Project Report Format

1. INTRODUCTION

Fleet vehicles are groups of motor vehicles owned or leased by a business, government agency, or other organization rather than by an individual or family. Typical examples include vehicles operated by car rental companies, taxicab companies, public utilities, public bus companies, and police departments. Fuel consumption: The amount of fuel used by each vehicle within a set time frame. So for a fleet vehicles we have to consider the fuel consumed by all the vehicles of that organization. For most fleets, fuel is a huge cost. From ensuring on-time delivery to meeting maintenance needs, managing a fleet of vehicles can be challenging—and expensive. In fact, one of the largest expenses in managing a fleet is the fuel used in the vehicles themselves. Vehicle fuel costs can represent as much as 60% of a fleet's total operating budget. Even a small increase in effective fuel consumption can benefit their bottom line. The most important factors that affect a vehicle's fuel consumption:1)Routing: The most efficient route is not always the route taken, despite best efforts. Traffic, personal preference, and construction can all cause drivers to take less than optimal routes that lead to worse fuel efficiency, 2)Driving behavior: How a vehicle is driven plays a large role in efficient fuel use. Harsh braking and acceleration, for example, can be big drivers of fuel consumption. The good news is that more efficient driving behavior can be encouraged through feedback, coaching, and training, 3)Maintenance: Tires issues, old oil, and other problems can lead to decreased fuel efficiency. This is why it's important to adhere to a preventative maintenance schedule and address small maintenance issues before they become larger problems, 4)Excessive loads: Heavy loads can also weight down a vehicle and decrease fuel efficiency and other real world parameters are road parameters, driver behavior, weather conditions, and vehicle parameters, etc. Today most of the government policies lean toward decreasing vehicle pollution and investing in vehicles with a smaller carbon footprint. Climate change is an important consideration in global fuel use and carbon emissions. Vehicles play a major part in reducing greenhouse gas emissions. So the need of fuel management is not only beneficial for the fleet organization but also for the society as a whole. So we have built a model to manage the fuel consumption of fleet vehicles using Machine Learning algorithms. The model was further evaluated with data collected from a vehicle on-road trip . A model was developed based on the parameters givenin the dataset like distance, vehicle speed, fuel specification(gas type etc to predict fuel consumption.

Machine learning techniques such as support vector machine (SVM) , random forest (RF) , and artificial neural networks (ANN) are widely applied to turn data into meaningful insights and solve complex problems .While the current approaches determine the fuel consumption of the vehicle, combining these techniques with data helps to identify parameters that may cause anomalies, the most deciding parameter of fuel consumption .Several previous models for both instantaneous and average fuel consumption have been proposed. A statistical model which is fast and simple compared to the physical load-based approach was developed to predict vehicle emissions and fuel consumption. However, these models are able to identify and learn trends in average fuel consumption with an adequate level of accuracy.

1.1. Project Overview

In the paper, we are enhancing the accuracy of the fuel consumption prediction model with Machine Learning to minimize Fuel Consumption. This will lead to an economic improvement for the business organization and satisfy the domain needs and also will be reducing the pollution caused. We propose a machine learning model to predict vehicle fuel consumption. The proposed model is based on the Decision Tree algorithm. The Fuel Consumption estimation is given as a function of distance travelled,vehicle speed, temperature inside and outside the vehicle, gas type and boolean variables like whether it is raining or not etc. The proposed model is applied and tested on a vehicle's On-Board Diagnostics Dataset. These types of fleets exist in various sectors including taxi, police vehicles, public transportation.

The observations were conducted on 10 features. Results achieved a higher accuracy with an R-Squared metric value of 0.97 than other related work using the same Decision Tree algorithm. We concluded that the Decision Tree has a great effect when used for fuel consumption prediction purposes. Data modeling can easily help to diagnose the reason behind fuel consumption with a knowledge of input parameters. Our model can compete with other machine learning algorithms for the same purpose which will help business organizations to manage the fuel consumption in a better way.

The performance of the Decision Tree Regression was evaluated using Mean Squared Error and Root Mean Squared Error. The model was further evaluated with data collected from a vehicle on-road trip. The study shows that Decision Tree Regression performed slightly better than other machine learning techniques such as Decision Tree Regression.

1.2. Purpose

Fuel economy is a measure of how far a vehicle will travel with a gallon of fuel; it is expressed in miles per gallon. This is a popular measure used for a long time by consumers in the United States; it is used also by vehicle manufacturers and regulators, mostly to communicate with the public. As a metric, fuel economy actually measures distance traveled per unit of fuel. Fuel consumption: The amount of fuel used by each vehicle within a set time frame. Fuel efficiency: The miles per gallon (MPG) of a driver/vehicle pair within a set time frame. So as we could see fuel consumption is the inverse of fuel economy. Fuel consumption is a fundamental engineering measure that is directly related to fuel consumed per 100 miles and is useful because it can be employed as a direct measure of volumetric fuel savings. Because fuel economy and fuel consumption are reciprocal, each of the two metrics can be computed in a straight-forward manner if the other is known. In mathematical terms, if fuel economy is X and fuel consumption is Y, their relationship is expressed by $XY = 1$. This relationship is not linear, as illustrated by in which fuel consumption is shown in units of gallons per 100 miles, and fuel economy is shown in units of miles per gallon.

In today's world an increase of fuel economy by 100 percent will significantly have a corresponding decrease in fuel consumption by 50 percent. Lets say-resulting decrease in fuel consumption per 100 miles and the total fuel saved in driving 10,000 miles .The dramatic decrease in the impact of increasing miles per gallon by 100 percent for a high-mpg vehicle is most visible in the case of increasing the miles per gallon rating from 40 mpg to 80 mpg, where the total fuel saved in driving 10,000 miles is only 125 gallons, compared to 500 gallons for a change from 10 mpg to 20 mpg. Likewise, it is instructive to compare the same absolute value of fuel economy changes—for example, 10-20 mpg and 40-50 mpg.

2. LITERATURE SURVEY

2.1. Existing Problem

Fuel is one of the fleet operator's biggest pain points. Being the second largest expense category (after labor costs), fuel seems to be an ever flowing source of stress and lost money. And if you don't control it the fuel management for fleet vehicles it will be a big headache. The fuel costs make up about 40-60% of India's total cost of transportation. The recent fuel price hike in India has got the fleet managers back on their toes again. Since March 22, there have been nine fuel pricehikes leading to the price of diesel crossing the RS.100 mark.

On average, trucks in India offer a fuel efficiency of between 1-6 km per litre of fuel. Several factors, such as road condition, vehicle maintenance, truckload, driving condition, etc., are responsible for the varying fuel efficiency of Indian trucks. Fuel costs make up anywhere between 40-60% of the total cost of operation for businesses and transporters. Therefore, Fleet fuel efficiency is crucial for business since even a marginal improvement or deterioration can make a big difference in the cost of operation, particularly for large truck fleet operators.

So it is now crucial for businesses to integrate technology with their transport operations to improve efficiency and transparency in their operations. Even marginal improvement in fleet performance will negatively impact fleet fuel efficiency and make operations better

So the solution proposed for this problem is- **Fuel Management system** which is a subdivision of a fleet management system that uses telematics-based tools and analytical software to capture fuel consumption data and improve fuel economy.

This system incorporates data about fuel transactions into analytics and learn what brands of fuel bring better economy, compare fuel usage across vehicles and generally improve the fuel buying and consuming behavior.

2.2. References

1) Name : Trip Based Modeling of Fuel Consumption in Modern Heavy-Duty Vehicles Using Artificial Intelligence.

Author Names : Sasanka Katreddi, Arvind Thiruvengadam

Published Year : 2021

2) Name : An Enhanced Fuel Consumption Machine Learning Model Used in Vehicles.

Author Names : B. Dhanalaxmi, M. Varsha, K. Roshan Chowdary and P. Mokshitha

Published Year: 2021

3) Name : Application of Machine Learning for Fuel Consumption Modelling of Trucks

Author Names : Federico Perrottaa, Tony Parrya and Luis C. Neves.

Published Year : 2017

4) Name : Minimising fuel consumption of vehicles as a function of path parameters

Author Names : Sagnik Choudhury, Y.V. Mahesh Kumar and Ankit Aggarwal

Published Year : 2011

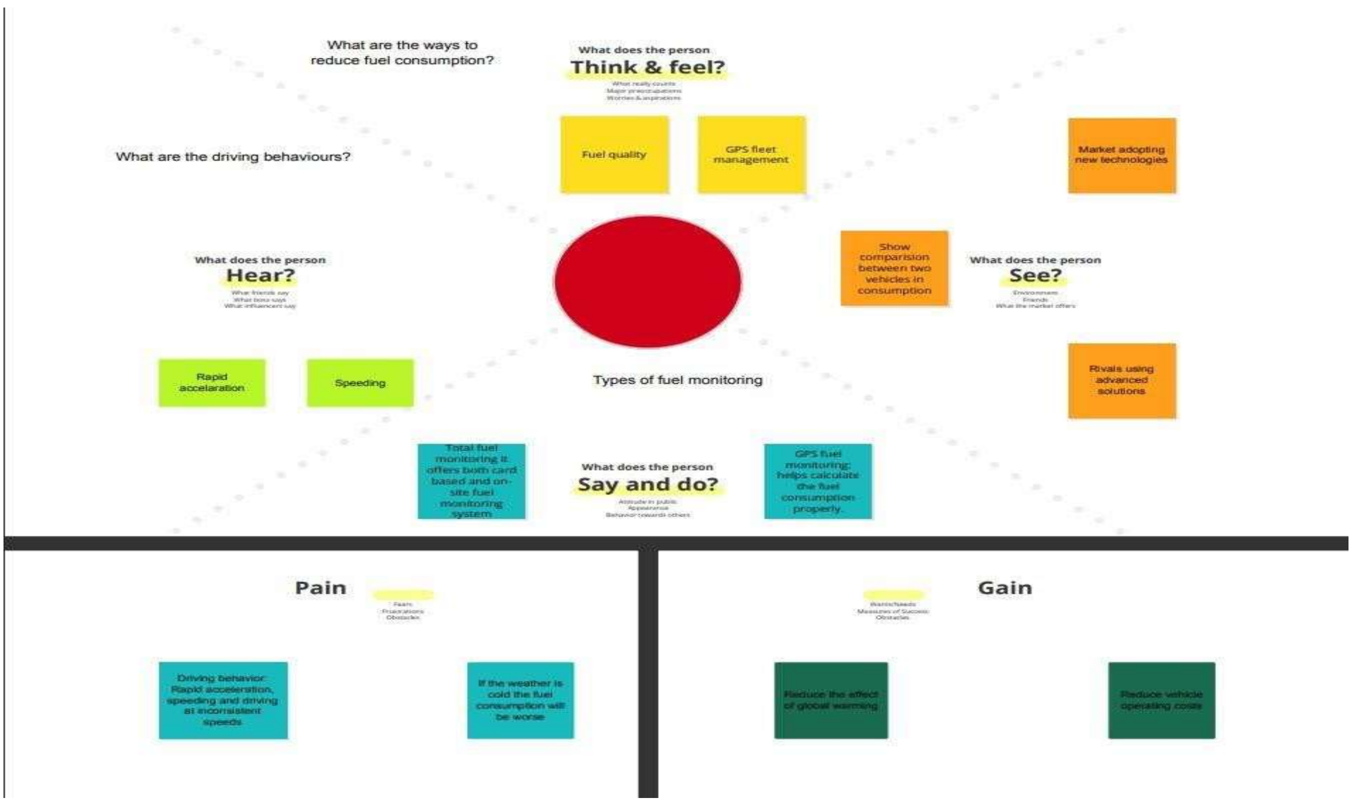
2.3. Problem Statement Definition

Predicting fuel consumption per trip based on gas type can help the fleet vehicle management to reduce the cost and time as well as prevent fraudulent activities on fuel usage.

So, the main aim of the project is to build 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

3. IDEATION & PROPOSED SOLUTION

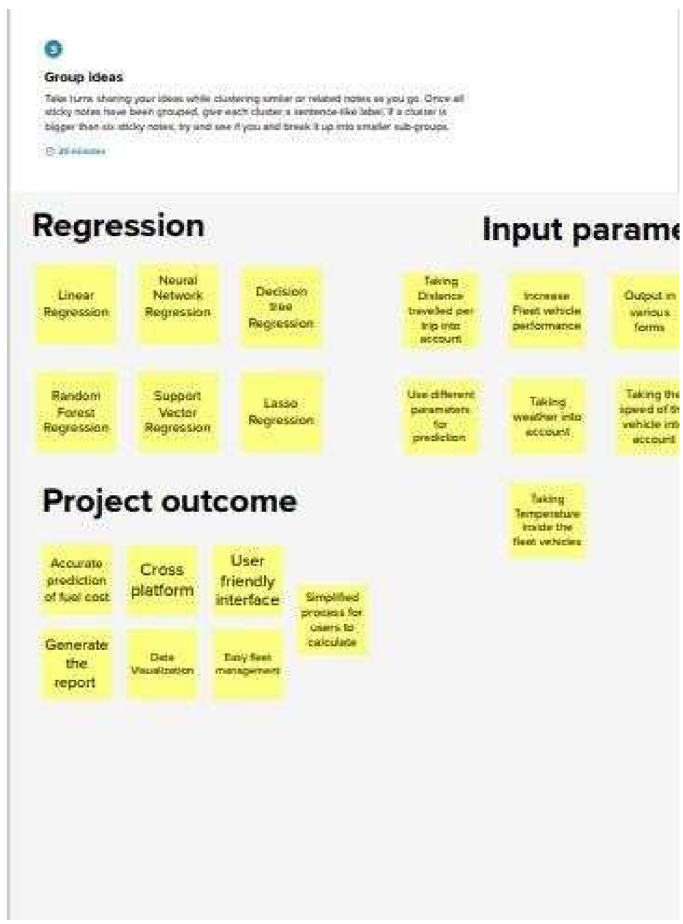
3.1. Empathy Map Canvas



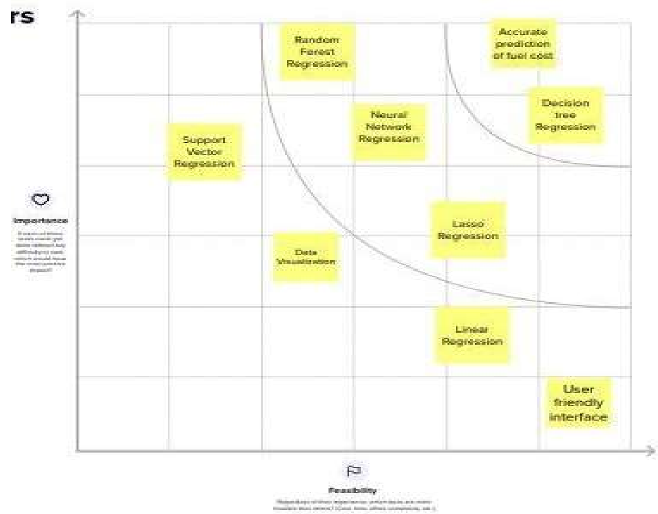
3.2. Ideation & Brainstorming

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

Step-2: Brainstorm, Idea Listing and Grouping



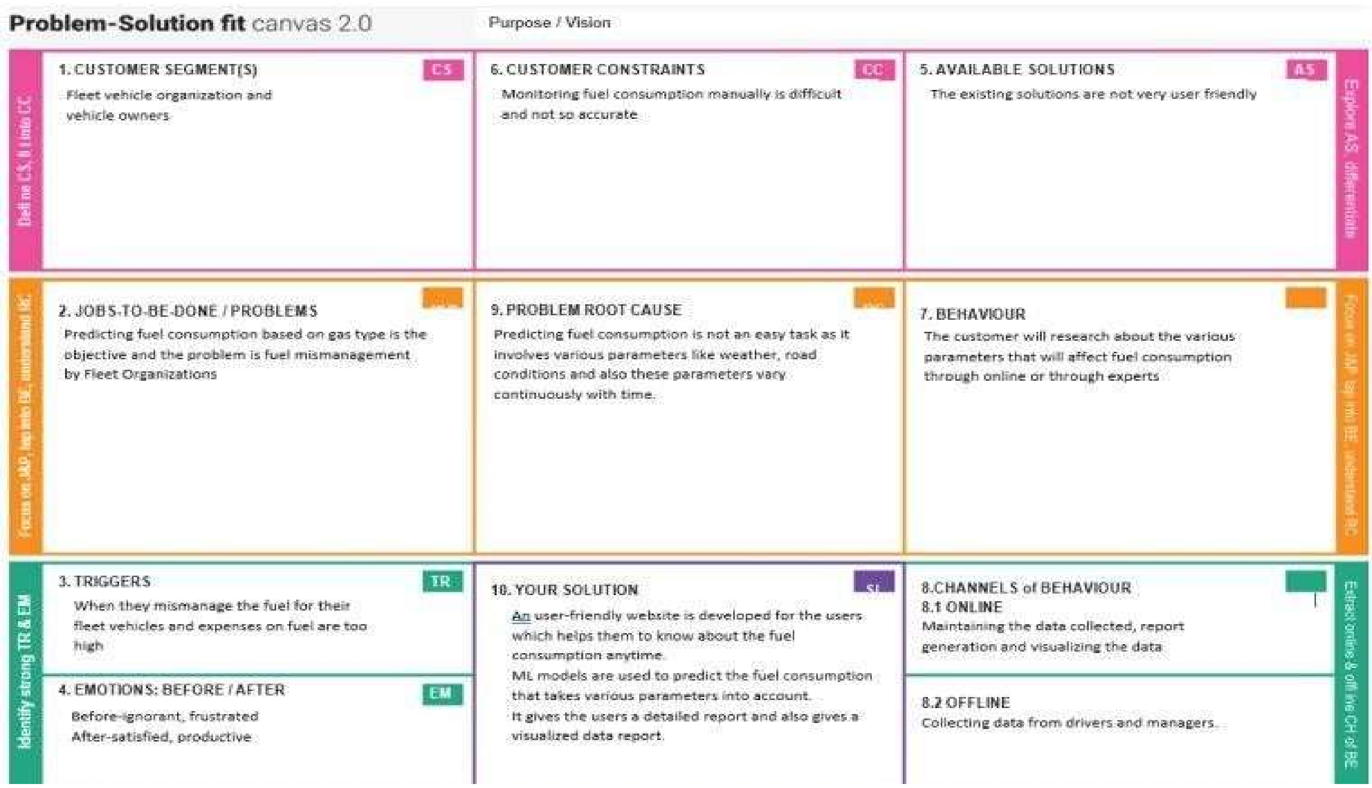
Step-3: Idea Prioritization



3.3. Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The problem statement is to predict the fuel consumption of fleet vehicles using Machine Learning. The ML model must be integrated with a web application.
2.	Idea / Solution description	Capture and prevent fuel theft and leakage. Fuel prediction for a vehicle will be given based on the gas type and other some parameters.
3.	Novelty / Uniqueness	Multiple ML models are deployed to predict the fuel consumption. Users can use the model to predict for various types of vehicles.
4.	Social Impact / Customer Satisfaction	The fleet manager will be satisfied knowing that he will prevent all the fraudulent activities in fuel management committed by his employees.
5.	Business Model (Revenue Model)	The detailed report generation and data visualization are given only for the premium users.
6.	Scalability of the Solution	This helps the user to know the fuel pilferage situations and also take some preventive steps to reduce the amount spent on fuel.

3.4. Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1. Functional Requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration and User Login	Registration through Form Registration through Gmail Login through Portal
FR-2	User Portal	Fuel prediction for a single vehicle Fuel prediction for multiple vehicles
FR-3	Output generation	Generating a report Visual representation of the report

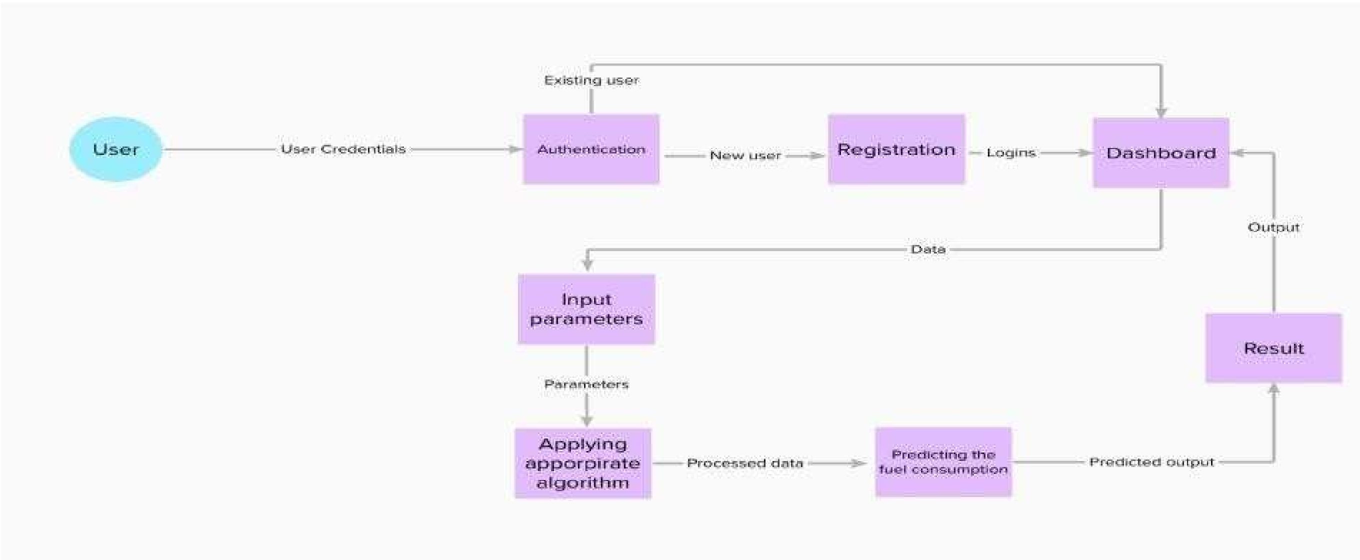
4.2. Non-Functional Requirements

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	An easy to use portal with an appealing User Interface. Users must be able to get their needed output easily from the portal.
NFR-2	**Security**	Each and every user provided with an unique username and password.
NFR-3	**Reliability**	The output of the model must be accurate and should be able to handle multiple requests.
NFR-4	**Performance**	After checking with multiple algorithms, a highly accurate ML model must be used for prediction.
NFR-5	**Availability**	Website must be portable and must be accessible 24*7.
NFR-6	**Scalability**	The generated report could be delivered to the user via their email.

5. PROJECT DESIGN

5.1. Data Flow Diagrams



5.2. Solution & Technical Architecture

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User interacts with the portal through the user interface	HTML, CSS, JavaScript
2.	User registration	New users register into the portal	Python
3.	User login	User logins into the portal through his/her user credentials	Python
4.	Database	The details collected from users for prediction are stored in a database where columns are mostly varchar type or float type	MySQL
5.	Cloud Database	Database Service on Cloud to store data in cloud	IBM DB2
6.	External API	Weather API used to provide input to discrete parameters like rainy, sunny etc	IBM Weather API
7.	Machine Learning Model	Various Machine Learning models are tested and the most accurate one is used	Linear regression, Random Forest, etc.

5.3. User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my username, email, password, organization name, and employee position	I can access my account / dashboard	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering username/email & password	I can access my account only through valid credentials	High	Sprint-1
	Dashboard	USN-3	As a user, I will be able to see the profile details and login details	I can view the details only after logging in	Medium	Sprint-2
Customer (Fleet Organization)		USN-4	After entering the dashboard, I will give the input parameters for prediction	I must give the input values in proper data form	Medium	Sprint-2
		USN-5	As a user, I will get the predicted output	I get the output for the needed vehicles	High	Sprint-2
		USN-6	As a user, I will get a detailed report of my output	I view a detailed report of the predicted output	Medium	Sprint-3
	Instruction	USN-7	As a user, I can go through the instructions in the site to carry out the prediction process.	I can see the instructions for easy use of portal	Low	Sprint-4
Developer	Portal configuration	USn-7	As a developer, I can configure the portal for easy use by customer	I can change the dashboard for easy and better use by the customer.	Medium	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation

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	7	6 Days	24 Oct 2022	29 Oct 2022	7	29 Oct 2022
Sprint-2	10	10 Days	31 Oct 2022	9 Nov 2022	10	9 Nov 2022
Sprint-3	5	2 Days	11 Nov 2022	12 Nov 2022	5	12 Nov 2022
Sprint-4	9	3 Days	14 Nov 2022	16 Nov 2022	9	16 Nov 2022

6.2. Sprint Delivery Schedule

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 username, email, password, organization name, and employee position.	4	High	Nandha Krishnan N Sathish Kumar P
Sprint-1	Login	USN-2	As a user, I can log into the application by entering username/email & password	3	High	Dev Karthikeyan A V Manimaran S
Sprint-2	Dashboard	USN-3	As a user, I will be able to see the profile details and login details	2	Medium	Nandha Krishnan N Dev Karthikeyan A V
Sprint-2	Entering input	USN-4	After entering the dashboard, I will give the input parameters for prediction	2	Medium	Manimaran S Sathish Kumar P
Sprint-2	Viewing output	USN-5	As a user, I will get the predicted output	6	High	Nandha Krishnan N Manimaran S
Sprint-3	Report generation	USN-6	As a user, I will get a detailed report of output	5	Medium	Sathish Kumar P Dev Karthikeyan A V
Sprint-4	Instructions	USN-7	As a user, I can go through the instructions in the site to carry out the prediction process.	4	Low	Manimaran S Sathish Kumar P
Sprint-4	Portal Configuration	USN-8	As a developer, I can configure the portal for easy use by customer	5	Medium	Dev Karthikeyan A V Nandha Krishnan N

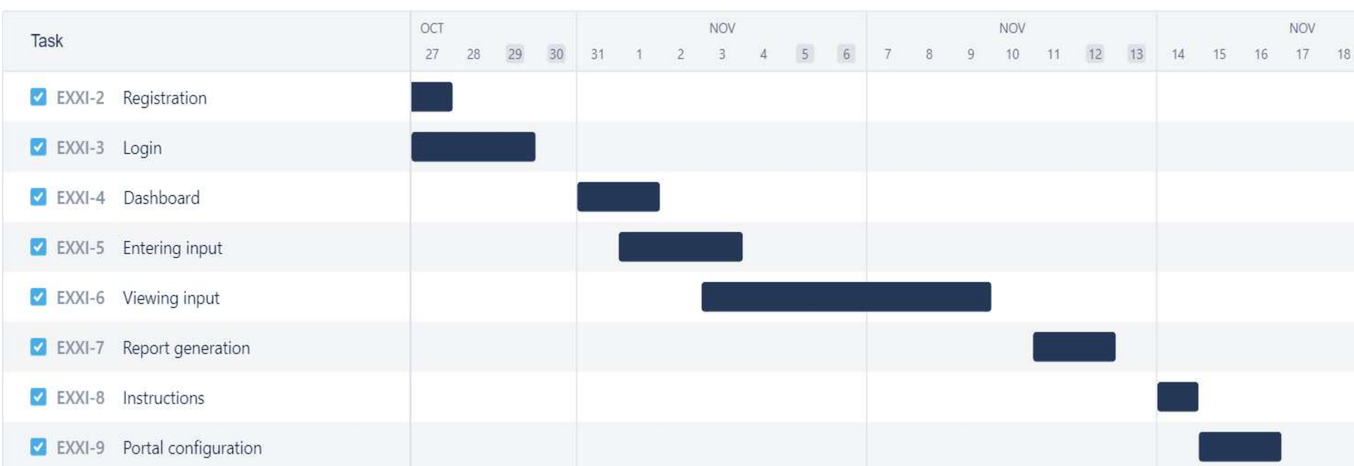
6.3. Reports from JIRA

AV=Velocity/Sprint duration

Sprint	Average velocity
Sprint 1	1.17
Sprint 2	1
Sprint 3	2.5
Sprint 4	3

Burn 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.



7. CODING & SOLUTIONING

7.1. Feature 1

Have done building a web application that is integrated to the model we built. A UI is provided for the uses where user has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

Building HTML Pages:

```
<html>
    <head>
            <title>
                    Prediction
            </title>

            <link href = 'https://fonts.googleapis.com/css?family=Montserrat' rel = 'stylesheet'>
            <style>
                    * {
                            box-sizing:border-box;
                    }
                    body {
                            font-family:'Montserrat';
                    }


                    .header {
                      top:0;
                      margin:0px;
                      left:0px;
                      right:0px;
                      position:fixed;
```

background-color:black;

```
            color:white;
            box-shadow:0px 8px 4px grey;
            overflow:hidden;
            padding:15px ;
            font-size:2vw;
            width:100%;
            text-align:left ;
            padding-left:100px ;
            opacity:0.9;
        }
        .header_text {
                font-size:40px;
                text-align:center;
        }
        .content {
                margin-top:100px;
        }
        .text {
                font-size:20px;
                margin-top:10px;
                text-align:center;
        }

input[type=number] , select {
width: 50%;
padding: 12px 20px;
margin: 8px  0;
display: inline-block;
border: 1px solid #ccc;
border-radius: 4px ;
```

```
box-sizing: border-box;
}

input[type=submit] {
width:50%;
background-color: #000000;
color: white;
padding: 14px 20px;
margin: 8px 0;
border: none;
border-radius: 4px;
cursor: pointer;
}

input[type=submit]:hover {
background-color: #5d6568;
color: #ffffff;
border-color: black;
}
form {
	margin-top: 20px;
}
.result {
color: black;
margin-top: 30px;
margin-bottom: 20px;
font-size:  25px;
color: red;
}
```

```
        </style>
    </head>
    <body align=center >
    <div class= "header" >
                <div> Car Fuel Consumption </div>
    </div>
    <div class= "content">
    <div class= "header_text"> Car Fuel Consumption Prediction </div>
    <div class= "text" > Fill in and below details to predict the consumption depending on the gas type . </div>
    <div class= "result" >
     {{ prediction_text }}
    </div >
    <form action = "{{url_for('y_predict')}}" method = "POST" >
        <input type= "number" step = "any" id="distance" name = "distance" placeholder = "distance(km)"
>
        <input type= "number" id = "speed" name = "speed" placeholder = "speed(km/h)">
        <input type= "number" id = "temp_inside" name = "temp_insidet" placeholder = " temp_inside (°C)
" >
        <input type= "number" id = "temp_outside" name = "temp_outside" placeholder = " temp_outside (
° C ) " >
        <input type= "number" id = "AC" name = "AC" placeholder = "AC" >
        <input type= "number" id = "rain" name = "rain" placeholder = "rain" >
        <input type= "number" id = "sun" name = "sun" placeholder = "sun" >
        <input type= "number" id =  "E10" name = "E10" placeholder = "E10" >
        <input type= "number" id = "SP98" name = "SP98" placeholder = "SP98" >
        <input type= "submit" value= "Submit" >
     </form>
    </div>
    </body>
</html>
```

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) Verify UI elements in the Home Page	http://127.0.0.1:5000	The Home page must be displayed properly	Working as expected	Pass
2)Check if user can give inputs	Enter the values of distance, Speed, Temp_out, Temp_in,Gas type,AC, rain and sun	Getting input successfully	Working as expected	Pass
3) Check if user cannot provide inputs in wrong format)	Enter the values of distance, Speed, Temp_out, Temp_in,Gas type,AC, rain and sun	User cannot give inputs in different format	Not working	Fail
4) Check if the route is working properly	http://127.0.0.1:5000/y_predict	The page should redirect to the results page	Working as expected	Pass
5) Verify UI elements in the Result Page	http://127.0.0.1:5000/y_predict	The Result page must be displayed properly	Working as expected	Pass
6) Check if the result is displayed properly	http://127.0.0.1:5000/y_predict	The result should be displayed properly	Working as expected	Pass

8.2. User Acceptance Testing

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved. The following are the detail and description for user acceptance testing.

Section	Total Cases	Not Tested	Fail	Pass
Client Application	6	0	1	5
Performance	3	0	0	3
Exception Reporting	2	0	0	2

9. RESULTS

The Following passage which explains the end result of the overall development of the cloud application.

9.1. Performance Metrics

```
Root Mean Squared Error:  0.02313126348804806
Mean Squared Error: 0.0005350553505535053
Mean Abolsute Error: 0.0025830258302583014
```

The three main metrics used to evaluate a Regression model are Mean Squared Error, Mean Absolute Error and Root Mean Squared Error. The average of the squared errors is calculated using the Mean Squared Error. In essence, it computes the difference between the estimated and real values, squares the answers, and then finds the average of those numbers. MSE can only presumptively assume non-negative values because the mistakes are squared. MSE is typically positive and not zero because most processes have inherent unpredictability and noise. Error in the Mean Square: 0.023131263488040806.

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES :

- One of the modern fuels with the highest efficiency and energy density is diesel. It offers better fuel economy than gasoline because it has more usable energy.
- This algorithm gives a better result.

DISADVANTAGES :

- Diesel fuel doesn't produce flashy high-speed performance, but it is thought to be more efficient because it transforms heat into energy rather than releasing it through the exhaust pipe like gas-powered vehicles do.
- Diesels still need regular maintenance to keep them running. You have to change the oil and the air, oil, and fuel filters.
- Although diesel fuel used to be cheaper than gasoline, it now often costs the same amount or more.

11. CONCLUSION

This paper presented a machine learning model that can be conveniently developed for each heavy vehicle in a fleet. The model relies on the predictors:distance, vehicle speed,temperature inside and outside the vehicle,gas type and if it is raining or not. The value for the variables are given in the dataset itself.

Moreover, the predictors can be easily computed based on these variables. The model predictors are aggregated over a fixed distance traveled. So with the Decision tree regression the features have been observed and model is trained in the structure of a tree to predict the fuel consumption.

So with the help of this model the fleet organization can predict the fuel consumption anytime they need.

12. FUTURE SCOPE

The fuel prediction is based mainly on the gas type of the fuel. So in future various real world parameters like road conditions, weather, driving behavior, routing, maintenance .Prediction based on these parameters could give a more accurate and real world application for fuel consumption maintenance. With better fuel consumption and its maintenance, the fleet managers can manage their business well and also this will reduce the cost of fleet organization and will definitely reduce the pollution in the environment.

13. APPENDIX

13.1. Source Codes

IBM Model training code:

```
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='TQcdbl5ER_u5pO18So7lfBIg4kVr0QjTlBGwSBy2heLc',

   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 = 'custommodeldeployment-donotdelete-pr-c6ofwyz61rvg98'

object_key = 'measurements2.xlsx'


body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']

import numpy as np

import pandas as pd

df=pd.read_excel("F:/trip-fuel-consumption/measurements2.xlsx")

print(df.head())

import seaborn as sns
```

```
sns.heatmap(df.isnull())

df.drop(['refill gas','refill liters','specials'],axis=1,inplace=True)

sns.heatmap(df.isnull())

temp_inside_mean=np.mean(df['temp_inside'])

df['temp_inside'].fillna(temp_inside_mean,inplace=True)

sns.heatmap(df.isnull())

x=df.drop(['consume','gas_type'],axis=1)

y=df['consume']

x.columns

x=x.values

y=y.values

dum1=pd.get_dummies(df['gas_type'])

print(dum1)

df=pd.concat([df,dum1],axis=1)

df.drop('gas_type',axis=1,inplace=True)

x1=df.drop('consume',axis=1)

y1=df['consume']

x1=x1.values

y1=y1.values

x_train,x_test,y_train,y_test=train_test_split(x1,y1,test_size=0.3,random
_state=42)

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
```

```
train_scaled = scaler.fit_transform(x_train)

test_scaled = scaler.transform(x_test)

from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeRegressor

tree_model = DecisionTreeRegressor()

tree_model.fit(train_scaled, y_train)

y_pred_1=tree_model.predict(train_scaled)

print(y_pred_1)

from sklearn import metrics

print(np.sqrt(metrics.mean_squared_error(y_train,y_pred_1)))

print(metrics.mean_squared_error(y_train,y_pred_1))

print(metrics.mean_absolute_error(y_train,y_pred_1))

y_pred_2=tree_model.predict(test_scaled)

print(y_pred_2)

from sklearn import metrics

print(np.sqrt(metrics.mean_squared_error(y_test,y_pred_2)))

print(metrics.mean_squared_error(y_test,y_pred_2))

print(metrics.mean_absolute_error(y_test,y_pred_2))

import joblib

joblib.dump(tree_model,'model2.save')

from ibm_watson_machine_learning import APIClient
import json
import numpy as np
wml_credentials = {
```

"url":"https://us-south.ml.cloud.ibm.com",

"apikey":"3NS6Mw_Q34G6EzqNroX-p-f-W_Z18pzigd2qz8FR2WuS"

}

```python
wml_client = APIClient(wml_credentials)

wml_client.spaces.list()

SPACE_ID="291e13c6-7940-4295-9ac7-1e262f72ca6b"

wml_client.set.default_space(SPACE_ID)

MODEL_NAME='PetrolConsumptionPredictor'

DEPLOYMENT_NAME='Car Petrol Consumption Prediction  '

BEST_MODEL=tree_model

software_spec_uid=wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')


# Setup model meta

model_props={

   wml_client.repository.ModelMetaNames.NAME:MODEL_NAME,

   wml_client.repository.ModelMetaNames.TYPE:'scikit-learn_1.0',

   wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid

}

model_details = wml_client.repository.store_model(model=BEST_MODEL,meta_props=model_props,training_data=x_train,training_target=y_train)

model_uid = wml_client.repository.get_model_uid(model_details)
```

Building HTML Page Code:

```html
<html>

    <head>

            <title>

                    Prediction

            </title>

            <link href = 'https://fonts.googleapis.com/css?family=Montserrat' rel = 'stylesheet'>

            <style>

                    * {

                            box-sizing:border-box;

                    }
```

```
body {
	font-family:'Montserrat';
}


.header {
  top:0;
  margin:0px;
  left:0px;
  right:0px;
  position:fixed;
  background-color:black;
  color:white;
  box-shadow:0px 8px 4px grey;
  overflow:hidden;
  padding:15px ;
  font-size:2vw;
  width:100%;
  text-align:left ;
  padding-left:100px ;
  opacity:0.9;
}
.header_text {
	font-size:40px;
	text-align:center;
}
.content {
	margin-top:100px;
}
.text {
```

```
            font-size:20px;
            margin-top:10px;
            text-align:center;
        }
input[type=number] , select {
width: 50%;
padding: 12px 20px;
margin: 8px  0;
display: inline-block;
border: 1px solid #ccc;
border-radius: 4px ;
box-sizing: border-box;
}

input[type=submit] {
width:50%;
background-color: #000000;
color: white;
padding: 14px 20px;
margin: 8px 0;
border: none;
border-radius: 4px;
cursor: pointer;
}

input[type=submit]:hover {
background-color: #5d6568;
color: #ffffff;
border-color: black;
```

```
            }
            form {
                    margin-top: 20px;
            }

            .result {
            color: black;
            margin-top: 30px;
            margin-bottom: 20px;
            font-size:  25px;
            color: red;
            }
            </style>
    </head>
    <body align=center >
    <div class= "header" >
                    <div> Car Fuel Consumption </div>
    </div>
    <div class= "content">
    <div class= "header_text"> Car Fuel Consumption Prediction </div>
    <div class= "text" > Fill in and below details to predict the consumption depending on the gas type . </div>
    <div class= "result" >
      {{ prediction_text }}
    </div >
    <form action = "{{url_for('y_predict')}}" method = "POST" >
            <input type= "number" step = "any" id="distance" name = "distance" placeholder = "distance(km)"
>
            <input type= "number" id = "speed" name = "speed" placeholder = "speed(km/h)">
            <input type= "number" id = "temp_inside" name = "temp_insidet" placeholder = " temp_inside (°C)
" >
```

```
            <input type= "number" id = "temp_outside" name = "temp_outside" placeholder = " temp_outside (
° C ) " >
            <input type= "number" id = "AC" name = "AC" placeholder = "AC" >
            <input type= "number" id = "rain" name = "rain" placeholder = "rain" >
            <input type= "number" id = "sun" name = "sun" placeholder = "sun" >
            <input type= "number" id =  "E10" name = "E10" placeholder = "E10" >
            <input type= "number" id = "SP98" name = "SP98" placeholder = "SP98" >
            <input type= "submit" value= "Submit" >
      </form>
    </div>
    </body>
</html>
```

Building Server Side Script Code:

```
import flask
import joblib
import requests
from flask import render_template, request
from flask_cors import CORS
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "3NS6Mw_Q34G6EzqNroX-p-f-W_Z18pzigd2qz8FR2WuS"
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
distance=float(input("Distance travelled:"))
speed=int(input("Speed of the vehicle:"))
temp_inside=int(input("Temperature inside the vehicle:"))
temp_outside=int(input("Temperature outside the vechicle:"))
AC=int(input("Is AC turned on? (Enter 0 if turned off, Enter 1 if turned on):"))
rain=int(input("Is it raining outside? (Enter 0 if not raining, Enter 1 if it is raining):"))
sun=int(input("Is the sun clearly visible? (Enter 0 if no, Enter 1 if yes):"))
E10=int(input("Is the type of petrol is E10? (Enter 0 if no, Enter 1 if yes):"))
SP98=int(input("Is the type of petrol is SP98? (Enter 0 if no, Enter 1 if yes):"))

'''distance, speed, temp_inside, temp_outside,AC, rain, sun, E10,SP98=15.4,39,22,23,0,0,0,0,1'''
features= [[distance, speed, temp_inside, temp_outside, AC, rain, sun,
    E10,
    SP98]]

payload_scoring = {"input_data": [{"field": [["distance", "speed", "temp_inside", "temp_outside",
    "specials", "AC", "rain", "sun", "E10",
    "SP98"]], "values":features}]}

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/e5d76088-9d1f-4513-8c63-22fd982a7f39/predictions?version=2022-11-15', json=payload_scoring,
  from flask import Flask,request,render_template
import joblib
app=Flask(_name_)
model=joblib.load("model2.save")
app=Flask(_name_)
@app.route('/')
def predict():
```

```
    return render_template('Manual_predict.html')

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

def y_predict():

    x_test=[[float(x) for x in request.form.values()]]

    print('actual',x_test)

    pred=model.predict(x_test)

    return render_template('Manual_predict.html',prediction_text=('Car fuel
Consumption(L/100km)\:',pred[0]))

if __name__=='_main_':

    app.run(host='0.0.0.0',debug=True)


main()
```

13.2. GITHUB AND PROJECT DEMO LINK:

1) **Github link : https://github.com/IBM-EPBL/IBM-Project-2488-1658472807**

2) **Project Demo Link: https://www.youtube.com/watch?v=SVDLSvSb0HA&authuser=1**