JAYAM COLLEGE OF ENGINEERING AND TECHONOLGY

TRIP BASED FUEL CONSUMPTION PREDICTION USING MACHINE LEARNING

APPLIED DATA SCIENCE

(PNT2022TMID40758)

TEAM MEMBERS

SURYA PRAKASH S

SAKTHIVEL J

MANIGANDAN R

KARTHIGA G

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Overview

Fuel efficiency is one of the quality indicators of transport vehicles. Fuel consumption is a significant part of the operating costs of a vehicle. Fuel consumption is the amount of fuel a vehicle uses to travel a particular distance.

Generally fuel consumption is the inverse of fuel economy. Therefore it is important to monitor fuel consumption. The fuel consumption prediction is one of the best ways to reduce the consumption. It helps users to be aware of vehicular conditions and manage the expenses accordingly. Since the increase in fuel rate has affected ordinary lives, it is essential to predict the consumption before use. In order to do this we have developed a model which enables users to predict the fuel consumption easily and accurately.

The model has been developed using Machine Learning. The solution is a web application that has been integrated with the Machine learning model. Initially the user must register to the website and get authorized. Users can access the web application to get the prediction result by providing basic inputs such as distance, mileage, vehicle weight, speed, gas type and road type. The model developed takes input from the user and processes it with different algorithms of machine learning such as Multi linear regression.

The web application serves two types of prediction which includes Single sample prediction for single vehicle and Multi sample prediction for number of vehicles.

Multi sample prediction is aimed at users like organizations where vehicular use is large(eg. Transportation). The input data given by these organizations are usually large in volume. To make it easy for them, the application is designed in such a way that an Excel sheet containing the input data can be given to the prediction model. The final prediction result will also be provided in the same page. The application offers additional features such as detailed report generation, view predicted history. This allows the user to get the complete information of the process carried out. The data provided by users is secured and used for optimal

prediction. With this the application developed serves a great purpose by helping users to reduce their vehicular expenses.

1.2 Purpose

Nowadys fuel expenses are considerably increased with the growing economy. A proper monitoring is required to reduce the expenses and lower the users burden. Many fraudulent activities take place in managing the fuel expenses in fleet management of organisations. This project is aimed at providing a User- friendly website to the users for fuel consumption prediction. It helps users such as layman and organisation to get the fuel consumption of a vehicle based on the trip data. The vehicle efficiency can also be analysed by the user based on the consumption for a particular trip. The estimation of fuel consumption is another one feature that a user needs to know and this can be delivered by the product developed here. It

mainly emphasize on the need to manage the fuel consumption and expenses of large fleet vehicles in transport and trade. In addition to prediction, it also offers a detailed report for the user which proves to be helpful for sharing predicted results to others. The website display the standard fuel price updated on daily basis automatically. This would help users to know the fuel estimation along with the prediction. Therefore, the website developed proves to be beneficial to the users.

Applied Data Science

Trip - Based Fuel Consumption Prediction

2. LITERATURE SURVEY

2.1 ExistingProblem

The need for fuel consumption prediction has urged many to build a solution. There are lot of machine learning models developed for various fuel based predictions. Unfortunately, the solutions given by people were just theoretical and not developed as a complete product. Previously there wasn't any application or working model for customers use. The algorithms developed gives only accurate values which a normal user cannot easily access and use. There are many journals and review papers on fuel consumption prediction but nothing has been implemented in the market for customers use. This has urged us to develop an application that is easy to use and delivers accurate prediction for users along with additional features.

1. **Fuel Consumption and Traffic Emissions Evaluation of Mixed TrafficFlow with Connected Automated Vehicles at Multiple TrafficScenarios**

1.

Bin Zhao et al. presented a paper which intends to analyses the impact of different proportions of connected automated vehicles (CAVs) on fuel consumption and traffic emissions. This paper studies fuel consumption and traffic emissions of

mixed traffic flow with CAVs at different traffic scenarios. The car following modes and proportional relationship of vehicles in the mixed traffic flow are analyzed. On this basis, different car-following models are applied to capture the corresponding car-following modes. Then, Virginia Tech microscopic (VT-micro) model is adopted to calculate the instantaneous fuel consumption and traffic emissions. Finally, based on three typical traffic scenarios, a basic segment with bottleneck zone, ramp of the freeway, and signalized intersection, a simulation platform is built based on Python and SUMO to obtain vehicle trajectory data, and the fuel consumption and traffic emissions in different scenarios are obtained. The

results show that in different traffic scenarios, the application of CAVs can reduce fuel consumption and traffic emissions. The higher the penetration rate, the more significant the reduction in fuel consumption and traffic emissions. In the three typical traffic scenarios, the advantages of CAVs are more evident in the signalized intersection. When the penetration rate of CAVs is 100%, the fuel consumption and traffic emissions reduction ratio is as high as 32%. It is noteworthy that the application of CAVs in urban transportation will significantly reduce fuel consumption and traffic emissions.

2. Fuel Consumption Prediction Model Using Maching Learning

Mohamed A. HAMED et al. team proposed the idea of enhancing the accuracy of the fuel consumption prediction of a model with Machine Learning to minimize fuel

consumption. This will lead to an economic improvement for the business and satisfy the domain needs. We propose a machine learning model to predict vehicle fuel consumption. The proposed model is based on the support Vector Machine algorithm. The Fuel Consumption estimation is given as a function of mass air flow, vehicle speed, revolutions per minute, and throttle position sensor features. The proposed model is applied and tested on a vehicle's Onboard diagnostics dataset. The observations were conducted on 18 features. Results achieved a higher accuracy with an R-squared metric value of 0.97 than other related work using the same support vector machine regression algorithm. In this study, they tried to enhance fuel consumption prediction using machine learning using the Support Vector Machine Algorithm to predict fuel consumption based on a legacy dataset containing onboard diagnostics data. Their aim is to achieve a good value for the R-squared metric using the SVM. The proposed model aims to predict fuel consumption using SVM. the proposed model consists of four phases: data preprocessing, feature weighting, feature selection, and SVM prediction model the proposed prediction model has been applied to a dataset with 8262 records, the dataset includes 18 fields, the fuel consumption dataset was gathered from 19

drivers using OBD scanner in vehicles, which was used for a previous dissertation for profiling automotive data in 2019. The Dataset gathered from 19 drivers had been collected depending on a vehicle model of the well-known Brazilian vehicle, A 2015 Chevrolet S10, which has a 2.5-liter flex-fuel engine of 206 hp. This Dataset

is gathered on an urban road in the city of Natal (Brazil). It was gathered at a distance of 18.8 kilometers for 34 minutes for each driver.

3. Development of a Fuel Consumption Prediction Model Based on MachineLearning Using Ship In-Service Data

Young-Long Kim et al. prepared a paper as interest in eco-friendly ships increases, methods for status monitoring and forecasting using in-service data from ships are being developed. Models for predicting the energy efficiency of a ship in real time need to process the operational data effectively and be optimized for such an application. This paper presents models that can predict fuel consumption using inservice data collected from a 13,000 TEU class container ship, alo with statistical and domain-knowledge methods to select the proper input variables for the models.

These methods prevent over fitting and multicollinearity while providing practical applicability. To implement the prediction model, either an artificial neural network

(ANN) or multiple linear regression (MLR) was applied, where the ANN-based models showed the best prediction accuracy for both variable selection methods. The goodness of fit of the models based on ANN ranged 0.9709 to 0.9936. Furthermore, sensitivity analysis of the draught under normal operating conditions indicated an optimal draught of 14.79 m, which was very close to the design draught of the target ship, and provides the optimal fuel consumption efficiency. These models could provide valuable information for ship operators to support decision making to maintain efficient operating conditions.

9

4. Vehicle Fuel Consumption Prediction Method Based on Driving Behavior Data Collected from Smartphones

Ying Yao et al. proposed this paper as vehicle energy consumption and pollutant emissions are key problems for the healthy and sustainable development of urban transportation. With the continuous growth of car ownership in China, the energy consumption of its private cars increased 4.2 times, from 13.12 to 68.34 million tons of standard coal, from 2005 to 2015. Based on the growth of the population, GDP, and the proportion of secondary and tertiary industries of China, the trend of future transportation energy consumption can be predicted. -e energy consumption of private cars will continue to increase before 2020 when it is expected to reach 117.38 million tons of standard coal. -therefore, reducing energy consumption has become one of the most important challenges in the transportation field.

Among many factors that affect the energy consumption of vehicles, driving behavior plays an important role. Research conducted by Ford Motor Company shows that improvement in driving behavior could improve fuel economy by 25% in the short term. Providing drivers with continuous eco-driving feedback in the long term could lead to a 10 percent reduction in fuel consumption. studied the influence of ecological driving behavior on fuel consumption and found that giving feedback on fuel consumption information to drivers could improve fuel economy by 10%. In addition, the eco-driving instructions given to drivers could improve fuel economy by approximately 15%. Zhao and Chang analyzed the influence of drivers' route choices on vehicle fuel consumption, and the results indicated that energy

consumption and exhaust emissions are significantly reduced by minimizing high-emission driving behavior.

This study proposes a vehicle fuel consumption prediction method based on Global Positioning System (GPS) data collected from a smartphone. Taxi drivers participated in this experiment.

By matching the driving behavior data of the mobile phone and the fuel consumption data of the OBD terminal, the driving behavior indexes that affect fuel consumption were screened, and the fuel consumption prediction models were constructed using machine learning algorithms. E-prediction model of drivers individual fuel consumption based on mobile phone data could not only further improve the real-time monitoring database of fuel consumption with strong error tolerance but also provide technical support for macro control of urban transportation energy consumption and effectiveness evaluation of the transportation energy policy.

5. Influence of road and traffic conditions on fuel consumption and fuel cost for different bus technologies

Ivković Ivan S.et al. presented paper on the influences of road and traffic conditions on fuel consumption and fuel costs of conventional diesel, parallel hybrid, and stoichiometric compressed natural gas buses in intercity bus service are analyzed. Calculation of fuel consumption and fuel costs for these three different bus technologies was conducted for road network of the Republic of Serbia. Three scenarios were considered. The first scenario includes bus traffic volume carried out

on the road network in 2014. The other two scenarios are characterized by the decrease i. e. increase of traffic volume by 20% with unchanged state of road infrastructure in comparison to the year 2014.

Obtained results show that in intercity bus service the greatest influence on the fuel consumption of buses has operating speed of the bus, followed by terrain type on which buses operate. The impact of other factors (international roughness index, fluctuation of traffic volume by 20%, and correction factors of fuel consumption) is less pronounced. This is cost saving for hybrid buses compared to diesel buses that are in the range of €3.33-7.27.

2.3 Problem definition statement

PS-1	A user	Predict my fuel consumption	I couldn't findany user-friendly website	All the other products were only available as software	Uncomfortable

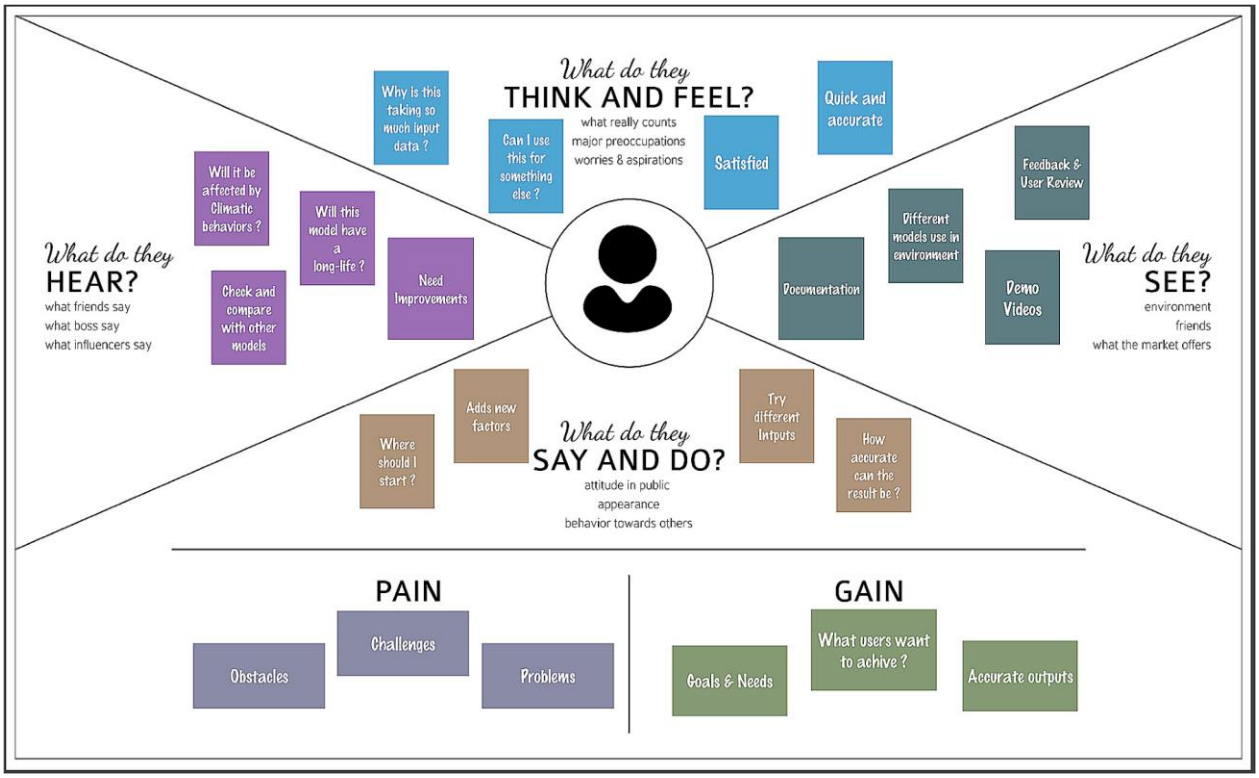
PS-2	A transportation manager, who manages more than 1000 vehiclesof our company	Find the fuel consumption of all our vehicles	I cannot findthe fuel consumpti onof all our vehicles at the same time	There is no option to enter multiple vehicle details	Annoying
PS-3	The owner ofan organization	Manage the records of fuel consumption of my fleets	I couldn't get a detailed report	All other existing solutions give only the prediction	Like I'm left with very minimal data

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally

created by Dave Gray and has gained much popularity within the agile community. Have the team members speak about the sticky notes as they place them on the empathy map. Ask questions to reach deeper insights so that they can be elaborated for the rest of the team. To help bring the user to life, you may even wish to sketch out the characteristics this person may have on the center of the face.



Ideation & 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


2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
Harish
Add or remove features
Share predictions with others
Report Generation
Train using own dataset
Enter multiple sample at a time
Import multiple input sample using excel sheet
KARTHI
use different parameters for prediction
output in various forms
other vehicular predictions
output accuracy
finding vehicle efficiency
cost estimation based on result
Logeshwaran
Receive Customer Suggestion / Feedbacks
Outputs Will be User Friendly
Cross Platform
Identify Terrain type
Identify Total Cost
Fleet Management Can be easy
Mugundaan
User friendly interface
Real time data can be correlated
Various conditions are used
Driver's driving consistency records
Simplified process for the user to calculate
Climatic conditions can be considered
3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
GROUP 1
User friendly interface
Identify Total Cost
Various conditions are used
Share predictions with others
visual representation of outputs
Enter multiple sample at a time
cost estimation based on result
output accuracy
use different parameters for prediction
Receive Customer Suggestion / Feedbacks
Driver's driving consistency records
GROUP 2
Train using own dataset
Cross Platform
Add or remove features
Simplified process for the user to calculate
Real time data can be correlated
finding vehicle efficiency
Climatic conditions can be considered
output in various forms
Report Generation
Fleet Management Can be Easy
Import multiple input sample using excel sheet
other vehicular predictions
Identify Terrain type


4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
Share predictions with others
Driver's driving consistency records
User friendly interface
cost estimation based on result
other vehicular predictions
Enter multiple sample at a time
Report Generation
output accuracy
visual representation of outputs
Add or remove features
Train using own dataset
Cross Platform
Various conditions are used
Identify Total Cost
output in various forms
Import multiple input sample using excel sheet
Receive Customer Suggestion / Feedbacks
use different parameters for prediction
Climatic conditions can be considered
finding vehicle efficiency
Identify Terrain type
Fleet Management Can be Easy
Simplified process for the user to calculate
Real time data can be correlated
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3. Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The problem statement is to predictfuel consumption of modern fleetvehicles using machine learning. A web application needs to be built which is integrated with the ML model. The solution shouldsatisfy the following user requirements: 1. User friendly interface 2. Process multiple samples simultaneously 3. Provide detailed report
2.	Idea / Solution description	The solution is a mobile responsive web application that can be used in both mobile and computers. Cumulative results of multiple ML models are used to achieve accurate prediction. The website provides a user-friendly interfaceand accepts multiple samples predicting them simultaneously. A detailed reportcan be generated along with the predicted output.

3.	Novelty / Uniqueness	1. Multiple ML modelsare used to predict thefuel consumption. 2. Results are generated in various forms. 3. Users can run multiple samples at a time.

4.	Social Impact / Customer Satisfaction	Fraudulent activities can be prevented in fleet management. Customers are satisfied in all aspects as the proposed solution is developed using multiple ML models.
5.	Business Model (Revenue Model)	The revenue is generated on subscription basis where large scale data processing and detailed report generation are allowed for only premium subscription.
6.	Scalability of the Solution	The application can further be extended to provide Application Programming Interface (API) which can be used by third party organizations such as Automobile Manufacturers, Logistics companies, etc.

4. Problem Solution Fit

Problem-Solution fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Companies and Organizations, developers and ordinary people

6. CUSTOMER CONSTRAINTS CC

Low accessibility to existing solution
Device compatibility

5. AVAILABLE SOLUTIONS AS

A software exists which gets the dataset and after training the model, predicts the result.
Various models have been developed, but have not been implemented and brought into use.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

High Fuel Expense
No Proper platform for fuel consumption prediction

9. PROBLEM ROOT CAUSE RC

Lack of awareness about fuel consumption
Existing solutions are minimal and does not meet user expectations

7. BEHAVIOUR BE

Research about variations in fuel consumption
Search for solutions online
Seek suggestions from others

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

Finding it difficult to manage fuel consumption of vehicles
Realizing that the fuel expense is significantly higher than estimated

4. EMOTIONS: BEFORE / AFTER EM

Before - frustration, confused
After - satisfied, feeling productive and smart

10. YOUR SOLUTION SL

A website is developed which uses combination of multiple ML models to predict the fuel consumption accurately.
The website has a user friendly interface and is mobile responsive.
It offers various functionalities such as detailed report generation, predicting results for multiple samples simultaneously.

8. CHANNELS of BEHAVIOUR CH

Online - Social Media, Forums, Blogs

Offline - Friends and Colleagues, Consultancy, Vehicle Manufacturers

Extract online & offline CH of BE

4.Requirement Analysis

Functional Requirements

Following are the functional requirements of the proposed solution.

FRNo.	FunctionalRequirement (Epic)	SubRequirement(Story/Sub-Task)
FR-1	User Registration/Login	Via Email Via Phone number
FR-2	User Dashboard	Single Sample Prediction Multiple Sample Prediction View User History
FR-3	Output Generartion	Visual Representation Report Generation

4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NF R-1	Usability	User-friendly interface to facilitate the user with easy processing Model provides visual representation of predictions
NF R-2	Security	Authentication - user can have his/herown private dashboard to have assured access
NF R-3	Reliability	The model is capable enoughto handle huge volume of data and run multiple samples

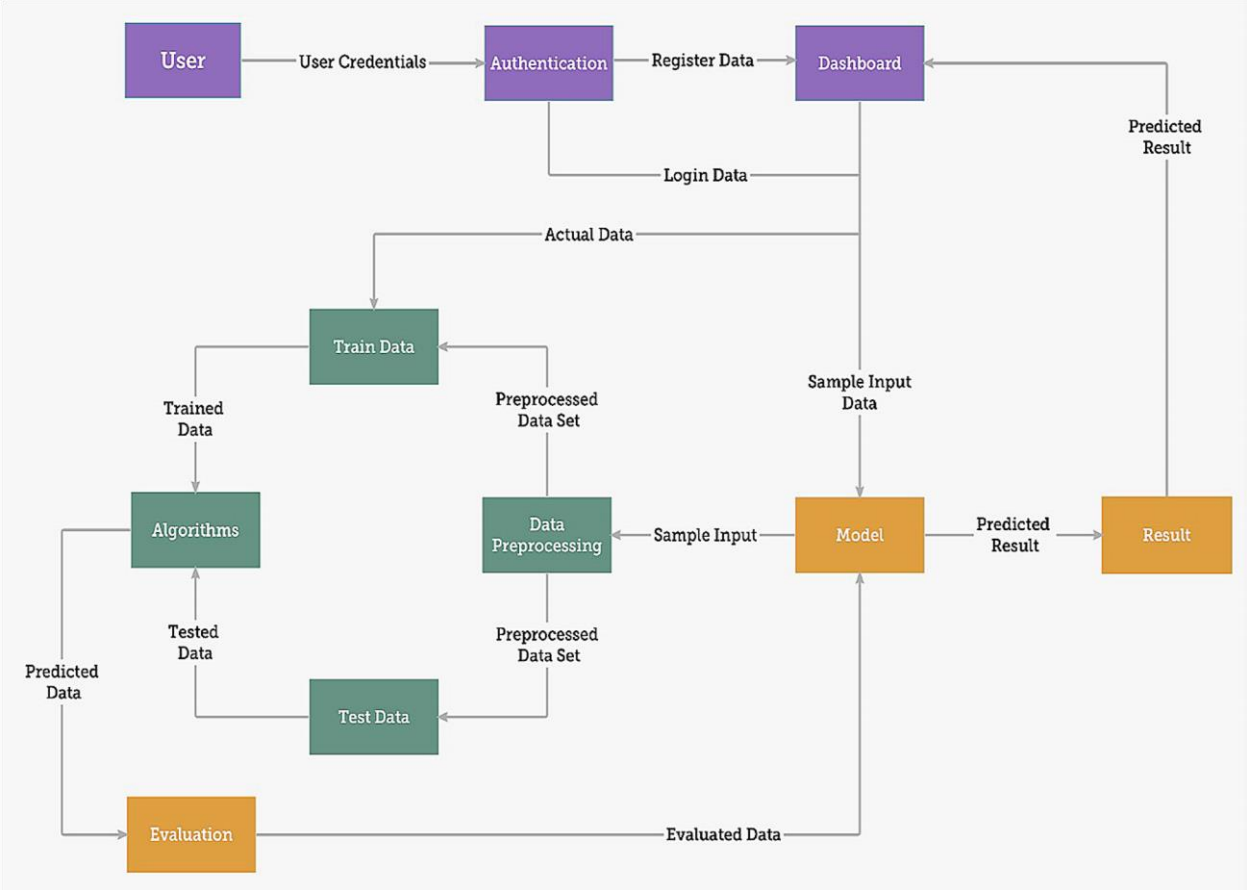
		simultaneously
NF R-4	Performance	As the model is combination of multiple ML algorithms, the accuracy is high
NH R-5	Availability	The websiteis also mobile-responsive and portable. It requires only basic configuration to run on any device.
NF R-6	Scalability	It can be extended furtherto provide APIwhich can be used by third party organisation suchas Automobile Manufactures, Logistics companies, etc.

5. Project Design

5.1 Data Flow Diagrams

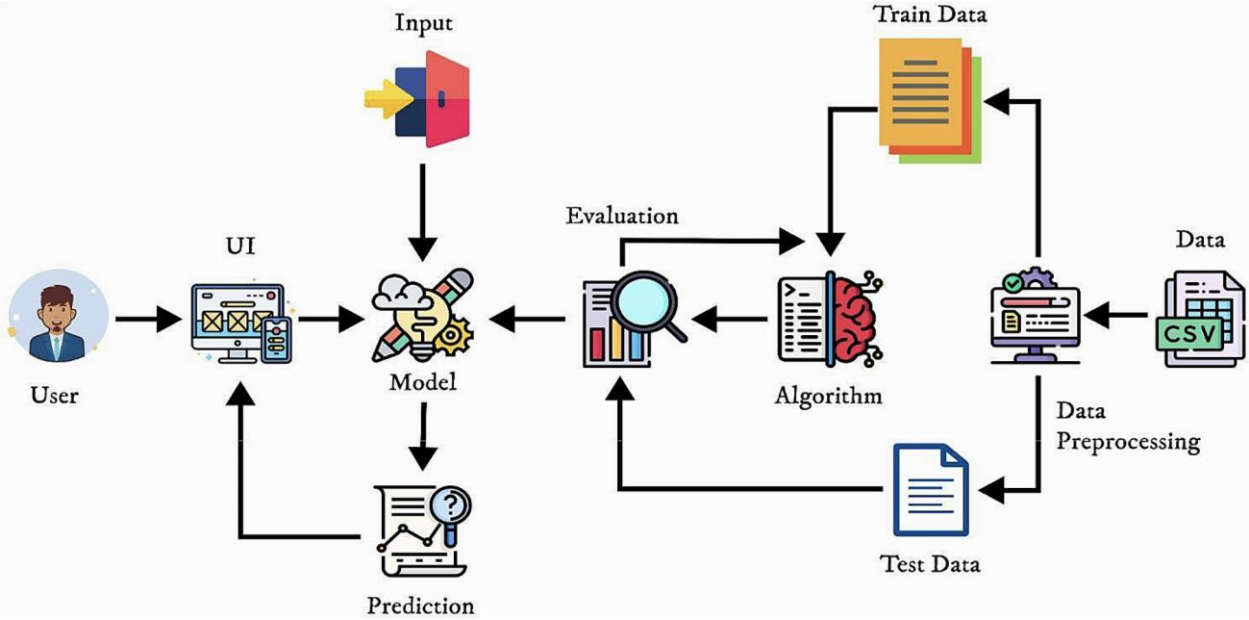
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the

system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored. The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system.



5.2 Solution and Technical Architecture

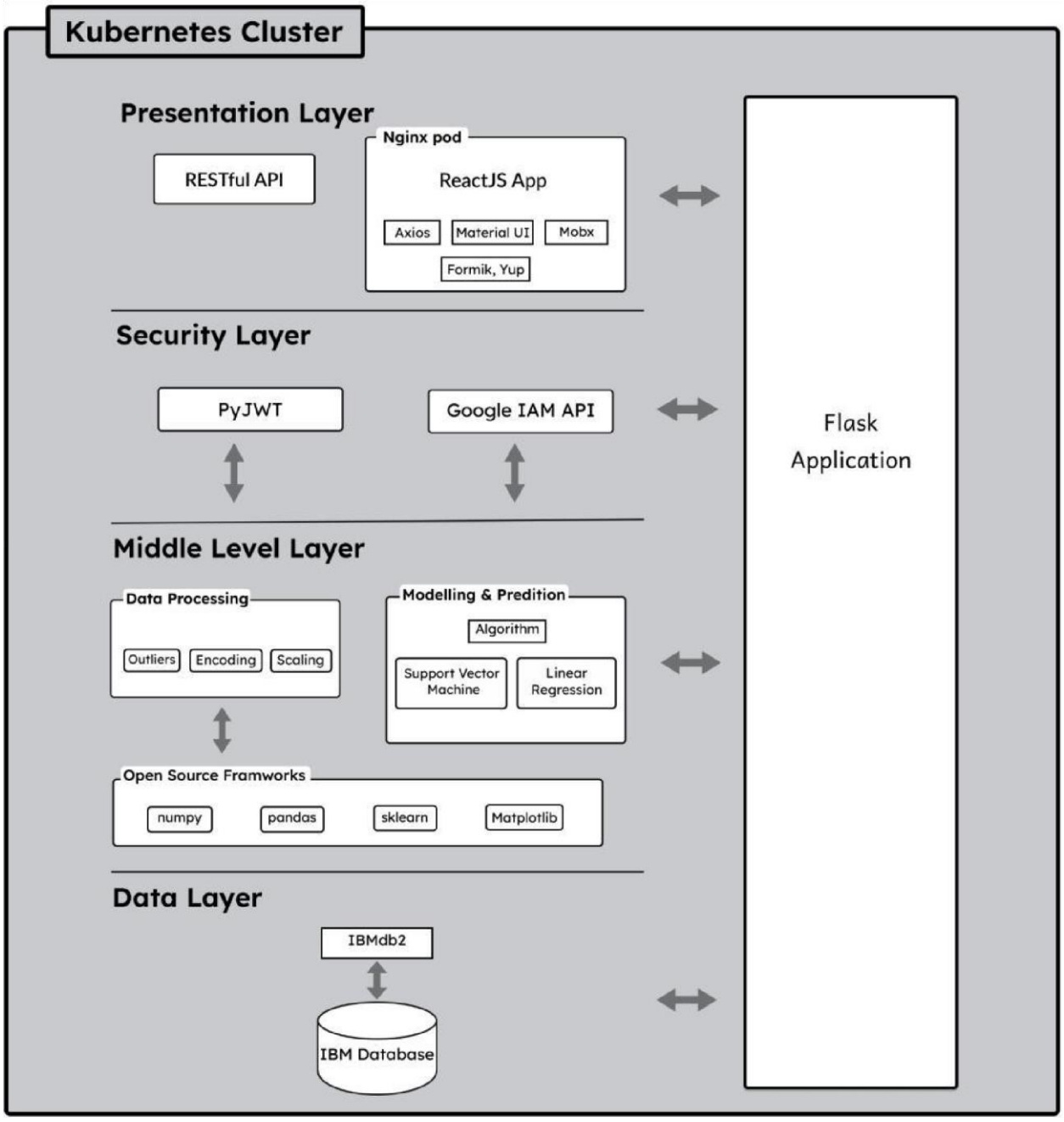
A solution architecture (SA) is an architectural description of a specific solution. SAs combine guidance from different enterprise architecture viewpoints (business, information and technical), as well as from the enterprise solution architecture (ESA).



Technical Architecture:

S.No	Component	Description	Technology
1.	Website	User interacts with the prediction model through website to predict the fuel consumption	Axios, Material ui, Mobx, Formik, Yup
2.	Cloud Database	The model is provided with data from IBM cloud databse	IBM Cloud DB,ibm_db(python package)
3.	API	Used to extend the service to other application	Flask Application
4.	JWT & Session	It is used for Handling JSON web tokens(singing, verifying, decoding)	PyJWT, Flask-Session

5.	Machine Learning Model	This model is developed to predict the fuel consumption usingML algorithms	Sklearn, Algorithms - MLR
6.	Data processing	Data is pre-processed and used for training the model which is then used for prediction	Pandas, Numpy, Matplotlib


Kubernetes Cluster
Presentation Layer
RESTful API
Nginx pod
ReactJS App
Axios
Material UI
Mobx
Formik, Yup
Security Layer
PyJWT
Google IAM API
Middle Level Layer
Data Processing
Outliers
Encoding
Scaling
Modelling & Predition
Algorithm
Support Vector Machine
Linear Regression
Open Source Framworks
numpy
pandas
sklearn
Matplotlib
Data Layer
IBMdb2
IBM Database
Flask Application

5.3 User Stories

User Type	Functional Requirement	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Customer	Registration / Login	USN-1	As a user, I can register or login to create a dashboard for my processing	I can access my account / dashboard	High	Sprint-3

	Dashboard	USN-2	Once I enter the dashboard I can input values for a single sample prediction	I can predict for single sample	High	Sprint-1
Customer (Organizatio n)		USN-3	Once I enter the dashboard I can input values via excels sheet for multiple	I can perform multiple sample prediction	Medium	Sprint-2

			sample prediction			

		USN- 4	As a user I can get visual representation of the prediction	I can have different forms of output	High	Sprint-1
		USN-5	As a user I can view the detailed report of my prediction	I can access details of my proccess and prediction	Medium	Sprint-1
	Documentation	USN-6	As a User I can refer to the documentation for support and guidence	I can use user manual forguidance	Medium	Sprint-1,2,3,4
Developer	Settings	USN-7	As a developer I can allow user to access dashboard's settings to edit the user profile and update it.	I can use the settings to edit the user profile details and update it.	Low	Sprint- 4

		USN- 8	As developerI can deploy the model in IBM Cloud using IBM Watson.	I can deploy the model in cloud and make it available as a service.	High	Sprint- 4

6. Project Design and 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 using email & password	4	High	Harish KumarR
Sprint-2		USN-2	As a user,I can registerusing Gmail	2	Medium	Karthikeyan S
Sprint-1		USN-3	As a user, I will receive an confirmation email once I registered for the application	1	Low	Harish Kumar R

	Login	USN-4	As a user, I can login to my dashboard through email-id and password	2	High	Mugundaan K
Sprint-2	Prediction Model	USN-6	Once I enter the dashboard I can input values for a	8	High	Logeshwaran K M
			single sample prediction			
Sprint-3		USN-7	I can input values via excel sheet for multi-sample prediction as per the template and perform prediction	6	Medium	Karthikeyan S
		USN-8	As a user, I can get a visual representation of the predicted value	4	Medium	Karthikeyan S

	Report Generation	USN-9	As a user, I can view the detailed report of my prediction	3	High	MugundaanK
Sprint-4	Documentation	USN-10	As a User, I can refer to the User manual documentation for support and guidance	6	Medium	Logeshwaran KM
	Deployment	USN-11	As developer , I can deploy the model in IBM Cloud using IBM Watson	7	High	Harish Kumar R

Sprint1:

It consists of several tasks such as

- Registration to the website
- OTP confirmation on successful registration
- Login to user dashboard

A fresh user who wants to access the website has to register himself/herself to the website by providing the required credentials. A verification OTP will be sent to the user's mail to authenticate. On successful registration a user dashboard will be

created to facilitate with website accessing. If a user has already registered to the website then he/she can login directly using the registered mail. This will take them directly to their dashboard.

Sprint2:

It consists of tasks such as

- Register via Gmail
- Perform Single sample prediction

A user can also register to the website via Gmail which makes the process easier by using existing gmail. Once the user entered into the dashboard he/she can do fuel consumption prediction for a single vehicle by providing the required inputs to the prediction model. The prediction result will also be available on the same page.

Sprint3:

It consists of tasks such as

- Perform Multi sample prediction
- Display visual representation of result
- Get a detailed report of prediction

This sprint aims to complete Multi sample prediction which is mainly offered for users such as transport and other organisations. The allows users to predict consumption for multiple vehicles at a single stretch. A sample datasheet will be provided to the users where they can enter the values for each input column and

this will be uploaded as an input file(format - .csv) to the prediction model. The result will be displayed in the same page as a table.

Here a user will also get a visual representation of their output via a detailed report. This detailed report will be generated based on the data provided by user and prediction done by the model. The report can be downloaded to facilitate user with more viability.

Sprint4:

This sprint covers tasks such as

- User manual or guides
- Deploy the model in Cloud

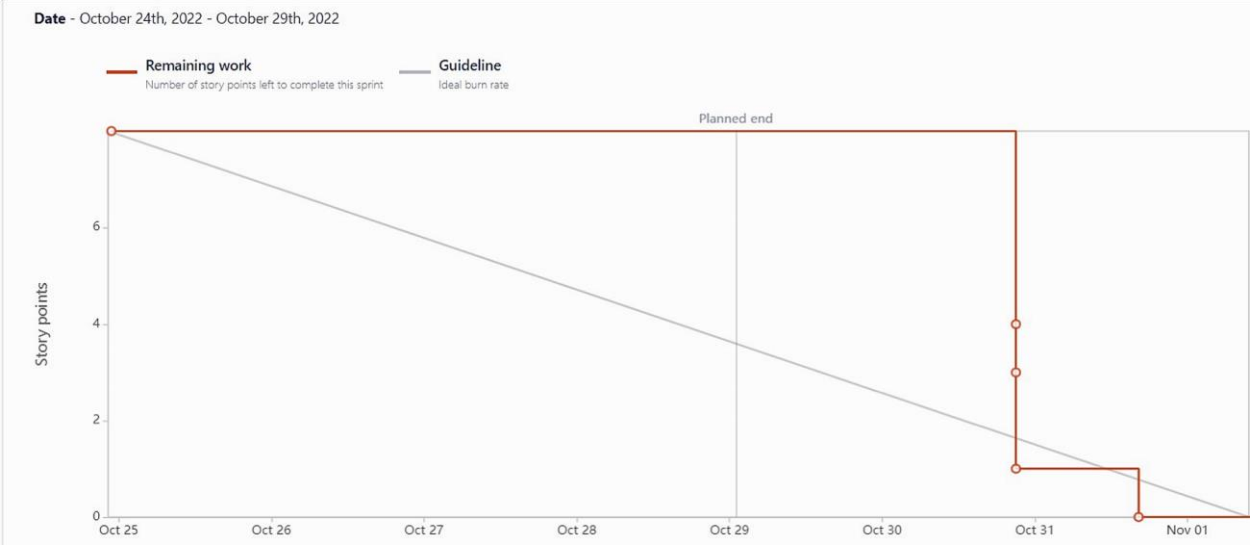
The user will be provided with a manual or guide to help them navigate and access different features of website throughout the process. The manual is prepared in such a way that any user can easily understand the flow of process by looking at the visuals itself. In addition to model developing we have also deployed the model in cloud using IBM Watson. This can can help user to access the website as a service from IBM cloud.

6.2 Sprint Delivery Schedule

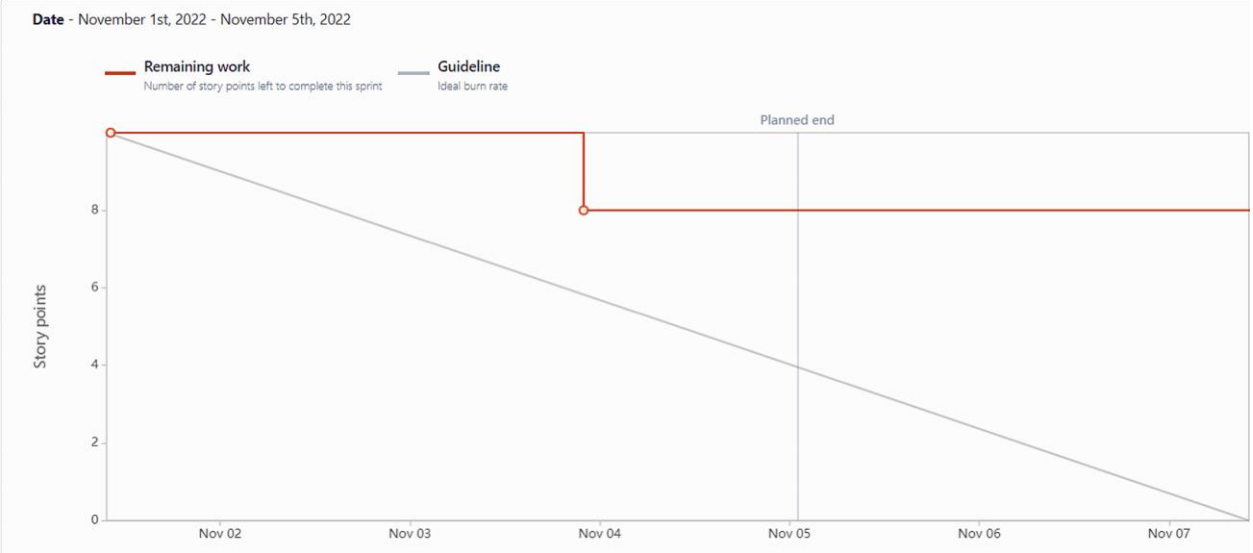
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	8	6 Days	24 Oct 2022	29 Oct 2022	8	28 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	13	6 Days	07 Nov 2022	12 Nov 2022	13	11Nov2022
Sprint-4	13	6 Days	14 Nov 2022	19 Nov 2022	13	19 Nov 2022

6.3 Reports from Jira

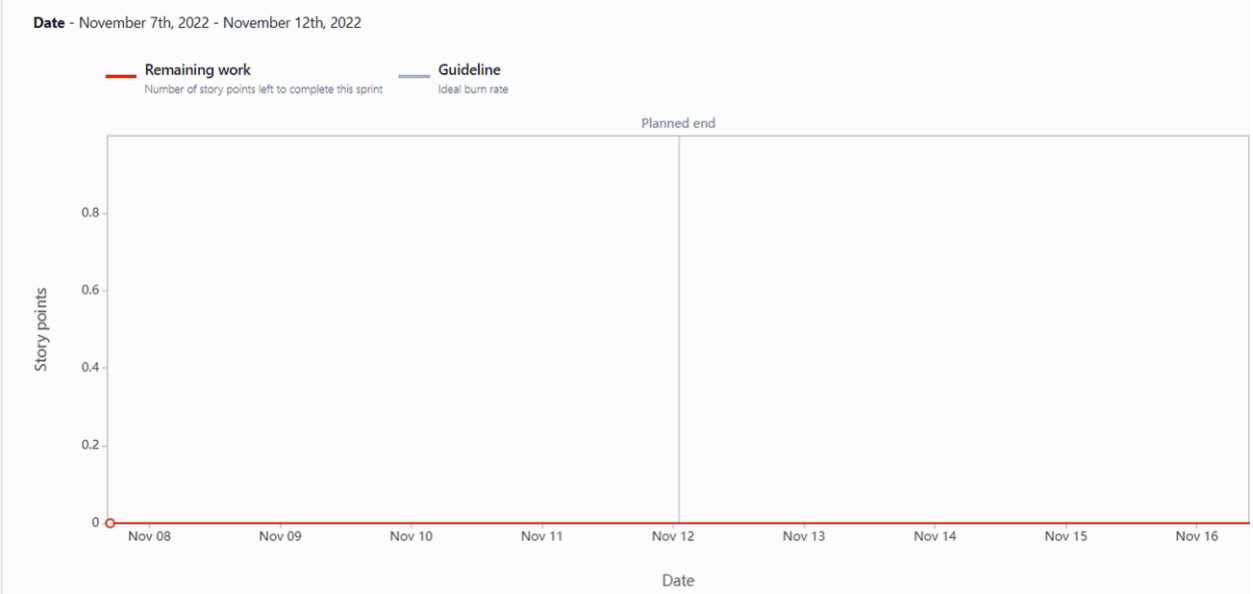
Sprint 1:



Sprint 2:



Sprint 3:


Date - November 7th, 2022 - November 12th, 2022
Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Planned end
Story points
0.8
0.6
0.4
0.2
0
Nov 08
Nov 09
Nov 10
Nov 11
Nov 12
Nov 13
Nov 14
Nov 15
Nov 16
Date

Sprint 4:


Date - November 14th, 2022 - November 19th, 2022
Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Story points
0.8
0.6
0.4
0.2
0
12:00 15 Nov
12:00 16 Nov
12:00 17 Nov
12:00 18 Nov
12:00 19 Nov
Date

7.CODING & SOLUTIONING

The project is intended to develop a website application for fuel consumption prediction. The application provides users with accurate prediction along with few features that users can make use of with great ease.

7.1 User Dashboard

To make users convienient to the website an user dashboard has been created where the progress of user will be managed. The user needs to register/login to the website inorder to access the dashboard. This dashboard consists of user profile, prediction tab, prediction history with a simple interface.

```
export default function Dashboard({ children }) {
    return (
       <Box sx={{bgcolor: "#F3EFEA"}}>
           <Header />
           <Grid container sx={{ minHeight: 'calc(100vh - 48px)'}}>
               {/* <Grid item xs={12}>
               </Grid> */}
               <Grid item container>
                   <Grid
                       item
                       xs={12}
                       md={3}
                       lg={1.5}
                       paddingLeft="10px"
                       sx={{
                           borderRight: {      You, 3 days ago • updated code …
                               sx: "none",
                               // s: "3px solid black",
                               // md: "3px solid black",
                               lg: "3px solid black",
                               height: "95%",
                           },
                       }}
                   >
                       <Sidebar />
                   </Grid>
                   <Grid item xs={12} md={9} lg={10} sx={{ height: "100%" }}>
                       <Outlet />
                   </Grid>
               </Grid>
           </Grid>
       </Box>
    );
}
```

7.2 Single and Multiple Sample Prediction

The prediction tab is designed to facilitate two types of uses.

Single sample prediction tab for user to predict the consumption for a single vehicle by entering the inputs manually.

Multi sample prediction tab for user who wants to predict for multiple vechicle simultaneously espeacially for oragnisation with large fleet vehicles. The user needs

to upload the input file containing input for each vehicle in a table format. This will deliver them with the same table that of input along with prediction column added at the end of the table.

```
return (
    <Grid container spacing={3} sx={{ marginBottom: 10 }}>
        <Grid item xs={12}>⋯
        </Grid>

        <Grid item xs={12}>
            <FormBox title="Multiple Sample Prediction" titleAction={<FormBoxAction/>}>
                <Grid container spacing={3}>
                    <Grid item xs={12}>
                        <FileUploader
                            handleChange={handleFileChange}
                            name="csvFile"
                            types={["CSV"]}
                            label="Upload or drop your CSV file here"
                            fileOrFiles={csvFile}
                            maxSize={3}
                        />
                    </Grid>
                    {csvFile && (
                        <Grid item xs={12}>
                            <Typography>{csvFile.name}</Typography>
                        </Grid>
                    )}
                    <Grid item xs={12}>
                        <Button variant="contained" disabled={csvFile === null} onClick={handleSubmit}>
                            Submit
                        </Button>
                        <Button
                            variant="outlined"
                            disabled={csvFile === null}
                            sx={{ marginLeft: 2 }}
                            onClick={() => {
                                setPredictedValues(null);
                                setCsvFile(null);
                            }}
                        >
                            Clear
                        </Button>
                    </Grid>
                </Grid>
            </FormBox>
        </Grid>
        {predictedValues && (
            <Grid item xs={12}>
                <PredictionTable data={predictedValues} onDownloadReport={handleDownloadReport}/>
            </Grid>
        )}
    </Grid>
);
```

7.3 Prediction History

Inorder to provide the user with better accessibility, we have included a prediction history tab which will allow user to access their recent predictions. This also

increases the usability of application. All the data related to prediction will be stored in the database and can be fetched to provide the prediction history.

```
export default observer(function SingleSamplePrediction() {
    const { enqueueSnackbar } = useSnackbar();
    const { dismissOnlyAction } = useSnackBarAction();
    const [show, setShow] = useState(false);
    const [predictedValue, setPredictedValue] = useState(0);
    const [id, setId] = useState(null);
    const [imperial, setImperial] = useState(false);
    const initialValues = {
        vehicleNumber: '',
        mileage: 0,
        payloadWeight: 0,
        vehicleWeight: 0,
        distance: 0,
        fuelType: "",
        roadType: "",
        averageSpeed: 0,
    };

    const registrationRegex = /[A-Za-z][A-Za-z]\s([0-9][0-9])\s[A-Za-z][A-Za-z]\s[0-9]

    const validationSchema = Yup.object({
        vehicleNumber: Yup.string().required("Vehicle Number is required").matches(reg
        mileage: Yup.number().min(1, "Mileage should be higher than 1").max(10, "Milea
        payloadWeight: Yup.number().min(0, "Payload Weight should be higher than 0"),
        vehicleWeight: Yup.number()
            .min(0.1, "Vehicle Weight should be higher than 0.1")
            .required("Vehicle Weight is required"),
        distance: Yup.number().min(1, "Distance should be higher than 1"),
        fuelType: Yup.string().required("Fuel Type is required"),
        roadType: Yup.string().required("Road Type is required"),
        averageSpeed: Yup.number().min(1, "Average Speed should be higher than 1").req
    });

    const predictSingleSample = (values) => {
        const payload = {
            mileage: parseInt(values.mileage),
            vehicleNumber: values.vehicleNumber,
            payloadWeight: parseInt(values.payloadWeight),
            vehicleWeight: parseInt(values.vehicleWeight),
            distance: parseInt(values.distance),
            fuelType: values.fuelType,
            roadType: values.roadType,
            averageSpeed: parseInt(values.averageSpeed),
        };
        apiModal.singleSamplePrediction(payload, function (res, success) {
            if (success) {
                setShow(true);
```

Screenshot

7.4 Display Current Fuel Price

The application will display the Fuel price (in rupees) updated everyday automatically. This will help user to be aware of the current fuel rate

variations. It is also used to provide the estimated cost of the fuel consumption given by the prediction model. It reduces the manual calculation of fuel expenses.

```
        <FormBox title="Today's Fuel Price" titleStyle={{ fontSize: 18 }} titleAction={<ImperialSwitch />}>
            <Grid container spacing={4}>
                <Grid item xs={12} display="flex" alignItems="center">
                        <Typography sx={{fontSize: 17}}>Current Price: </Typography>
                        <Typography sx={{pl: 1, fontSize: 18, fontWeight: 600}}>Rs. {imperial ? getImperialGallonPrice(getFuelPrice())
                    {/* <Box paddingLeft={1}>
                        <Typography>
                            {'Rs. ' + (imperial
                                ? `${getImperialGallonPrice(getFuelPrice() * fuelQuantity)}`
                                : `${lodash.round(getFuelPrice() * fuelQuantity, 2)}`)}
                        </Typography>
                    </Box> */}
                </Grid>
                <Grid item xs={10}>
                    <TextField
                        name="quantity"
                        value={fuelQuantity}
                        onChange={handleFuelQuantityChange}
                        type="number"
                        inputProps={{ min: 0 }}
                        placeholder={`Enter Fuel Quantity in ${imperial ? "Gallon" : "Litre"} `}
                        variant="standard"
                        fullWidth
                    />
                </Grid>
                <Grid item xs={2}>
                    <IconButton onClick={handleClose}>
                        <Clear />
                    </IconButton>
                </Grid>
                {show && (
                    <Grid item xs={12} display='flex'>
                        <Typography>Total Price is: </Typography>
                        <Typography paddingLeft={1}>
                            {imperial
                                ? getImperialGallonPrice(getFuelPrice() * fuelQuantity)
                                : lodash.round(getFuelPrice() * fuelQuantity, 2)}
                        </Typography>
                    </Grid>
                )}
            </Grid>
        </FormBox>
    );
});
```

Screenshot

7.5 Detailed Report Generation

A detailed report of the prediction done by the user will be generated to give a great clarity to the user about the process.

The report consists of

- User name
- Date of prediction
- Input values
- Prediction result
- Fuel consumption estimation(in rupees)
- Visual representation of Consumption

The report will be available offline where the user can download and make use of it for storing and sharing

```
getMultipleSamplePredictionReport = (id, callback) => {
    axiosClient
        .get(EndPoints.multipleSamplePredictionReport + "/" + id, { responseType: "a
        .then((res) => {
            const url = window.URL.createObjectURL(new Blob([res.data]));
            const link = document.createElement("a");
            link.href = url;
            link.setAttribute("download", `report.pdf`);
  You, 3 days ago • updated code …
            // Append to html link element page
            document.body.appendChild(link);

            // Start download
            link.click();

            // Clean up and remove the link
            link.parentNode.removeChild(link);
            if (callback) callback(res, true);
        })
        .catch((err) => {
            if (callback) callback(err, false);
        });
};
```

8. TESTING

1. Test Cases

TestCase	Features	Description	Steps to Execute	Expected Results
LoginPage_FCP_001	Functional	verify user is redirected to login page when not logged in	1.Enter URL and click go	Application should be redirected to login page
LoginPage_FCP_002	UI	Verify user is able to see the login box	1.Enter URL and click go	Application should show below UI elements: a.email text box b.password text box c.Login button d.New User? Signup

LoginPage_FCP_003	Functional	Verify user is able to see the signup box	1.Enter Url and click go 2.Enter valid credential 3.Click on the login button button	Application should be able to login and redirected to dashboard page
LoginPage_FCP_004	Functional	Verify user is able to sign in with google	1. Enter URL and Click go 2. Click on signin with google button 3. Give accessto the google account	Application Should redirect to google oauth page and be able to allow access
LoginPage_FCP_005	Functional	Verify the error message is received if the email does not exist	1.Enter URL and Click go 2.Enter the wronguser credential	Application should display an error message saying that user does notexist

SignUpPage_FCP_006	UI	Verify user is able to see the sign upbox	1.Ent erURL and Click go	Application should show below UI elements: a.email text box b.password textbox c.mobiletextbox d. nametextbox e.SignupButton
SignUpPage_FCP_007	Functional	Verify user is able to sign up with credentials	1. Enter URL and Click go 2. Enter the user details 3. Click on the signup page 4. Click on register button 5.	Application should register the user and send otp if the entered email is valid

Prediction_FCP_008	Functional	Verify user is able to enter input details for Single Sample Prediction	1. Enter URL and click go 2. Click on Prediction Button on Side Menu 3. Click on Single Sample Prediction Tab 4. Enter valid inputs 5. Click on Predict button 6.	Application should display a new box showing the predicted value and a button to download the report
Prediction_FCP_009	Functional	Verify user is able to enter input details for Multi Sample	1. Enter URL and click go 2. Click on Prediction Button on Side Menu	Application should display a table with predicted values

8.2 User Acceptance Testing

1. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level and how they were resolved

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
ByDesign	5	4	2	3	14
Duplicate	1	0	0	0	1
External	2	3	0	1	6
Fixed	1	4	3	2	10
NotReproduced	0	0	1	0	1
Skipped	0	0	1	0	1
Won'tFix	0	1	0	1	2
Totals	9	11	7	7	35

2.Test Case Analysis

Section	TotalCases	NotTested	Fail	Pass

Authentication	9	0	0	9
Prediction	2	0	0	2
PredictionHistory	2	0	0	2
ReportGeneration	2	0	0	2
ProfileUpdation	1	0	1	0

9. RESULTS

9.1 Performance metrics

S.No.	Parameter	Values
1.	Metrics	**RegressionModel:** MAE-8.203 MSE-128.398 RMSE-11.331 R2score-0.721
2.	Tune the Model	**Hyperparameter Tuning**-0.929 Validation Method-CrossValidation

Performance Metrics

```
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
```

```
mae = mean_absolute_error(y_test, pred)
mae
```

```
8.203867369235024
```

```
mse= mean_squared_error(y_test,pred)
mse
```

```
128.39806863056307
```

```
rmse = np.sqrt(mse)
rmse
```

```
11.331287156830996
```

```
r2_score(y_test,pred)
```

```
0.721307989574244
```

Hyperparameter tuning

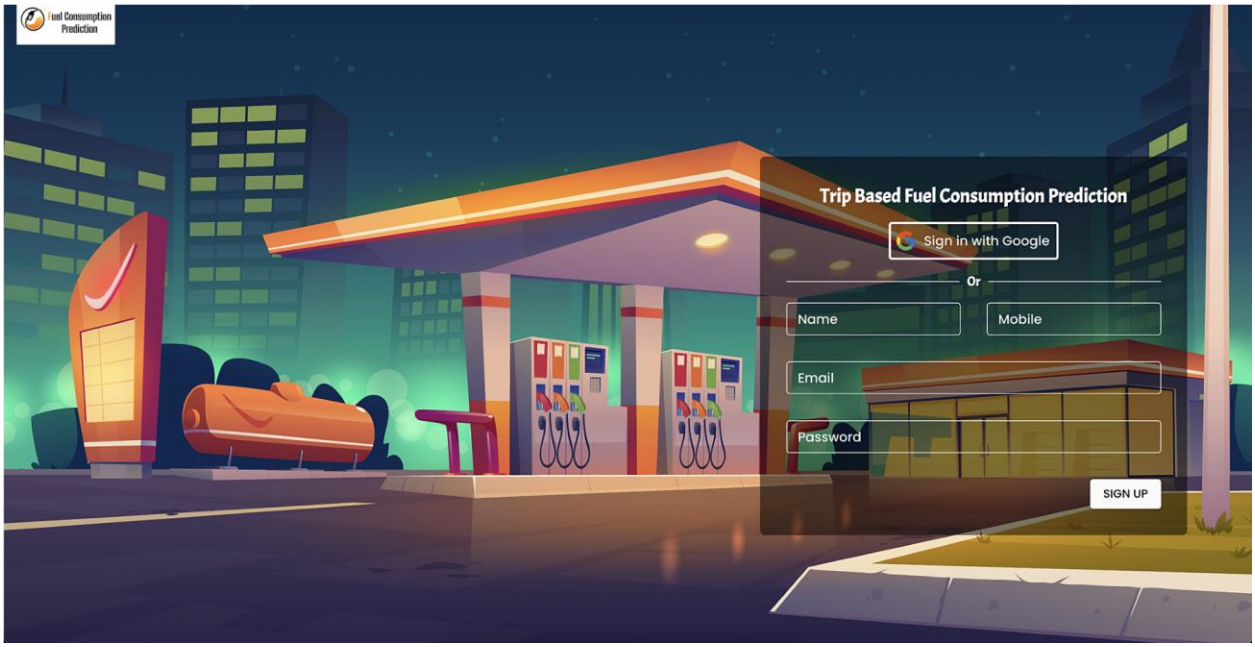
```
[252]: rcv.score(x_test, pred)

[252]: 0.9292735740693612
```

Validation

```
: from sklearn.model_selection import cross_val_score

: lin_score = cross_val_score(rcv, x, y, cv=5, scoring="neg_mean_absolute_error")

  Fitting 5 folds for each of 1 candidates, totalling 5 fits
  Fitting 5 folds for each of 1 candidates, totalling 5 fits
  Fitting 5 folds for each of 1 candidates, totalling 5 fits
  Fitting 5 folds for each of 1 candidates, totalling 5 fits
  Fitting 5 folds for each of 1 candidates, totalling 5 fits

: lin_score_n = -lin_score
  lin_score_n

: array([8.75539133, 9.01726042, 8.87656065, 8.67217387, 8.75847804])

: np.mean(lin_score_n)

: 8.815972860563146
```

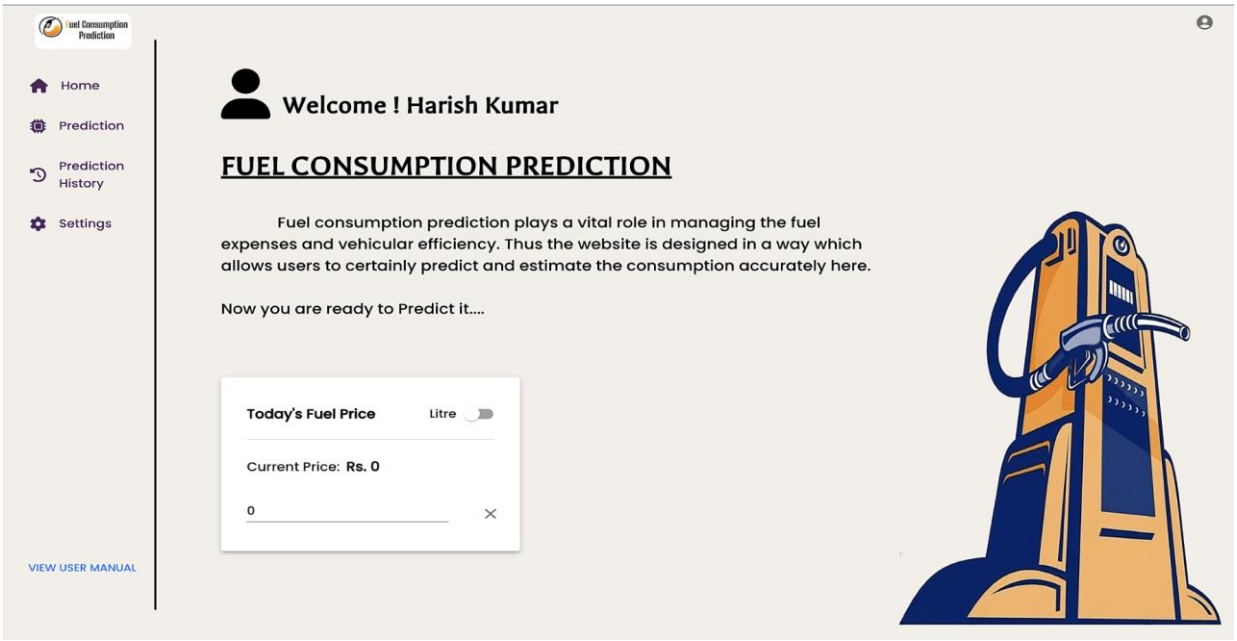
Outputs:

Signup

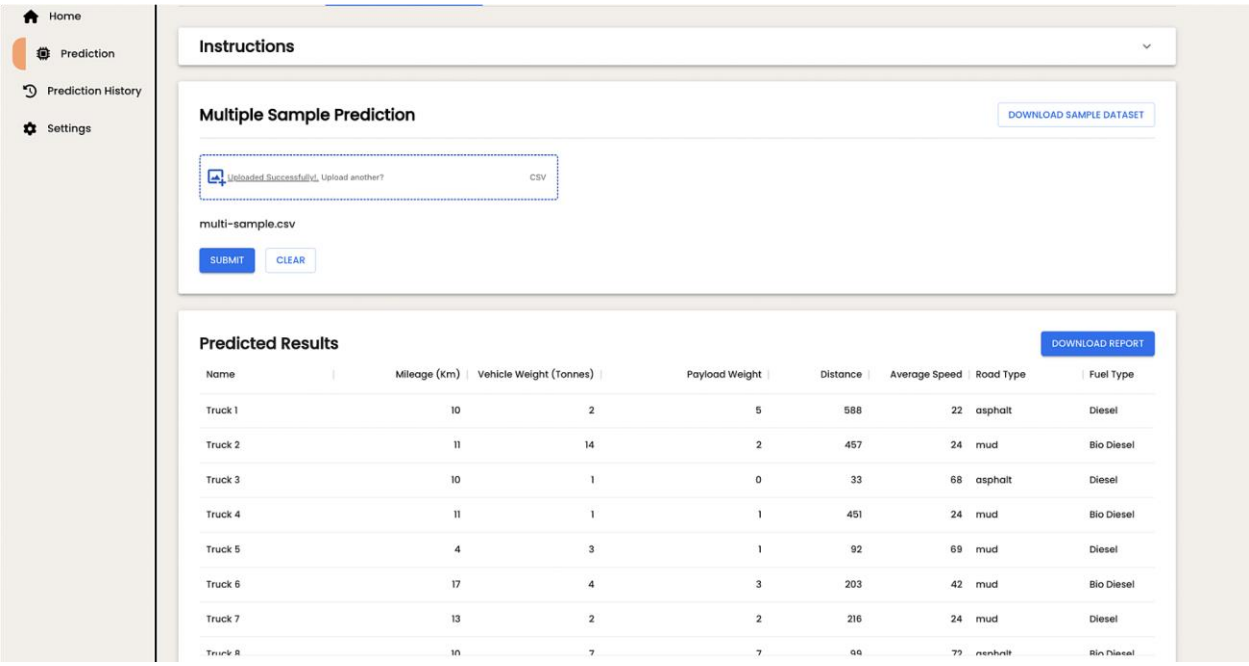


Dashboard

Trip - Based Fuel Consumption Prediction

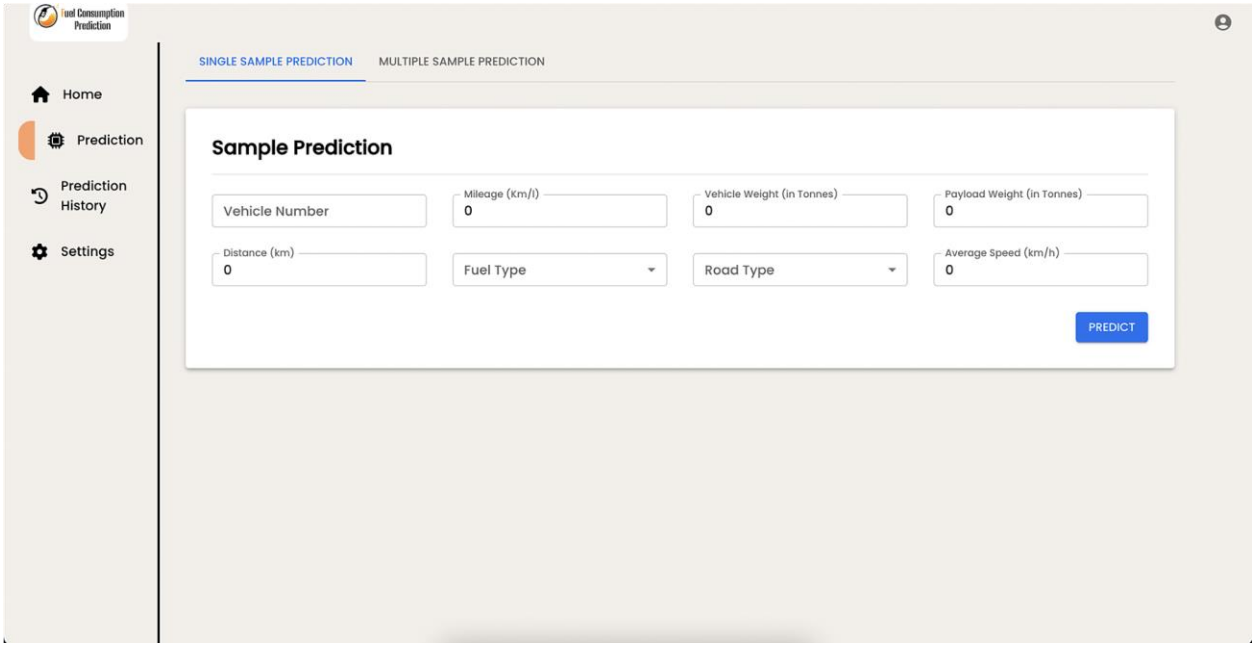


Multiple Sample Prediction

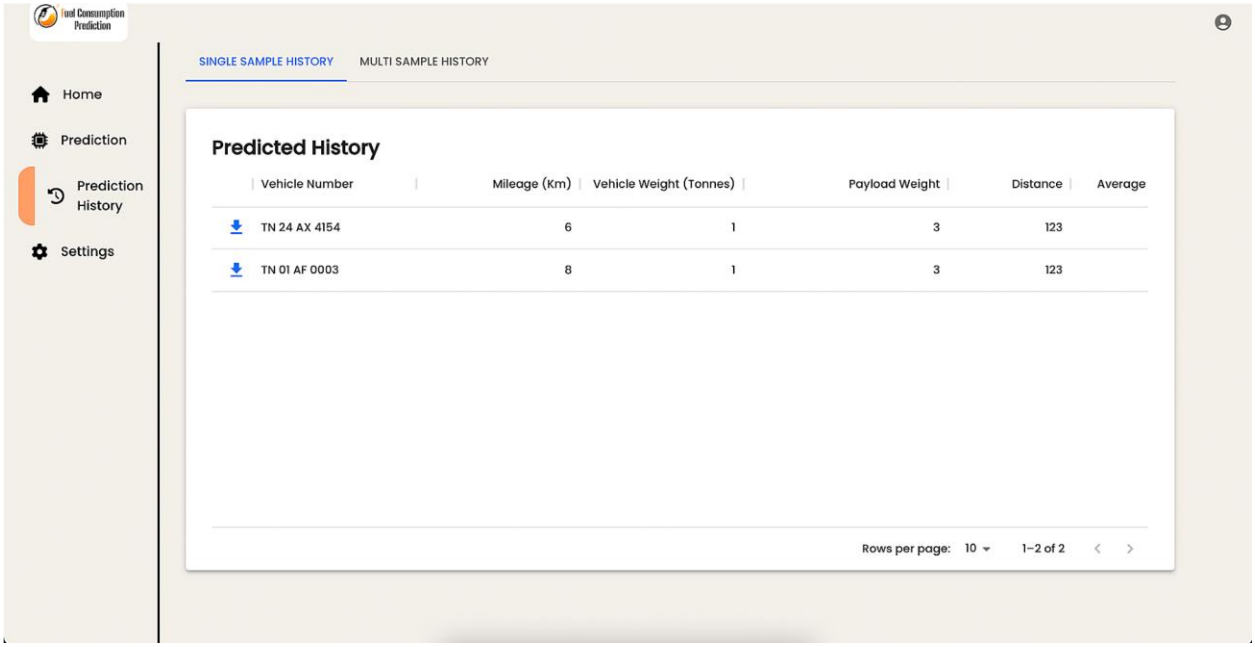


Single Sample Prediction

Trip - Based Fuel Consumption Prediction



Single Sample History


Fuel Consumption Prediction
Home
Prediction
Prediction History
Settings
SINGLE SAMPLE HISTORY
MULTI SAMPLE HISTORY
Predicted History
Vehicle Number
Mileage (Km)
Vehicle Weight (Tonnes)
Payload Weight
Distance
Average
TN 24 AX 4154
6
1
3
123
TN 01 AF 0003
8
1
3
123
Rows per page: 10
1–2 of 2

Report

Fuel Consumption Prediction

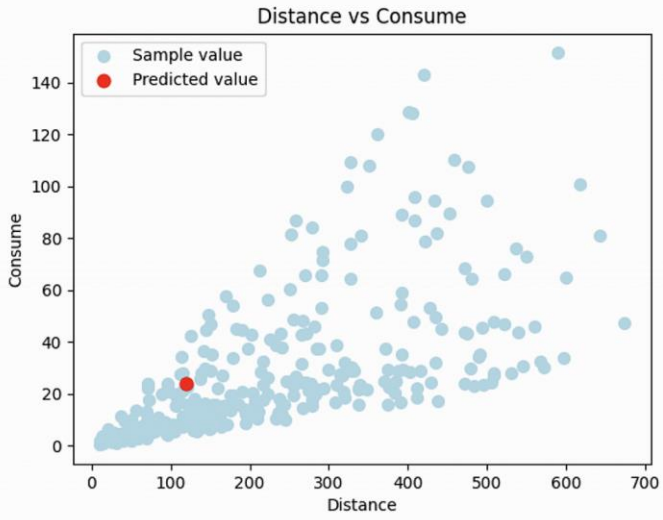
Harish Kumar
18/11/2022

Prediction Report

Vehicle Number:	TN 24 AX 4154	**Average Speed (Km/hr):**	75
Mileage (Km/l):	8	**Payload Weight(Tonnes):**	3
Distance (Km):	120	**Vehicle Weight(Tonnes):**	1
Road Type	asphalt	**Today's fuel price (Rupees):**	94.22
Fuel Type:	Bio Diesel		

Predicted Value (Litres)	**23.85** Rs.2247.26

The Following Scatter plot show the relationship between distance and consumption. The X-Axis shows the distance travelled and the Y-Axis shows the Amount of fuel consumed, The green dot represents the other vehicles, and the red square represents the predicted value.



Page 1/1 ©FCP

10. ADVANTAGES & DISADVANTAGES

1. Advantages

- Easy to use User Interface
- Mobile responsive UI
- Supports both single and multi sample predictions
- Enables users to access the prediction history
- Provides detailed report

2. Disadvantages

- Predicted values are not exact
- Gas types restricted to only two types
- Actual fuel consumption cannot be updated

11. CONCLUSION

Fuel has become an indispensable part of our daily life. The world faces a burgeoning demand of fuel which is expected to continue for quite a longtime.

Therefore it is very important to manage the fuel consumption and reduce the expenses on it. The web application developed ensures to provide accurate fuel consumption prediction and update the user to current demand. It also offers various features to users to keep them intact under the growing fuel scenario. The application proves to be worthy for organisation especially for those in transport and servicing. It is also deployed on cloud making it available as a service to users at an affordable rate. Any individual who wants to predict the fuel consumption for a particular trip can access to this application having a easy to use interface. Hence, the web application is released successfully ensuring complete security and usability.

12. FUTURE SCOPE

This project can be further extended to provide RESTapi, which a developer can use to access the services provided by the application. This allows a developer to integrate the application's services into another application. Organizations can integrate the application into their products. A subscription based payment model can be added, which restricts the free tier users from using Multiple Sample

Predictions and various other features.

13. APPENDIX

13.1 Source Code

models.py class

User(db.Model):

__tablename__ = 'users'

id = db.Column(db.Integer, primary_key=True) name = db.Column(db.String, nullable=False) mobile = db.Column(db.String, unique=True, nullable=True)

email = db.Column(db.String, unique=True, nullable=False) password = db.Column(db.String, nullable=True) isActivated = db.Column("is_activated", db.Boolean, default=False) def serialize(self): return {c: getattr(self, c) for c in inspect(self).attrs.keys()}

class UserMultiplePrediction(db.Model):

__tablename = "users_multiple_prediction" id = db.Column(db.Integer, primary_key=True) userId = db.Column("user_id", db.Integer,

ForeignKey(User.id), nullable=False)

createdAt = db.Column("created_at", db.Date, nullable=False, server_default=func.now())

class UserSinglePrediction(db.Model): __tablename = 'users_single_prediction' id =

db.Column(db.Integer, primary_key=True) vehicleNumber =

db.Column("vehicle_number", db.String, nullable=True) userId = db.Column("user_id",

db.Integer,

ForeignKey(User.id), nullable=False) mileage = db.Column(db.Numeric, nullable=True) fuelType = db.Column('fuel_type', db.String, nullable=True) distance = db.Column(db.Numeric, nullable=True) roadType = db.Column('road_type', db.String, nullable=True)

averageSpeed = db.Column('avg_speed', db.Numeric, nullable=True) vehicleWeight = db.Column('vehicle_weight', db.Numeric, nullable=False) payloadWeight =

db.Column('payload_weight', db.Numeric, nullable=True) predicted = db.Column(db.Numeric, nullable=False)

actual = db.Column(db.Numeric, nullable=True)

userMultiplePredictionId = db.Column("user_multiple_prediction_id", db.Integer, ForeignKey(UserMultiplePrediction.id), nullable=True)createdAt = db.Column("created_at", db.Date, nullable=False, server_default=func.now())

routes.prediction.py

```
@app.route('/predict/single', methods=['POST']) @token_required def
singleSamplePrediction(): data = request.get_json()
  vehicle_number = data['vehicleNumber'] or 'TN XX YY ZZZZ' vehicle_weight =
  data['vehicleWeight'] or 0
  payload_weight = data['payloadWeight'] or 0 mileage = data['mileage'] or 0 distance =
  data['distance'] or 0 avg_speed = data['averageSpeed'] or 0 road_type = data['roadType']
  or 0 fuel_type = data['fuelType'] or 0 userId = g.get('userId')


  predicted = model.predict(
    [[mileage, vehicle_weight, payload_weight, distance, avg_speed, road_type, fuel_type]])
fuelTypeString = dataset.inverse_transform('fuel_type', [fuel_type])[0] roadTypeString =
dataset.inverse_transform('road_type', [road_type])[0] userPrediction =
UserSinglePrediction(userId=userId, vehicleNumber=vehicle_number, mileage=mileage,
fuelType=fuelTypeString, roadType=roadTypeString, distance=distance, averageSpeed=avg_speed,
  vehicleWeight=vehicle_weight, payloadWeight=payload_weight, predicted=float(predicted[0]),
  userMultiplePredictionId=None)


  try:
    db.session.add(userPrediction) db.session.commit()
    return jsonify({'predictedValue': float(predicted[0]), 'id': userPrediction.id}), 200 except
  Exception as except: db.session.rollback()
    print(e)
    return msgResponse("Something went wrong"), 400
```

SinglePrediction.py

```
<form onSubmit={handleSubmit}>
<Grid container spacing={4}>
<Grid item xs={12} sm={6} md={3} xl={3}>
<ValidatedTextField name="vehicleNumber" label="Vehicle
Number" />
(in Tonnes)" />
</Grid>
```

```
<Grid item xs={12} sm={6} md={3} xl={3}>
  <ValidatedTextField name="mileage" label="Mileage (Km/l)" />
</Grid>
<Grid item xs={12} sm={6} md={3} xl={3}>
  <ValidatedTextField name="vehicleWeight" label="Vehicle Weight
</Grid>
<Grid item xs={12} sm={6} md={3} xl={3}>
  <ValidatedTextField name="payloadWeight" label="Payload
Weight (in Tonnes)" />
</Grid>
<Grid item xs={1
<InputLabel size="small" id="fuel-type-select-label"> Fuel Type
</InputLabel> <Select labelId="fuel-type-select-label" id="fuel-type-select"
value={values.fuelType} size="small" name="fuelType" error={touched.fuelType &&
errors.fuelType && true} label="Fuel Type" onChange={handleChange}
>
  {mainstore.metadata.dataSetColumnMapping.fuelType.map((fuelType) => {
    return (
     <MenuItem key={fuelType.name} value={fuelType.value}>
   );
  })
  {fuelType.label}
</MenuItem>
     </Select>
{touched.fuelType && errors.fuelType && (
<FormHelperText
error={true}>{errors.fuelType}</FormHelperText>
)}
</FormControl>
</Grid>
```

```
<Grid item xs={12} sm={6} md={3} xl={3}>
<FormControl fullWidth>
<InputLabel size="small" id="road-type-select-label"> Road Type
</InputLabel>  <Select  labelId="road-type-select-label"  id="road-type-select"  value={values.roadType}
name="roadType" label="Road Type" size="small"
error={touched.roadType && errors.roadType && true} onChange={handleChange} fullWidth>
      {mainstore.metadata.dataSetColumnMapping.roadType.map((roadType) => { return (
                        <MenuItem key={roadType.name}

          value={roadType.value}>
      );
  })}
    {roadType.label}
</MenuItem>
</Select>
{touched.roadType && errors.roadType && (
<FormHelperText
error={true}>{errors.roadType}</FormHelperText>
)}
</FormControl>
</Grid>
<Grid item xs={12} sm={6} md={3} xl={3}>
<ValidatedTextField name="averageSpeed" label="Average Speed
          (km/h)" />
          </Grid>
          <Grid item container xs={12} justifyContent="flex-end" alignItems="center">
             <Grid item>
                <Button variant="contained" type="submit"> Predict </Button>
             </Grid>
```

```
</Grid>
</Grid>
</form>
```

MultiSampleHistory.jsx

```
export default function MultiSampleHistory() {
  const [predictionHistory, setPredictionHistory] = useState([]); useEffect(() => {
    apiModal.getMultipleSamplePredictionHistory(function (res, success) { if (success) {
      setPredictionHistory(res.data); }}); }, []);
  return <PredictionTable data={predictionHistory} />;} const PredictionTable = ({ data }) => { const
  [pageSize, setPageSize] = useState(10); const handleDownloadReport = (id, row) => {
    apiModal.getMultipleSamplePredictionReport(id, function(res, success) { if(success){ console.log(res)
})}
  const columns = [
    { field: "sample_count", headerName: "Vehicle Count",type: "number",width: 100, }, { field:
      "created_at", headerName: "Predicted At",type: "number", width: 150,
    }, {
      field: "id",headerName: "",disableColumnMenu: true,sortable: false, renderCell: ({ id, row,
      formattedValue }) => { return (
        <IconButton onClick={() => handleDownloadReport(id, row)}>
          <Download color="primary" />
        </IconButton>
      );
    },
  },
  ];
  return (
    <> <Paper sx={{ height: "500px", padding: 4 }}>
      <Typography variant="h5" fontWeight={600}> Multi Sample Predicted History </Typography>
```

```
<DataGrid rows={data} sx={{ border: "none", }} columns={columns} pageSize={pageSize}
  rowsPerPageOptions={[5, 10, 25, 50]} onPageSizeChange={(pageSize) => setPageSize(pageSize)}
  />
</Paper> </>
);
};
```

Modes/Api/store.js

```
import { observable } from "mobx";

const mainstore = observable({
  userInfo: {userId: "", email: "", name: "",mobile: "",token: "",},
metadata: {dataSetColumnMapping: {fuelType: [ { name: "",  value: "",  label: "", },],
  roadType: [ { name: "",value: "",label: "",}, ],},
  fuelPrice: { id: "",cityId: "", petrol: "",diesel: "", currency: "",date: "",createdAt: 0 changeText:
"",petrolDiff: "", dieselDiff: "",cngDiff: null,cngPrice: null, }, }, userRegistration: {
  otpToken: "", }); export default mainstore;
```

13.2 Github & Demo link

Github - https://github.com/IBM-EPBL/IBM-Project-45492-1660730462

Demo link - https://drive.google.com/file/d/1bF-VhHB3uagiOIzXY8QSHevK6dD3hi74/view?usp=share_link

14. REFERENCES

1. Bin Zhao Et Al - Fuel Consumption And Traffic Emission Evaluation Of Mixed Traffic Flow With Connected Automated VesiclesAt Multiple TrafficScenarios, [January 2022]

2. Young-Rong Kim Et Al - Development Of A Fuel Consumption Prediction Model Based On MachineLearning Using Ship In-Service Data, [January 2021]

3. Mohamed A. Hamed Et Al - Fuel Consumption Prediction Model Using Machine Learning, [November 2021]

4. Ahmet Gurcan Et Al - Fuel Consumption Models Applied To Automobiles Using RealTime Data, [July 2022]

5. Sasanka Katreddi And Arvind Thiruvengadam - Trip-Based Modelling Of Fuel Consumption In Modern Heavy-Duty Vehicles Using Artificial Intelligence, [December 2021]

6. Ivković Ivan S.Et Al - Influence Of Road And Traffic Conditions On Fuel Consumption And Cost For Different Bus Technologies, [January 2019]

7. Tony Sandberg- Linköping University, Master's thesis, Tony Sandberg. Heavy truck modeling for fuel consumption simulations and measurements, [March 2020]

8. Federico Perrotta Et Al - Application Of Machine Learning For Fuel Consumption modeling Of Trucks, [December 2019]

9. Jonas Lindberg- Kth Institute of Technology, School Of ComputerScience And Communications (Csc) - Master's Thesis, Fuel Consumption Prediction For Heavy Vehicles Using MachineLearning On Log Data, [February 2017]

10. A. Nikolaos Peppes, Evgenia Adamopoulou, and Konstantinos Demestichas - vol.4704, Machine LearningApplied to Sensor Data Analysis ed: MDPI: sensors, "Driving

Behaviour Analysis Using Machine and Deep Learning Methodsfor Continuous Streams of Vehicular Data,", [July 2022]