Team ID	PNT2022TMID31735
Project Name	Machine Learning Based Vehicle Performance Analyzer

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

[1] X. Yuan, L. Li, H. Gou, and T. Dong, “Energy and environmental impact of battery electric vehicle range in China,” Applied Energy, vol. 157, pp. 75–84, 2015.

[2] View at: Publisher Site | Google ScholarM. Ceraolo and G. Pede, “Techniques for estimating the residual range of an electric vehicle,” IEEE Transactions on Vehicular Technology, vol. 50, no. 1, pp. 109–115, 2001.

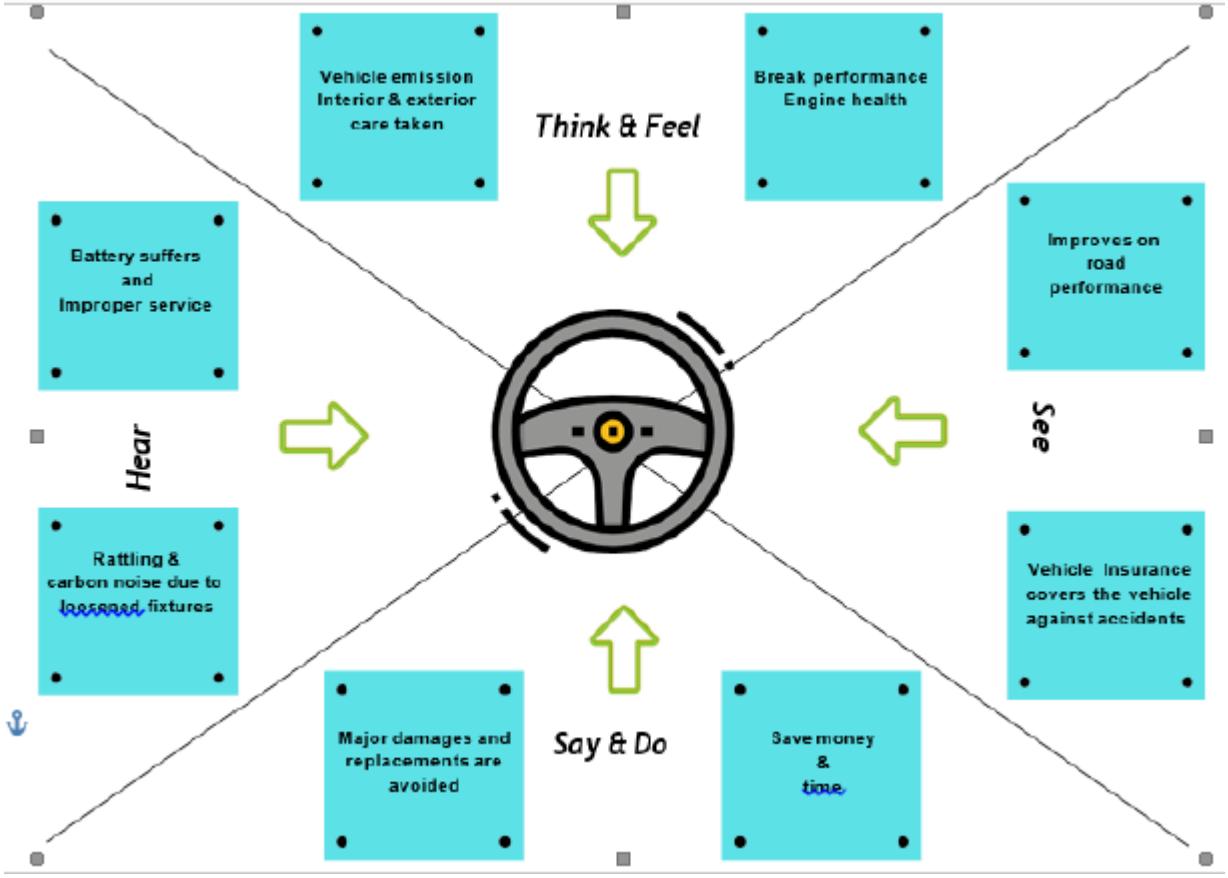
[3] View at: Publisher Site | Google ScholarA. Shekhar, V. Prasanth, P. Bauer, and M. Bolech, “Generic methodology for driving range estimation of electric vehicle with on-road charging,” in Proceedings of the IEEE Transportation Electrification Conference and Expo, ITEC '15, June 2015.

[4] View at: Google ScholarP. Ondruska and I. Posner, “Probabilistic attainability maps: Efficiently predicting driver-specific electric vehicle range,” in Proceedings of the 25th IEEE Intelligent Vehicles Symposium, IV '14, pp. 1169–1174, June 2014.

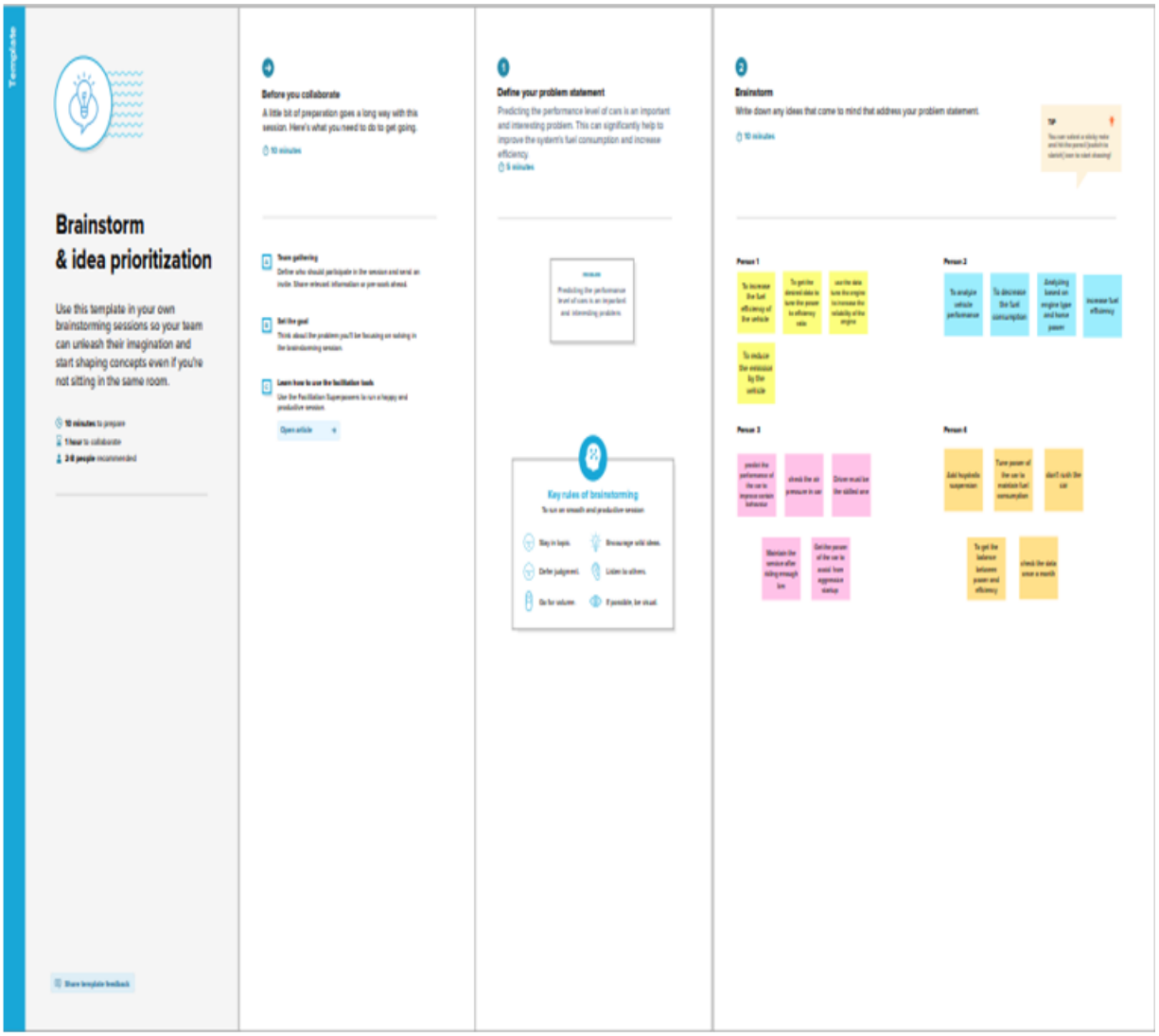
[5] View at: Google ScholarC. Bingham, C. Walsh, and S. Carroll, “Impact of driving characteristics on electric vehicle energy consumption and range,” IET Intelligent Transport Systems, vol. 6, no. 6, pp. 29–35, 2012.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2) Ideation & Brainstroming



Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, 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.

20 minutes

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

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The objective of this project is to predict the price of used cars using the various Machine Learning models by using User Interface (UI).
2.	Idea / Solution description	To train the system with the dataset using a regression model and it will be integrated to the web-based application where the user is notified with the status.
3.	Novelty / Uniqueness	By using the optimal regression model to predict the value in a less amount to time and predict its value.
4.	Social Impact / Customer Satisfaction	The customer can get an idea about the resale value of their vehicle to predict the performance. By knowing the vehicle brand, fuel type, kilometeres driven .
5.	Business Model (Revenue Model)	The web-based application has a friendly UI for the customer to enter their vehicles detail and the system predicts the value within few seconds.
6.	Scalability of the Solution	Machine learning approaches, this project proposed a scalable framework for predicting values for different type of used cars. The solution given by the trained system is efficient and is nearly accurate value of the vehicle.

3.4 Proposed Solution fit

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

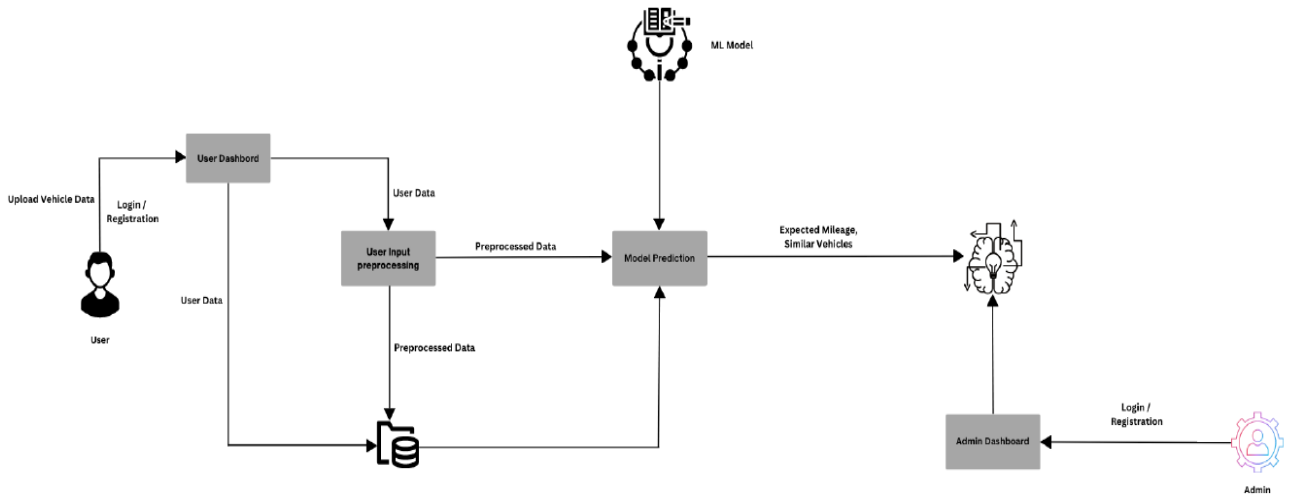
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Reset Password	Reset password through Gmail Reset password through Mobile number
FR-4	Feedback	The user can submit the feedback through a contact form in the website or through Gmail

4.2 Non-Functional requirements

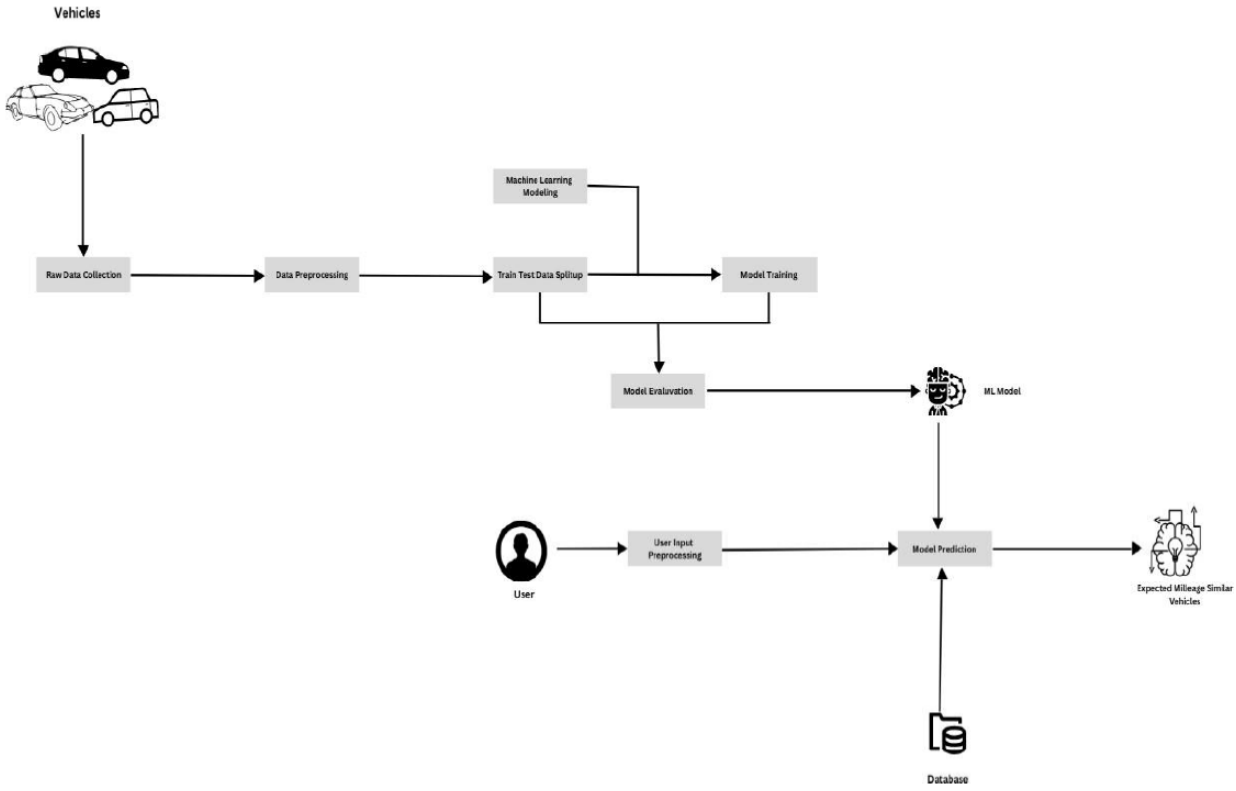
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The analyzer allows the user to improve performance based on the results provided. It is easy to use with just the data required
NFR-2	**Security**	The security is improved by using vehicle alarm, wheel lock, vehicle lock and also GPS tracker
NFR-3	**Reliability**	The reliability rating is good due to best performance, less frequency of problem occurrence and cost for repairing is low.
NFR-4	**Performance**	The vehicle is upgraded in their quality and infrastructure to provide better performance like good mileage, smooth travel
NFR-5	**Availability**	The data required is collected by research persons and this data can be used to provide better results.
NFR-6	**Scalability**	Better scalability since our model analyses all information provides better refined solution. With less change to the vehicle, we could achieve maximum performance.

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	Nisha T

Sprint-1	Design	USN-2	As a user, I can able to Enter the data of the vehicle	20	High	Leodas L
Sprint-2	Result	USN-3	As a user, I can get the predicted performance of the vehicle using the given data	20	High	Bavadharni T
Sprint-3	Design	USN-4	As a user, I want the good user experience.	10	Low	Nissan Nawaz SA
Sprint-3	Result	USN-5	As a user, I want the website is fast	10	Low	Samuel V
Sprint-4	Result	USN-6	As a user, I expect the prediction is highly accuracy.	10	Medium	Leodas L

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	25 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	08 Nov 2022	12 Nov 2022	20	13 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	20 Nov 2022

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…  from sklearn.metrics import r2_score,mean_squared_error

In [132…  acc = r2_score(y_test, y_pred)

In [133…  acc

Out[133…  0.8570363544939325

In [134…  err=np.sqrt(mean_squared_error(y_test,y_pred))

In [135…  err

Out[135…  2.7436940578959117
```

10. ADVANTAGES & DISADVANTAGES

Advantages:

- It made easy to predict the performance of the Vehicle
- It can accessible everyone 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 permanently
- In Program API Key is Publicly 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>Vehicle Performance Analyzer</title>
        <meta charset="utf-8">
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
        <header>
            <h1>Vehicle Performance Analyzer</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">
```

```
            <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>
```

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

        <input type="submit" value="Predict" class="submit-btn btn">
        <p class="result" id="result">{{Prediction}} <span class="answer">{{mpg}}</span></p>
    </form>
</div>


</body>
</html>
```

main.css

```
body {
    font-family: 'Times New Roman', Times, serif;
    background-image: url("bgp.jpg");
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center center;
    padding: 0;
    margin: 0;
}

* {
    box-sizing: border-box;
}

header {
    align-items: center;
    justify-content: center;
    width: 100%;
    height: 40px;
    display: flex;
}

h1 {
    position: absolute;
    text-align: center;
    top: 2px;
    color: #ffffff;
}

form {
    width: max-content;
    background-color: rgba(0, 0, 0, 0.5);
    padding-left: 50px;
    padding-right: 50px;
    border-radius:20px;
```

```
    padding-top: 10px;
}

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

.field {
    padding: 2px 0;
}

.form-container {
    font-family: 'Times New Roman', Times, serif;
    display: flex;
    align-items: center;
    justify-content: center;
    padding: 3vh;
    min-height:max-content;
    color: #ffffff;

}

.field input[type=number]{
    width:200px;
    background-color:#eeeef0;
    font-size:15px;
    padding:5px 10px;
    color:black;
    border-radius:20px;
    border:none;
```

```
}

.submit-btn{
    font-size:15px;
    padding:10px 30px;
    color:black;
    background-color:white;
    border-radius:20px;
    border:none;
    display:block;
    margin:20px auto;
    cursor: pointer;
}

.submit-btn:hover{
    color: white;
    background-color: black;
    border: none;
}

.result{
    text-align:center;
    font-size:20px;
    color:white;
    margin-top: 0;
    text-decoration: underline;
}

.answer{
    color:#ffffff;
    font-weight:bolder;
}
```

```
.makers{
    margin-top: 0%;
    text-align: center;
    color: white;
    font-size:15px;
}
```

App.py:

```
from flask import Flask, render_template,request
import pickle

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)]]
    output=model.predict(t1)
    return render_template("index.html",prediction="The predicted MPG of the vehicle is ", mpg=str(output[0]))

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

Github Link:

https://github.com/IBM-EPBL/IBM-Project-46950-1660795123