
       [-0.76894131, -0.83869402, -0.57921701, -0.85249475, -0.09615916,
         0.12084763,  0.47613487],
       [-0.76894131, -0.77039845, -0.93005317, -0.80170972,  0.52547581,
         1.64734406,  1.70487001],
       [-1.37203254, -1.14114584, -0.20139345, -0.70132069, -0.79549851,
        -0.89681665,  1.70487001],
```

```
[-0.76894131, -0.99479818, -0.93005317, -1.15602393,  0.48662363,
  0.37526371,  1.70487001],
[-0.76894131, -0.30208595, -0.33633044, -0.40015361, -0.40697665,
  1.64734406, -0.75260027],
[-0.76894131, -0.45819012, -0.33633044, -0.06946035,  0.33121488,
  1.64734406, -0.75260027],
[-0.76894131, -0.7313724 , -0.44428003, -0.33519601,  1.18596297,
  1.64734406, -0.75260027],
[ 1.64342359,  1.91264189,  1.76868654,  0.75491072, -2.15532501,
 -1.4056488 , -0.75260027],
[-0.16585009, -0.03866017, -0.74114139,  0.71593615,  1.76874576,
  0.88409585,  0.47613487],
[ 1.64342359,  1.60043356,  1.28291339,  1.5249536 , -1.06746381,
  0.12084763, -0.75260027],
[-0.76894131, -0.45819012, -0.49825482, -0.1580389 ,  0.0203974 ,
  1.64734406, -0.75260027],
[ 1.64342359,  1.27846872,  1.2289386 ,  1.85919001, -0.40697665,
 -0.13356844, -0.75260027],
[-0.76894131, -0.78015496, -0.49825482, -0.54306034, -0.01845479,
  0.12084763,  0.47613487],
[-0.76894131, -0.64356381,  0.14944271, -0.31157506, -0.60123758,
 -0.64240058,  0.47613487],
[ 0.43724114,  0.42965232,  0.14944271,  0.13604189, -0.21271572,
 -0.13356844, -0.75260027],
[ 0.43724114,  0.37111326,  0.01450573,  0.60846084, -0.01845479,
 -1.15123272, -0.75260027],
[-0.76894131, -0.72161589, -0.25536825, -0.6517167 , -0.21271572,
 -1.4056488 ,  1.70487001],
[ 1.64342359,  1.91264189,  2.03856051,  2.39892865, -1.57254222,
 -1.15123272, -0.75260027],
[ 0.43724114,  0.43940883, -0.39030523,  0.33800099,  0.64203237,
  0.62967978, -0.75260027],
[-0.76894131, -0.86796355, -0.98402797, -1.03791919,  1.14711079,
  0.37526371,  1.70487001],
[-0.76894131, -0.6338073 , -0.49825482, -0.82414962,  0.37006707,
 -0.89681665, -0.75260027],
[ 1.64342359,  2.07850256,  1.76868654,  2.15208975, -1.37828129,
 -1.15123272, -0.75260027],
[-0.76894131, -0.93625912, -1.01101537, -1.26822343, -0.67894195,
  1.13851192,  1.70487001],
[ 1.64342359,  1.60043356,  1.44483777,  1.69974861, -0.98975944,
 -0.64240058, -0.75260027],
[-0.76894131, -0.3508685 , -0.39030523, -0.29031621,  0.37006707,
  1.13851192, -0.75260027],
[-0.76894131, -0.86796355, -0.39030523, -0.77808877, -0.01845479,
 -0.64240058,  0.47613487],
[ 1.64342359,  2.36144136,  2.52433366,  2.39538551, -1.57254222,
 -0.64240058, -0.75260027],
[-0.76894131, -0.3508685 , -0.33633044,  0.1313177 ,  1.92415451,
  1.64734406, -0.75260027],
[-0.76894131, -1.05333724, -0.93005317, -1.00366882,  1.53563265,
 -1.15123272,  0.47613487],
[ 0.43724114,  0.61502602, -0.17440605,  0.71003092,  1.34137172,
  0.37526371, -0.75260027],
[-0.76894131, -0.45819012, -0.44428003, -0.03993416,  0.68088456,
  0.88409585, -0.75260027],
[ 0.43724114,  0.42965232,  1.63374955,  0.61554713, -0.83435069,
  0.62967978, -0.75260027],
[ 1.64342359,  2.61511063,  3.25299338,  1.77297355, -2.15532501,
 -1.4056488 , -0.75260027],
[-0.76894131, -1.12163282, -0.95704057, -1.54813166,  0.95284986,
 -1.15123272,  1.70487001],
[ 0.43724114,  0.1272005 , -0.52524222,  0.04864439,  0.09810177,
  0.62967978, -0.75260027],
[-0.76894131, -0.87772006, -0.44428003, -0.76155411,  1.34137172,
 -0.64240058,  1.70487001],
```

```
[-0.76894131, -0.87772006, -0.33633044, -0.75092468,  0.564328  ,
 -0.89681665,  1.70487001],
[ 0.43724114,  0.37111326, -0.12043126,  0.59783142,  0.64203237,
  0.62967978, -0.75260027],
[ 0.43724114,  0.43940883, -0.12043126, -0.15921995, -0.21271572,
 -0.64240058, -0.75260027],
[-0.76894131, -0.3508685 , -0.39030523, -0.22299651,  0.95284986,
  1.64734406, -0.75260027],
[ 1.64342359,  1.12236455,  0.95906463,  0.62027132, -1.96106408,
 -1.4056488 , -0.75260027],
[ 0.43724114,  0.61502602, -0.12043126,  0.48681296,  0.564328  ,
 -0.38798451, -0.75260027],
[-0.76894131, -0.51672918, -0.39030523, -0.25134164, -0.01845479,
  1.13851192,  1.70487001],
[-0.76894131, -0.65332032, -0.47126742,  0.06517905,  1.53563265,
 -0.89681665,  0.47613487],
[-0.76894131, -0.78015496, -0.79511619, -0.84895161, -0.40697665,
  1.64734406,  1.70487001],
[-0.76894131, -0.86796355, -0.25536825, -1.03673814,  1.34137172,
 -1.15123272, -0.75260027],
[ 0.43724114,  0.6930781 ,  0.14944271,  0.04510125, -0.79549851,
 -1.15123272, -0.75260027],
[-0.76894131, -0.93625912, -0.93005317, -1.14421345,  1.92415451,
 -1.15123272, -0.75260027],
[ 0.43724114, -0.23379038,  0.77015285,  0.57421047,  0.09810177,
  0.62967978,  0.47613487],
[ 1.64342359,  0.71259112,  0.14944271,  1.34189126,  1.34137172,
  0.37526371, -0.75260027],
[ 1.64342359,  1.27846872,  1.2289386 ,  1.81076706, -0.79549851,
 -0.38798451, -0.75260027],
[-0.76894131, -0.69234636, -0.79511619, -0.90446084, -0.01845479,
 -0.64240058,  0.47613487],
[ 0.43724114,  0.1272005 , -0.39030523, -0.05764987,  0.564328  ,
 -0.38798451, -0.75260027],
[-0.76894131, -0.30208595,  0.01450573, -0.14622843, -0.44582883,
  1.13851192, -0.75260027],
[-0.76894131, -0.6338073 , -0.49825482, -0.62455261,  0.17580614,
 -0.89681665, -0.75260027],
[-0.76894131, -0.70210287, -0.25536825, -0.27141945, -0.21271572,
 -0.13356844,  0.47613487],
[ 1.64342359,  1.12236455,  0.95906463,  1.4375561 , -0.60123758,
 -0.38798451, -0.75260027]])
```
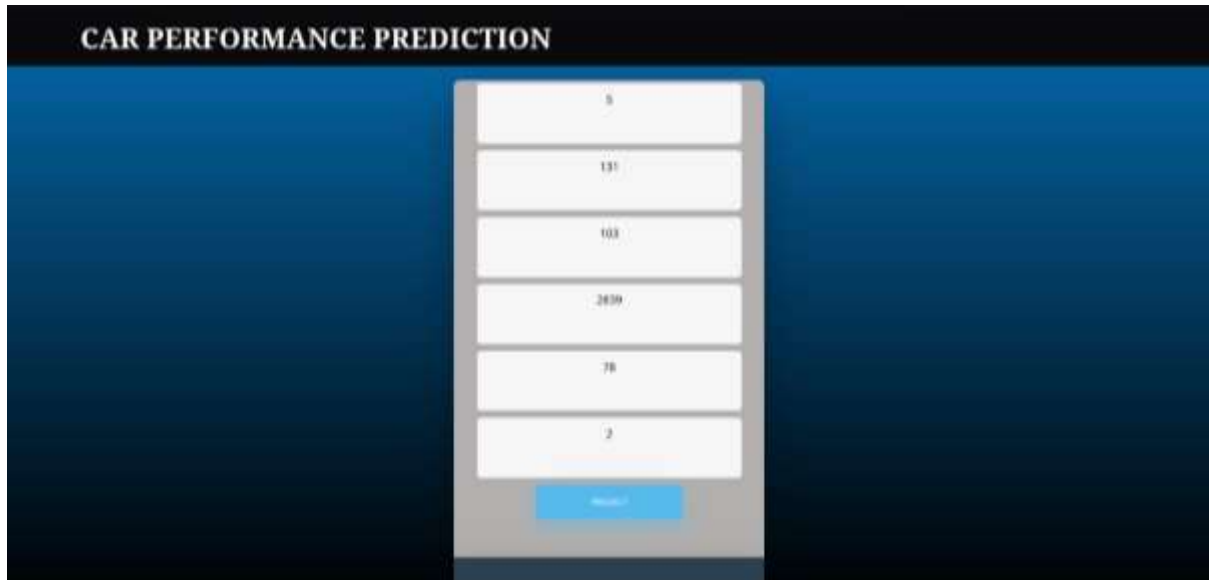
Predicted Output:

```
In [129]:   1  y_pred = rf.predict(x_test)

In [130]:   1  y_pred

Out[130]:  array([15.95333333, 38.6       , 31.13333333, 27.21333333, 15.84666667,
                  14.01666667, 32.68333333, 26.1       , 24.59666667, 14.36666667,
                  35.80333333, 19.54666667, 24.75333333, 18.08       , 28.08333333,
                  15.        , 33.05       , 37.68333333, 18.44666667, 23.15333333,
                  17.1       , 18.73       , 29.83666667, 28.93333333, 26.99666667,
                  26.12666667, 34.61       , 26.51333333, 35.71666667, 23.57666667,
                  15.58666667, 34.23333333, 14.46333333, 13.65       , 23.53333333,
                  13.15       , 34.19666667, 11.7       , 33.33666667, 20.40333333,
                  28.61333333, 29.48333333, 37.87       , 11.16666667, 20.58666667,
                  21.13       , 31.33666667, 19.14       , 18.07333333, 17.82666667,
                  18.26666667, 17.97666667, 25.73       , 27.51333333, 25.50666667,
                  14.08333333, 26.15666667, 25.84666667, 19.56666667, 20.40333333,
                  21.79333333, 25.53       , 13.96666667, 13.4       , 27.88       ,
                  23.11666667, 28.21666667, 15.81333333, 30.93333333, 36.49333333,
                  24.40333333, 21.46666667, 19.47       , 31.68666667, 14.93333333,
                  14.61666667, 28.9       , 22.06       , 13.5       , 17.49       ])
```

8.2 User Accepting Testing.

INPUT:

Output:

9.RESULTS

Model Evaluation

```
1 from sklearn.metrics import r2_score,mean_squared_error
```

```
1 acc = r2_score(y_test, y_pred)
```

```
1 acc
```

0.8570363544939325

```
1 err=np.sqrt(mean_squared_error(y_test,y_pred))
```

```
1 err
```

2.7436940578959117

10.ADVANTAGES AND DISADVANTAGES.

Advantages:

- Prediction of vehicle performance is faster and easier.
- Easy and wide range of access from everyday user to machinist in workshops.

Disadvantages:

- Accuracy need to be increased
- The dataset should be wide to for every region
- Values such as standard climate conditions need to be considered on the long run of vehicles

11.CONCLUSION.

The automotive industry is extremely competitive. With increasing fuel prices and picky consumers. Automobile makers are constantly optimizing their processes to increase fuel efficiency. The performance analysis of the car is based on the various parameters. These are the factors on which the health of the car is analyzed, improved to gain the competitive advantage. This application will solve the problems in evaluation of the vehicle

12.FUTURE SCOPE

- Developing the CSS and Animation of the Website
- Developing the High Accuracy Model
- Developing the Code to make API key highly secure

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-21225-1659775342/tree/main/Project%20Design%20%26Planning
(Consist of source code , demo video and related data files)