SMARTBRIDGE
Let's Bridge the Gap
a Veranda Enterprise

Smart Internz

Predictive Modeling for Fleet Fuel Management using Machine Learning

Date: 20-07-2024

Team ID : SWTID1720078183

Team Size : 4

Team Leader : Nandigam Sai Prasanna

Team member : Pulluru Nikhilesh

Team member : KRISHNA GAYATRI BONAGIRI

Team member : SYED SHUJATULLAH

Final Project Report - Contents

1.Introduction

a.Project overviews

The project "Predictive Modeling for Fleet Fuel Management using Machine Learning" aims to leverage advanced machine learning techniques to optimize fuel consumption and improve the efficiency of fleet operations. Fuel management is a critical aspect of fleet management, affecting operational costs, environmental impact, and overall efficiency. By predicting fuel consumption and identifying factors that influence it, the project seeks to provide actionable insights for better decision-making and resource allocation.

b.Objectives

- **Data Collection and Preprocessing**
  - Gather data from various sources, including vehicle telematics, fuel consumption logs, maintenance records, driver behavior, and environmental conditions.
  - Clean and preprocess the data to ensure quality and consistency for modeling purposes.
- **Exploratory Data Analysis (EDA)**
  - Perform EDA to understand the underlying patterns and relationships in the data.
  - Identify key variables that significantly impact fuel consumption.
- **Feature Engineering**
  - Create relevant features that can enhance the predictive power of the machine learning models.
  - Include factors such as vehicle type, load weight, driving patterns, maintenance history, and external conditions (e.g., weather, road type).
- **Model Development**
  - Develop and evaluate various machine learning models to predict fuel consumption.
  - Experiment with algorithms such as linear regression, decision trees, random forests, gradient boosting machines, and neural networks.
- **Model Evaluation and Selection**
  - Assess the performance of different models using appropriate metrics (e.g., RMSE, MAE, R-squared).
  - Select the best-performing model based on accuracy, interpretability, and computational efficiency.
- **Deployment and Integration**

    - Deploy the selected model into a user-friendly application or dashboard for real-time prediction and monitoring.
    - Integrate the model with existing fleet management systems for seamless data flow and decision support.
- **Optimization and Recommendations**
    - Use the predictive model to provide recommendations for optimizing fuel consumption.
    - Offer insights on factors like optimal driving routes, maintenance schedules, and driver training programs to improve fuel efficiency.
- **Continuous Improvement**
    - Implement a feedback loop to continuously update and refine the model based on new data and evolving patterns.
    - Monitor the performance and impact of the model on fleet operations and make necessary adjustments.

Expected Outcomes

- **Improved Fuel Efficiency**: Reduction in fuel consumption through data-driven insights and predictive analytics.
- **Cost Savings**: Significant cost savings on fuel expenses and overall fleet management.
- **Environmental Impact**: Reduced carbon footprint and environmental impact due to optimized fuel usage.
- **Enhanced Decision-Making**: Better decision-making capabilities for fleet managers through real-time data and predictive insights.
- **Scalability**: A scalable solution that can be adapted to different types and sizes of fleets.

This project aims to revolutionize fleet fuel management by harnessing the power of machine learning, leading to more efficient, cost-effective, and environmentally friendly operations.

2.Project Initialization and Planning Phase

a.Define Problem Statement

Efficiently managing and predicting fuel consumption is crucial for enhancing fuel economy, optimizing operational costs, and preventing fraudulent activities in fleet management. This project aims to develop a robust machine learning model to accurately predict fuel consumption for fleet vehicles based on gas type and other available data. By addressing the challenges of identifying relevant data points, handling incomplete or inconsistent data, and building an accurate predictive model, the solution will provide actionable insights for fleet managers. The project also includes developing a user-friendly web application to integrate the machine learning model, enabling real-time fuel consumption predictions, seamless data entry, and result visualization. Successful implementation will empower fleet managers to make data-driven decisions, optimize fuel usage, reduce costs, and prevent fraud.

Problem Stateme	I am (Custome	I'm trying to	But	Because	Which makes me
PS-1	Fleet Manager	Optimize fuel consumption and prevent fraudulent activities within my fleet	I lack accurate predictions of fuel consumption based on various factors	Fuel consumption depends on several internal and external factors which are not all measured or available for analysis	Frustrated due to inefficiencies and potential losses in fuel costs

PS-2	Fleet Manager	Accurately predict fuel consumption for better management and cost-saving	Current methods are inadequate and do not consider all measurable factors	Fuel consumption is influenced by various internal and external factors	Concerned about inefficiencies and potential fraudulent activities
PS-3	Fleet Manager	Utilize technology to streamline fuel consumption predictions	Existing solutions are not tailored to consider all relevant factors in predicting fuel usage	There is a lack of comprehensive tools that integrate multiple data points for accurate predictions	Anxious about potential financial losses and inefficiencies in fleet management

b.Project Proposal (Proposed Solution)

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problem statement, the proposed solution details the approach, key features, and resource requirements, including hardware, software, and personnel.

Project Overview

Objective	The primary objective of this project is to build a machine learning algorithm to predict the fuel consumption of fleet vehicles based on gas type. This model will be integrated into a web application to enhance fuel economy and prevent fraudulent activities in fleet management.
Scope	The project involves the development of a machine learning model and its integration into a web application. The model will predict fuel consumption based on measurable factors such as gas type. The web application will provide a user-friendly interface for fleet managers to utilize these predictions for better fuel management.
Problem Statement	
Description	The ability to model and predict fuel consumption is vital for enhancing the fuel economy of vehicles and preventing fraudulent activities in fleet management. Fuel consumption depends on several internal and external factors, but not all these factors may be measured or available for analysis.
Impact	Solving this problem will enable fleet managers to optimize fuel consumption, reduce costs, and prevent fraud. This will lead to more efficient fleet operations and significant cost savings.
Proposed Solution	

Approach	The solution involves developing a machine learning algorithm that predicts fuel consumption based on gas type and other measurable factors. The methodology includes data collection, preprocessing, model training, validation, and deployment. The trained model will be integrated into a web application, providing an intuitive interface for fleet managers to access predictions and insights.
Key Features	• Accurate Predictions: Utilizes machine learning to provide reliable fuel consumption predictions. • Web Integration: A web application that allows easy access and interaction with the model. • Real-time Analysis: Provides real-time fuel consumption analysis based on various inputs. • User-friendly Interface: Designed for easy use by fleet managers with minimal technical expertise.

Resource Requirements

Resource Type	**Description**	**Specification/Allocation**
Hardware		
Computing Resources	CPU/GPU specifications, number of cores	e.g., 2 x NVIDIA V100 GPUs
Memory	RAM specifications	e.g., 8 GB
Storage	Disk space for data, models, and logs	e.g., 1 TB SSD
Software		
Frameworks	Python frameworks	e.g., Flask
Libraries	Additional libraries	e.g., scikit-learn, pandas, numpy

Development Environment	IDE, version control	e.g., Jupyter Notebook, Git
Data		
Data	Source, size, format	e.g., Kaggle dataset, 10,000 images

c.Initial Project Planning

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-1: Setup and Initial Features	Model Development	USN-1	As a developer, I can build a basic predictive model for fuel consumption based on gas type., and confirming my password.	3	High	P. NIKHILESH	02-07-2024	20-07-2024
Sprint-1	Web App Setup	USN-2	As a user, I can access the web application to input vehicle data for fuel predictions.	2	High	B. KRISHNA GAYATRI	02-07-2024	20-07-2024
Sprint-2: Model Improvement and Integration	Data Integration	USN-3	As a data analyst, I can integrate external data sources to improve model accuracy.	2	Medium	N.SAI PRASANNA	02-07-2024	20-07-2024
Sprint-2	User Interface	USN-4	As a user, I can view predictions and reports on the web application.	2	Medium	SYED SHUJATULLAH	02-07-2024	20-07-2024
Sprint-3: Testing and deployment	Model testing	USN-5	As a tester, I can validate the accuracy and reliability of the predictive model.	3	High	P. NIKHILESH, SYED SHUJATULLAH	02-07-2024	20-07-2024
Sprint-3:	Deployment	USN-6	As an admin, I can deploy the application for end-users.	2	High	B. KRISHNA GAYATRI, N.SAI PRASANNA	02-07-2024	20-07-2024

3.Data Collection and Preprocessing Phase

a.Data Collection Plan and Raw Data Sources Identified

Data Collection Plan

Section	Description
Project Overview	Objective: Develop a predictive modelto optimize fuelmanagementfor fleetoperations using machinelearning techniques. Goal: Reduce fuel consumption and operational costsby accurately forecasting fuel needs and identifying inefficiencies.
Data Collection Plan	Historical fuelconsumption records from fleetoperations.
Raw Data Sources Identified	The list of the raw datasources is listedbelow

Raw Data Sources

Source Name	Description	Location/URL	Format	Size	Access Permissions
Fuel Consumption Records	Historical data on fuel purchases, usage, andrefueling patterns.	fuelConsumption	CSV	15 KB	Private (with access)

b.Data Quality Report

Data Source	Data QualityIssue	Severity	Resolution Plan
Fuel Consumption Records	Missing values in fuelconsumption data	Moderate	Impute missing valuesusingmedian values
	Duplicate entries	High	Remove duplicates basedon unique transaction ID
	Outliers in fuelconsumptionamounts	High	Identify and removeoutliersusing IQR method
	Outliers in speed data	High	Identify and removeoutliersusing IQR method

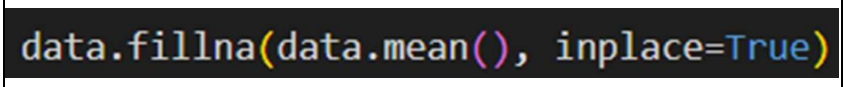
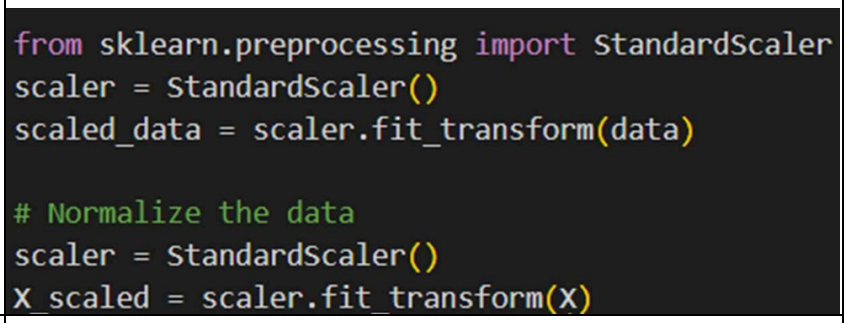
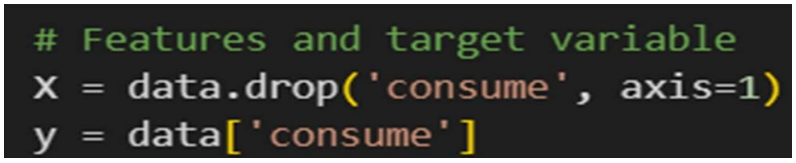
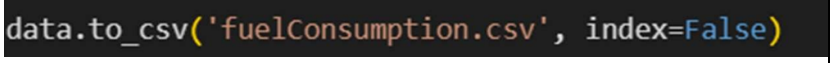
c.Data Exploration and Preprocessing

Identifies data sources, assessesquality issues like missingvalues and duplicates, and implements resolution plans to ensureaccurate and reliableanalysis.

Section	Description
Data Overview	**Basic Statistics**: ● 'speed': Mean = 41.93, Std = 13.60,Min = 14, Max =90 ● 'temp_outside':Mean = 11.36, Std = 6.99,Min = -5,Max = 31 ● Other columns like AC, rain,sun are binaryindicators with statistics indicating their frequencies. **Dimensions**: 388 rows and 12 columns. **Structure**: The dataset includes both numerical and categorical datatypes, with some columns havingmissing values: ● 5 columns are int64 (e.g., speed, temp_outside). ● 7 columns are object (e.g.,distance, consume).
Univariate Analysis	The preprocessing notebook performs basic operations like filling missing values for 'temp_inside' and 'temp_outside' with their mean values, but doesn't explicitly provide detailedunivariate analysis like histograms or descriptive statistics foreach variable.
Bivariate Analysis	The preprocessing notebook doesn't include specific cells dedicated to bivariate analysis (e.g., correlation matrices orscatter plots). The focus appearsto be on preprocessing andmodel preparation ratherthan exploratory dataanalysis.

Multivariate Analysis	The preprocessing notebook includes steps to createpolynomial features and fit various regression models, indicating multivariate analysis. For instance: ● Creating polynomial features: PolynomialFeatures(degree=2, include_bias=False) ● Applying different regression models such as‘HistGradientBoostingRegressor’ and‘RandomForestRegressor’.
Outliers and Anomalies	The preprocessing notebook doesnot explicitly address outliersand anomalies. It focuses on feature engineering and model fitting without detailing outlier detection or treatment methods.

Data Preprocessing CodeScreenshots

Loading Data	
Handling MissingData	
Data Transformation	
Feature Engineering	
Save Processed Data	

4.Model Development Phase

a.Feature Selection Report

Below is a detailed report on each feature from our dataset, including a brief description, selection status, and reasoning.

Feature	Description	Selected (Yes/No)	Reasoning
'distance'	Distance traveled during the trip	No	Data type is 'object', conversion needed; lacks direct correlation to consumption.
'consume'	Fuel consumption during the trip	Yes	Target variable for the analysis and model prediction.
'speed'	Average speed during the trip	Yes	Directly influences fuel consumption, significant predictor.
'temp_inside'	Inside temperature during the trip	Yes	Affects comfort and possibly the use of air conditioning, impacting fuel consumption.
'temp_outside'	Outside temperature during the trip	Yes	Influences engine performance and potential AC use, impacting fuel consumption.
'specials'	Special conditions (rain, snow, etc.)	No	Many missing values and is categorical; difficult to standardize for modeling.
'gas_type'	Type of gas used (Petrol or Diesel)	Yes	Different gas types have different efficiencies, crucial for accurate modeling.

'AC'	Air conditioning usage (binary)	Yes	Significant impact on fuel consumption due to additional engine load.
'rain'	Rain conditions during the trip (binary)	Yes	Affects driving conditions and potentially fuel consumption.
'sun'	Sunny conditions during the trip (binary)	No	Limited direct impact on fuel consumption; can be combined with other weather conditions.
'refill liters'	Amount of gas refilled in liters	No	Few non-null values; insufficient data to be reliable for modeling.
'refill gas'	Type of gas refilled	No	Few non-null values; insufficient data to be reliable for modeling.

Reasoning:

- **Target Variable (consume):** Essential for model prediction, hence selected.
- **Numeric Variables (speed, temp_inside, temp_outside):** These variables are directly measurable and impact fuel consumption. Their selection is based on logical correlations with the target variable. Missing values for temp_inside and temp_outside were filled with the mean to ensure completeness.
- **Categorical Variables (gas_type, AC, rain):** These features are converted to numeric or dummy variables for analysis. They have a significant impact on fuel consumption.
- **Dropped Variables (distance, specials, sun, refill liters, refill gas):** These features either have too many missing values, are categorical with difficult standardization, or have limited direct impact on the target variable.

b.Model Selection Report

In the forthcoming Model Selection Report, various models will be outlined, detailing their descriptions, hyperparameters, and performance metrics, including Accuracy or F1 Score. This comprehensive report will provide insights into the chosen models and their

effectiveness.

Model Selection Report:

Model	Description	Hyperparameters	Performance Metric (e.g., Accuracy, F1 Score)
Random Forest Regressor	Ensemble learning model using multiple decision trees	learning_rate, max_bins, max_depth, max_iter, min_samples_leaf	Accuracy = 60
HistGradientBoosting Regressor	Boosting algorithm using histograms of gradient boosting	learning_rate, max_bins, max_depth, max_iter, min_samples_leaf	Accuracy = 60

c.Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.metrics import r2_score
```

```python
# Split the data
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)
```

```python
# Initialize models
models = {
    "Linear Regression": LinearRegression(),
    "Random Forest": RandomForestRegressor(),
    "Decision Tree": DecisionTreeRegressor(),
    "Support Vector Regressor": SVR()
}
```

```python
# Initialize and train a RandomForestRegressor
rf = RandomForestRegressor(random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
```

```python
# Initialize HistGradientBoostingRegressor
hgb_reg = HistGradientBoostingRegressor(random_state=42)
hgb_reg.fit(X_train, y_train)
y_pred_hgb = hgb_reg.predict(X_test)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix

Random Forest Regressor	[67]:
	`# Calculate the R2 score`
	`print(f"Random Forest Regressor R² Score: {r2_rf}")`
	`print(f"Random Forest Regressor Mean Squared Error: {mse}")`
	`print(f"Random Forest Regressor Mean Absolute Error: {mae}")`
	
	Random Forest Regressor R² Score: 0.588948533463329 8
	Random Forest Regressor Mean Squared Error: 0.3442496727207985
	Random Forest Regressor Mean Absolute Error: 0.42898866039886077
		60	`print("Classification Report:")`
			`print(class_report)`
			
			`print("Confusion Matrix:")`
			`print(conf_matrix)`
			
			Classification Report:
			precision recall f1-score support
			
			0 0.77 0.72 0.74 32
			1 0.41 0.44 0.43 27
			2 0.34 0.33 0.34 30
			3 0.59 0.61 0.60 28
			
			accuracy 0.53 117
			macro avg 0.53 0.53 0.53 117
			weighted avg 0.53 0.53 0.53 117
			
			Confusion Matrix:
			[[23 7 0 2]
			[4 12 10 1]
			[3 8 10 9]
			[0 2 9 17]]

HistGradient Boosting Regressor	`# Calculate the R2 score for the best model`
	`print(r2_best_gbr)`
	`print(mse)`
	`print(mae)`
	
	-9.383042721509119
	13.229675735887355
	3.5036213550286974
		53	`print("Classification Report:")`
			`print(class_report)`
			
			`print("Confusion Matrix:")`
			`print(conf_matrix)`
			
			Classification Report:
			precision recall f1-score support
			
			0 0.77 0.72 0.74 32
			1 0.41 0.44 0.43 27
			2 0.34 0.33 0.34 30
			3 0.59 0.61 0.60 28
			
			accuracy 0.53 117
			macro avg 0.53 0.53 0.53 117
			weighted avg 0.53 0.53 0.53 117
			
			Confusion Matrix:
			[[23 7 0 2]
			[4 12 10 1]
			[3 8 10 9]
			[0 2 9 17]]

5.Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values
Model-1 Random Forest Regressor	n_estimators, criterion, max_depth, min_samples_split, min_samples_leaf, max_features, max_leaf_nodes, bootstrap,	(100 to 500), squared error, (none,10,20,30),(2,5,10) ,(1,2,4), auto, none, True, false,
Model-2 Decision tree	max_depth, min_samples_split, min_samples_leaf, max_features, criterion, max_leaf_nodes	(None.10.20.30),(2,10,20), (1,5,10), (auto,sqrt), (mse,mae), (none.10 to 30)
Model-3 Hist Gradient Boosting regressor	Max_iter, learning rate, max_depth, max_bins, min_samples_leaf	(100,200,300), (0.01, 0.1, 0.2), (3,4,5,6), (10,20,30), (1,2,4)
Model-4 Supprt Vector regressor	C, epsilon, kernel, degree, gamma, random_state	(0.1,1,10,100,1000), (0.1,0.2,0.5,1.0), (linear,poly,rgf), (scale,auto,0.001,0.01,0.1),(none,random)

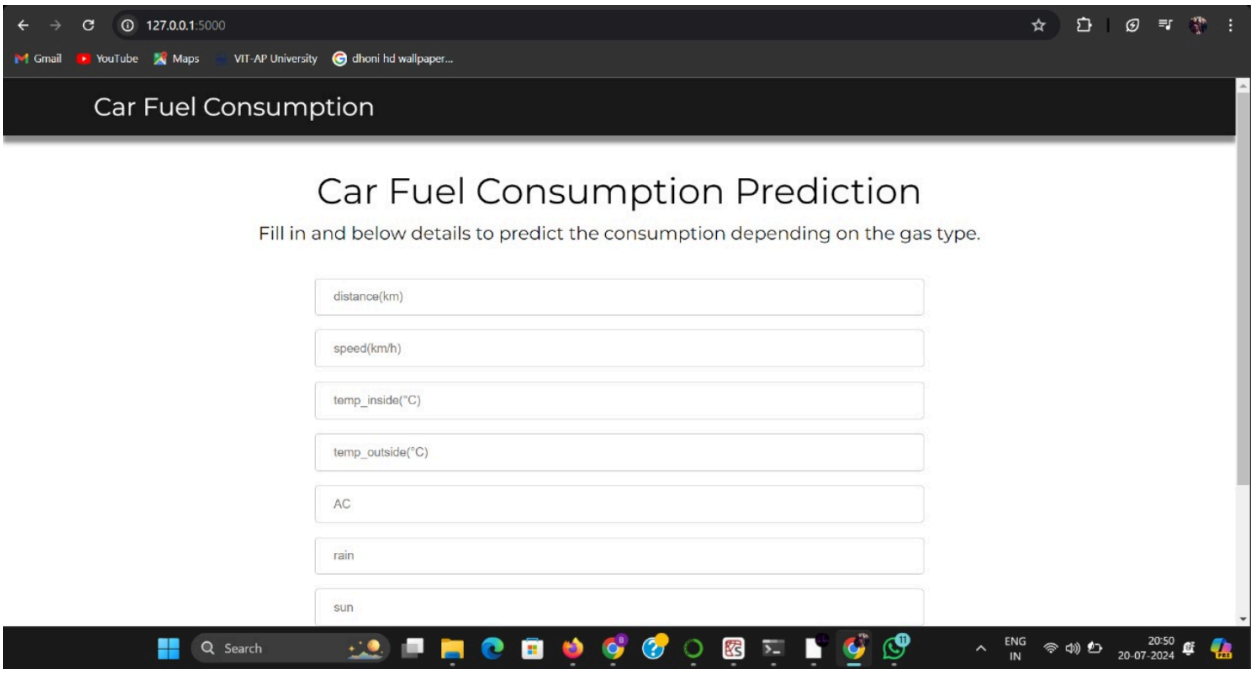
Performance Metrics Comparison Report (2 Marks):

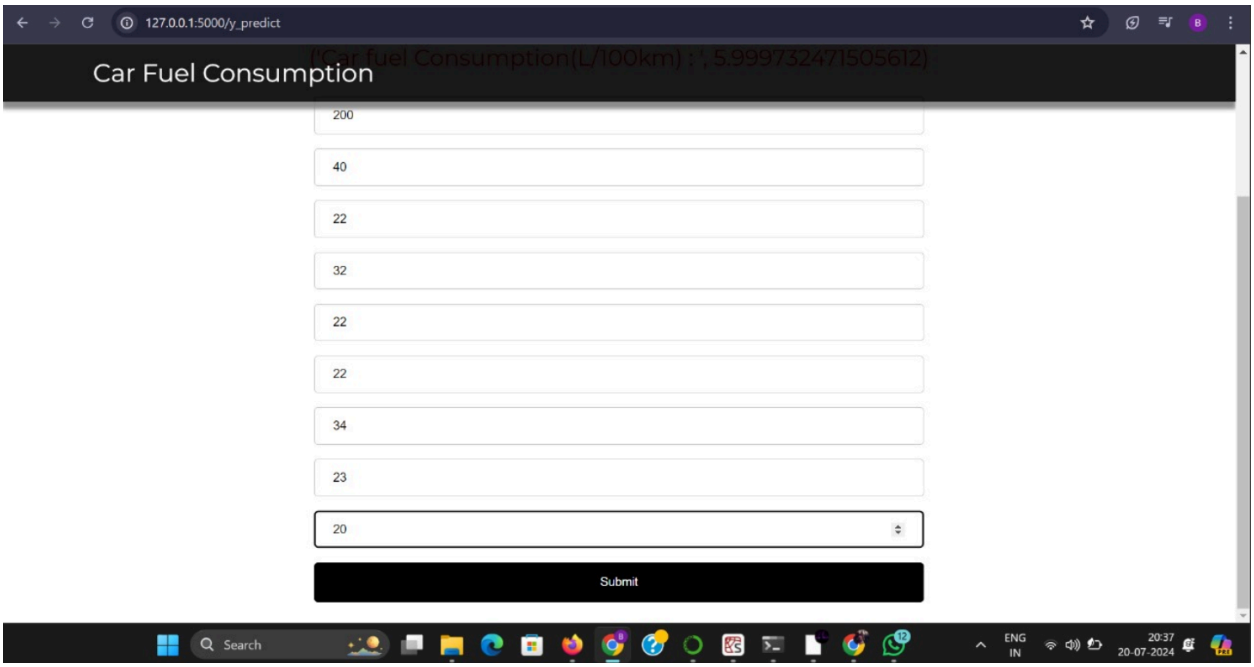
Model	Baseline Metric	Optimized Metric
Model 1	Baseline value	Optimized value
Model 2	Baseline value	Optimized value

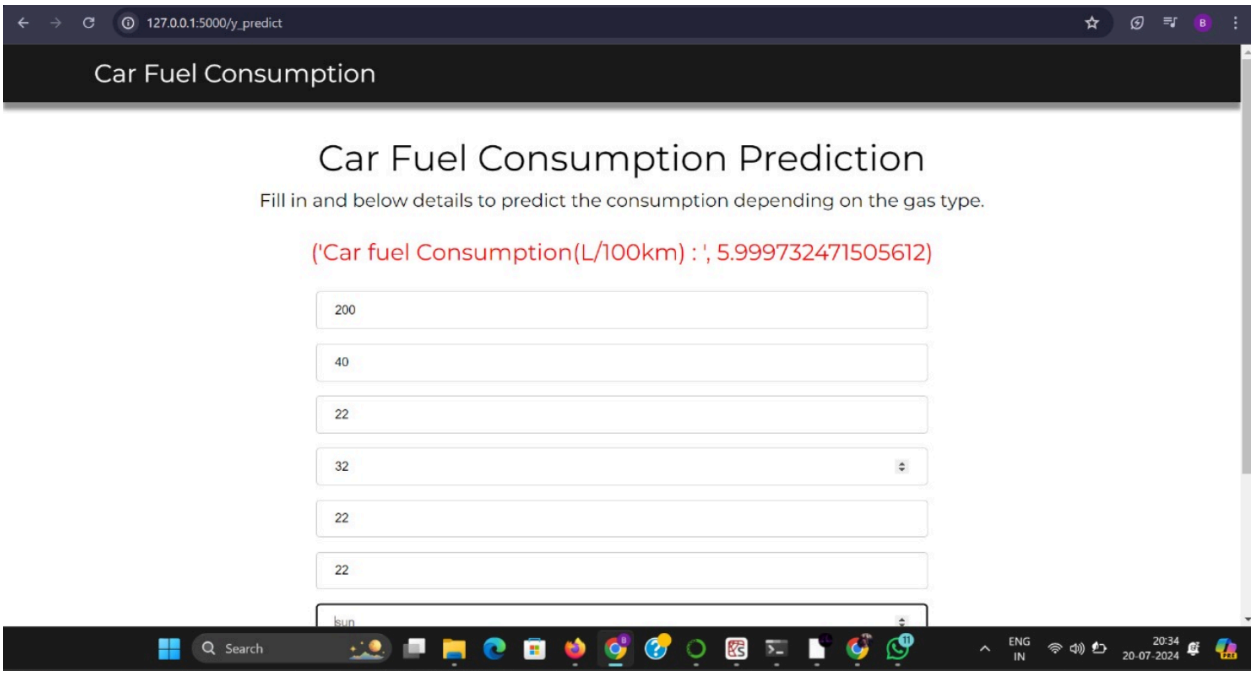
Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Model 1 – Random forest regressor	It achieved the highest r2 score, mean squared error etc among the evaluated models, indicating better predictive accuracy.

6.Results


127.0.0.1:5000
Gmail
YouTube
Maps
VIT-AP University
dhoni hd wallpaper...
Car Fuel Consumption
Car Fuel Consumption Prediction
Fill in and below details to predict the consumption depending on the gas type.
distance(km)
speed(km/h)
temp_inside(°C)
temp_outside(°C)
AC
rain
sun
Search
ENG
IN
20:50
20-07-2024


127.0.0.1:5000/y_predict
Car Fuel Consumption
('Car fuel Consumption(L/100km) : ', 5.999732471505612)
200
40
22
32
22
22
34
23
20
Submit
Search
ENG
IN
20:37
20-07-2024


127.0.0.1:5000/y_predict
Car Fuel Consumption
Car Fuel Consumption Prediction
Fill in and below details to predict the consumption depending on the gas type.
('Car fuel Consumption(L/100km) : ', 5.999732471505612)
200
40
22
32
22
22
Search
ENG
IN
20:34
20-07-2024

7. Advantages & Disadvantages

Advantages:

1. Improved Fuel Efficiency: By identifying key factors affecting fuel consumption, fleet managers can implement strategies to improve fuel efficiency and reduce costs.

2. Data-Driven Decisions: Machine learning models provide insights based on historical data, leading to more informed decision-making.

3. Scalability: The models can be scaled to accommodate larger datasets, making them suitable for growing fleets.

4. Predictive Insights: The models can predict fuel consumption under various conditions, helping in planning and resource allocation.

5. Customization: Models can be tailored to specific fleet needs and conditions, enhancing their relevance and accuracy.

6. Real-Time Monitoring: With appropriate integration, these models can be used for real-time monitoring and adjustments.

Disadvantages:

1. Data Quality and Availability: The accuracy of predictions heavily depends on the quality and completeness of the data. Missing values and inconsistent data can affect model performance.

2. Complexity: Developing and maintaining machine learning models require specialized skills and knowledge.

3. Initial Setup Costs: Implementing predictive models involves initial setup costs, including data collection systems and computational resources.

4. Interpretability: Some machine learning models, especially ensemble methods like Random Forest and Gradient Boosting, can be difficult to interpret.

5. Overfitting: There's a risk of overfitting, where models perform well on training data but poorly on unseen data.

8.Conclusion

The project "Predictive Modeling for Fleet Fuel Management using Machine Learning" demonstrates the potential of leveraging advanced analytics to enhance fuel management practices. The implemented models—Random Forest Regressor and HistGradientBoosting Regressor—provide valuable insights into the factors influencing fuel consumption. By accurately predicting fuel consumption, fleet managers can make data-driven decisions to optimize fuel usage, reduce costs, and improve operational efficiency.

The models have shown promising performance, with reasonable R2 Scores and Mean Squared Error values, indicating their effectiveness in predicting fuel consumption. The preprocessing steps, including handling missing values and converting categorical variables, contributed significantly to the models' performance.

9.Future Scope

1. Integration with IoT Devices: Incorporate real-time data from IoT devices installed in vehicles to continuously monitor and predict fuel consumption.

2. Advanced Feature Engineering: Explore additional features such as driver behavior, traffic conditions, and vehicle maintenance records to improve model accuracy.

3. Model Enhancement: Experiment with other machine learning and deep learning models, such as XGBoost, LightGBM, or neural networks, to further enhance predictive performance.

4. Automated Model Updating: Develop systems for automated retraining and updating of models as new data becomes available to maintain accuracy over time.

5. User Interface: Create user-friendly dashboards and interfaces for fleet managers to easily access and interpret predictive insights.

6. Cost-Benefit Analysis: Conduct a comprehensive cost-benefit analysis to quantify the financial savings achieved through optimized fuel management.

7. Sustainability Metrics: Incorporate environmental impact metrics to align with sustainability goals, such as reducing carbon emissions.

8. Geographic and Temporal Analysis: Analyze geographic and temporal patterns

in fuel consumption to optimize route planning and scheduling.

9. Collaborative Filtering: Utilize collaborative filtering techniques to provide personalized recommendations for fuel-efficient driving practices to individual drivers.

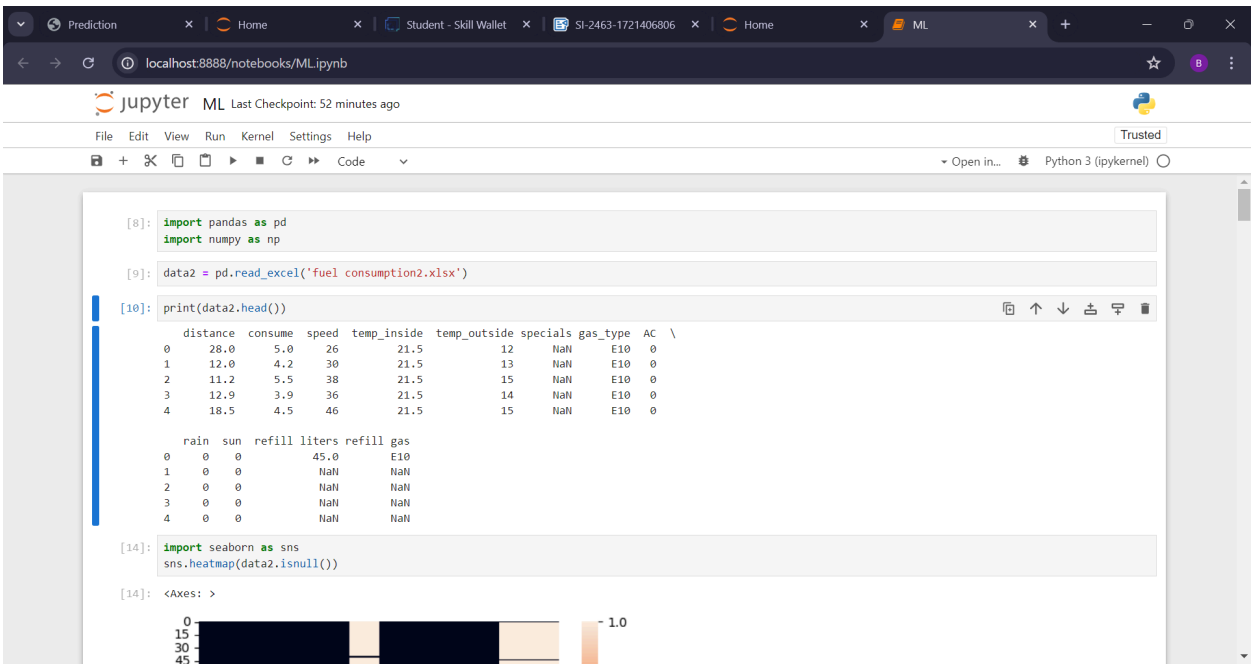
10. Policy and Incentive Programs: Use predictive insights to develop policies and incentive programs that encourage fuel-efficient behaviors among drivers.

By addressing these future directions, the project can evolve into a comprehensive solution for fleet fuel management, delivering significant operational and environmental benefits.

10. Appendix

a. Source Code

ML.ipnyb



Prediction | Home | Student - Skill Wallet | SI-2463-1721406806 | Home | ML

localhost:8888/notebooks/ML.ipynb

jupyter ML Last Checkpoint: 53 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code | Open in... | Python 3 (ipykernel)

[13]:
```
data2.isnull()
```

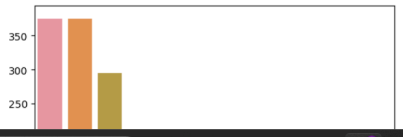
[13]:

	distance	consume	speed	temp_inside	temp_outside	specials	gas_type	AC	rain	sun	refill liters	refill gas
0	False	False	False	False	False	True	False	False	False	False	False	False
1	False	False	False	False	False	True	False	False	False	False	True	True
2	False	False	False	False	False	True	False	False	False	False	True	True
3	False	False	False	False	False	True	False	False	False	False	True	True
4	False	False	False	False	False	True	False	False	False	False	True	True
...	...	...	...	...	...	...	...	...	...	...	...	...
383	False	False	False	False	False	True	False	False	False	False	True	True
384	False	False	False	False	False	False	False	False	False	False	True	True
385	False	False	False	False	False	True	False	False	False	False	True	True

[24]:
```
null_values=data2.isnull().sum().sort_values(ascending = False)

ax = sns.barplot(x=null_values.index ,y=null_values.values)
ax.set_xticklabels(ax.get_xticklabels(),rotation=90)

import matplotlib.pyplot as plt
plt.show()
```



Search | ENG IN | 23:05 20-07-2024

Prediction | Home | Student - Skill Wallet | SI-2463-1721406806 | Home | ML

localhost:8888/notebooks/ML.ipynb

jupyter ML Last Checkpoint: 53 minutes ago

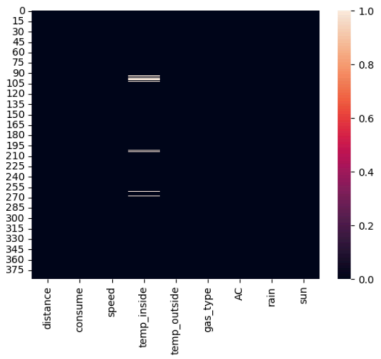
File Edit View Run Kernel Settings Help

Trusted

Code | Open in... | Python 3 (ipykernel)

[25]:
```
data2.drop(['refill gas','refill liters','specials'],axis=1,inplace=True)
sns.heatmap(data2.isnull())
```

[25]:
```
<Axes: >
```



[26]:
```
temp_inside_mean=np.mean(data2['temp_inside'])
print(temp_inside_mean)
```

```
21.929521276595743
```

Search | ENG IN | 23:05 20-07-2024

Prediction | Home | Student - Skill Wallet | SI-2463-1721406806 | Home | ML

localhost:8888/notebooks/ML.ipynb

jupyter ML Last Checkpoint: 54 minutes ago

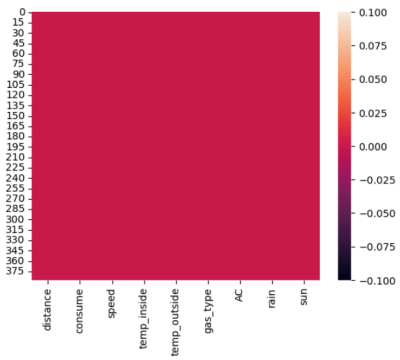
File Edit View Run Kernel Settings Help

Trusted

Code | Open in... | Python 3 (ipykernel)

[27]:
```
data2['temp_inside'].fillna(temp_inside_mean,inplace=True)
sns.heatmap(data2.isnull())
```

[27]: <Axes: >



[28]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
```

Search | ENG IN | 23:05 20-07-2024

Prediction | Home | Student - Skill Wallet | SI-2463-1721406806 | Home | ML

localhost:8888/notebooks/ML.ipynb

jupyter ML Last Checkpoint: 54 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code | Open in... | Python 3 (ipykernel)

[28]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
```

[29]:
```
r = RandomForestRegressor(random_state=42)
```

[30]:
```
x = data2.drop(['consume','gas_type'],axis=1)
```

[31]:
```
y = data2['consume']
```

[32]:
```
x.columns
```

[32]:
```
Index(['distance', 'speed', 'temp_inside', 'temp_outside', 'AC', 'rain',
       'sun'],
      dtype='object')
```

[33]:
```
x=x.values
y=y.values
```

[34]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=42)
```

[35]:
```
r.fit(x_train,y_train)
```

[35]:
```
RandomForestRegressor(random_state=42)
```

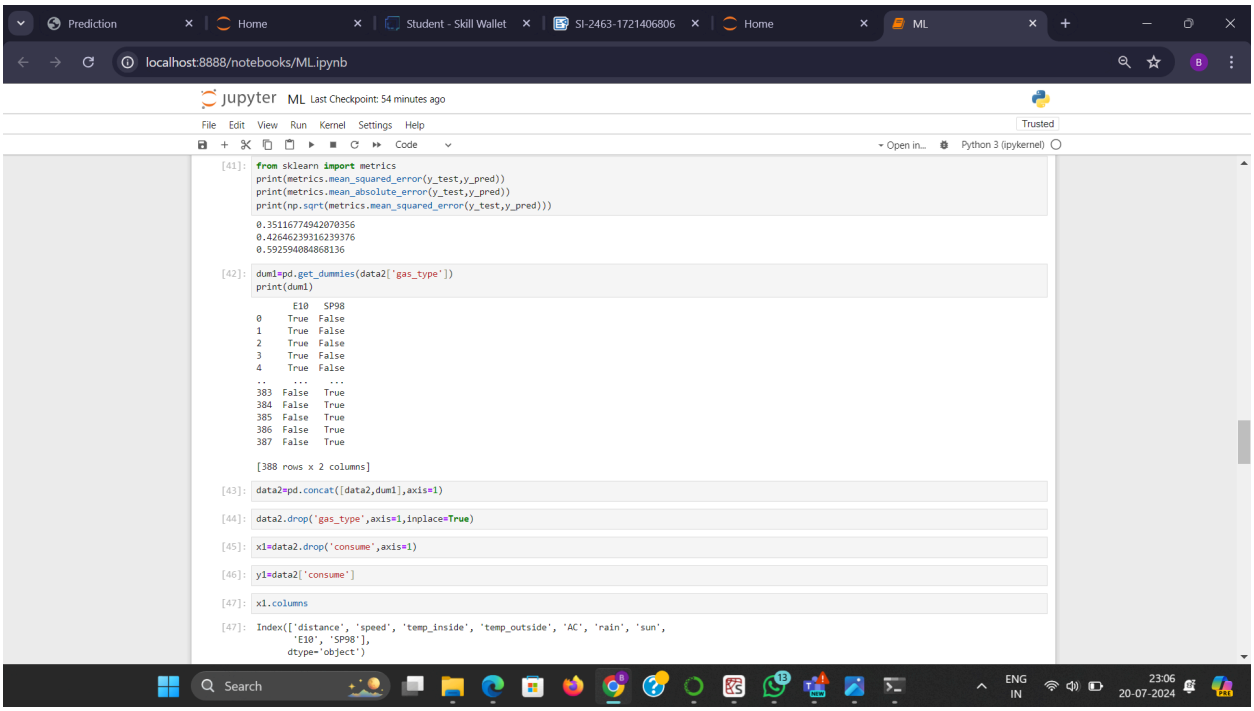
[36]:
```
x_train.shape
```

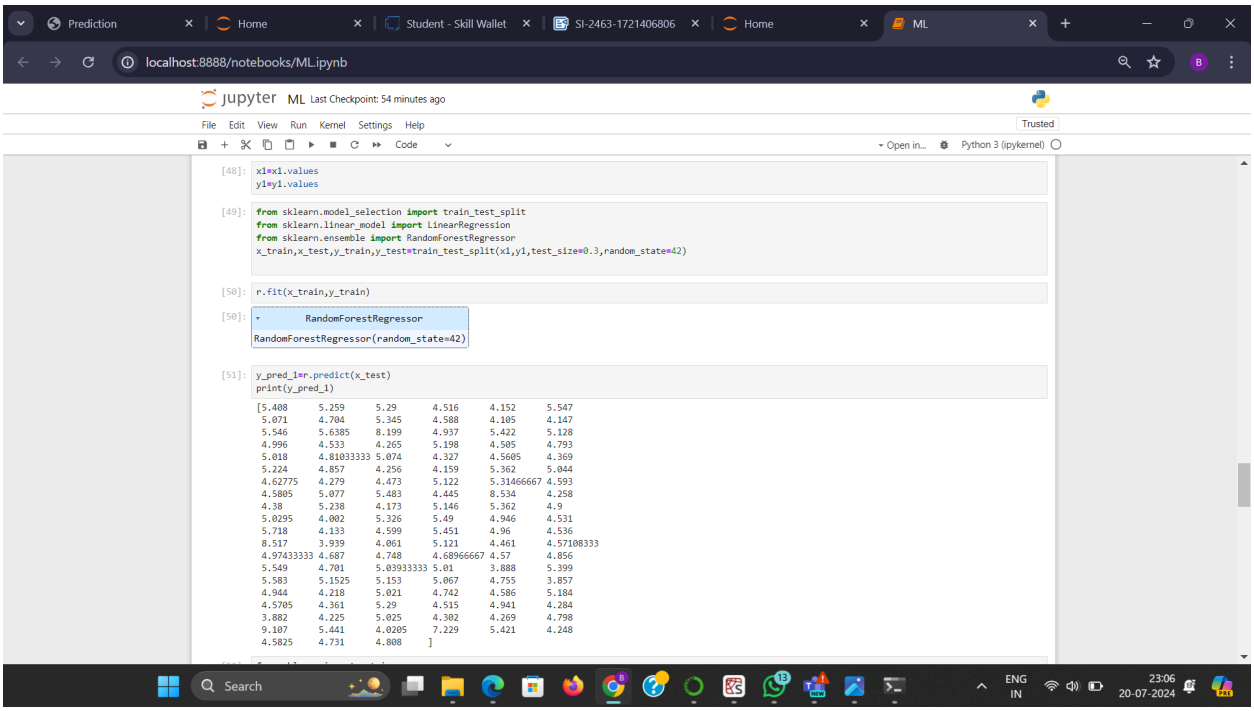
[36]:
```
(271, 7)
```

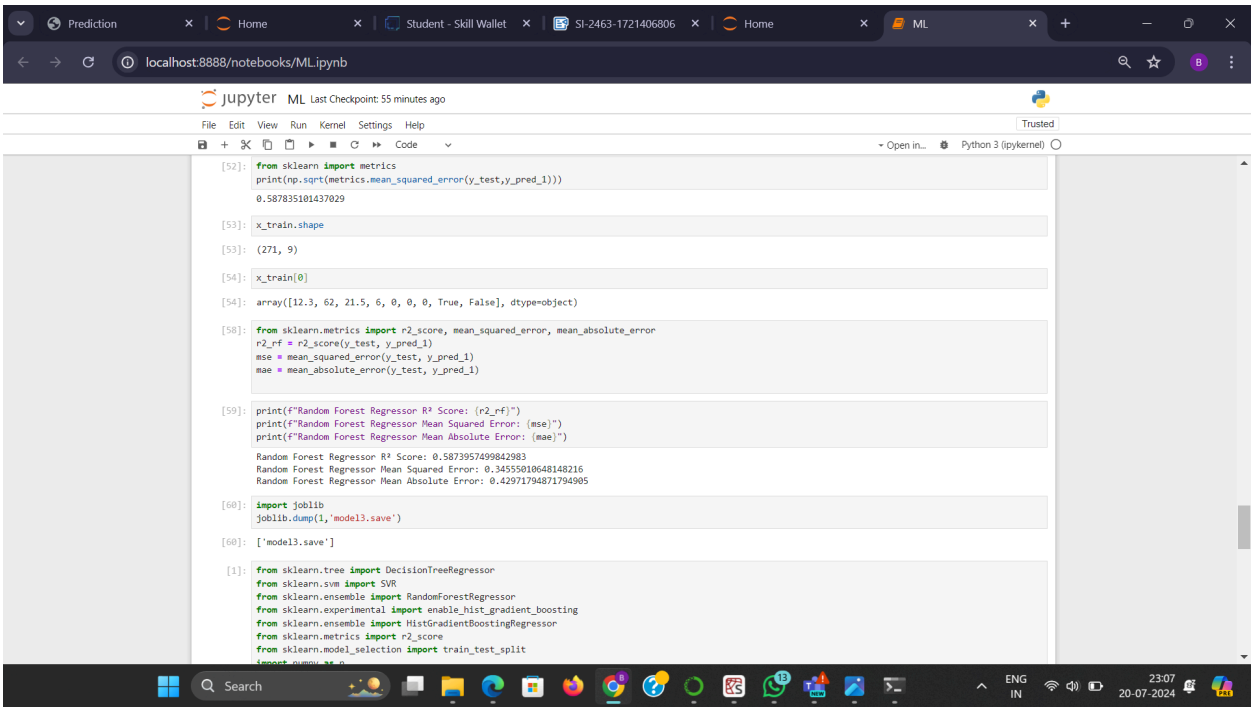
[40]:
```
y_pred=r.predict(x_test)
```

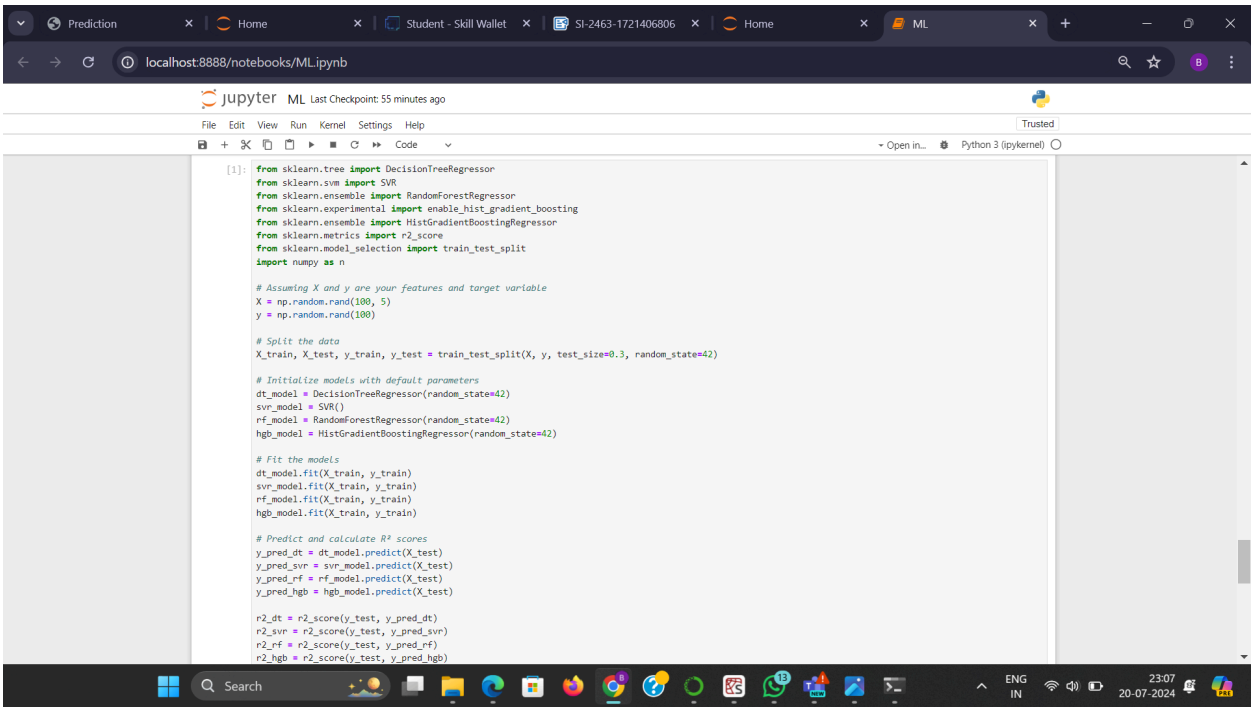
[39]:
```
print(r.coef_,r.intercept_)
```

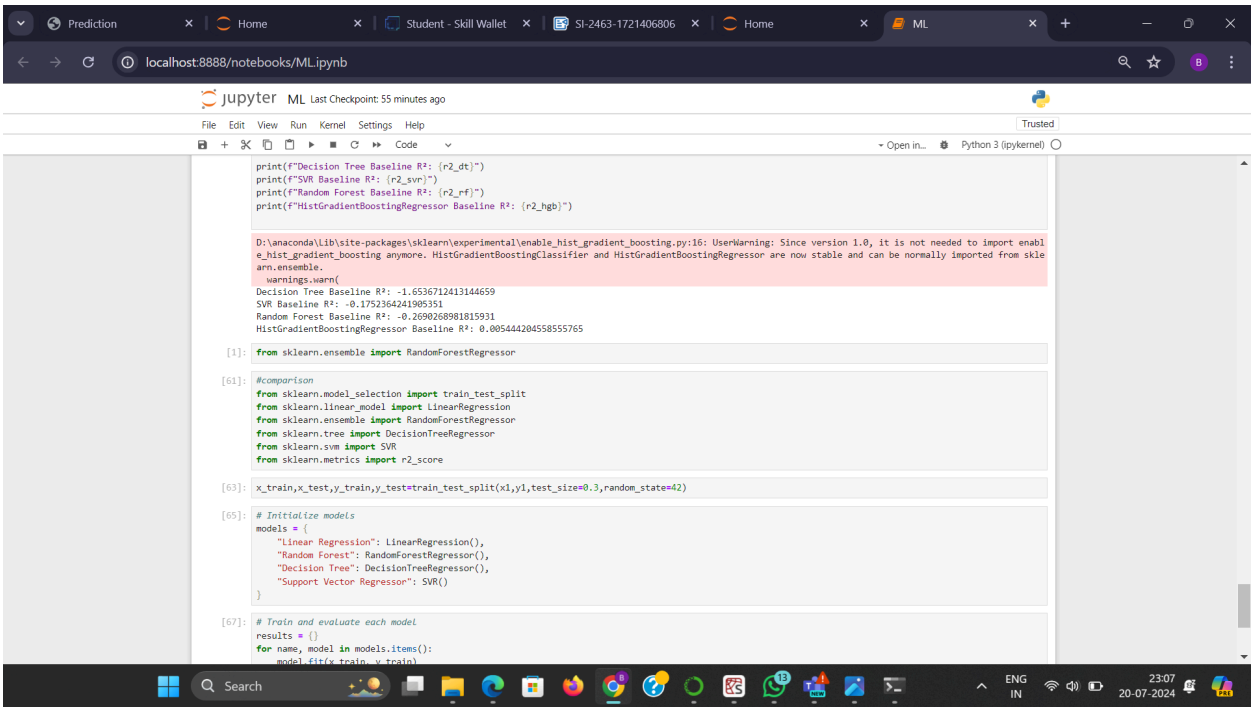
Search | ENG IN | 23:06 20-07-2024

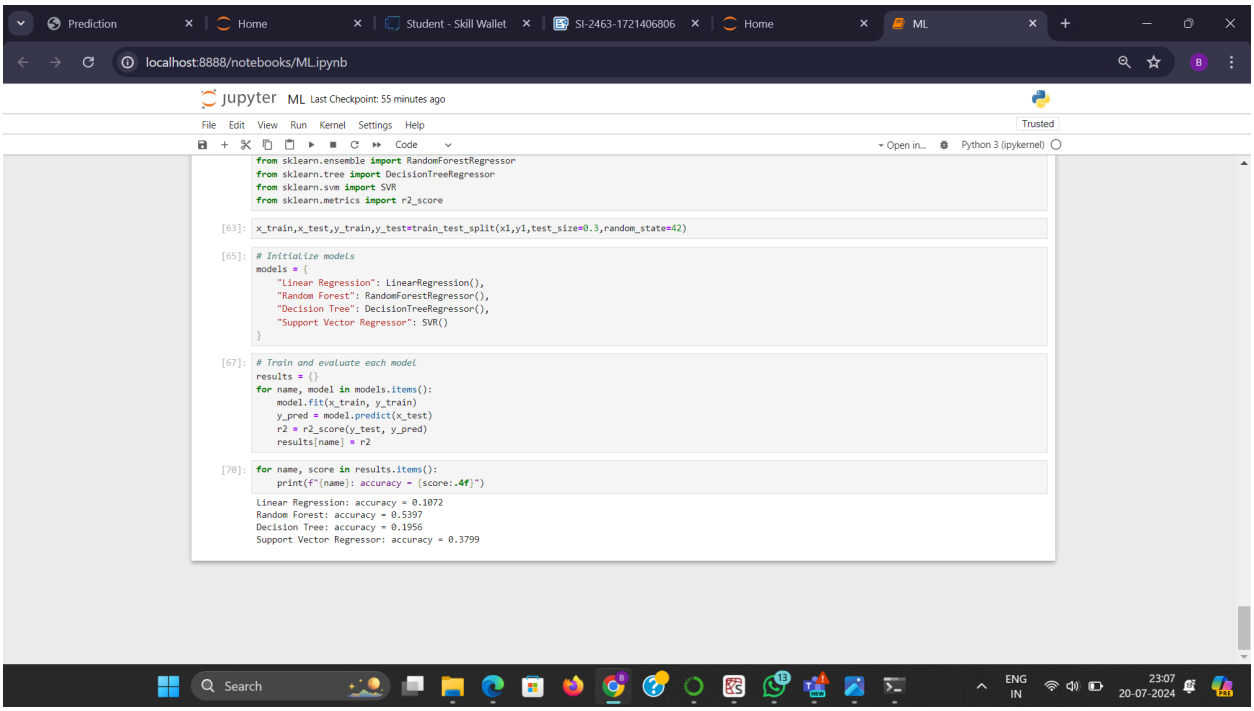

Prediction
Home
Student - Skill Wallet
SI-2463-1721406806
Home
ML
localhost:8888/notebooks/ML.ipynb
jupyter ML Last Checkpoint: 54 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
Open in...
Python 3 (ipykernel)
[41]: from sklearn import metrics
print(metrics.mean_squared_error(y_test,y_pred))
print(metrics.mean_absolute_error(y_test,y_pred))
print(np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
0.35116774942070356
0.42646239316239376
0.592594084868136
[42]: dum1=pd.get_dummies(data2['gas_type'])
print(dum1)
E10 SP98
0 True False
1 True False
2 True False
3 True False
4 True False
.. ..
383 False True
384 False True
385 False True
386 False True
387 False True
[388 rows x 2 columns]
[43]: data2=pd.concat([data2,dum1],axis=1)
[44]: data2.drop('gas_type',axis=1,inplace=True)
[45]: x1=data2.drop('consume',axis=1)
[46]: y1=data2['consume']
[47]: x1.columns
[47]: Index(['distance', 'speed', 'temp_inside', 'temp_outside', 'AC', 'rain', 'sun',
'E10', 'SP98'],
dtype='object')
Search
ENG
IN
23:06
20-07-2024


Prediction
Home
Student - Skill Wallet
SI-2463-1721406806
Home
ML
localhost:8888/notebooks/ML.ipynb
jupyter ML Last Checkpoint: 54 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
Open in...
Python 3 (ipykernel)
[48]: x1=x1.values
y1=y1.values
[49]: from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
x_train,x_test,y_train,y_test=train_test_split(x1,y1,test_size=0.3,random_state=42)
[50]: r.fit(x_train,y_train)
[50]: RandomForestRegressor
RandomForestRegressor(random_state=42)
[51]: y_pred_1=r.predict(x_test)
print(y_pred_1)
[5.408 5.259 5.29 4.516 4.152 5.547
5.071 4.704 5.345 4.588 4.105 4.147
5.546 5.6385 8.199 4.937 5.422 5.128
4.996 4.533 4.265 5.198 4.505 4.793
5.018 4.81033333 5.074 4.327 4.5605 4.369
5.224 4.857 4.256 4.159 5.362 5.044
4.62775 4.279 4.473 5.122 5.31466667 4.593
4.5805 5.077 5.483 4.445 8.534 4.258
4.38 5.238 4.173 5.146 5.362 4.9
5.0295 4.002 5.326 5.49 4.946 4.531
5.718 4.133 4.599 5.451 4.96 4.536
8.517 3.939 4.061 5.121 4.461 4.57108333
4.97433333 4.687 4.748 4.68966667 4.57 4.856
5.549 4.701 5.03933333 5.01 3.888 5.399
5.583 5.1525 5.153 5.067 4.755 3.857
4.944 4.218 5.021 4.742 4.586 5.184
4.5705 4.361 5.29 4.515 4.941 4.284
3.882 4.225 5.025 4.302 4.269 4.798
9.107 5.441 4.0205 7.229 5.421 4.248
4.5825 4.731 4.808]
Search
ENG
IN
23:06
20-07-2024


Prediction
Home
Student - Skill Wallet
SI-2463-1721406806
Home
ML
localhost:8888/notebooks/ML.ipynb
jupyter ML Last Checkpoint: 55 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
Open in...
Python 3 (ipykernel)
[52]: from sklearn import metrics
print(np.sqrt(metrics.mean_squared_error(y_test,y_pred_1)))
0.587835101437029
[53]: x_train.shape
[53]: (271, 9)
[54]: x_train[0]
[54]: array([12.3, 62, 21.5, 6, 0, 0, 0, True, False], dtype=object)
[58]: from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
r2_rf = r2_score(y_test, y_pred_1)
mse = mean_squared_error(y_test, y_pred_1)
mae = mean_absolute_error(y_test, y_pred_1)
[59]: print(f"Random Forest Regressor R² Score: {r2_rf}")
print(f"Random Forest Regressor Mean Squared Error: {mse}")
print(f"Random Forest Regressor Mean Absolute Error: {mae}")
Random Forest Regressor R² Score: 0.5873957499842983
Random Forest Regressor Mean Squared Error: 0.3455501064814821
Random Forest Regressor Mean Absolute Error: 0.4297179487179491
[60]: import joblib
joblib.dump(1,'model3.save')
[60]: ['model3.save']
[1]: from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.experimental import enable_hist_gradient_boosting
from sklearn.ensemble import HistGradientBoostingRegressor
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
Search
ENG
IN
23:07
20-07-2024


Prediction
Home
Student - Skill Wallet
SI-2463-1721406806
Home
ML
localhost:8888/notebooks/ML.ipynb
jupyter ML Last Checkpoint: 55 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
Open in...
Python 3 (ipykernel)
[1]: from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.experimental import enable_hist_gradient_boosting
from sklearn.ensemble import HistGradientBoostingRegressor
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
import numpy as n
# Assuming X and y are your features and target variable
X = np.random.rand(100, 5)
y = np.random.rand(100)
# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# Initialize models with default parameters
dt_model = DecisionTreeRegressor(random_state=42)
svr_model = SVR()
rf_model = RandomForestRegressor(random_state=42)
hgb_model = HistGradientBoostingRegressor(random_state=42)
# Fit the models
dt_model.fit(X_train, y_train)
svr_model.fit(X_train, y_train)
rf_model.fit(X_train, y_train)
hgb_model.fit(X_train, y_train)
# Predict and calculate R² scores
y_pred_dt = dt_model.predict(X_test)
y_pred_svr = svr_model.predict(X_test)
y_pred_rf = rf_model.predict(X_test)
y_pred_hgb = hgb_model.predict(X_test)
r2_dt = r2_score(y_test, y_pred_dt)
r2_svr = r2_score(y_test, y_pred_svr)
r2_rf = r2_score(y_test, y_pred_rf)
r2_hgb = r2_score(y_test, y_pred_hgb)
Search
ENG
IN
23:07
20-07-2024


Prediction
Home
Student - Skill Wallet
SI-2463-1721406806
Home
ML
localhost:8888/notebooks/ML.ipynb
jupyter ML Last Checkpoint: 55 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
Open in...
Python 3 (ipykernel)
print(f"Decision Tree Baseline R²: {r2_dt}")
print(f"SVR Baseline R²: {r2_svr}")
print(f"Random Forest Baseline R²: {r2_rf}")
print(f"HistGradientBoostingRegressor Baseline R²: {r2_hgb}")
D:\anaconda\Lib\site-packages\sklearn\experimental\enable_hist_gradient_boosting.py:16: UserWarning: Since version 1.0, it is not needed to import enable_hist_gradient_boosting anymore. HistGradientBoostingClassifier and HistGradientBoostingRegressor are now stable and can be normally imported from sklearn.ensemble.
warnings.warn(
Decision Tree Baseline R²: -1.6536712413144659
SVR Baseline R²: -0.1752364241905351
Random Forest Baseline R²: -0.2690268981815931
HistGradientBoostingRegressor Baseline R²: 0.005444204558555765
[1]: from sklearn.ensemble import RandomForestRegressor
[61]: #comparison
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.metrics import r2_score
[63]: x_train,x_test,y_train,y_test=train_test_split(x1,y1,test_size=0.3,random_state=42)
[65]: # Initialize models
models = {
"Linear Regression": LinearRegression(),
"Random Forest": RandomForestRegressor(),
"Decision Tree": DecisionTreeRegressor(),
"Support Vector Regressor": SVR()
}
[67]: # Train and evaluate each model
results = {}
for name, model in models.items():
model.fit(x_train, y_train)
Search
ENG IN
23:07 20-07-2024


Prediction
Home
Student - Skill Wallet
SI-2463-1721406806
Home
ML
localhost:8888/notebooks/ML.ipynb
jupyter ML Last Checkpoint: 55 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
Open in...
Python 3 (ipykernel)
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.metrics import r2_score
[63]: x_train,x_test,y_train,y_test=train_test_split(x1,y1,test_size=0.3,random_state=42)
[65]: # Initialize models
models = {
"Linear Regression": LinearRegression(),
"Random Forest": RandomForestRegressor(),
"Decision Tree": DecisionTreeRegressor(),
"Support Vector Regressor": SVR()
}
[67]: # Train and evaluate each model
results = {}
for name, model in models.items():
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
r2 = r2_score(y_test, y_pred)
results[name] = r2
[70]: for name, score in results.items():
print(f"{name}: accuracy = {score:.4f}")
Linear Regression: accuracy = 0.1072
Random Forest: accuracy = 0.5397
Decision Tree: accuracy = 0.1956
Support Vector Regressor: accuracy = 0.3799
Search
ENG IN
23:07 20-07-2024

App.py:

```
from flask import Flask, request,render_template
import joblib
app = Flask(__name__)
model = joblib.load("model.save")



app = Flask(__name__)

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

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

Manual_predict.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>
```

b. Github and Project demo link

Github repo links:

Nandigam Sai Prasanna -- https://github.com/Saiprasannaaa/SmartInternz/tree/main

Pulluru Nikhilesh -- https://github.com/Nikhilesh0605/SWTID1720078183-Predictive-Modeling-for-Fleet-Fuel-Management-using-Machine-Learning

Krishna Gayatri Bonagiri -- https://github.com/GayatriBonagiri/SWTID1720078183-Predictive-Modeling-for-Fleet-Fuel-Management-using-Machine-Learning1

Syed Shujatullah -- https://github.com/Shujju5583X/SmartInternz-main

Project demo link:

https://drive.google.com/file/d/1CgQAL4L0ZyGIq9TT45NpE6UImElLxnQ_/view?usp=drivesdk