MACHINE LEARNING BASED VEHICLE PERFORMANCE ANALYZER

Bonafide record of work done by

FARHAAN N - **811519104029**

ARSHATHUL MOHAMED HAQ B - **811519104010**

GOKUL SS - **811519104033**

GOKUL RAJ R - **811519104035**

KEVIN PATRICK C - **811519104055**

INDUSTRY MENTOR : NIDHI FACULTY

MENTOR : GURU PRAKASHBACHELOR OF ENGINEERING

TEAM ID: PNT2022TMID10856

BRANCH: COMPUTER SCIENCE AND ENGINEERING

NOVEMBER 2022 DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF ENGINEERING (AUTONOMOUS INSTITUTION)

SAMAYAPURAM, TRICHY – 621 112

INDEX OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Project Overview

Predicting the performance level of cars is an important and interesting problem. The main goal is to predict the performance of the car to improve certain behaviours of the vehicle. 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 towards understanding the vehicle's performance.

1.2 Purpose

The use of Machine Learning techniques (supervised and unsupervised) on automotive engine sensor data to discover drivers' usage patterns, and to perform classification through a distributed online sensing platform. Such platforms can be used in different domains, such as fleet management, insurance market, fuel consumption optimization, CO2 emission reduction, amongothers. Thus, the main purpose of the project is to predict the performance of the car to improve certain behaviors of the vehicle using various machine learning algorithms.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing Problem

The potential for processing car sensing data has increased in recent years due to the development of new technologies. Having this type of data is important, for instance, to analyze the way drivers behave when sitting behind the steering wheel. Very little has been done to analyze car usage patterns based on car engine sensor data, and, therefore, it has not been explored to its full potential by considering all sensors within a car engine. Aiming to bridge this gap, the use of Machine Learning techniques (supervised and unsupervised) on automotive engine sensor data to discover drivers' usage patterns, and to perform classification through a distributed online sensing platform. Such platforms can be used in different domains, such as fleet management, insurance market, fuel consumption optimization, CO2 emission reduction, amongothers.

2.2 Problem Definition

Thus, by going through the existing problem and gaining knowledge from the various papers in the literature survey. We can frame the problem definition as follows:
"To predict the performance of the car to improve certain behaviors of the vehicle using various machine learning algorithms"

2.3 References

ML based Real-Time Vehicle Data Analysis for Safe Driving Modeling

In the paper “Machine Learning Based Real-Time Vehicle Data Analysis for Safe Driving Modeling” Machine learning approach to analyze and predict the vehicle performance in real time. The focus is on analyzing the data which is collected from the vehicle using the OBD-II scanner and eventually providing the driver’s safety solutions The meta features of the vehicle are analyzed in the cloud and are then shared to the concerned parties. The proposed system consists of an OBD-II scanner and a mini dash cam which continuously send data tothe cloud server where data analysis is done.

Random Forest

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected bymost trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees.

Decision Tree

The decision tree and its variants are the other learning algorithms that divide the input space into regions and has separate parameters for each region. They are classified as a non- parametric supervised learning method which is widely used in classification and regression, as well as in representing decisions and decision making. The structure of a decision tree is a flowchart, in which each internal node represents a “test” on

an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. Besides, the paths from root to leaf represent classification rules.

Naïve Bayes

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

Support Vector Mechanism

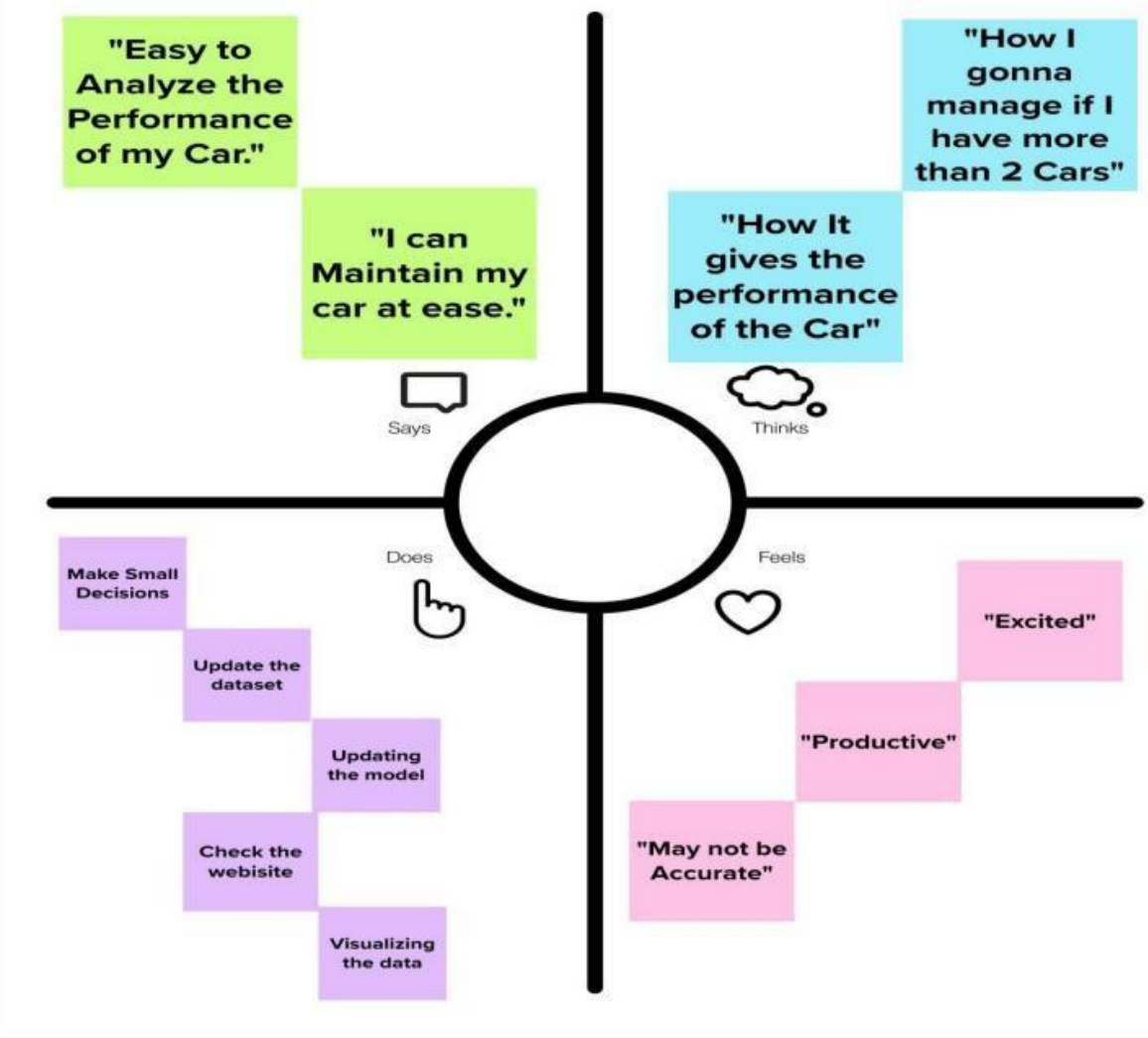
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms,which is used for Classification as well as Regression problems. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in thefuture. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called supportvectors, and hence the algorithm is termed as Support Vector Machine.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map

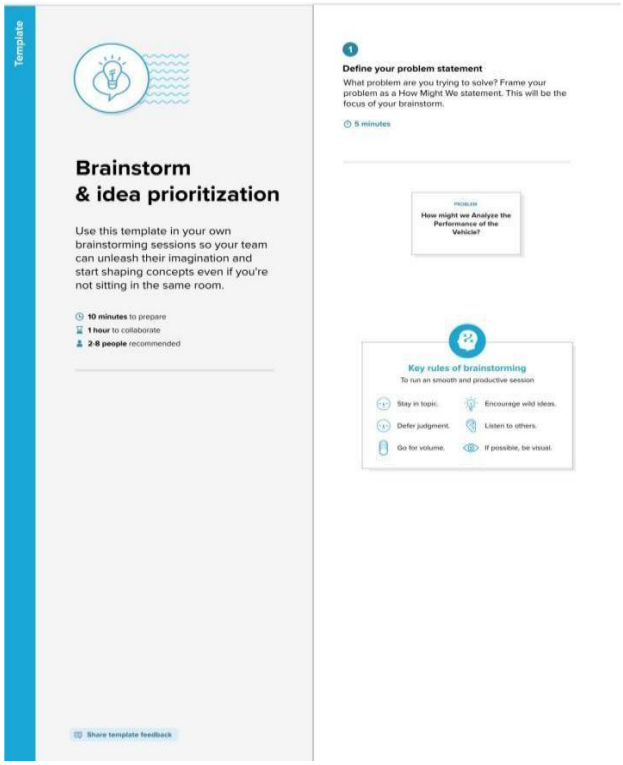
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



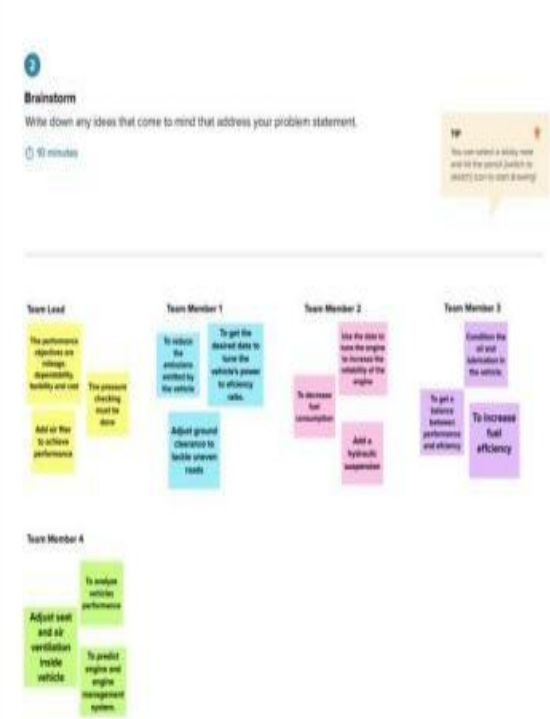
3.2 Ideation and Brainstorming

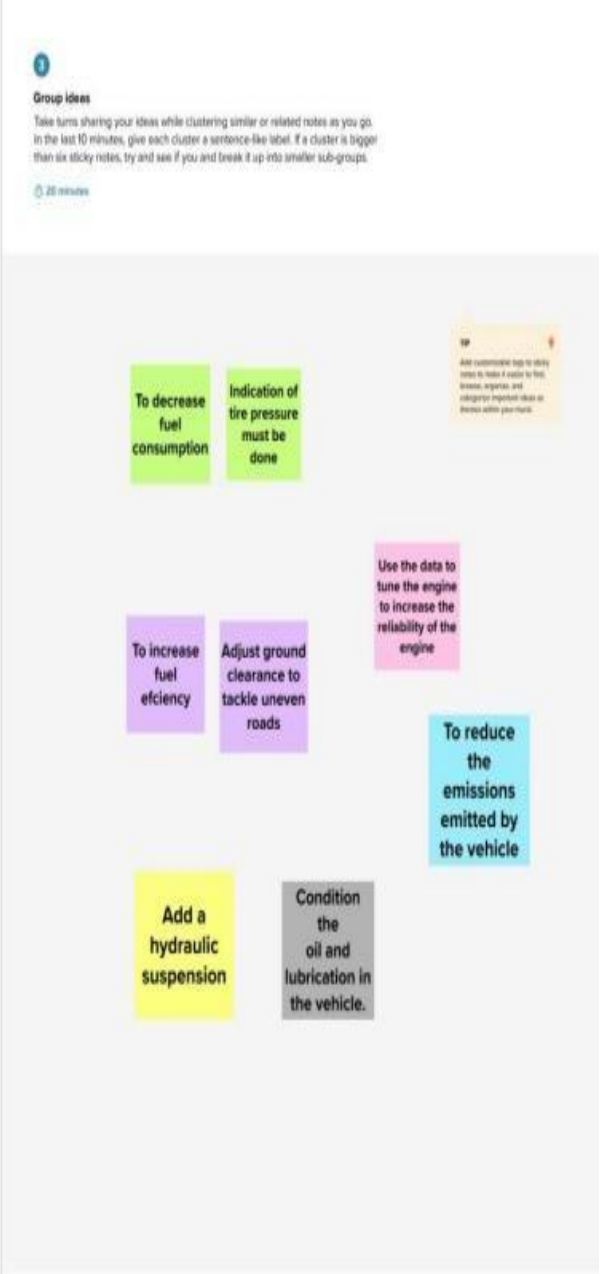
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. 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.

Step-1: Team Gathering, Collaboration and Select the Problem Statement

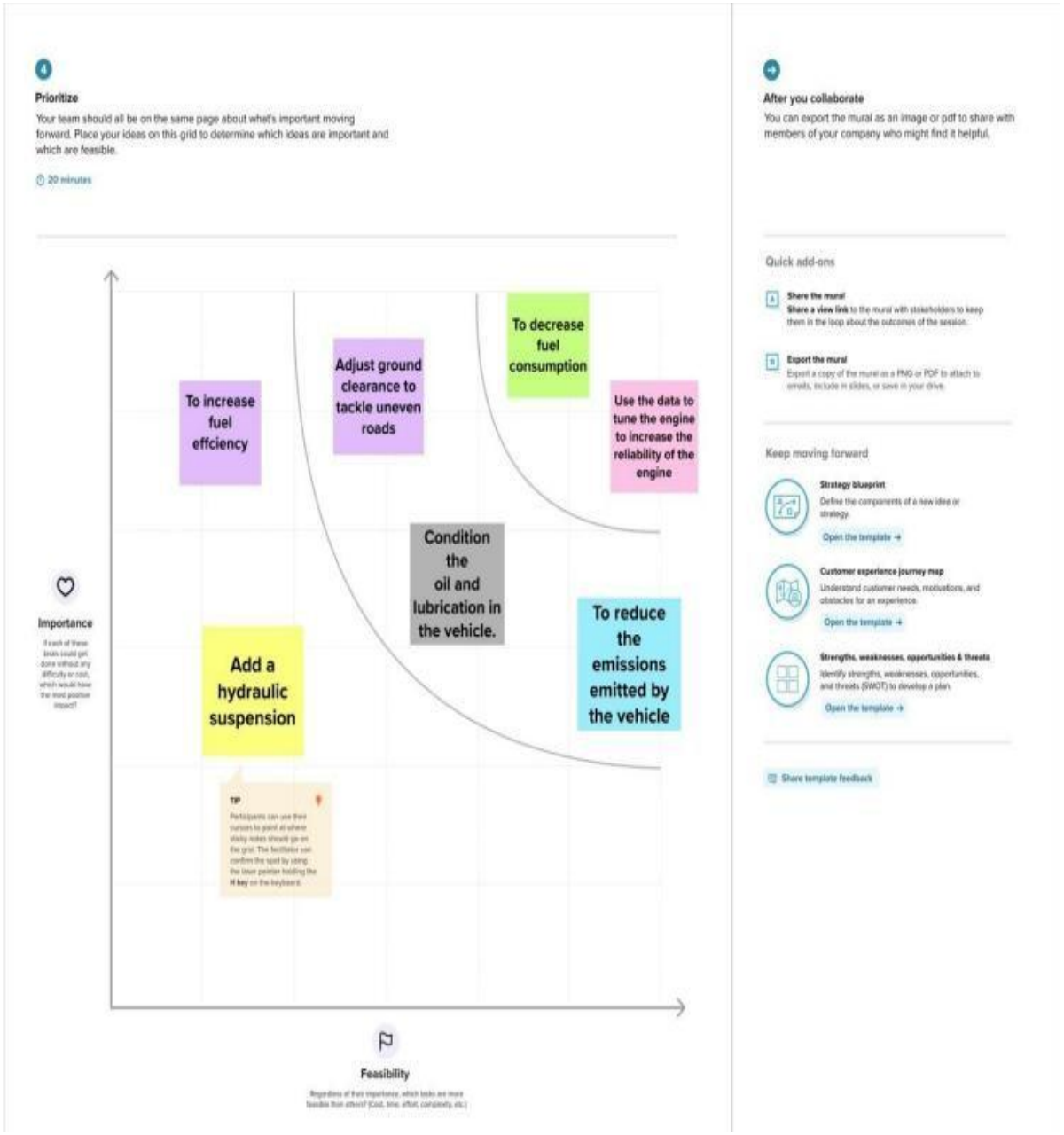


Step-2: Brainstorm, Idea Listing and Grouping





Step-3: Idea Prioritization



3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	**Machine Learning based Vehicle Performance analyzer** Predicting the performance level of cars is an important and interesting problem. The main goal is to predict the performance of the car to improve certain behaviors of the vehicle. The Performance of the vehicle denotes the mileage. 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 on-going 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 towards understanding the vehicle's performance.
2.	Idea/Solution description	They are various ideas to improve the Vehicle Performance. Analyzing these different aspects and qualities giving a general and at the same time a refined solution to improve the performance of the vehicle. To improve the mileage strength and efficiency and comfort, we have modified some parts and upgraded Some qualities to provide better performance.
3.	Novelty/Uniqueness	Unlike other Vehicle Performance Analyzers out there, Our analyzer focuses on perfecting the vehicle's performance, bring out it's full potential and improve on the possible areas.

4.	Social Impact/ Customer Satisfaction	Analyzing a vehicle's performance can benefit in many ways. One of the biggest benefit is that , the consumption of the fuel(petrol/diesel, etc) can be reduced in a drastic manner, which in-turn reduces the cost for the fuel and also reduces the emissions from the engine(exhaust gases).
5.	Business Model(Revenue Model)	This application aims to improve the performance of the vehicle and mainly, reduce the emissions. The main business model aims to get moderate profit and provide maximum performance.
6.	Scalability of the Solution	The main perks of this project is that, it can be hosted on bigger cloud platforms such as IBM Watson, etc. and can be accessed from across the globe.

3.4 Problem Solution Fit

The problem solution fit is the solution one has found to address the problem of the customer.

CHAPTER 4

REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Input	User Input have been inserted into the Webapp.

Non-functional Requirements:

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

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	**Reliability**	The reliability rating is good due to best performance, less frequency of problem occurrence and cost for repairing is low.
NFR-3	**Performance**	The vehicle is upgraded in their quality and infrastructure to provide better performance like good mileage, smooth travel due to good suspension and better engine performance.
NFR-4	**Availability**	The data required is collected by research persons and this data can be used to provide better results.
NFR-5	**Scalability**	Our project has better scalability since our model analyses all information provides better refined solution. With less change to the vehicle, we could achieve maximum performance.

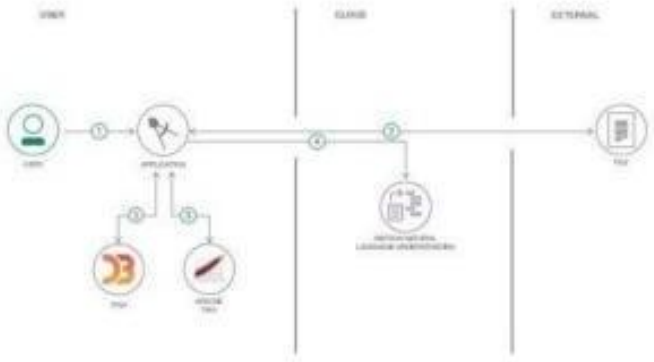
CHAPTER 5

PROJECT DESIGN

5.1 Dataflow Diagram

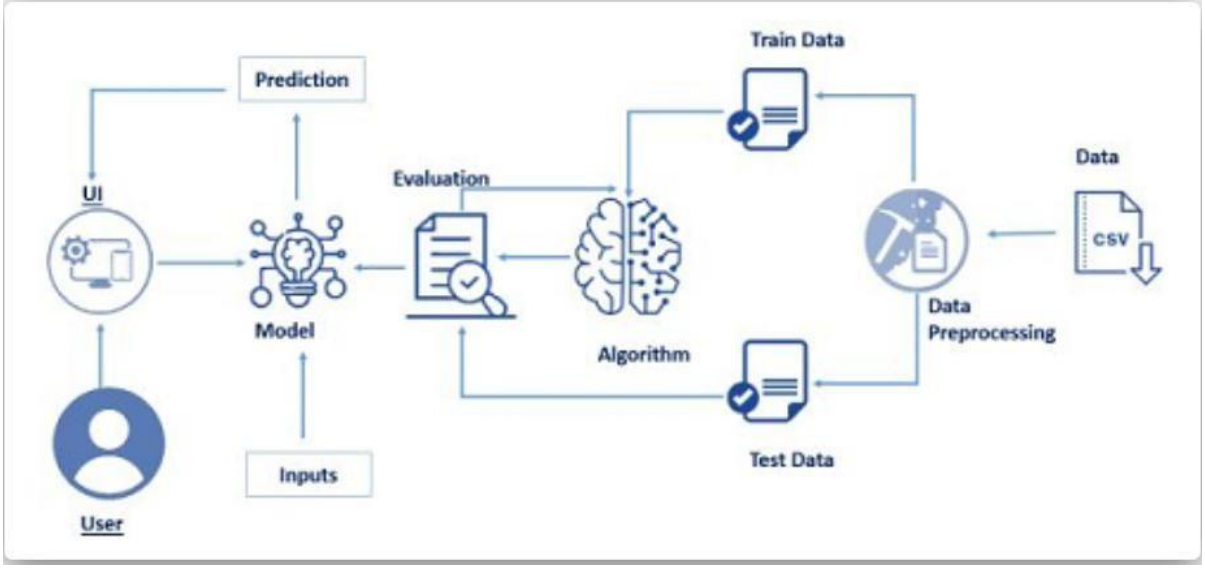
A Data Flow Diagram (DFD) is a traditional visual representation of how information flows within a system. A neat and clear DFD can thus depict the right amount of the system requirements graphically. It not only shows how data enters and leaves the system, but also what changes the information and where the data is stored.

Flow



1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.

5.2Technical Architecture



5.3 User Stories

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

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Preprocessing	USN-1	Collecting the Dataset, Handling Null values, Handling Irrelevant Data.	10	Low	C.Kevin Patrick
Sprint-2	Data Visualization	USN-2	Univariate Analysis, Bivariate Analysis, Multivariate Analysis.	10	Medium	Gokul Raj R
Sprint-3	Data Modeling and Evaluation	USN-3	Training the model with dataset and Evaluating the model.	20	High	Farhaan N
Sprint-4	Html Page for User interaction	USN-4	As a user, Input can be given in Index Page and Output shown in Output page.	10	Medium	Gokul S S
Sprint-5	Model Deployment	USN-5	Model is deployed in Flask and model is trained in IBM cloud and IBM watson	10	High	Arshathul Mohammad Haq B

CHAPTER 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 Preprocessing	USN-1	Collecting the Dataset, Handling Null values, Handling Irrelevant Data.	10	Low	C.Kevin Patrick
Sprint-2	Data Visualization	USN-2	Univariate Analysis, Bivariate Analysis, Multivariate Analysis.	10	Medium	Gokul Raj R
Sprint-3	Data Modeling and Evaluation	USN-3	Training the model with dataset and Evaluating the model.	20	High	Farhaan N
Sprint-4	Html Page for User interaction	USN-4	As a user, Input can be given in Index Page and Output shown in Output page.	10	Medium	Gokul S S
Sprint-5	Model Deployment	USN-5	Model is deployed in Flask and model is trained in IBM cloud and IBM watson	10	High	Arshathul Mohammad Haq B

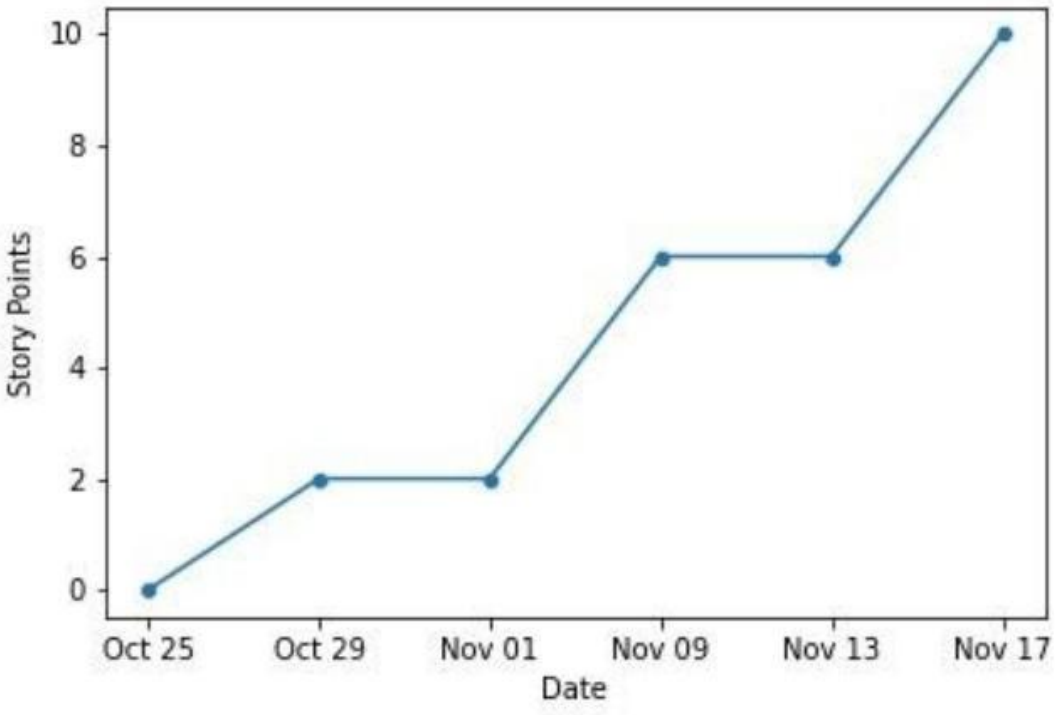
6.2 Sprint Delivery Schedule

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	10	6 Days	24 Oct 2022	31 Oct 2022	10	04 Nov 2022
Sprint-2	10	6 Days	31 Oct 2022	07 Nov 2022	10	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	14 Nov 2022	10	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	21 Nov 2022	5	19 Nov 2022
Sprint-5	10	6 Days	21 Nov 2022	28 Nov 2022	5	25 Nov 2022

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)

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



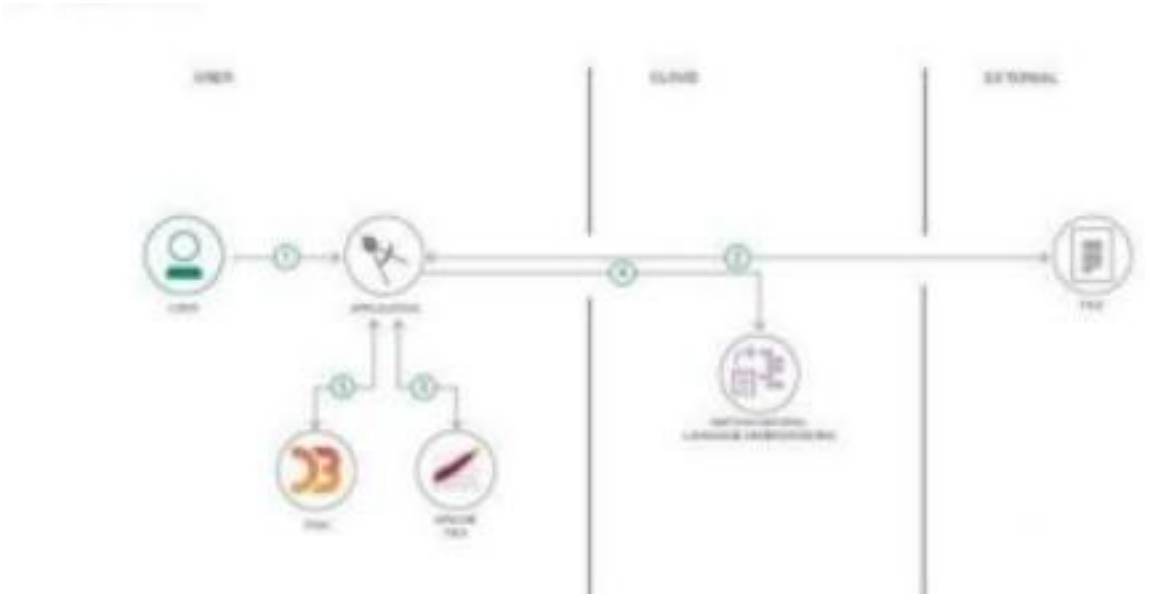
Burndown Chart : A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

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



CHAPTER 8

TESTING

8.1 Test Cases

				Date	03-Nov-22				
				Team ID	PNT2022TMID10856				
				Project Name	Project - Machine Learning based				
				Maximum Marks	4marks				
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
HomePage_TC_OO1	Functional	Home Page	Verify user is able to enter the data into the text field in the webpage and click the button.		1.Enter URL 2.Enter the values	[11 ,415,200,4000,20,72,2,"Ford"]	Page Refresh	Working as expected	Pass
HomePage_TC_OO2	Functional	Home Page	Verify if the user is able to view the output after the submit button has been clicked		1.Click the submit button		Vehicle Performance with mileage 21.1	Working as expected	Pass

8.2 User Acceptance Testing

1. PurposeofDocument

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. DefectAnalysis

Thisreportshowsthenumberofresolvedor closed bugs at each severity level, and how they were resolved

Resolution	Severity1	Severity2	Severity3	Severity4	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'tFix	0	0	0	0	0
Totals	4	2	1	1	13

3. TestCaseAnalysis

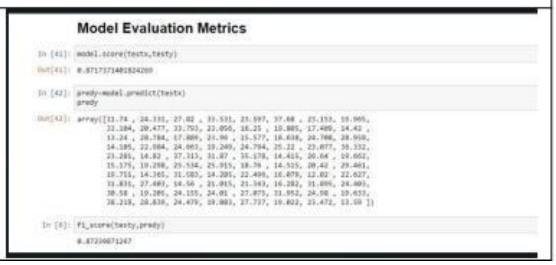
Thisreportshowsthenumberoftestcasesthathavepassed,failed,anduntested

Section	TotalCases	Not Tested	Fail	Pass
PrintEngine	4	0	0	4
ClientApplication	4	0	0	4
Security	1	0	0	1
OutsourceShipping	0	0	0	0
ExceptionReporting	1	0	0	1
FinalReportOutput	4	0	0	4
VersionControl	1	0	0	1

CHAPTER 9

RESULTS

9.1 Performance Metrics

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

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES :

- Using the Random Forest Algorithm in the model helps to perform both classification aswell as regression tasks.
- A random forest produces good predictions that can be easily understood
- It can handle large datasets easily
- Random Forest Algorithm provides a higher-level accuracy in predicting outcomes.

DISADVANTAGES :

- The main limitation of using random forest algorithm in the model is that a large numberof trees can make the algorithm too slow and ineffective for real-time predictions.
- The random forest algorithm is quite slow to create predictions once it is trained.
- It is very difficult to find the place for placing temperature sensor and pressure sensor.

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 the factors, like horsepower, cylinder count, fueltype, and engine type, the health of the car is forecast.

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 web page 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 SCOPE

The data_set used for this model is an old vehicle data_set, thus the model's accuracy would drop when the details of vehicles released in recent times are given as input.

Thus, in the future we propose to use the latest data_set set containing vehicle information to help train the model. We also plan to use other classification algorithms such as SVM and Decision Tress instead of Random Forest and measure if any accuracy gain occurs.

Finally, we propose to scale the machine learning model to also analyse the performance of a larger range of vehicles.

CHAPTER-13

Source Code

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=, initial-scale=1.0">
    <title>Car Performance Prediction</title>
    <style>
      html {
    height: 100%;
  }
  body
    { margin:0;
    padding:0;
    font-family: sans-serif;
    background: linear-gradient(#141e30, #243b55);
  }

  .login-box {
```

```
    position: absolute;
    top: 49%;
    left: 50%;
    width: 400px;
    padding: 40px;
    transform: translate(-50%, -50%);
    background: rgba(0,0,0,.5);
    box-sizing: border-box;
    box-shadow: 0 15px 25px rgba(0,0,0,.6);
    border-radius: 10px;
  }

  .login-box h2
    { margin: 0 0
    30px;
    padding: 0;
    color: #fff;
    text-align: center;
  }

  .login-box .user-box
    {position: relative;
  }
```

```
.login-box .user-box input
  {width: 100%;
  padding: 10px 0;
  font-size: 16px;
  color: #fff;
  margin-bottom: 30px;
  border: none;
  border-bottom: 1px solid #fff;
  outline: none;
  background: transparent;
}
.login-box .user-box select
  {width: 100%;
  padding: 10px 0;
  font-size: 16px;
  color: #03e8f489;
  margin-bottom: 30px;
  border: none;
  border-bottom: 1px solid #fff;
  outline: none;
  background: transparent;
}
.login-box .user-box select option{
```

```
    color: #03e8f489;
    background-color: #141e30;
  }

  .login-box .user-box label
    {position: absolute;
    top:0;
    left: 0;
    padding: 10px 0;
    font-size: 16px;
    color: #fff;
    pointer-events: none;
    transition: .5s;
  }
  .login-box .user-box select:focus ~ label,
  .login-box .user-box select:valid ~ label
    {top: -20px;
    left: 0;
    color: #03e8f489;
    font-size: 12px;
  }

  .login-box .user-box input:focus ~ label,
```

```
.login-box .user-box input:valid ~ label
  {top: -20px;
  left: 0;
  color: #03e9f4;
  font-size: 12px;
}

.login-box form button
  {position: relative;
  display: inline-block;
  padding: 10px 20px;
  color: #03e9f4;
  background-color: #141e30;
  font-size: 16px;
  text-decoration: none;
  text-transform: uppercase;
  overflow: hidden;
  transition: .5s;
  margin-top: 40px;
  letter-spacing: 4px
}

.login-box button:hover {
```

```
  background: #03e9f4;
  color: #fff;
  border-radius: 5px;
  box-shadow: 0 0 5px #03e9f4,
              0 0 25px #03e9f4,
              0 0 50px #03e9f4,
              0 0 100px #03e9f4;
}

.login-box button span
  {position: absolute;
  display: block;
}

.login-box button span:nth-child(1)
  {top: 0;
  left: -100%;
  width: 100%;
  height: 2px;
  background: linear-gradient(90deg, transparent, #03e9f4);
  animation: btn-anim1 1s linear infinite;
}
```

```
@keyframes btn-anim1
 {0% {
   left: -100%;
 }
 50%,100% {
   left: 100%;
 }
}

.login-box button span:nth-child(2)
 {top: -100%;
 right:  0;
 width: 2px;
 height: 100%;
 background: linear-gradient(180deg, transparent, #03e9f4);
 animation: btn-anim2 1s linear infinite;
 animation-delay: .25s
}

@keyframes btn-anim2
 {0% {
   top: -100%;
 }
```

```
  50%,100% {
    top: 100%;
  }
}

.login-box button span:nth-child(3)
  {bottom: 0;
  right: -100%;
  width: 100%;
  height: 2px;
  background: linear-gradient(270deg, transparent, #03e9f4);
  animation: btn-anim3 1s linear infinite;
  animation-delay: .5s
}

@keyframes btn-anim3
  {0% {
    right: -100%;
  }
  50%,100% {
    right: 100%;
  }
}
```

```
  .login-box button span:nth-child(4)
    {bottom: -100%;
    left: 0;
    width: 2px;
    height: 100%;
    background: linear-gradient(360deg, transparent, #03e9f4);
    animation: btn-anim4 1s linear infinite;
    animation-delay: .75s
  }

  @keyframes btn-anim4
    {0% {
      bottom: -100%;
    }
    50%,100% {
      bottom: 100%;
    }
  }

    </style>
</head>
<body>
```

```
<div class="login-box">
   <h2>CAR PERFORMANCE PREDICTION</h2>
   <form action="/getting" method="post">
    <div class="user-box">
     <input type="text" name="name" required="">
     <label>Enter Name of the car</label>
    </div>
    <div class="user-box">
     <select name="cylinder" id="cylinder">
         <option value="6">6</option>
         <option value="7">7</option>
         <option value="8">8</option>
     </select>
     <label>Select No. of Cylinder</label>
    </div>
    <div class="user-box">
     <input type="number" name="displacement" required="">
     <label>Enter Displacement</label>
    </div>
    <div class="user-box">
     <input type="number" name="hp" required="">
     <label>Enter Horse Power</label>
    </div>
```

```
<div class="user-box">
  <input type="number" name="weight" required="">
  <label>Enter Weight</label>
</div>
<div class="user-box">
  <input type="number" name="acceleration" required="">
  <label>
    Enter Acceleration
  </label>
</div>
<div class="user-box">
  <input type="number" name="year" required="">
  <label>
    Enter Model Year
  </label>
</div>
<div class="user-box">
  <select name="origin" id="origin">
      <option value="1">1</option>
      <option value="2">2</option>
      <option value="3">3</option>
  </select>
  <label>Select Orgin</label>
```

```
        </div>

        <button type="submit" href="/getting">
          <span></span>
          <span></span>
          <span></span>
          <span></span>
          Predict Performance
        </button>
      </form>
    </div>
</body>
</html>
```

OUTPUT.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=, initial-scale=1.0">
    <title>Car Performance Prediction</title>
    <style>
```

```
    html
  { height:
  100%;
}
body
  { margin:0;
  padding:0;
  font-family: sans-serif;
  background: linear-gradient(#141e30, #243b55);
}

.login-box
  { position:
  absolute;top: 49%;
  left: 50%;
  width: 400px;
  padding: 40px;
  transform: translate(-50%, -50%);
  background: rgba(0,0,0,.5);
  box-sizing: border-box;
  box-shadow: 0 15px 25px rgba(0,0,0,.6);
  border-radius: 10px;
}
```

```
.login-box h2
  { margin: 0 0
  30px;
  padding: 0;
  color: #fff;
  text-align: center;
}

.login-box .user-box
  {position: relative;
}


.login-box .user-box label
  {position: absolute;
  top:0;
  left: 0;
  padding: 10px 0;
  font-size: 16px;
  color: #fff;
  pointer-events: none;
  transition: .5s;
}
```

```
.login-box form button
 {position: relative;
 display: inline-block;
 padding: 10px 20px;
 color: #03e9f4;
 background-color: #141e30;
 font-size: 16px;
 text-decoration: none;
 text-transform: uppercase;
 overflow: hidden;
 transition: .5s;
 margin-top: 40px;
 letter-spacing: 4px
}

.login-box button:hover
 {background: #03e9f4;
 color: #fff;
 border-radius: 5px;
 box-shadow: 0 0 5px #03e9f4,
             0 0 25px #03e9f4,
             0 0 50px #03e9f4,
             0 0 100px #03e9f4;
```

```
}

.login-box button span
 {position: absolute;
 display: block;
}

.login-box button span:nth-child(1)
 {top: 0;
 left: -100%;
 width: 100%;
 height: 2px;
 background: linear-gradient(90deg, transparent, #03e9f4);
 animation: btn-anim1 1s linear infinite;
}

@keyframes btn-anim1
 {0% {
  left: -100%;
 }
 50%,100% {
  left: 100%;
 }
```

```
}

.login-box button span:nth-child(2)
 {top: -100%;
 right:  0;
 width: 2px;
 height: 100%;
 background: linear-gradient(180deg, transparent, #03e9f4);
 animation: btn-anim2 1s linear infinite;
 animation-delay: .25s
}

@keyframes btn-anim2
 {0% {
  top: -100%;
 }
 50%,100% {
  top: 100%;
 }
}

.login-box button span:nth-child(3)
 {bottom: 0;
```

```
    right: -100%;
    width: 100%;
    height: 2px;
    background: linear-gradient(270deg, transparent, #03e9f4);
    animation: btn-anim3 1s linear infinite;
    animation-delay: .5s
  }

  @keyframes btn-anim3
    {0% {
      right: -100%;
    }
    50%,100% {
      right: 100%;
    }
  }

  .login-box button span:nth-child(4)
    {bottom: -100%;
    left: 0;
    width: 2px;
    height: 100%;
    background: linear-gradient(360deg, transparent, #03e9f4);
```

```
    animation: btn-anim4 1s linear infinite;
    animation-delay: .75s
  }

  @keyframes btn-anim4
    {0% {
      bottom: -100%;
    }
    50%,100% {
      bottom: 100%;
    }
  }

    </style>
</head>
<body>
    <div class="login-box">
        <h2>CAR PERFORMANCE PREDICTION</h2>
        <form>
          <div class="user-box">
            <label>Your Mileage is {{pred}} kmpl</label>
          </div>
          <button href="#">
```

```
            <span></span>
            <span></span>
            <span></span>
            <span></span>
            Never let the future disturb you...
          </button>
        </form>
      </div>
</body>
</html>
```

```
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 =
"GgbMbG9DNUGFtfjqv_PkRSdpgOnNBonWDs7yKAp3SgRI"
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__)

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

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


    cB = request.form["cB"]
    cy = request.form["cylinder"]

disp = request.form["disp"]
    hP = request.form["hP"]
    weight =
request.form["W"]
    Acc = request.form["Acc"]
    mY =
request.form["mY"]
    origin = request.form["orgin"]

    t = [[11
, int(cy),int(disp),int(hP),int(weight),int(Acc),int(mY),int(origin)]]
    print(t)


# NOTE: manually define and pass the array(s) of values to be scored in the next line

payload_scoring = {"input_data": [{"field": ["car name" ,
"cylinders" , "displacement" ,"horsepower","weight" ,
"acceleration" ,"model year" ,"orgin"], "values":
t}]}

    response_scoring =
requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/2e34925b-a557-48a1-8d5e-b3ba
cfd6ded0/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]
```

```
    return render_template('index.html' ,
prediction_text=out)

if(_name_== "_main_"):
    app.run(debug = False)
```

```
'''



app = Flask(_name_,template_folder='templates')
@app.route('/')
def index():

return render_template('index.html')
@app.route('/output',methods=['POST'])
def output():

 name = request.form['name']
    name = name.split(' ')[0]
    temp =
pd.read_csv('Temp_file.csv')
    for i in range(len(temp["Brand"])):
        if
temp["Brand"].iloc[i] == name:
            name =
temp["Encoded"].iloc[i]
    cyl = request.form['cylinder']
    disp =
request.form['displacement']
    hp = request.form['hp']
    w = request.form['weight']

acc = request.form['acceleration']
    year = request.form['year']
    origin =
request.form['origin']

    payload_scoring = {"input_data":
[{"field": ["cylinders" , "displacement"
,"horsepower","weight" , "acceleration" ,"model year"
,"orgin","Brand"], "values":[[int(cyl) ,
int(disp),int(hp),int(w),int(acc),int(year),int(origin),int(name)]]}]}
    response_scoring =
requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/2e34925b-a557-48a1-8d5e-b3ba
cfd6ded0/predictions?version=2022-11-18', json=payload_scoring,
    headers={'Authorization':
'Bearer ' + mltoken})
    print("Scoring response")
    prediction =
response_scoring.json()
    return render_template('output.html' ,
pred=prediction['predictions'][0]['values'][0][0])
app.run()
```

Splitting the variables

In [ ]:
```
x = data.drop('mpg',axis=1)
y = data.mpg
```

Splitting the data

In [29]:
```
from sklearn.model_selection import train_test_split
```

In [30]:
```
trainx,testx,trainy,testy = train_test_split(x,y,test_size=0.2)
```

In [31]:
```
testx
```

Out[31]:

	cylinders	displacement	horsepower	weight	acceleration	model year	origin	Brand
25	8	360.0	215	4615	14.0	70	1	11
236	4	140.0	89	2755	15.8	77	1	11
55	4	97.0	60	1834	19.0	71	2	29
337	4	107.0	72	2290	17.0	80	3	13
173	4	119.0	97	2545	17.0	75	3	8
...	...	...	...	...	...	...	...	...
227	6	225.0	100	3630	17.7	77	1	22
357	4	119.0	100	2615	14.8	81	3	8
152	6	225.0	95	3264	16.0	75	1	22
170	4	140.0	78	2592	18.5	75	1	23
5	8	429.0	198	4341	10.0	70	1	11

80 rows × 8 columns

Model Training

In [4]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import f1_score
```

In [40]:
```
model = RandomForestRegressor().fit(trainx,trainy)
```

Model Evaluation Metrics

In [41]:
```
model.score(testx,testy)
```

Out[41]:
```
0.8717371401824269
```

In [42]:
```
predy=model.predict(testx)
predy
```

Out[42]:
```
array([11.74 , 24.331, 27.82 , 33.531, 23.597, 37.68 , 23.153, 19.965,
       33.104, 20.477, 33.793, 23.056, 16.25 , 19.805, 17.409, 14.43 ,
       13.24 , 28.784, 17.889, 23.96 , 15.577, 18.638, 24.708, 28.958,
       14.105, 22.984, 24.663, 19.249, 24.794, 25.22 , 23.077, 36.332,
       23.281, 14.82 , 37.313, 31.87 , 35.178, 14.415, 26.64 , 19.662,
       15.175, 19.298, 25.534, 25.915, 18.76 , 14.515, 20.42 , 29.461,
       19.751, 14.365, 31.583, 14.285, 22.496, 16.079, 12.02 , 22.627,
       31.831, 27.403, 14.56 , 21.015, 21.343, 16.282, 31.895, 24.403,
       30.58 , 19.205, 24.155, 24.01 , 27.075, 31.952, 24.98 , 19.633,
       38.218, 28.839, 24.479, 19.083, 27.737, 19.022, 23.472, 13.59 ])
```

In [6]:
```
f1_score(testy,predy)
```

```
0.87239871267
```

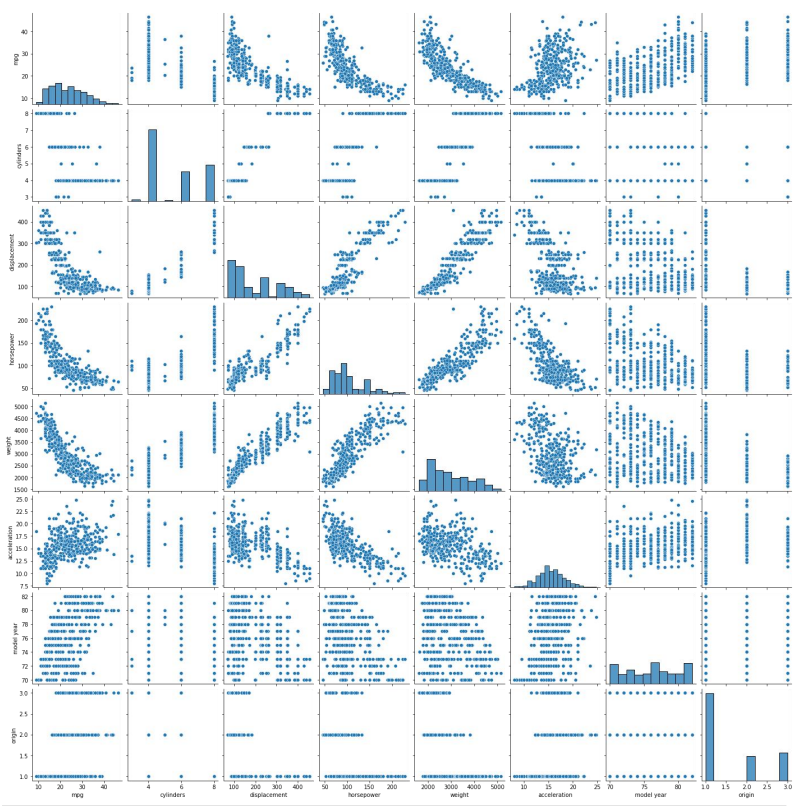
In [84]:
```
file = open('binary.pkl','wb')
pickle.dump(model,file)
```

In [85]:
```
file.close()
```

Visualizing the data

In [62]:
```
sns.pairplot(data)
```

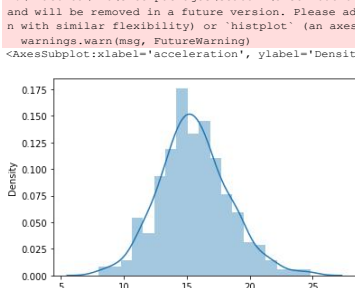
Out[62]:
```
<seaborn.axisgrid.PairGrid at 0x1db78f97a30>
```



In [63]:
```
sns.countplot(data.cylinders)
```

```
D:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keywor
d arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

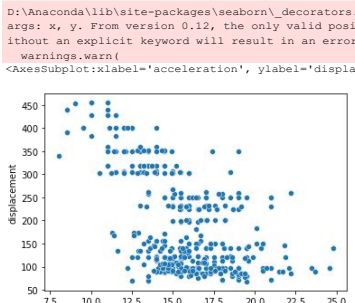
Out[63]:
```
<AxesSubplot:xlabel='cylinders', ylabel='count'>
```



In [64]:
```
sns.countplot(data["model year"])
```

```
D:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keywor
d arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

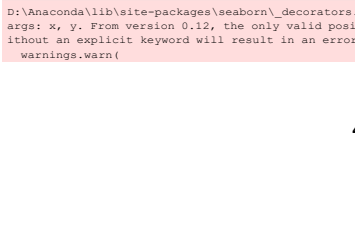
Out[64]:
```
<AxesSubplot:xlabel='model year', ylabel='count'>
```



In [65]:
```
sns.countplot(data.origin)
```

```
D:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keywor
d arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[65]:
```
<AxesSubplot:xlabel='origin', ylabel='count'>
```



In [66]:
```
sns.distplot(data.displacement)
```

```
D:\Anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function
and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functio
n with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[66]:
```
<AxesSubplot:xlabel='displacement', ylabel='Density'>
```


In [67]:
```
sns.distplot(data.horsepower)
```

```
D:\Anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function
and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functio
n with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[67]:
```
<AxesSubplot:xlabel='horsepower', ylabel='Density'>
```


In [68]:
```
sns.distplot(data.weight)
```

```
D:\Anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function
and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functio
n with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[68]:
```
<AxesSubplot:xlabel='weight', ylabel='Density'>
```


In [69]:
```
sns.distplot(data.acceleration)
```

```
D:\Anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function
and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functio
n with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[69]:
```
<AxesSubplot:xlabel='acceleration', ylabel='Density'>
```


In [70]:
```
sns.scatterplot(data.acceleration,data.displacement)
```

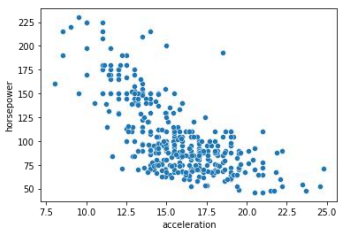
```
D:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword
args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments w
ithout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[70]:
```
<AxesSubplot:xlabel='acceleration', ylabel='displacement'>
```


In [71]:
```
sns.scatterplot(data.acceleration,data.horsepower)
```

```
D:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword
args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments w
ithout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
Out[71]: <AxesSubplot:xlabel='acceleration', ylabel='horsepower'>
```



Importing the libraries

In [8]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
```

Importing the data

In [9]:
```
data = pd.read_csv('car performance.csv')
```

In [10]:
```
data
```

Out[10]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino
...	...	...	...	...	...	...	...	...	...
393	27.0	4	140.0	86	2790	15.6	82	1	ford mustang gl
394	44.0	4	97.0	52	2130	24.6	82	2	vw pickup
395	32.0	4	135.0	84	2295	11.6	82	1	dodge rampage
396	28.0	4	120.0	79	2625	18.6	82	1	ford ranger
397	31.0	4	119.0	82	2720	19.4	82	1	chevy s-10

398 rows × 9 columns

In [11]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   mpg            398 non-null    float64
 1   cylinders      398 non-null    int64  
 2   displacement   398 non-null    float64
 3   horsepower     398 non-null    int64  
 4   weight         398 non-null    int64  
 5   acceleration   398 non-null    float64
 6   model year     398 non-null    int64  
 7   origin         398 non-null    int64  
 8   car name       398 non-null    object 
dtypes: float64(3), int64(5), object(1)
memory usage: 28.1+ KB
```

No Null Values Found

In [12]:
```
data.isnull().sum()
```

Out[12]:
```
mpg             0
cylinders       0
displacement    0
horsepower      0
weight          0
acceleration    0
model year      0
origin          0
car name        0
dtype: int64
```

In [152…
```
l = []
for i in data["car name"]:
    l.append(i.split(' ')[0])
```

In [153…
```
data.insert(9,"Brand",l)
```

Handling Irrelevent Values

In [154…
```
make_typo_correction = {
        'vw': 'volkswagen',
        'chevy': 'chevrolet',
        'maxda': 'mazda',
        'vokswagen': 'volkswagen',
        'toyouta': 'toyota',
        'chevroelt': 'chevrolet'
    }
data['Brand'] = data['Brand'].replace(make_typo_correction)
```

In [155…
```
data.Brand.unique()
```

Out[155]:
```
array(['chevrolet', 'buick', 'plymouth', 'amc', 'ford', 'pontiac',
       'dodge', 'toyota', 'datsun', 'volkswagen', 'peugeot', 'audi',
       'saab', 'bmw', 'hi', 'mercury', 'opel', 'fiat', 'oldsmobile',
       'chrysler', 'mazda', 'volvo', 'renault', 'honda', 'subaru',
       'capri', 'mercedes-benz', 'cadillac', 'mercedes', 'triumph',
       'nissan'], dtype=object)
```

In [156…
```
temp_file = pd.DataFrame(data.Brand.unique(),columns=["Brand"])
```

In [157…
```
data.drop('car name',axis=1,inplace=True)
```

Label Encoding

In [158…
```
from sklearn.preprocessing import LabelEncoder
```

In [160…
```
temp_file
```

Out[160]:

	Brand
0	chevrolet
1	buick
2	plymouth
3	amc
4	ford
5	pontiac
6	dodge
7	toyota
8	datsun
9	volkswagen
10	peugeot
11	audi
12	saab
13	bmw
14	hi
15	mercury
16	opel
17	fiat
18	oldsmobile
19	chrysler
20	mazda
21	volvo
22	renault
23	honda
24	subaru
25	capri
26	mercedes-benz
27	cadillac
28	mercedes
29	triumph
30	nissan

In [161…
```
data["Brand"] = LabelEncoder.fit_transform(data,data["Brand"])
```

```
D:\Anaconda\lib\site-packages\sklearn\preprocessing\_label.py:117: UserWarning: Pandas doesn't allow columns to be created via a new attribute name - see https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access
  self.classes_, y = _unique(y, return_inverse=True)
```

In [162…
```
temp_file["Encoded"] = pd.DataFrame(data["Brand"].unique())
```

In [163…
```
temp_file.to_csv('Temp_file.csv')
```

In [164…
```
temp_file
```

Out[164]:

	Brand	Encoded
0	chevrolet	6
1	buick	3
2	plymouth	22
3	amc	0
4	ford	11
5	pontiac	23
6	dodge	9
7	toyota	27
8	datsun	8
9	volkswagen	29
10	peugeot	21
11	audi	1
12	saab	25
13	bmw	2
14	hi	12
15	mercury	17
16	opel	20
17	fiat	10
18	oldsmobile	19
19	chrysler	7
20	mazda	14
21	volvo	30
22	renault	24
23	honda	13
24	subaru	26
25	capri	5
26	mercedes-benz	16
27	cadillac	4
28	mercedes	15
29	triumph	28

30	nissan	18

In [166…
```
def correlation(car, threshold):
    col_corr  =  set()
    corr_matrix = car.corr()
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if abs(corr_matrix.iloc[i,j]) > threshold:
                colname = corr_matrix.columns[i]
                col_corr.add(colname)
    return col_corr
```

In [168…
```
correlation(data,0.9)
```

Out[168]:
```
{'displacement', 'weight'}
```

Github link : https://github.com/IBM-EPBL/IBM-Project-8480-1658920560

Video Link : https://www.youtube.com/watch?v=FG42PDhAf3M