Team ID	PNT2022TMID19412
Team Members	P Gokul krishnan, P Balamurugan, M Tamilarasan, K Srijayakanth

Machine Learning Based Vehicle Performance analyser

Documentation

Table of Contents

1. INTRODUCTION

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 vechicle (miles per gallon). Now a days fuel prices are increasing and atuomobile industries try to optimize the vehicle for running them using less fuel. This application help 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.

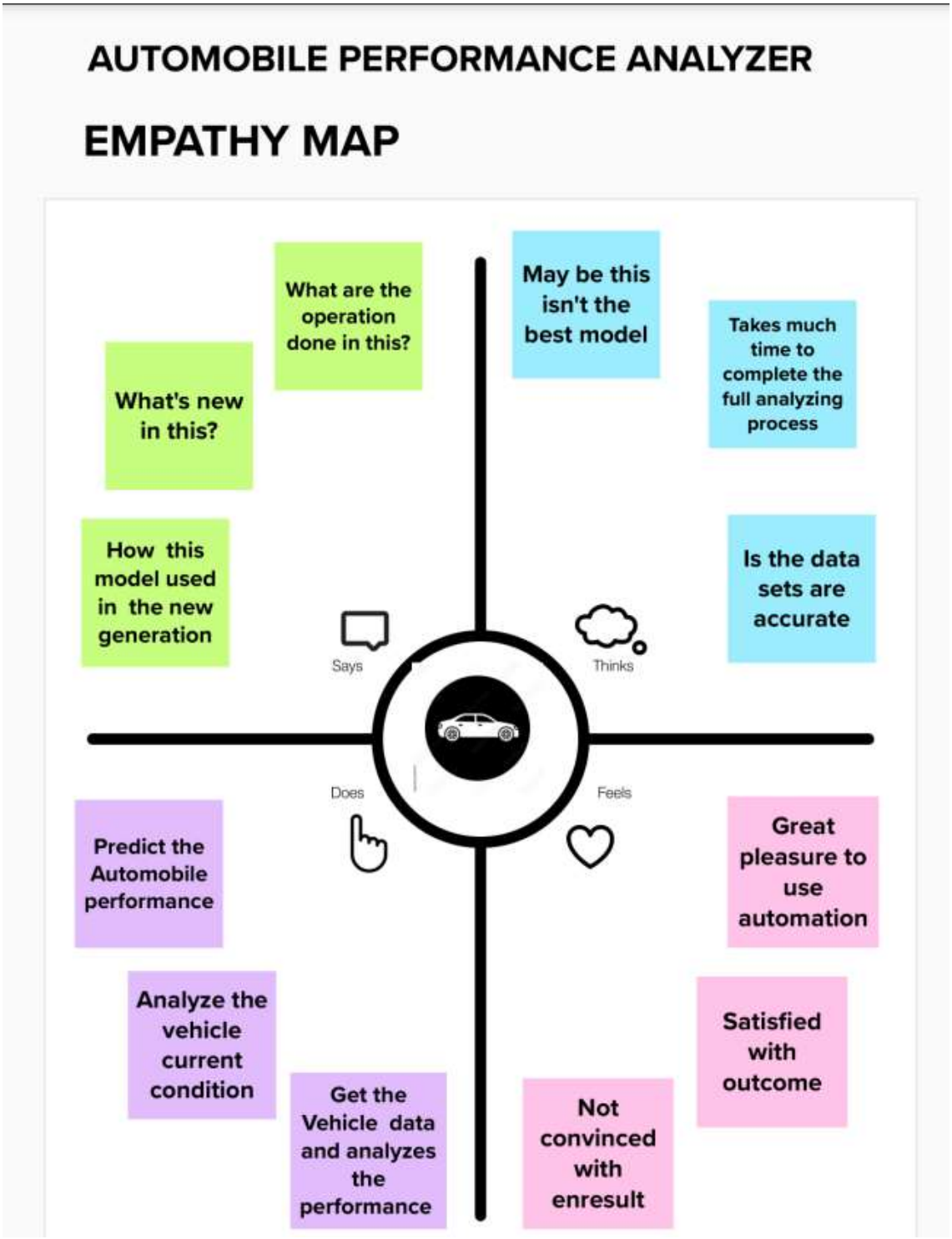
2.2 References

https://www.researchgate.net/publication/273951090_Simulation_for_prediction_of_vehicle_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 & Brainstroming

Template

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 →

Team Gathering Members

M Niyas Khan
P Cibi
R Narasimman
T Sivaprakash

Goal

To discuss the brainstorm idea and choose the idea that is cost efficient, flexible, easy to analyse the performance of the vehicle

1 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

Vehicle user or manufacture trying to analyse the performance of the vehicle But, it is hard to analysis. Because it needs a knowledge of the engineering and vehicle, it takes time to do it manually, which makes users feel fear, worried about the vehicle

Key rules of brainstorming

To run an smooth and productive session

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

GOKUL KRISHNAN P

SELF DRIVING AI USER FRIENDLY

FUEL TANK CAPACITY

ENGINE HORSE POWER

CAR MILEAGE PERMORMANCE

TAMILARASAN M

ENGINE TYPE(PETROL/ DIESEL)

FEEDBACK CAN BE GIVEN BY THE CUSTOMER

CAR WEIGHT AND STURCTURE (OR) DIMENSION

GEAR AUTOMATIC (OR) MANUAL

SRIJAYAKANTH K

ACCELERATION OF THE CAR

NO OF CYCLINDER

INTEGRATED SAFETY

POWER TRAIN INNOVATION

BALAMURUGAN P

LOCAL MARKET REQUIREMENT

CAR MATERIAL

DRIVING DYNAMIC

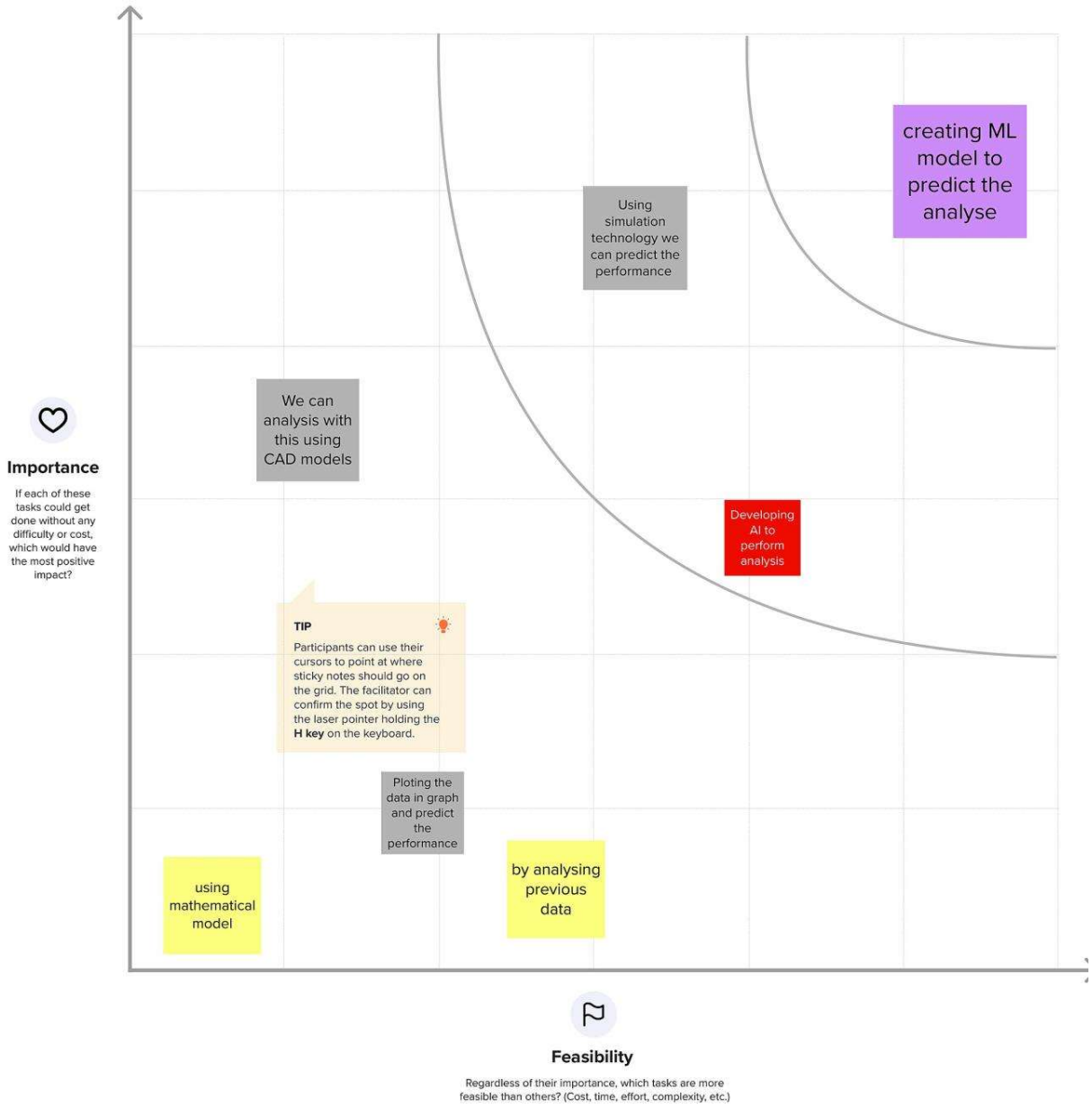
VEHICLE ENERGY MANAGEMENT

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.

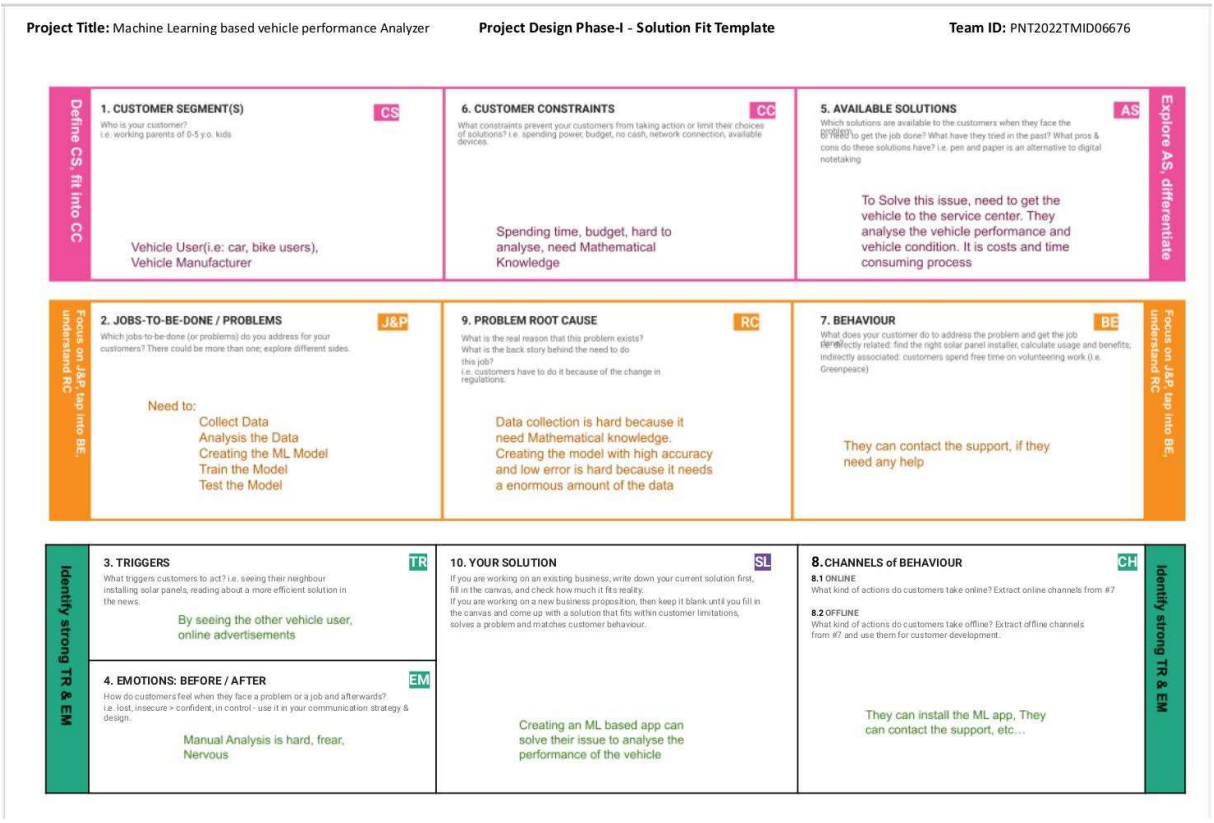
20 minutes



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Vehicle user or manufacture trying to analyse the performance of the vehicle But, it is hard to analysis. Because it needs a knowledge of the engineering and vehicle, it takes time to do it manually, which makes users feel fear, worried about the vehicle
2.	Idea / Solution description	Dataset of the Vehicle performance need to be collected and need to analyse the data. Based on the data analysis Machine Learning Model should be created and need to test the accuracy of the model and the error of the model.
3.	Novelty / Uniqueness	Using this Machine Learning project we can develop the app in that app we can frequently update the dataset and train the model, So the user can get the accurate data
4.	Social Impact / Customer Satisfaction	The Social impact for this product is good, It make people life easier by perform analyse of the vehicle
5.	Business Model (Revenue Model)	Alige Model, MVP (Minimum Viable Product) Model
6.	Scalability of the Solution	It can be further developed to provide app integration, We can further develop the project to bring more accuracy.

3.4 Proposed Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Collecting the details about the vehicle	Collecting Miles per galon, number of cylinder, displacement, horsepower, weight, acceleration, model_year, car name
FR-2	Launch the website	launch the website in the browser and enter the collected values and click submit
FR-3	View the result	Then the user can able to view the performance score
FR-4	Taking decision	By the result user can take decision to give the service to the vehicle or not based on the performance score

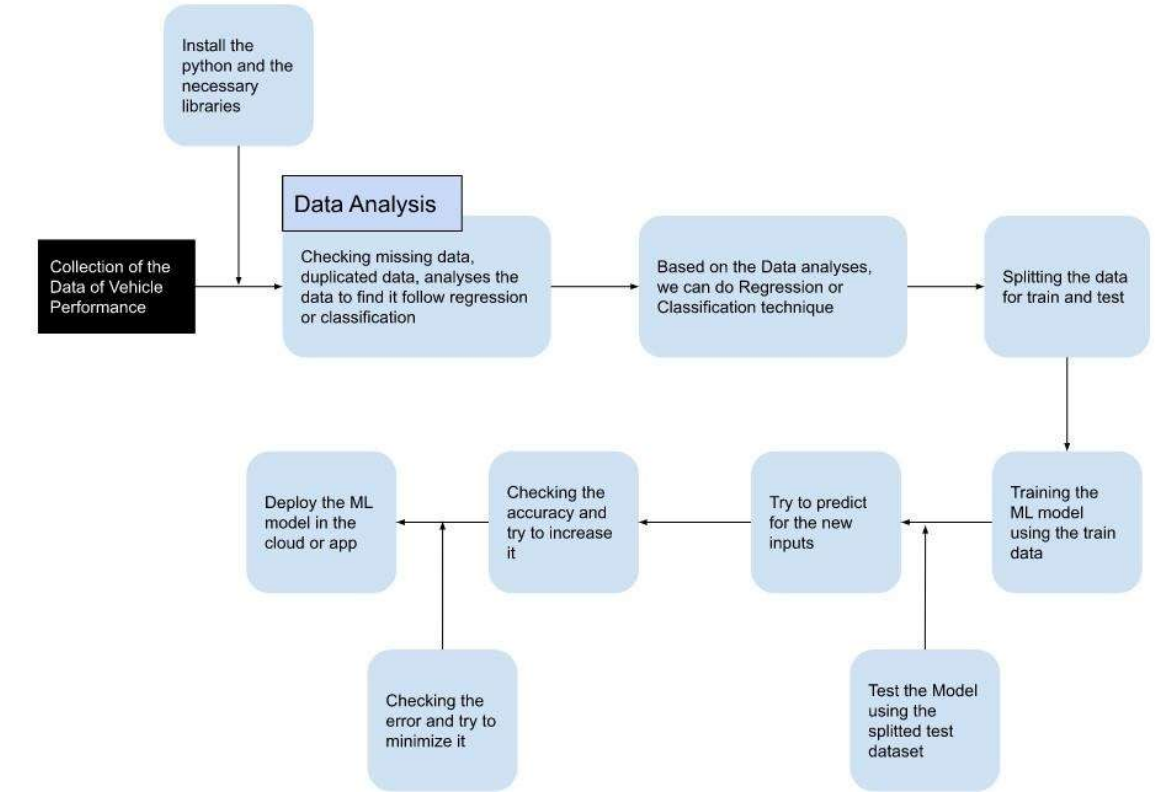
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It is hard for the people does not know about the terms used for the car. It should be easy to understand the website
NFR-2	**Security**	User data is completely secure. It will not get stored in the server Https provides good encryption to the data
NFR-3	**Reliability**	The website is reliable. It has to be more accuracy
NFR-4	**Performance**	The page load time, and the ML model predicting time should be with in 5sec
NFR-5	**Availability**	The user can able to use the website 24hours
NFR-6	**Scalability**	Can be scalable up to 2,00,000 ML product request predicting at a second Can be scaled to used database to track the previous previous statistics

5. PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer(web user)	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

			of the vehicle			
	Result	USN-3	As a user, I can get the predicted performance of the vehicle using the given data	I get the vehicle performance value	High	Sprint-2
	Design	USN-4	As a user, I want the good user experience.	I get easy understanding of website	Low	Sprint-3
	Result	USN-5	As a user, I want the website is fast	I get results faster	Low	Sprint-3
	Result	USN-6	As a user, I expect the prediction is highly accuracy.	I get most high accrued value	Medium	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

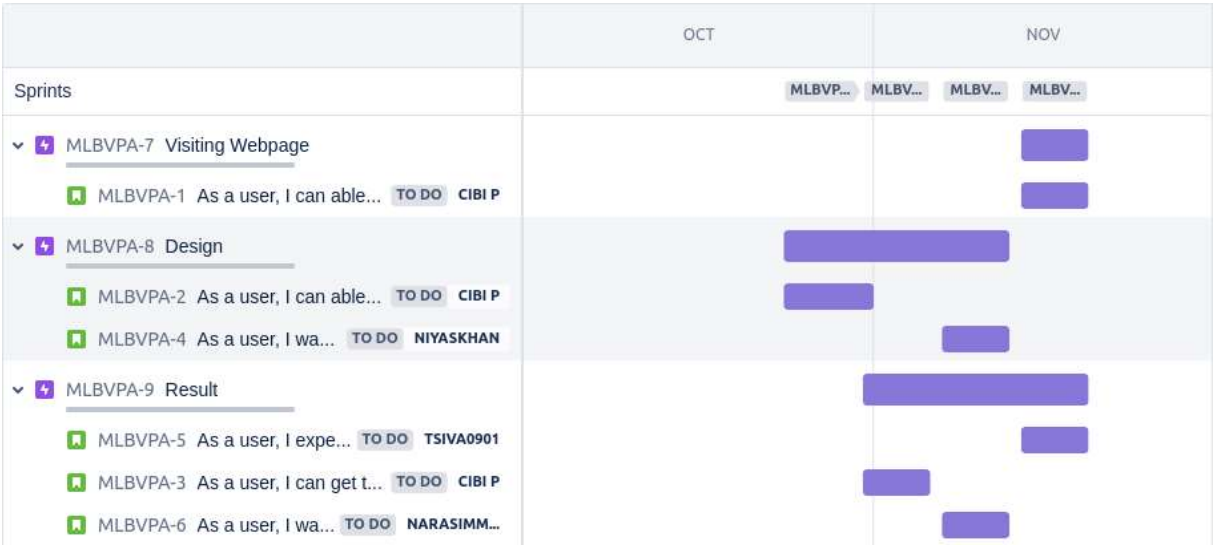
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Visiting Webpage	USN-1	As a user, I can able to view the website using the good domain name	10	High	Cibi P

Sprint-1	Design	USN-2	As a user, I can able to Enter the data of the vehicle	20	High	Cibi P
Sprint-2	Result	USN-3	As a user, I can get the predicted performance of the vehicle using the given data	20	High	Cibi P
Sprint-3	Design	USN-4	As a user, I want the good user experience.	10	Low	Niyas Khan M
Sprint-3	Result	USN-5	As a user, I want the website is fast	10	Low	Narasimman R
Sprint-4	Result	USN-6	As a user, I expect the prediction is highly accuracy.	10	Medium	Siva Prakash T

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	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA

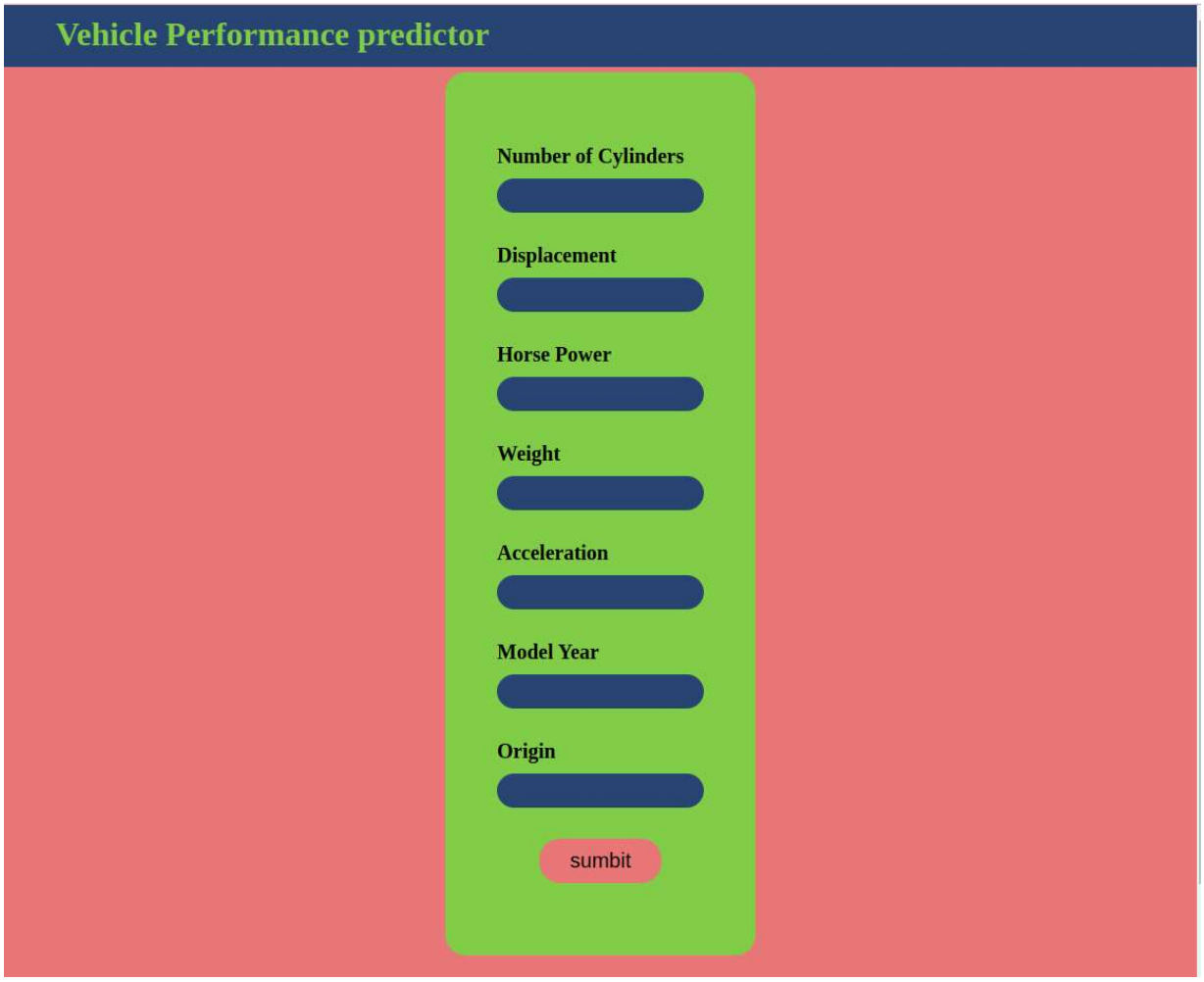


7. CODING & SOLUTION

7.1 Feature 1

Created a GUI based interface for the easy interation of the user using the HTML, CSS, PythonFlask. Code in Appendex

Output



7.2 Feature 2

implementation of the Model integrated with HTML and CSS in Appendex

8. TESTING

8.1 Test Cases

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]])
```

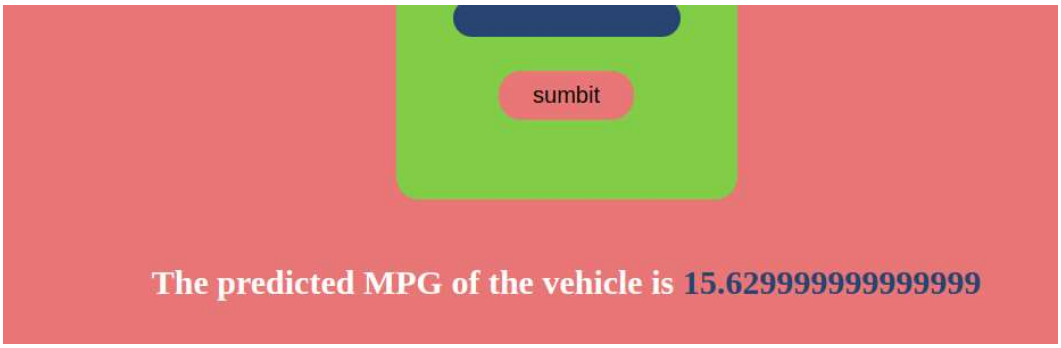
Predicted Output:

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

In [130]: 1 y_pred

Out[130]: array([15.95333333, 30.6       , 31.13333333, 27.21333333, 15.84666667,
                 14.01666667, 32.60333333, 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, 29.40333333, 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      , 31.60666667, 14.93333333,
                 14.61666667, 28.9       , 22.06      , 13.5       , 17.49      ])
```

8.2 User Accepting Testing

Input:

Output:



9. RESULTS

9.1 Performance Metrics

Model Evaluation

```
In [131]: 1 from sklearn.metrics import r2_score,mean_squared_error

In [132]: 1 acc = r2_score(y_test, y_pred)

In [133]: 1 acc
Out[133]: 0.8570363544939325

In [134]: 1 err=np.sqrt(mean_squared_error(y_test,y_pred))

In [135]: 1 err
Out[135]: 2.7436940578959117
```

10. ADVANTAGES & DISADVANTAGES

Advantages:

- It made easy to predict the performance of the Vehicle
- It can we accessible every one who want to predict the performance of the vehicle using the internet

Disadvantages:

- Need to Increase the Accuracy of the Project
- Cost for deploying in IBM CLOUD for permenently
- In Program API Key is Publicaly available, it may give rise to the security risk

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

13. APPENDEX

Source Code

Code:
index.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>ML Based Vehicle Performance Predictor</title>
        <meta charset="utf-8">
        <link rel="stylesheet" href="/static/css/main.css">
    </head>
    <body>
        <header>
            <h1>Vehicle Performance predictor</h1>
        </header>
        <div class="form-container">
            <form action="/model" method="POST">
                <div class="field">
                    <label for="no_of_cylinders">
                        <p class="cylinders field-name">Number of Cylinders</p>
                    </label>
                    <input type="number" id="no_of_cylinders input" name="no_of_cylinders">
                </div>

                <div class="field">
                    <label for="displacement">
                        <p class="displacement field-name">Displacement</p>
                    </label>
                    <input type="number" id="displacement input" name="displacement">
                </div>

                <div class="field">
                    <label for="horsepower">
                        <p class="horsepower field-name">Horse Power</p>
                    </label>
                    <input type="number" id="horsepower input" name="horsepower">
                </div>

                <div class="field">
                    <label for="weight">
                        <p class="weight field-name">Weight</p>
                    </label>
                    <input type="number" id="weight input" name="weight">
                </div>
```

```
                <div class="field">
                    <label for="acceleration">
                        <p class="acceleration field-name">Acceleration</p>
                    </label>
                    <input type="number" id="acceleration input" name="acceleration">
                </div>

                <div class="field">
                    <label for="model_year">
                        <p class="model_year field-name">Model Year</p>
                    </label>
                    <input type="number" id="model_year input" name="model_year">
                </div>

                <div class="field">
                    <label for="origin">
                        <p class="origin field-name">Origin</p>
                    </label>
                    <input type="number" id="origin input" name="origin">
                </div>

                <input type="submit" value="sumbit" class="submit-btn btn">
            </form>
        </div>
        <p class="result" id="result">{{y}} <span class="answer">{{z}}</span></p>
    </body>
</html>
```

main.css

```css
body {
    background-color: #E97777;
    padding: 0;
    margin: 0;
}

* {
    box-sizing: border-box;
}

header {
    width: 100%;
    background-color: #274472;
    margin: 0;
    height: 60px;
}

header h1 {
    margin: 0;
    position: absolute;
    top: 10px;
    left: 50px;
    color: #82CD47;
}

form {
    width: max-content;
    background-color: #82CD47;
    padding: 50px;
    border-radius:20px;
}

.field-name {
    padding: 10px 0;
    margin: 0;
    font-size: 20px;
    font-weight:bolder;
}

.field {
    padding: 10px 0;
}
```

```css
.form-container {
    display: flex;
    align-items: center;
    justify-content: center;
    height: 92vh;
    min-height:max-content;
}

.field input[type=number]{
    width:200px;
    background-color:#274472;
    font-size:20px;
    padding:5px 10px;
    color:white;
    border-radius:20px;
    border:none;
}

.submit-btn{
    font-size:20px;
    padding:10px 30px;
    background-color:#E97777;
    border-radius:20px;
    border:none;
    display:block;
    margin:20px auto;
}

.result{
    text-align:center;
    font-size:30px;
    font-weight:bolder;
    padding:20px;
    color:white;
}

.answer{
    color:#274472;
}
```

App.py

```python
from flask import Flask, render_template,request
#import pickle
import numpy as np

import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = ""
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}


app=Flask(__name__)
#model=pickle.load(open('RFregression.pkl','rb'))

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

@app.route('/model',methods=["GET","POST"])
def result():
    no_of_clynder=request.form["no_of_cylinders"]
    displacement=request.form["displacement"]
    horsepower=request.form["horsepower"]
    weight=request.form["weight"]
    acceleration=request.form["acceleration"]
    model_year=request.form["model_year"]
    origin=request.form["origin"]

    t1=[[int(no_of_clynder),float(displacement),int(horsepower),int(weight),float(acceleration),int(model_year),int(origin)]]

    # 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"]], "values": t1}]}

    response_scoring = requests.post('', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    prediction=response_scoring.json()["predictions"][0]["values"][0][0]
    print(prediction)

    return render_template("index.html",y="The predicted MPG of the vehicle is ", z=str(prediction))

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

Github Link

https://github.com/IBM-EPBL/IBM-Project-31806-1660205332