# TRIP BASED MODELING OF FUEL CONSUMPTION IN MODERN FLEET VEHICLES USING MACHINE LEARNING

Bonafide record of work done by

**Team ID: PNT2022TMID12710**

**DEEPTI RAVI KUMAR** - 718019Z210 (19Z210)
**K NARESH** - 718019Z221 (19Z221)
**KOUSIK NIBITH RAM V P** - 718019Z253 (19Z253)
**RAJESH G** - 718020Z432 (20Z432)

GUIDE: Dr. Saranya K G

## Professional Readiness for Innovation, Employability, and Entrepreneurship

**BACHELOR OF ENGINEERING**
**BRANCH:** COMPUTER SCIENCE AND ENGINEERING
PSG College of Technology

November 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

# Table of Contents

# Chapter 1

# INTRODUCTION

## 1.1 Project Overview

To predict the fuel consumption of fuel in fleet vehicles, which can help in improving the fuel economy and also can be useful in preventing fraudulent activities, by using Supervised Regression Machine Learning algorithms like linear, or Support Vector Regression to predict the consumption of fuel in fleet vehicles based on various parameters like fuel type, weather, etc., the result of which will be displayed to the user via a web application with the ML models integrated with it. The results provided by the application will help the customer to plan ahead for future trips and find the best fuel type and vehicle in the fleet, thus it can improve the fuel economy of the fleet and allows the user to find the most appropriate fuel type for any particular vehicle type. This can allow the entire fleet to save fuel and thus reduce greenhouse emissions by using the optimal amount of carbon-based fuels, which is currently a diminishing resource. The customer will be able to find the most efficient combination of vehicles and fuel types in the fleet, and thus will be able to save expenses. By predicting the expenses for future trips, the customer is then able to plan ahead and also can prevent fraudulent activities.

## 1.2 Purpose

The main purpose of this project is to predict the fuel consumption of fleet vehicles using Machine Learning algorithms to predict the consumption based on the gas type. A web application is built using flask which is integrated with the ML model to predict and then display the results. This can be used in many different domains by just using the appropriate dataset, for example, airlines, railways, and other vehicles. The model is trained to predict fuel consumption using the distance of a trip, the average speed during the trip, the temperature in the vehicle and engine, the type of fuel used, and the general weather during the trip.

# Chapter 2

# LITERATURE SURVEY

## 2.1 Existing Problem

Technological advancements have given rise to new types of fuel and more variations in fleet vehicles, engines, and other vital components. Though the data can be collected easily, not much is being done with this data on the consumer's side. It is extremely difficult to find the optimal type of fuel and the other optimal parameters. The full potential of the data collection capabilities of the various sensors available to the user has not been achieved. There is also the possibility of fraudulent activity regarding the consumption of fuel during a trip in order to acquire more allowances than they are actually supposed to. Thus, the user needs to be able to predict fuel consumption with certain parameters that would help them find the best type of fuel and prevent any malicious activities.

## 2.2 References

1. Influence of road and traffic conditions on fuel consumption and fuel cost for different bus technologies (Published Year 2017)
   By: Ivan S. Ivkovi, Snežana M. Kaplanovi, and Branko M. Milovanovi
   a. Numerous studies and experimental research confirmed that CNG and hybrid buses have significant environmental advantages compared to conventional diesel buses.
   b. The purpose of this research is to highlight the potential benefits of the application of energy-efficient vehicle technologies in intercity vehicle service.
   c. Potential benefits of the application of energy-efficient bus technologies in intercity bus service. Precisely, the focus is on the fuel consumption of diesel, hybrid, and compressed natural gas
   d. (CNG) buses according to various road and traffic conditions
   e. Fuel consumption of buses depends on many factors, including road type, speed, acceleration, road grade, load mass, air conditioning, driving style, etc.,
   f. There are numerous studies that confirm that speed is one of the most important factors which significantly affects fuel consumption.

2. Effects of driving style on the fuel consumption of city buses under different road conditions and vehicle masses (Published Year 2015)
   By: Hongjie Ma, Hui Xie, Denggao Huang, Shuo Xiong

a. The variance in fuel consumption caused by driving style (DS) difference exceeds 10% and reaches a maximum of 20% under different road conditions, even for experienced bus drivers.
b. To study the influence of DS on fuel consumption, a method for summarizing DS characteristic parameters on the basis of a vehicle-engine combined model is proposed.
c. The author proposes 26 DS characteristic parameters related to fuel consumption in the accelerating, normal running, and decelerating processes of vehicles.
d. This study also calculates the minimum sample size necessary for analyzing the effect of DS characteristics on fuel consumption.
e. The analysis results can be employed to evaluate the fuel consumption of drivers, as well as to guide the design of the Driver Advisory System for Eco-driving directly.

3. Fuel consumption and emission characteristics in an asymmetric twin-scroll turbocharged diesel engine with two exhaust gas recirculation circuits (Published Year 2019)
   By: Dengting Zhu, Xinquian Zheng
   a. This paper is the first known presentation of an asymmetric twin-scroll turbocharged engine with two exhaust gas recirculation circuits for emission and energy improvements.
   b. At the high-speed range, the turbine's larger scroll has an exhaust pressure that is higher than the intake pressure, leading to poor fuel economy.
   c. A test bench experiment was performed to validate numerical models of the asymmetric twin-scroll turbocharged engine with one and two exhaust gas recirculation circuits.
   d. Based on the models, both the influences of critical turbine parameters (turbine asymmetry, efficiency and throat area) on engine emission and fuel consumption characteristics, and the EGR valves and the wastegate control strategy were studied, and they were different from the asymmetric twin-scroll turbocharged engine with one exhaust gas recirculation circuit.
   e. The maximum exhaust gas recirculation rate and fuel economy improvements were approximately 8.59% and 1.98%.
   f. The new technology of the asymmetric twin-scroll turbocharged engine with two exhaust gas recirculation circuits described in this report has the potential to provide substantial gains in engine emission and energy.

## 2.3 Problem Definition

The problem statement based on the existing problem, and the various papers researched here in the literature survey can be defined as:
"To build a Machine Learning algorithm to predict the fuel consumption of fleet vehicles based on the gas type. A web application is built which is integrated with ML model"

# Chapter 3

# IDEATION & PROPOSED SOLUTION

## 3.1 Empathy Map Canvas

The empathy map provides the developer with a base to understand the users' pains and gains and develop a deep understanding of the user experiences. Figure 1 shows the empathy map for the Trip Based Modeling of Fuel Consumption in Modern Fleet Vehicles Using Machine Learning.

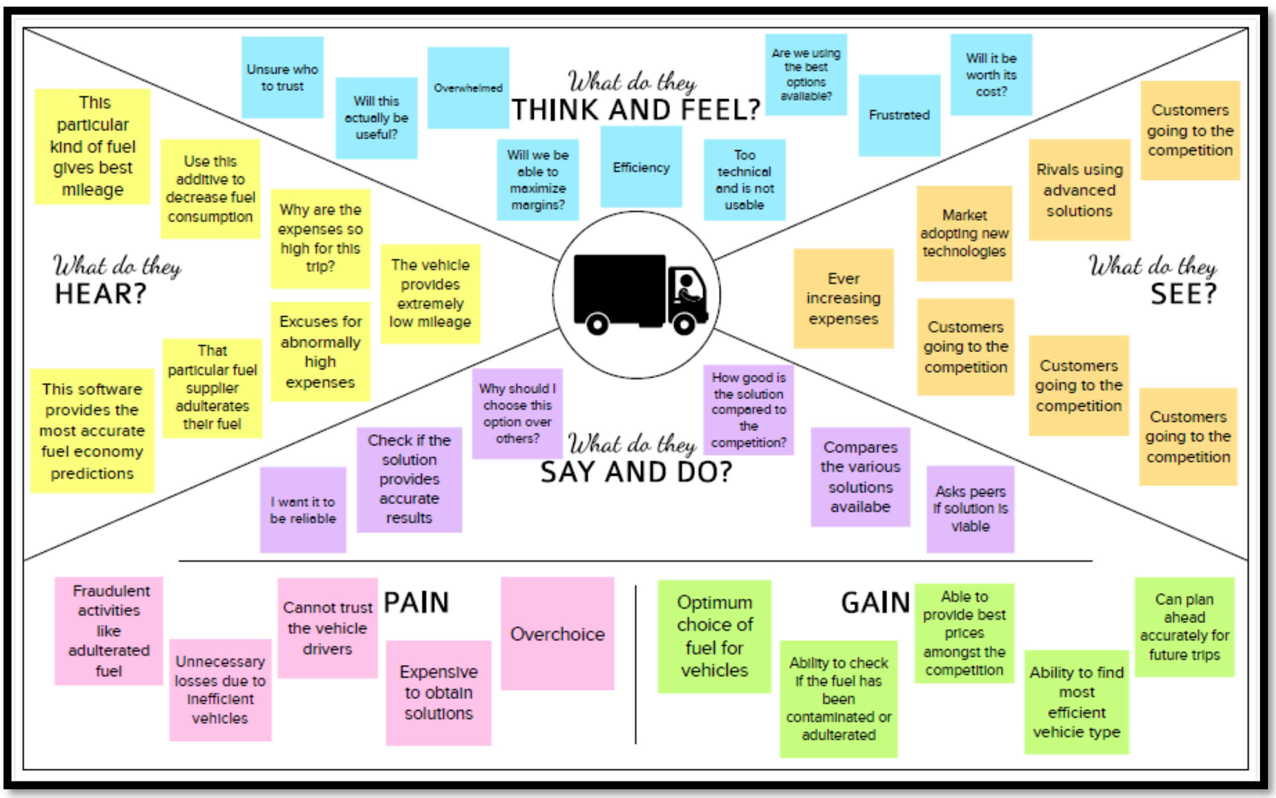



**Fig. 1 – Empathy Map**

## 3.2 Ideation & Brainstorming

In this phase of the project, the developers are to brainstorm and come up with a list of ideas, and then finding the most feasible and the ideas of highest importance and priority. The general idea in this stage is to develop a large number of ideas to solve the problem at hand, and then filter the best ideas and combine them to produce the most optimal solution. Figure 2 shows the various ideas of the team, and the priority given to each of these ideas in the solution to be developed.

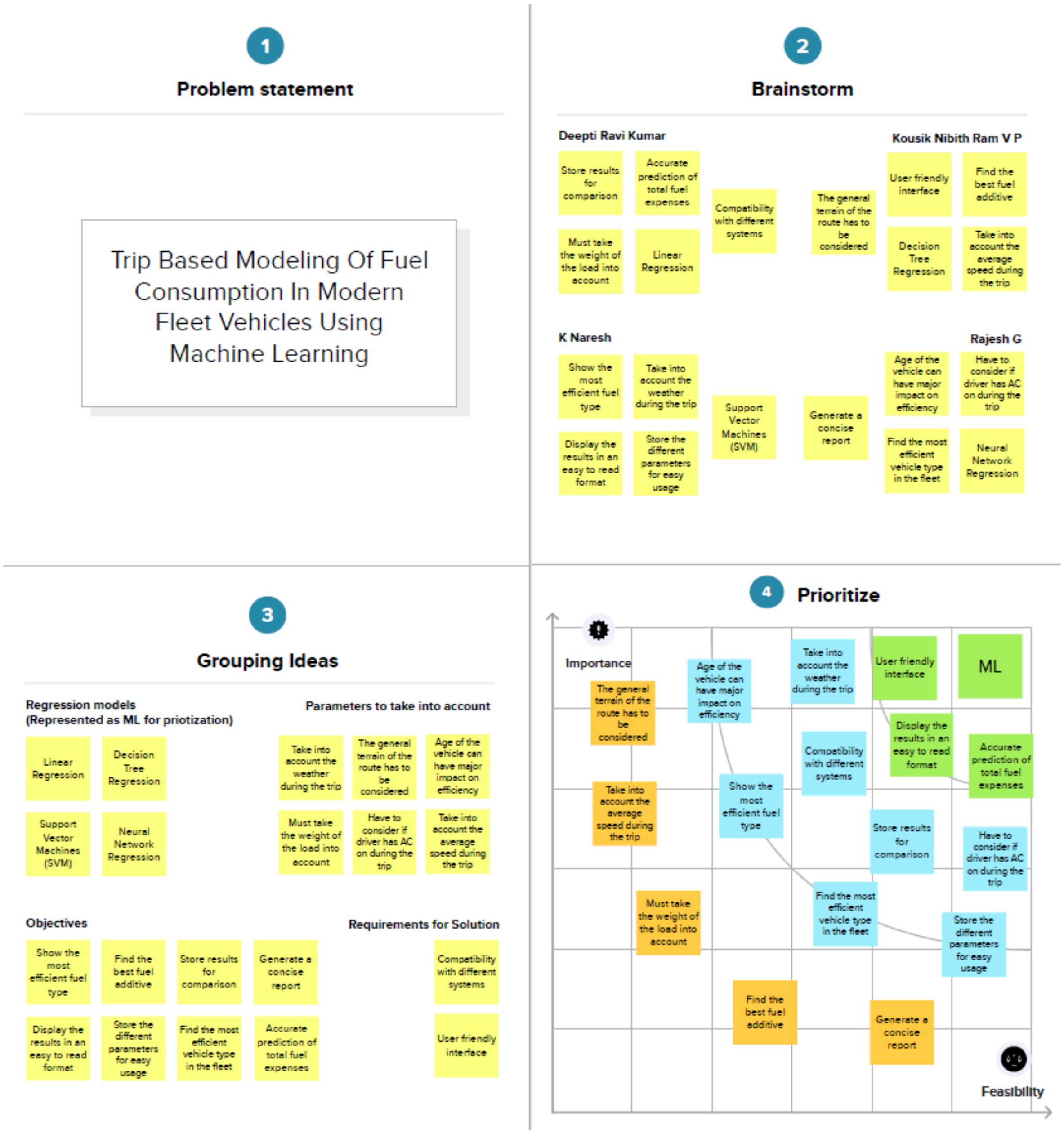



**Fig. 2 – Ideation and Brainstorming**

## 3.3 Proposed Solution

| S. No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | Predict the fuel consumption of fuel in fleet vehicles, which can help in improving the fuel economy and also can be useful in preventing fraudulent activities. |
| 2. | Idea / Solution description | Use Supervised Regression Machine Learning algorithms like linear, or Support Vector Regression to predict the consumption of fuel in fleet vehicles based on various parameters like fuel type, weather, etc.<br>The result of these algorithms will be displayed to the user via a web application with the ML models integrated with it. |
| 3. | Novelty / Uniqueness | Several machine learning models can be used to obtain prediction results.<br>These results can then be displayed to the user intuitively and easily. |
| 4. | Social Impact / Customer Satisfaction | The results provided by the application will help the customer to plan ahead for future trips and find the best fuel type and vehicle in the fleet, thus it can improve the fuel economy of the fleet and allows the user to find the most appropriate fuel type for any particular vehicle type.<br>This can allow the entire fleet to save fuel and thus reduce greenhouse emissions by using the optimal amount of carbon-based fuels, which is currently a diminishing resource. |
| 5. | Business Model / Financial Benefits | By using the application, the customer will be able to find the most efficient combination of vehicles and fuel types in the fleet, and thus will be able to save expenses.<br>By predicting the expenses for future trips, the customer is then able to plan ahead and also can prevent fraudulent activities. |
| 6. | Scalability of the Solution | The majority of this solution can also be applied to similar vehicles like trains and airplanes if the appropriate data is available.<br>The application is to be made as a web application and thus can be made available to any user. |

## 3.4 Problem Solution fit

The problem solution fit allows the developers to make sure that the solution that is being developed is actually a product that solves the given problem statement. It helps identify behavioral patterns and recognize what would work and why. It is a template to help identify solutions with higher chances of solution adoption, reduce time spent on testing and get a better overview of the current situation. Figure 3 shows the Problem Solution fit for the product being developed here.

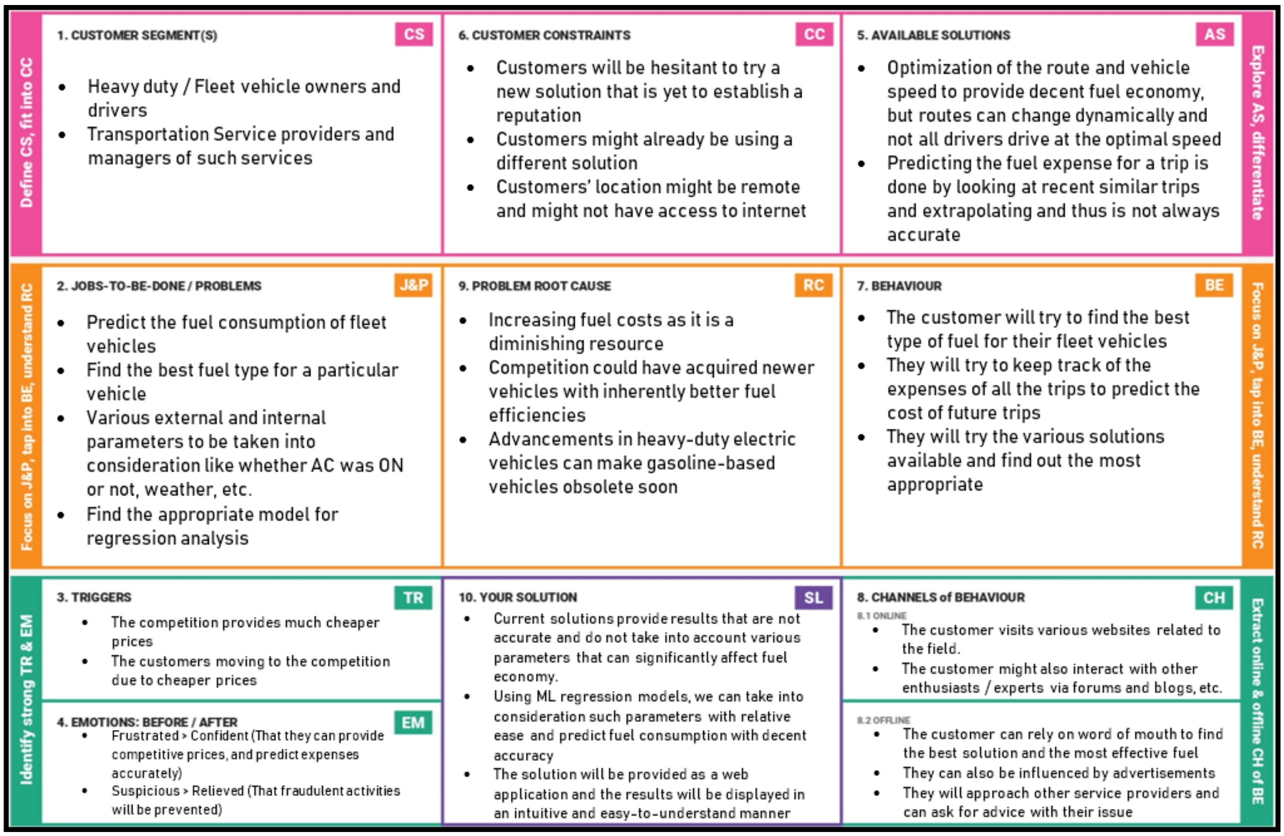



**Fig. 3 – Problem Solution Fit**

# Chapter 4

# REQUIREMENT ANALYSIS

## 4.1 Functional requirements

| FR No. | Functional Requirement | Sub Requirement |
|---|---|---|
| FR-1 | User Registration/Login | Via Email<br>Via Phone number |
| FR-2 | User Dashboard | Single Sample Prediction Multiple Sample Prediction Current trip status<br>View User History |
| FR-3 | Output Generation | Visual Representation<br>Report Generation |

## 4.2 Non-functional requirements

| NFR No. | Non-functional Requirements | Description |
|---|---|---|
| NFR-1 | Usability | A user-friendly interface that makes processing easier for the user Predictions are visually represented by the model. |
| NFR-2 | Security | Authentication - A user can have a private dashboard for secure access. |
| NFR-3 | Reliability | The model is able to run numerous samples simultaneously and handle massive amounts of data. |
| NFR-4 | Performance | The model's accuracy is good because it combines several ML methods. |
| NFR-5 | Availability | The website is also mobile-friendly and portable. It only requires basic configuration to run on any device. |
| NFR-6 | Scalability | It can be further expanded to offer API that third party organization like automakers, logistics firms, etc. can use. |

# Chapter 5
# PROJECT DESIGN

## 5.1 Data Flow Diagrams

A data flow diagram (DFD) is used to map out the flow of data in a system. It uses a defined set of symbols to represent the input, output, storage, etc., of data in any system. Figure 4 shows the Data Flow Diagram representing the system being developed here.

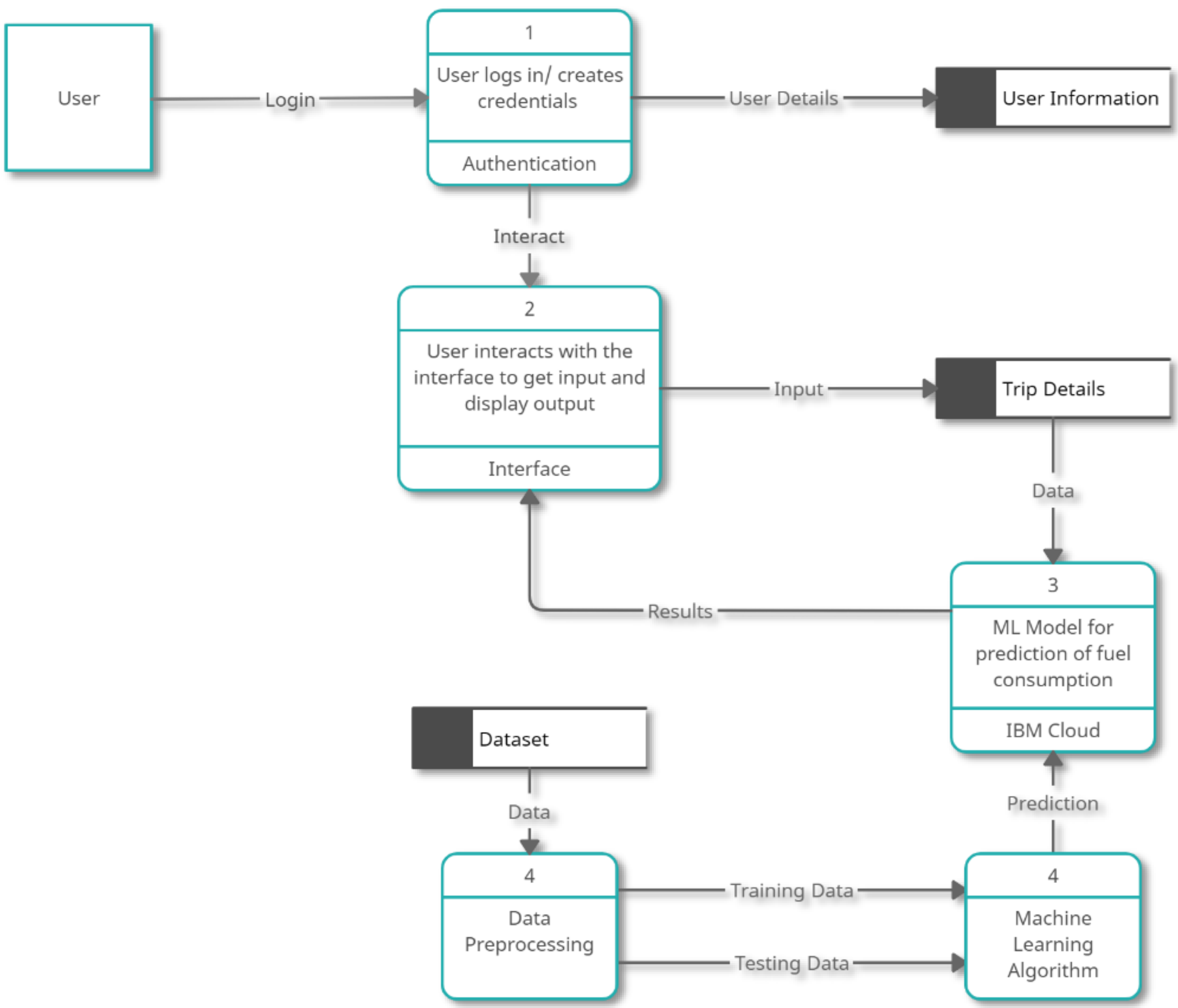



**Fig. 4 – Data Flow Diagram**

## 5.2 Solution & Technical Architecture

The solution architecture shows the rough draft of the system design of the solution. The technical architecture represents the flow between various technologies used here in this project. Figure 5 shows the solution architecture of the project and figure 6 shows the technical architecture.

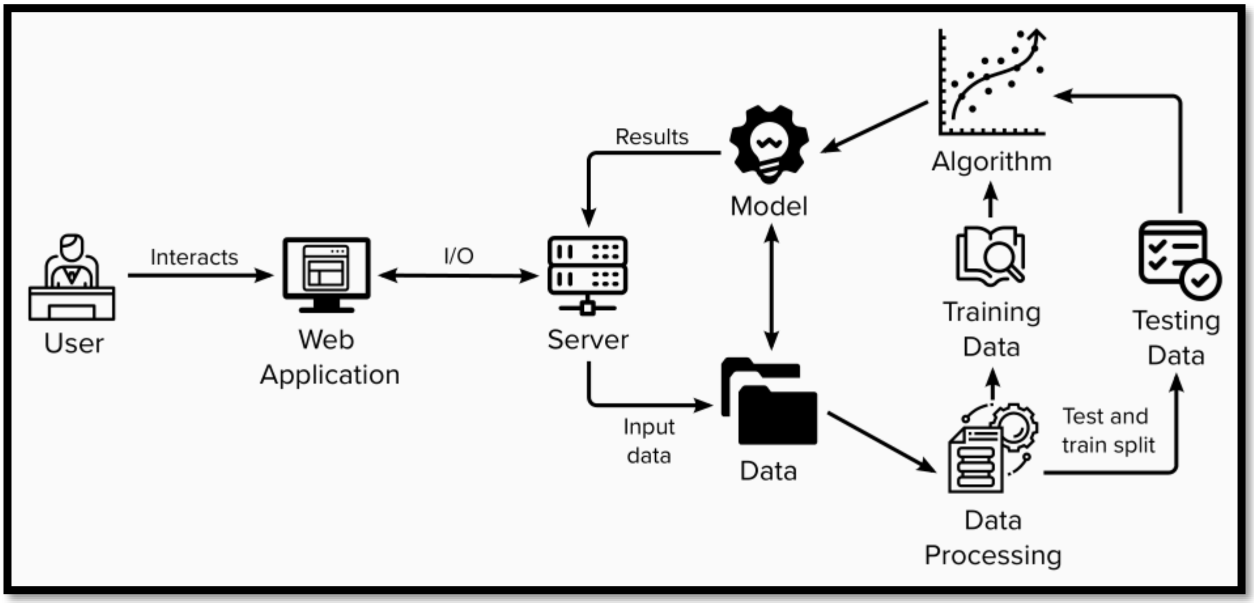



**Fig. 5 – Solution Architecture**

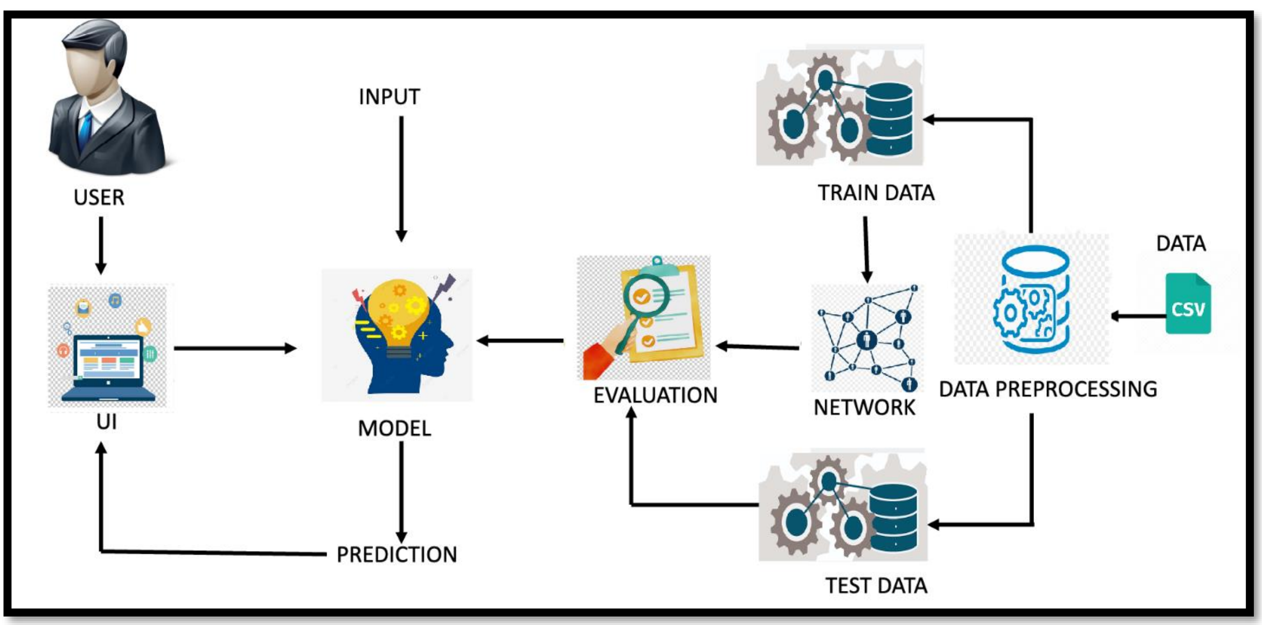



**Fig. 6 – Technology Architecture**

## 5.3 User Stories

A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer. In the following table, the various user stories are represented.

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Customer | Registration | USN-1 | As a user, I can register for the application by entering my email, and password, and confirming my password. | I can access my account/dashboard | High | Sprint-1 |
| | | USN-2 | As a user, I will receive a confirmation email once I have registered for the application | I can receive a confirmation email & click confirm | High | Sprint-1 |
| | Login | USN-3 | As a user, I can log into the application by entering my email & password | I can log in with my id and password | High | Sprint-1 |
| | User Interface | USN-4 | As a user, I can access the dashboard and provide input values for prediction | I can use the app and perform the functionality | High | Sprint-2 |
| | | USN-5 | As a user, I can perform multiple predictions at the same time | I can use predict the value for multiple trips at the same time | Medium | Sprint-3 |
| | | USN-6 | As a user, I can view the report for a particular trip easily | The report should be easily readable and not convoluted | High | Sprint-2,3 |
| | | USN-7 | As a user, I should be able to view all past trip details | I can view the prediction history | Medium | Sprint-3,4 |
| | | USN-8 | As a user, I should be able to have different graphics for visual representation | I can get visual representations of the result | Medium | Sprint-4 |
| | | USN-9 | As a user, I should be able to compare different trip details and results | I can compare the results of different trips | Medium | Sprint-4 |

# 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 | Registration | USN-1 | As a user, I can register for the application by entering my email, and password, and confirming my password. | 5 | High | Deepti Ravi Kumar |
| Sprint-1 | | USN-2 | As a user, I will receive a confirmation email once I have registered for the application | 3 | High | K Naresh |
| Sprint-1 | Login | USN-3 | As a user, I can log into the application by entering my email & password | 5 | High | Kousik Nibith Ram V P |
| Sprint-2 | User Interface | USN-4 | As a user, I can access the dashboard and provide input values for prediction | 5 | High | Rajesh G |
| Sprint-3 | | USN-5 | As a user, I can perform multiple predictions at the same time | 3 | Medium | Deepti Ravi Kumar |
| Sprint-2,3 | | USN-6 | As a user, I can view the report for a particular trip easily | 5 | High | Deepti Ravi Kumar, K Naresh, Kousik Nibith Ram V P, Rajesh G |
| Sprint-3,4 | | USN-7 | As a user, I should be able to view all past trip details | 5 | Medium | K Naresh |
| Sprint-4 | | USN-8 | As a user, I should be able to have different graphics for visual representation | 5 | Medium | Kousik Nibith Ram V P |
| Sprint-4 | | USN-9 | As a user, I should be able to compare different trip details and results | 3 | Medium | Rajesh G |

## 6.2 Sprint Delivery Schedule

| Sprint | Total Story Points | Duration | Sprint Start Date | Story Points Completed (as on Planned End Date) | Sprint Release Date |
|---|---|---|---|---|---|
| Sprint-1 | 13 | 6 Days | 24 Oct 2022 | 13 | 29 Oct 2022 |
| Sprint-2 | 10 | 6 Days | 31 Oct 2022 | 10 | 05 Nov 2022 |
| Sprint-3 | 13 | 6 Days | 07 Nov 2022 | 13 | 12 Nov 2022 |
| Sprint-4 | 8 | 6 Days | 14 Nov 2022 | 8 | 19 Nov 2022 |

## 6.3 Reports from JIRA

The average velocity of the team per iterative unit can be calculated using the following formula. Assume we have a 6-day sprint duration, and the velocity of the team is 30,

***Average Velocity (AV) = Sprint Duration / Velocity = 30 / 6 = 5***

The following graph, Figure 7, represents the burndown chart, which represents the amount of work left versus the amount of time required to complete the project. The burndown chart can be developed if the project consists of some measurable progress over a certain unit of time.

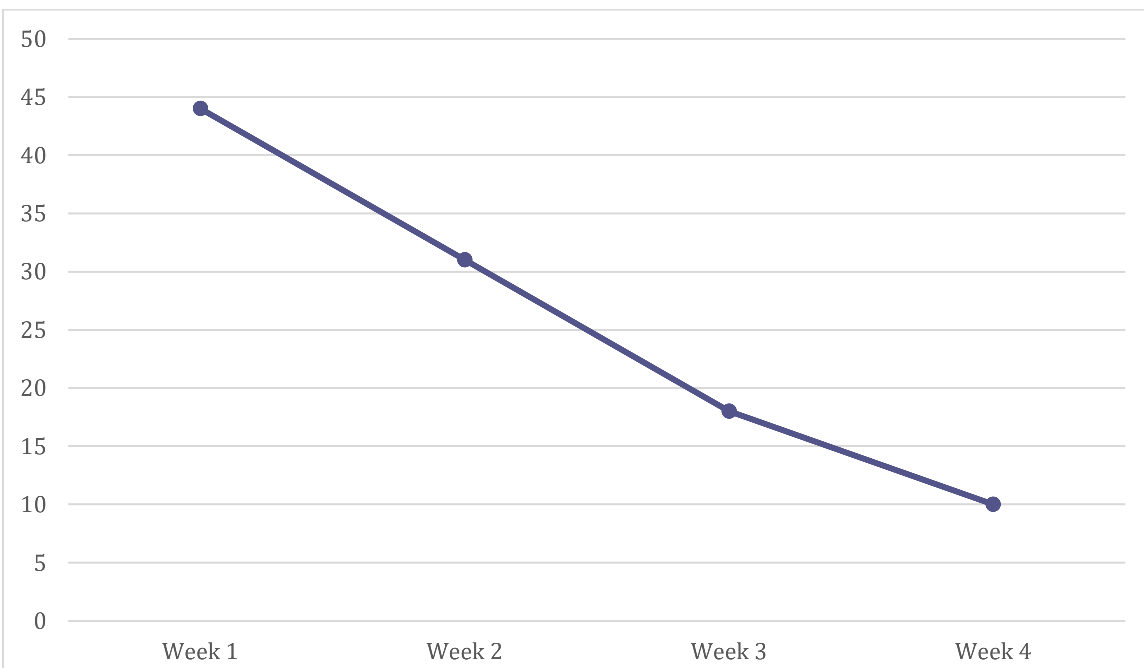



**Fig. 7 – Technology Architecture**

# Chapter 7

# CODING & SOLUTIONING

## 7.1 Feature 1: Front End

```
<!DOCTYPE html>
<html>
<head>
    <title>CarFuelConsumption</title>
    <meta charset="utf-8">
    <link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>
    <style>
        * {
            box-sizing: border-box;
        }
        body {
            font-family: 'Montserrat';
            text-align: center;
        }
        .title {
            text-align: center;
            color: #34568B;
        }
        table,
        th,
```

```
        td {
          border: 1px solid;
        }
        input {
          width: 30%;
          padding: 12px 20px;
          margin: 8px 0;
          box-sizing: border-box;
          border: 2px solid black;
          border-radius: 4px;
          color: white;
        }
        .btn {
          padding: 15px 32px;
          text-align: center;
          text-decoration: none;
          display: inline-block;
          font-size: 16px;
          background-color: #34568B;
          color: #ffffff;
          border-color: #34568B;
        }
      </style>
    </head>
```

```
<body>
  <div class="title">
    <h1>Car Fuel Consumption Prediction</h1>
  </div>
  <div class="login">
    <form action="{{url_for('predict')}}" method="post">
      <label for="distance">Distance</label><br>
      <input type="text" id="distance" placeholder="Distance" name="distance"><br>

      <label for="speed">Speed</label><br>
      <input type="text" id="speed" placeholder="Speed" name="speed"><br>

      <label for="temp_inside">Temperature Inside</label><br>
      <input type="text" id="temp_inside" placeholder="Temperature Inside"
name="temp_inside"><br>

      <label for="temp_outside">Temperature Outside</label><br>
      <input type="text" id="temp_outside" placeholder="Temperature Outside"
name="temp_outside"><br>

      <label for="AC">AC</label><br>
      <input type="text" id="AC" placeholder="AC" name="AC"><br>

      <label for="rain">Rain</label><br>
```

```
            <input type="text" id="rain" placeholder="Rain" name="rain"><br>


            <label for="sun">Sun</label><br>
            <input type="text" id="sun" placeholder="Sun" name="sun"><br>


            <label for="gas_types">Gas Types</label><br>
            <input type="text" id="gas_types" placeholder="Gas Types" name="gas_types"><br>


            <br>
            <button type="submit" class="btn" align="center">Predict</button>
        </form>
        <br>
        {{ prediction_text}}
    </div>
</body>
</html>
```

## Flask Server Code:

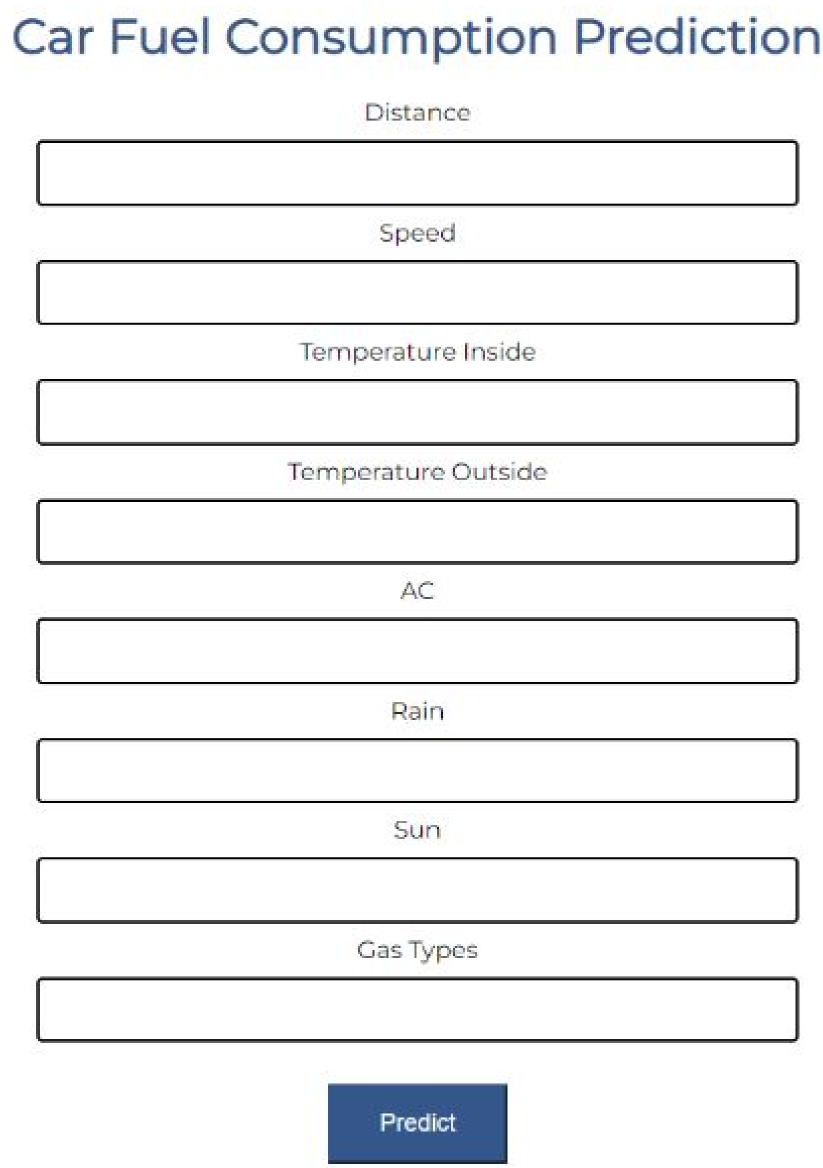
```
import numpy as np
from flask import Flask,jsonify,request,render_template
import pickle


app = Flask(_name_)
model=pickle.load(open('model.pkl','rb'))
```

```
@app.route('/')
def Home():
    return render_template('Manual_predict.html')


@app.route('/predict', methods=['POST'])
def predict():
    float_features=[float(x) for x in request.form.values()]
    features=[np.array(float_features)]
    prediction = model.predict(features)
    return render_template("Manual_predict.html", prediction_text = "Car Fuel
Consumption(L/100km) : {}".format(prediction))


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


Car Fuel Consumption Prediction
Distance
Speed
Temperature Inside
Temperature Outside
AC
Rain
Sun
Gas Types
Predict

# Chapter 8

# TESTING

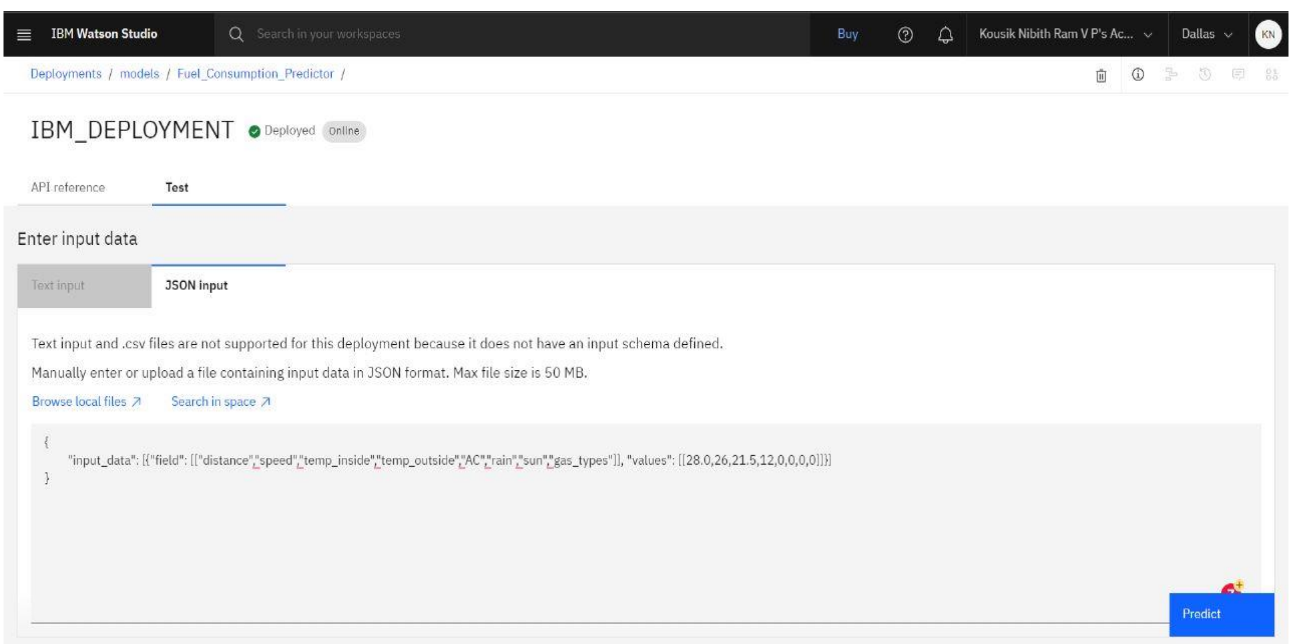
## 8.1 Test Cases

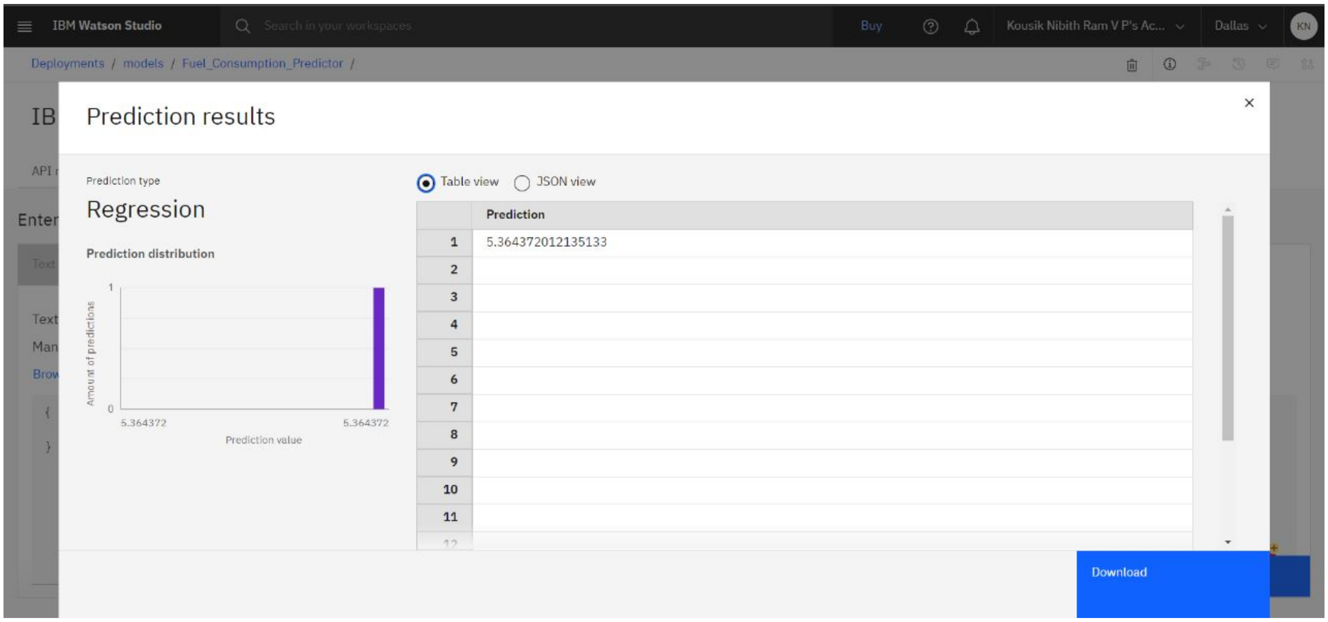
The below mention chart conveys the user test cases and its details of the test cases

| Steps To Execute | Test Data | Expected Result | Actual Result | Status |
|---|---|---|---|---|
| 1)Run the Fuel Consumption Prediction page | http://127.0.0.1:5000 | Local hosted the Web app Successfully | Working as expected | Pass |
| 2)Enter the values to be predicted | Enter the values of distance, Speed, Temp_out, Gas type, rain and sun | Predicted Successfully | Working as expected | Pass |
| 3)Predicted value displaying(output) | Value of fuel consumption litre (output) | The value Fuel consumption came Successfully | Working as expected | Pass |

## 8.2 User Acceptance Testing

The modelling structure will be similar to the traditional four-stage Travel Demand Forecast Model (TDFM) for the requirements of this study area. Traffic Analysis Zones (TAZs) contains 15 internal zones and 8 external zones, but Traffic Analysis Districts (TADs) contain 157 internal zones. Demography observation and trip's survey were estimated with data from the Shiraz city; centre of Fars province, Iran country, 1999 Sharif university data, and travel observation data has been collected (2009). Each individual TAZ is divided to about 10 districts based on road's boundaries, which has the close equivalency factor for contained districts. All the statistics and demographic data controlled and tested as well as both zonal layers. The main advantage of this method is that it can provide the opportunity to consider more concentrated centroids compared to just 15 centroids in TAZs, furthermore, implementing this type of model, precise identification of different zonal applications (residential, industrial, public and private) can be easily feasible. On the other hand, one of the evident disadvantages of this type of modelling is that considering the excessive number of attributes can cause increasing the time, effort and money. This model is essentially a conventional four-stage model with a number of additional sub models. Fuel price and socio-economic conditions affect trip generation. Fuel consumption shows the passing distance to reach a location and intensity of the activity, and socio-economic factors determine the magnitude and extent of population activity. Author developed transportation networks via ArcGIS9.2 & ArcGIS9.3 software and travel demand models are developed by using CUBE5 and TransCAD4.5 transportation planning software packages in GIS platform. All these software packages are implemented at the Transportation System Engineering (TSE) laboratory with the license purchased by the Indian Institute of Technology Bombay. The methodology used for planning travel demand model and its application for evaluating appropriate land use and sustainability of the model can be explained in following steps: 1. Generation and Creation of network for case study and define all necessary attributes. 2. Define all characteristics and attribute of network for travel demand model generation. 3. Generation and Validation of base year OD Matrices. 4. Development and Calibration of Travel Demand Model. 5. Model Application. 6. Fuel Indices and Evaluation of Alternatives. 2.1. Base year OD matrix generation This process starts with the last 15 years Home Interview Survey (HIS) data as input. HIS obtained sample size magnitude equal to 4.3% of the number of households, i.e. to select one sample out of 23 households(literature standards mentioned that for more than 1 million population cities, it could be more than 4%sample size magnitude)

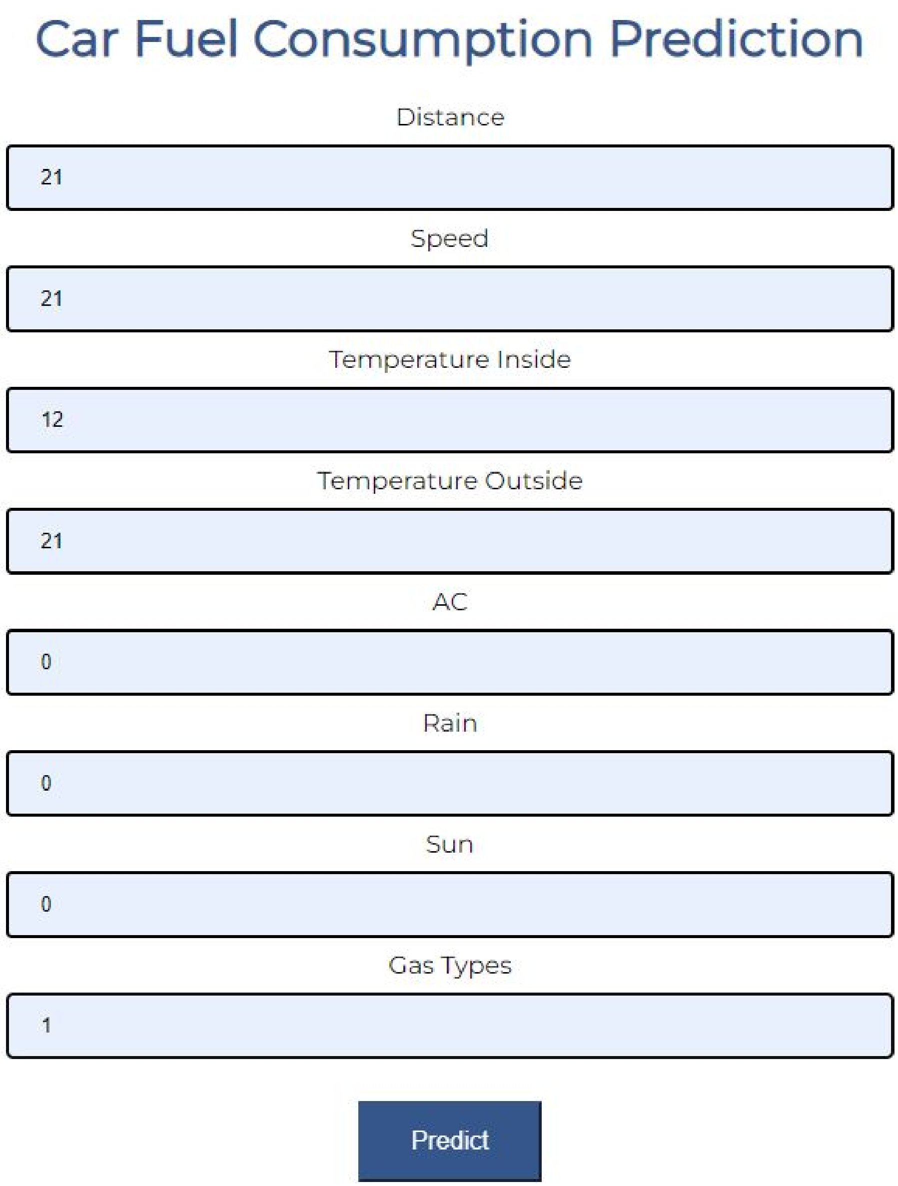


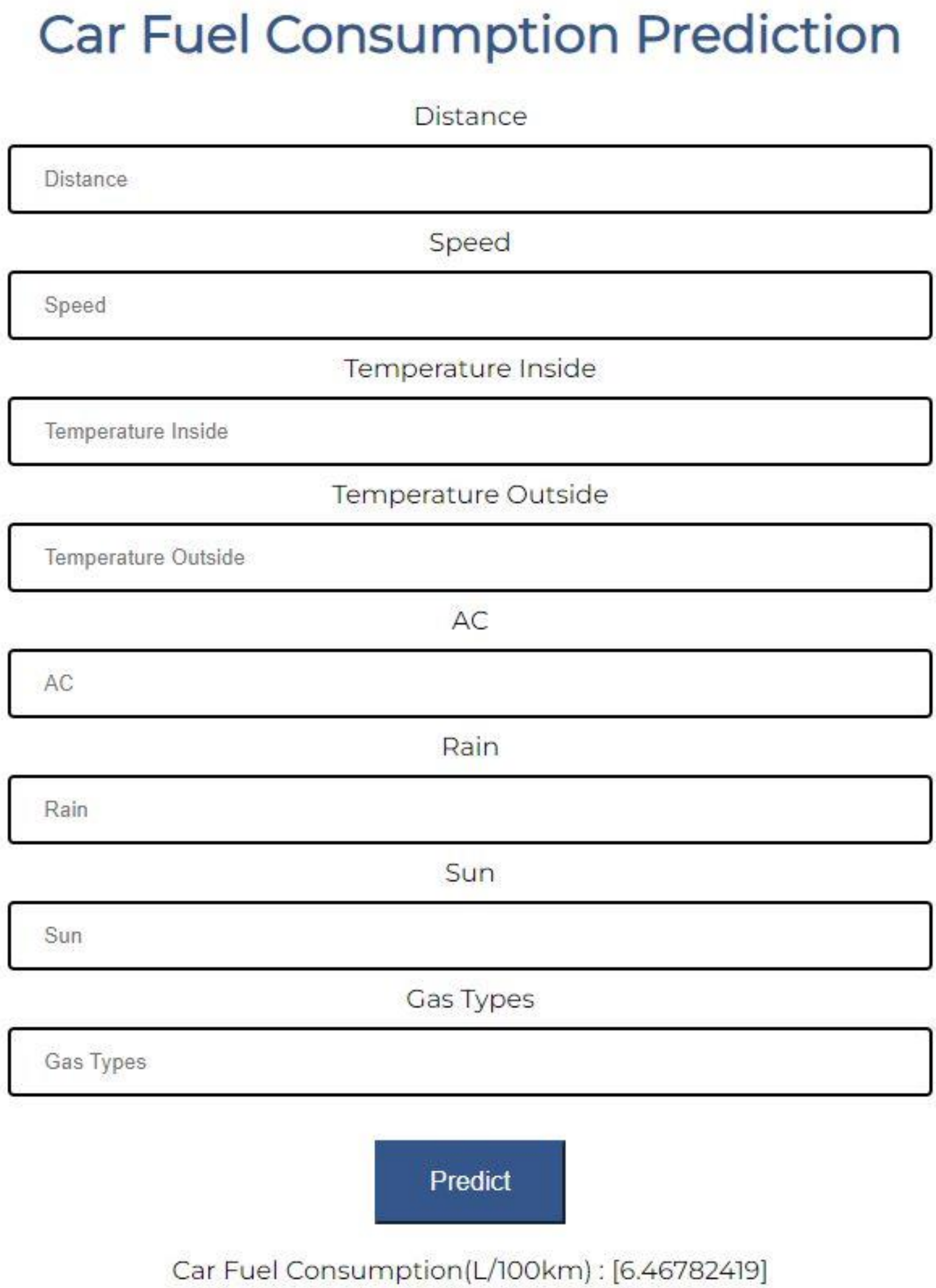

IBM Watson Studio
Deployments / models / Fuel_Consumption_Predictor /
Prediction results
Prediction type
Regression
Prediction distribution
Table view
JSON view
Prediction
1 5.364372012135133
Amount of predictions
Prediction value
5.364372
5.364372
Download

# Chapter 9

# RESULTS

## 9.1 Results




Car Fuel Consumption Prediction
Distance
Distance
Speed
Speed
Temperature Inside
Temperature Inside
Temperature Outside
Temperature Outside
AC
AC
Rain
Rain
Sun
Sun
Gas Types
Gas Types
Predict
Car Fuel Consumption(L/100km) : [6.46782419]

## 9.2 Performance Metric

Mean Absolute Error-0.6689496242764843

Mean Squared error-0.747694688029747

# Chapter 10

# ADVANTAGES & DISADVANTAGES

## 10.1 Advantages

The following are the various advantages of using the solution:

- Linear Regression is a model that works extremely well for linearly separable data.
- It is extremely easy to implement, and interpret. It is also highly efficient to train and use for regression.
- Linear regression can handle overfitting, Overfitting is a concept in data science, which occurs when a statistical model fits exactly against its training data. When this happens, the algorithm, unfortunately, cannot perform accurately against unseen data, defeating its purpose, very well by the use of techniques like dimensionality reduction, regularization, and cross-validation.
- It usually provides satisfactory results while being easy to implement and understand too.

## 10.2 Disadvantages

The following are the various disadvantages of using the solution:

- The assumption that the dependent and independent variables are linear is made.
- Though it can handle overfitting well, it is highly prone to overfitting and is easily affected the presence of any noise in the data.
- It is extremely sensitive to outliers, so if the data contains any exceptions whatsoever, the results may not be as accurate as required.

# Chapter 11
# CONCLUSION

## Conclusion

The prediction of fuel consumption is an extremely difficult and interesting challenge to solve. The main purpose of the project was to try and get accurate predictions of fuel consumption based on various parameters like the weather, the temperature, the average speed of the vehicle on the trip, the type of fuel, etc. These components have been analyzed using Linear Regression, a popular Machine Learning model, hailed for its simplicity and efficiency to optimize the efficiency of the vehicles. This is highly vital in today's world due to the quickly diminishing quantities of carbon-based fuels that most fleet vehicles use currently, as other alternatives are not as efficient or do not provide equivalent ranges in one fill. Thus, by optimizing the fuel type for each particular vehicle, we can minimize the amount of fuel used, while maximizing the performance of the vehicles. The data that has been produced using the Linear Regression algorithm is then displayed to the user in an intuitive and easy-to-read manner to make sure that the user can understand the results of the machine-learning model and make appropriate use of the resultant data.

# Chapter 12

# FUTURE SCOPE

## Future Scope

The dataset used for the prediction can be made significantly larger for better results and a more complex model, or an ensemble model (where multiple machine learning algorithms are used as a single entity), can be used in order to develop more accurate results and do so in a significantly shorter amount of time. This project can also be scaled up to different types of vehicles like aircraft, trains, and also military vehicles, where the general amount of fuel used is much higher when compared to consumer quantities.

# Chapter 13

# APPENDIX

## 13.1 Source Code+

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings('ignore')

# df=pd.read_excel('fuel consumption2.xlsx')

import os, types

import pandas as pd

from botocore.client import Config

import ibm_boto3

def _iter_(self): return 0

# @hidden_cell

# The following code accesses a file in your IBM Cloud Object Storage. It includes your
credentials.

# You might want to remove those credentials before you share the notebook.

cos_client = ibm_boto3.client(service_name='s3',

    ibm_api_key_id='JHBSOUQOqJ_lW-oAYZ9eXQ23cAjjXvjzL--jB2itM3n2',

    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",

    config=Config(signature_version='oauth'),
```

```
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')
bucket = 'ibmfuelprediction-donotdelete-pr-nalr08rg5fqali'
object_key = 'fuel consumption2.xlsx'
body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
df = pd.read_excel(body.read())
df.head()
print(df.shape)
df.info()
df.describe()
df.isnull().sum()
df.drop(['specials','refill liters','refill gas'],axis=1,inplace=True)
df.head(2)
mn = df.temp_inside.mean()
mn
med = df.temp_inside.median()
med
df['temp_inside']=df.temp_inside.fillna(mn)
df.isnull().sum()
df.head(5)
df[['distance','consume','speed','temp_inside','temp_outside']].mean()
df[['distance','consume','speed','temp_inside','temp_outside']].median()
df[['gas_type','AC','rain','sun']].mode()
df.describe()
df.head()
```

```
sns.histplot(df.distance)

sns.kdeplot(df.distance,shade=True)

sns.histplot(df.speed)

sns.kdeplot(df.speed,shade=True)

sns.histplot(df.temp_inside)

sns.kdeplot(df.temp_inside,shade=True)

sns.histplot(df.temp_outside)

sns.kdeplot(df.temp_outside,shade=True)

df.head(1)

plt.hist(df.gas_type)

plt.figure(figsize=(7,5))

df.gas_type.value_counts().plot(kind='barh')

plt.hist(df.temp_outside)

df.temp_inside.value_counts().plot(kind='barh')

plt.hist(df.temp_inside)

df.head(2)

sns.barplot(x='gas_type',y='consume',data=df)

plt.figure(figsize=(12,5))

sns.boxplot(x='temp_outside',y='consume',data=df,palette='rainbow')

df.head(2)

sns.barplot(x='gas_type',y='consume',data=df)

sns.barplot(x='AC',y='consume',data=df)

sns.barplot(x='rain',y='consume',data=df)

sns.barplot(x='sun',y='consume',data=df)
```

```
sns.heatmap(df.corr(),annot=True)
sns.pairplot(df)
df.head(2)
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df['gas_types']=le.fit_transform(df.gas_type)
df.drop('gas_type',axis=1,inplace=True)
df.head(2)
x=df.drop(['consume'],axis=1)
y=df.consume
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,mean_squared_error
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=42)
linear_reg=LinearRegression()
linear_reg.fit(x_train,y_train)
y_pred=linear_reg.predict(x_test)
mean_absolute_error(y_test,y_pred)
mean_squared_error(y_test,y_pred)
np.sqrt(mean_squared_error(y_test,y_pred))
import pickle
pickle.dump(linear_reg,open('model.pkl','wb'))
import joblib
joblib.dump(linear_reg,'model.save')
```

```
y
x
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "ju1Yk5i0N4urMBRLYVCJo4oM4cmX1nqdUNrCMJya2atq"
}
client = APIClient(wml_credentials)
def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    return (next(item for item in space['resources'] if item['entity']['name'] ==
space_name)['metadata']['id'])
space_uid = guid_from_space_name(client, 'models')
print("Space UID = " + space_uid)
client.set.default_space(space_uid)
client.software_specifications.list()
#Set Python Version
software_spec_uid = client.software_specifications.get_uid_by_name("runtime-22.1-py3.9")
software_spec_uid
model_details = client.repository.store_model(model = linear_reg, meta_props={
    client.repository.ModelMetaNames.NAME: "Fuel_Consumption_Predictor",
    client.repository.ModelMetaNames.TYPE: "scikit-learn_1.0",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
}
```

```
)
model_id = client.repository.get_model_id(model_details)
model_id
x_train
linear_reg.predict(x_train)
import requests

#NOTE: you must manually set API_KEY below using information retrieved from your IBM
Cloud account.
API_KEY = "ju1Yk5i0N4urMBRLYVCJo4oM4cmX1nqdUNrCMJya2atq"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

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

# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"field":
[["distance","speed","temp_inside","temp_outside","AC","rain","sun","gas_types"]], "values":
[[28.0,26,21.5,12,0,0,0,0]]}]}

response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/a02df083-f2eb-4743-aa66-
a605746c5264/predictions?version=2022-11-16', json=payload_scoring,
```

```
 headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())
```

## 13.2 GitHub & Project Demo Link

The GitHub link containing the final deliverables and the project demo is given below:

Link – GitHub - IBM-EPBL/IBM-Project-13229-1659514706: Trip Based Modeling of ...

Project Demo Link - https://drive.google.com/drive/folders/1QVCSIolFKltoe_S3vxQqZjXme-76EPRU?usp=sharing