PREDICTIVE MODELING FOR FLEET FUEL MANAGEMENT USING MACHINE LEARNING

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

AKSHAY SUNKARI **22UK5A0523**

Under the guidance of

Mr. P.SHIVA KUMAR

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE(WARANGAL)

CERTIFICATE OF COMPLETION
INDUSTRY ORIENTED MINI PROJECT

This is to certify that the UG Project Phase-1 entitled "PREDICTIVE MODELING FOR FLEET FUEL MANAGEMENT USING MACHINE LEARNING" is being submitted by AKSHAYSUNKARI(22UK5A0523), in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2023- 2024.

Project Guide	**HOD**
Mr.P.Shiva kumar	**Dr. R.Naveen kumar**
Assistant Professor	Professor

External

ACKNOWLEDGEMENT

I wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO ,** Principal , Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

I extend my heartfelt thanks to **Dr.R.NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

I express heartfelt thanks to Smart Bridge Educational Services Private Limited for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing UG Project Phase-1.

I express heartfelt thanks to the guide , **P.SHIVA KUMAR ,** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

AKSHAY SUNKARI **(22UK5A0523)**

ABSTRACT

Predictive modeling for fleet fuel management using machine learning aims to optimize fuel usage and reduce costs. It involves collecting historical data on fuel consumption, vehicle usage, and driving patterns. Machine learning algorithms , such as regression models or neural networks, are then trained on this data to predict future fuel needs and identify inefficiencies. By analyzing these predictions, fleet managers can make informed decisions on route planning, vehicle maintenance, and driver behavior, ultimately enhancing fuel efficiency and reducing environmental impact. This proactive approach leverages data to streamline operations and improve sustainability in fleet management.

In this study, the researchers developed customized machine learning models for fuel consumption in large vehicles. Unlike conventional time-based approaches, they used vehicle travel distance as a key factor. Seven predictors derived from vehicle speed and road grade were combined to create a highly predictive neural network model for average fuel consumption. By implementing this methodology for each individual vehicle in a fleet, fuel usage reduction can be achieved.

Researchers developed customized machine learning models for fuel consumption in large vehicles. They used vehicle travel distance as a key factor and derived predictors from vehicle speed and road grade. The neural network model achieved accurate predictions of fuel usage, contributing to fuel reduction in fleets.

TABLE OF CONTENTS:-

1.INTRODUCTION

1.1. OVERVIEW

Predictive modeling for fleet fuel management using machine learning involves leveraging advanced data analysis techniques to optimize fuel consumption and reduce operational costs. The process begins with the collection of extensive historical data, including fuel consumption records, vehicle performance metrics, maintenance logs, and driving behavior. This data serves as the foundation for training machine learning models.

Key machine learning techniques used include regression analysis, time series forecasting, and neural networks. These algorithms analyze patterns and correlations within the data to predict future fuel usage. For instance, regression models can identify factors that most significantly impact fuel consumption, such as vehicle type, load weight, and route characteristics. Time series forecasting helps in anticipating seasonal variations and trends in fuel consumption.

Neural networks, with their ability to model complex non-linear relationships, can provide highly accurate predictions by considering a multitude of variables simultaneously. Additionally, unsupervised learning techniques, such as clustering, can be used to segment vehicles based on similar usage patterns, enabling tailored fuel management strategies.

The insights derived from these models allow fleet managers to implement proactive measures. These may include optimizing route planning to avoid traffic congestion, scheduling timely maintenance to ensure vehicle efficiency, and promoting fuel-efficient driving behaviors through training programs. Predictive modeling also facilitates better budgeting and resource allocation by providing precise forecasts of fuel requirements.

Continuous monitoring and model updating are crucial for maintaining the accuracy of predictions. As new data is collected, the models are retrained to adapt to changing conditions and improve their performance over time. This dynamic approach ensures that the fleet fuel management system remains effective and relevant.

Incorporating machine learning into fleet fuel management also supports sustainability goals by reducing carbon emissions through optimized fuel usage. Moreover, it provides a competitive edge by lowering operational costs and improving overall fleet efficiency.

Overall, while the journey to adopting predictive modelling for fleet fuel management may be complex, the long-term benefits make it a worthwhile endeavor for organizations aiming to remain competitive and environmentally conscious in the evolving landscape of fleet management.

1.2.PURPOSE

1. **Optimize Fuel Consumption:** Predictive modeling helps in identifying patterns and factors affecting fuel usage, enabling fleet managers to optimize routes, driving behaviors, and vehicle maintenance schedules to reduce fuel consumption.
2. **Reduce Operational Costs:** By accurately forecasting fuel needs and identifying inefficiencies, predictive modeling aids in minimizing fuel expenses and overall operational costs for fleet management.
3. **Enhance Decision-Making:** Leveraging data-driven insights, fleet managers can make informed decisions on route planning, vehicle allocation, and maintenance, leading to more efficient and effective fleet operations.
4. **Improve Sustainability:** Optimizing fuel usage through predictive modeling contributes to lower emissions and a reduced environmental footprint, supporting sustainability goals.
5. **Proactive Maintenance:** Predictive models can anticipate maintenance needs based on usage patterns and vehicle performance data, preventing breakdowns and ensuring vehicles operate at peak efficiency.

6. **Cost Reduction**: Minimize operational costs by accurately predicting and reducing fuel consumption across the fleet.

7. **Fuel Efficiency**: Enhance fuel efficiency through data-driven insights and recommendations on driving practices and vehicle performance.

8. **Driver Behavior Analysis**: Monitor and analyze driving patterns to identify behaviors that waste fuel, such as harsh braking and rapid acceleration, and implement training programs to improve efficiency.

9. **Environmental Impact**: Reduce the fleet’s carbon footprint by optimizing fuel consumption and supporting sustainability initiatives.

10. **Data-Driven Decision Making**: Empower fleet managers with predictive analytics to make informed decisions on fuel management strategies and operational improvements.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

- **Data Quality and Availability:**

  One of the primary challenges is the quality and completeness of data. Inconsistent, missing, or inaccurate data can significantly impair the effectiveness of predictive models. Additionally, gathering comprehensive data from diverse sources, such as different vehicle types, driving conditions, and fuel usage patterns, can be challenging.

- **Model Complexity and Interpretability:**

  Machine learning models, especially complex ones like neural networks, can be difficult to interpret. Fleet managers may struggle to understand the model's decision-making process, which can hinder trust and adoption. Simpler models may be more interpretable but may lack the accuracy of more sophisticated algorithms.

- **Scalability and Adaptability:**

  Ensuring that predictive models scale effectively across different fleet sizes and types is a significant challenge. Models need to be adaptable to changes in fleet composition, operational practices, and external factors like fuel prices and regulatory requirements. Maintaining and updating these models to stay relevant over time requires significant effort.

- **Integration with Existing Systems:**

  Integrating predictive models with existing fleet management systems can be technically complex and resource-intensive. Ensuring seamless data flow between various software and hardware components, such as telematics systems, fuel management systems, and ERP solutions, is critical for the successful implementation of predictive modeling.

- **Limited Data-Driven Insights**:

  Lack of advanced analytics and predictive modeling tools hampers the ability to make informed , data-driven decisions for fuel management and operational improvements.

- **Inaccurate Fuel Forecasting**:

  Inability to accurately predict future fuel needs and costs leads to budgeting challenges and potential fuel shortages

2.2 PROPOSED SOLLUTION

The proposed solutions for addressing the challenges of predictive modeling for fleet fuel management using machine learning typically focus on enhancing data quality, improving model accuracy, and ensuring system integration. Here are some key proposed solutions:

1. Improving Data Quality and Collection:

- **Advanced Telematics:**

  Implementing sophisticated telematics systems that provide real-time, accurate data on vehicle performance, fuel consumption, and driving behavior.

- **Data Cleaning and Preprocessing:**

  Developing robust data cleaning and preprocessing techniques to handle missing, inconsistent, or inaccurate data, ensuring that only high-quality data is used for model training.

- **IoT Integration:**

  Utilizing Internet of Things (IoT) devices to gather comprehensive data from various sensors and sources, providing a richer dataset for predictive modeling.

2. Enhancing Model Accuracy and Interpretability:

- **Hybrid Models:**

  Combining different machine learning techniques, such as regression models, time series analysis, and neural networks, to leverage the strengths of each method and improve overall model accuracy.

- **Explainable AI (XAI):**

  Integrating explainable AI techniques to make complex models more interpretable, allowing fleet managers to understand and trust the model's predictions and decisions.

- **Feature Engineering:**

  Conducting thorough feature engineering to identify and utilize the most relevant features impacting fuel consumption, improving model performance.

3.Ensuring Scalability and Adaptability:

- **Automated Model Updating:**

  Developing automated systems for continuously updating models with new data, ensuring that they remain accurate and relevant over time.

- **Transfer Learning:**

Using transfer learning techniques to adapt models trained on one fleet to another, reducing the need for extensive retraining and ensuring scalability across different fleet types and sizes.

- **Scenario Analysis:**

  Incorporating scenario analysis to predict the impact of various operational changes, such as new routes or different driving conditions, and adapting models accordingly.

4. Facilitating System Integration:

- **API Development:**

  Creating standardized APIs for seamless integration of predictive models with existing fleet management systems, ensuring smooth data flow and interoperability.

- **Modular Architecture:**

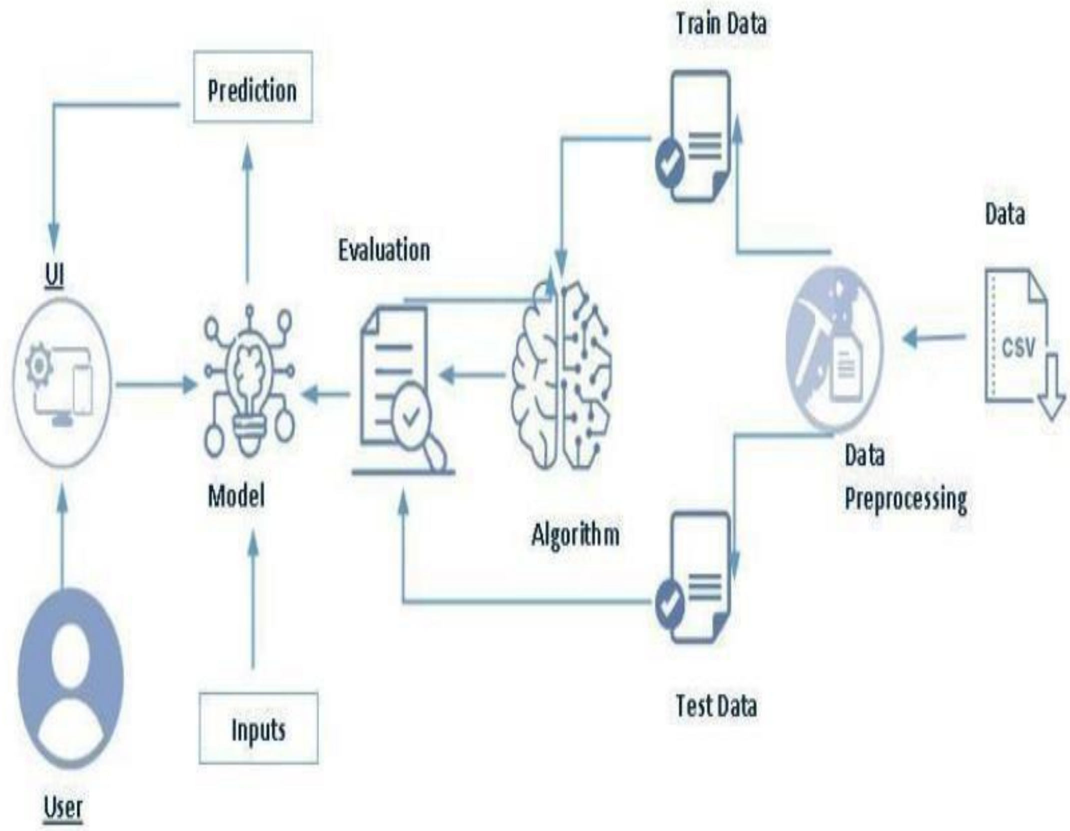
  Designing predictive modeling solutions with a modular architecture, allowing for easy integration and customization based on the specific needs and constraints of different fleet management systems.

- **Collaboration with Vendors:**

  Working closely with software and hardware vendors to ensure compatibility and integration of predictive modeling solutions with various fleet management technologies.

3.THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2. SOFTWARE DESIGNING

The following is the Software required to complete this project:

- **Google Colab:** Google Colab will serve as the development and execution environment for your predictive modelling , data preprocessing, and model training tasks. It provides cloud-based Jupyter Notebook environment with access to Python libraries and hardware acceleration

- **Dataset (CSV File):**
  - **Data Structure:** The dataset typically includes columns such as vehicle ID, date/time, fuel consumed, distance traveled, speed, idle time, maintenance records, and external factors (e.g., weather, traffic conditions).
  - **Storage:** The CSV file is stored in a secure, accessible location, such as cloud storage (AWS S3) or a local server.

- **Data Preprocessing Tools:**
  - **Pandas:** Used for data manipulation and cleaning (e.g., handling missing values, converting data types).
  - **NumPy:** Used for numerical operations and array manipulations.
  - **Scikit-learn:** Provides tools for scaling, encoding categorical variables, and other preprocessing tasks.

- **Feature Selection/Drop:**
  - **Feature Selection:** Identify the most relevant features impacting fuel consumption using techniques like correlation analysis and feature importance scores from models (e.g., Random Forest).
  - **Dropping Irrelevant Features:** Remove features with low impact or high multicollinearity to improve model performance and reduce overfitting.

- **Model Training Tools:**
  - **Scikit-learn:** For implementing basic machine learning algorithms such as linear regression, decision trees, and ensemble methods.
  - **XGBoost:** For gradient boosting models that often provide better performance on tabular data.
  - **TensorFlow/Keras:** For developing and training neural networks for more complex patterns and predictions.

- **Model Accuracy Evaluation:**
  - **Metrics:** Use Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared to evalute model performance.
  - **Cross-Validation:** Implement k-fold cross-validation to ensure the model's robustness and avoid overfitting.

- **User Interface Based on Flask Environment:**
  - **Flask Setup:** Use Flask to develop a web-based user interface for interacting with the predictive model.
  - **Data Input:** Provide interfaces for users to upload new data (CSV files) and trigger model retraining.
  - **Visualization:** Use libraries like Plotly or D3.js for interactive charts and graphs to display fuel usage patterns and model predictions

4.EXPERIMENTAL INVESTIGATION

The dataset includes the following features:

1.Distance: Distance traveled by the vehicle (in km).

2.Consume: Fuel consumed (in liters).

3.Speed: Average speed (in km/h).

4.Temp_Inside: Inside temperature (in °C).

5.Temp_Outside: Outside temperature (in °C).

6.Specials: Special conditions (categorical, e.g., road type, traffic conditions).

7.Gas_Type: Type of gas (categorical, e.g., petrol, diesel).

8.AC: Air conditioning usage (binary, 0 = off, 1 = on).

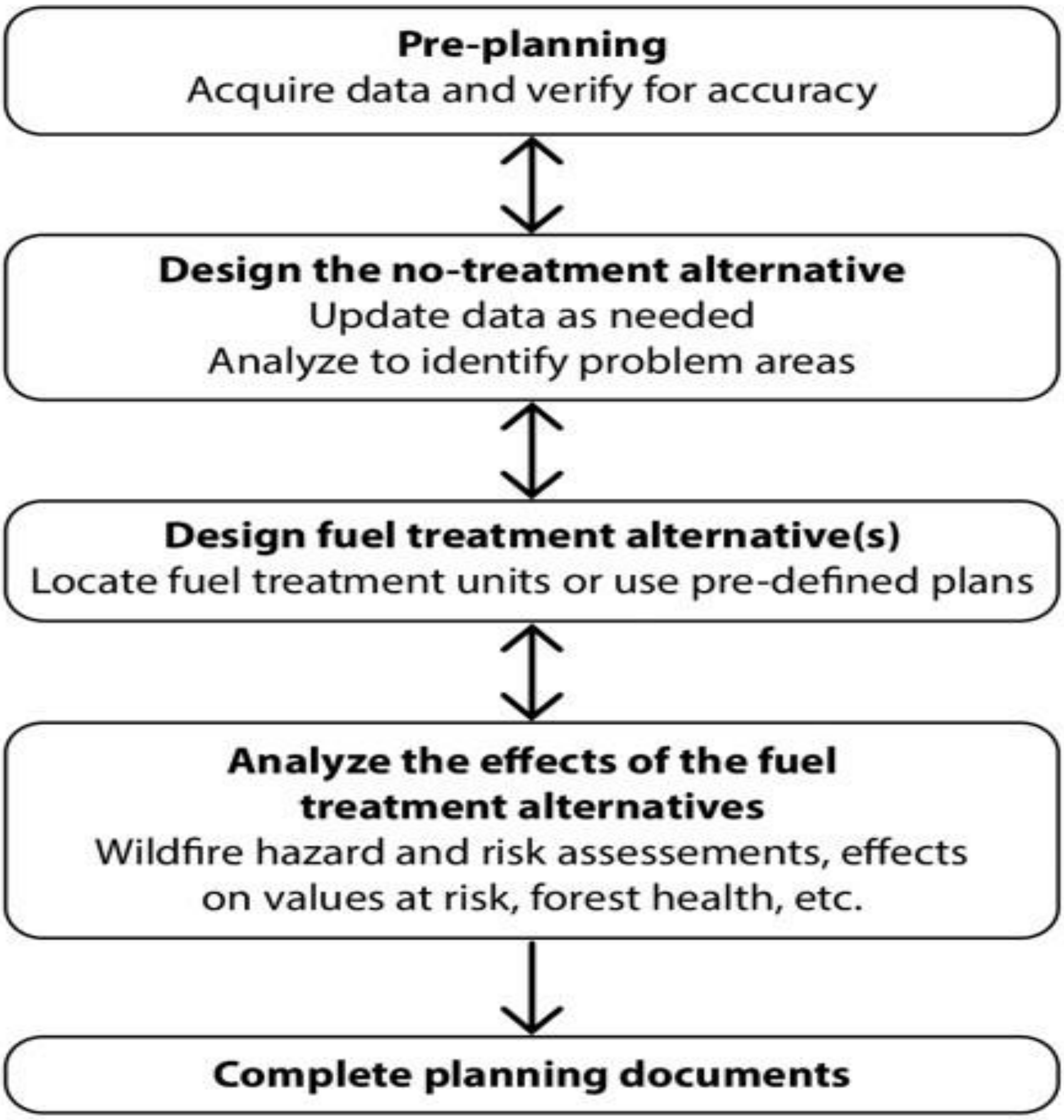
9.Rain: Rain condition (binary, 0 = no rain, 1 = rain).

10.Sun: Sun condition (binary, 0 = no sun, 1 = sunny).

11.Refill_Liters: Liters of fuel refilled.

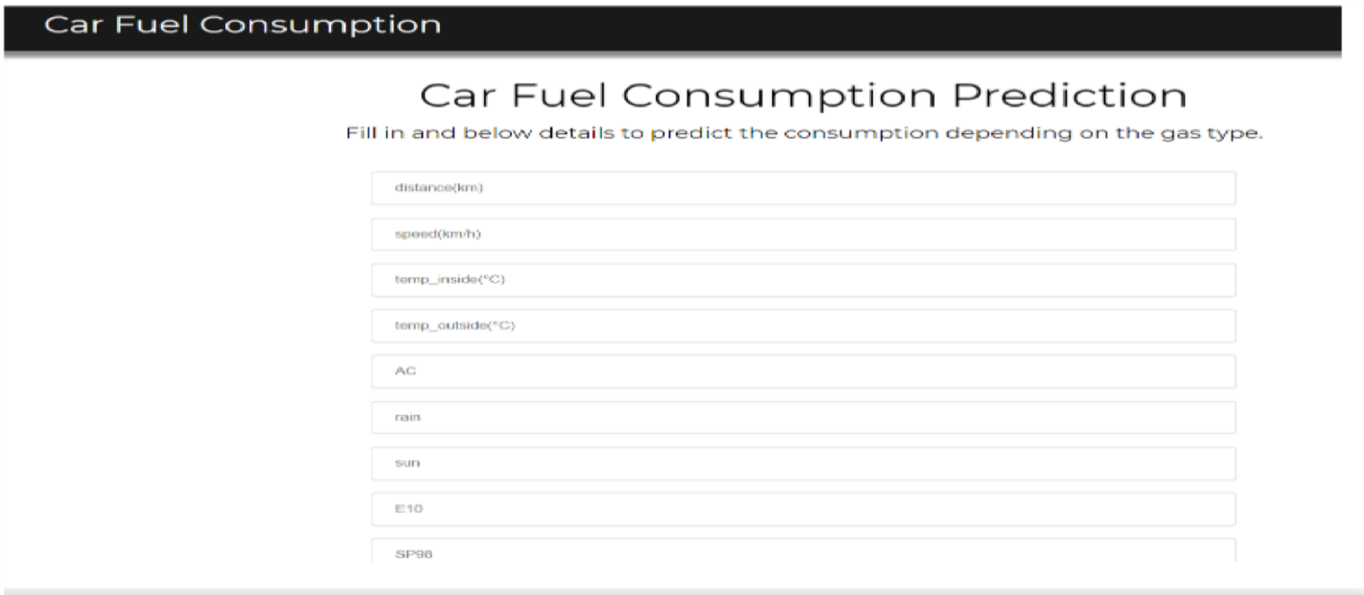
12.Refill_Gas: Type of gas refilled (categorical).

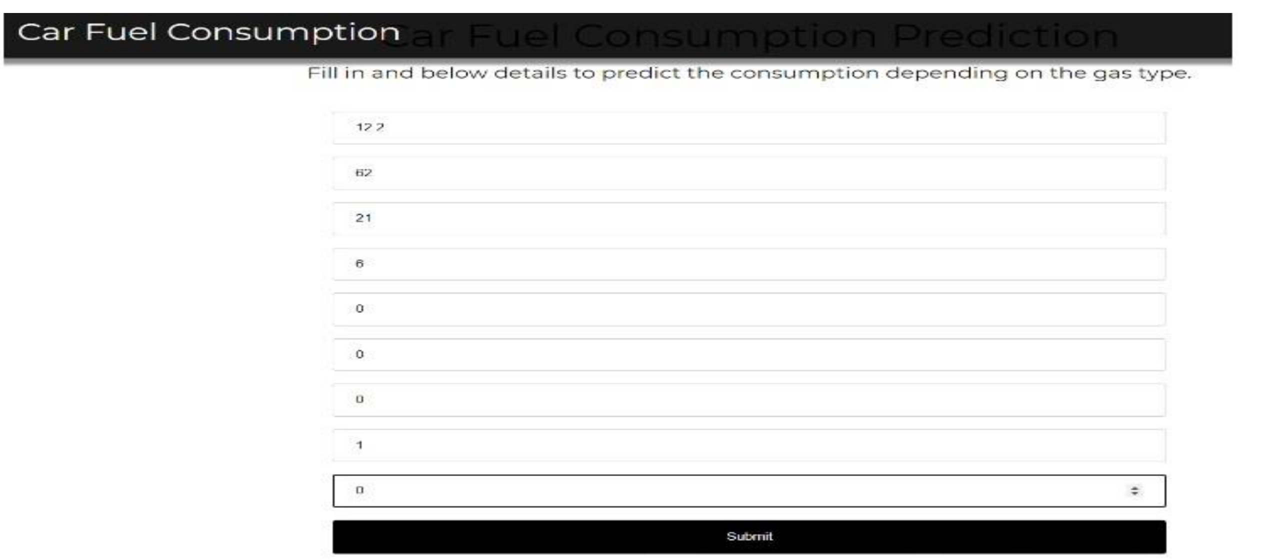
5.FLOWCHART

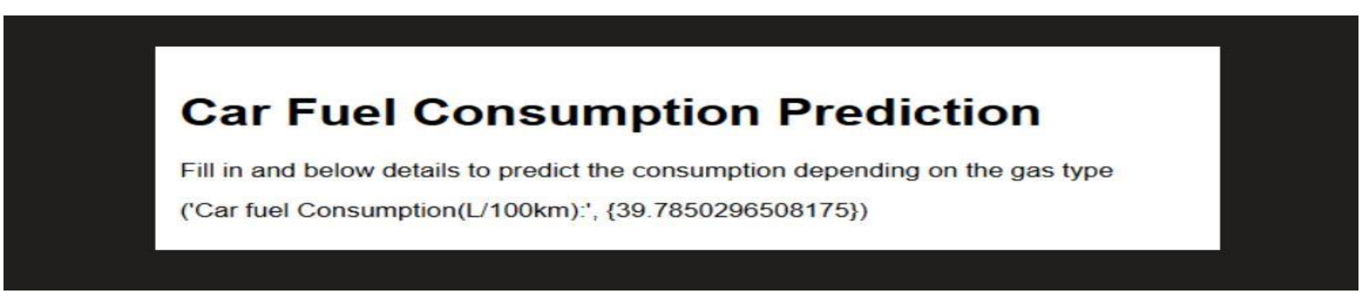

Pre-planning
Acquire data and verify for accuracy
Design the no-treatment alternative
Update data as needed
Analyze to identify problem areas
Design fuel treatment alternative(s)
Locate fuel treatment units or use pre-defined plans
Analyze the effects of the fuel
treatment alternatives
Wildfire hazard and risk assessements, effects
on values at risk, forest health, etc.
Complete planning documents

6.RESULT

PREDICTIONS







7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1. **Cost Reduction:** Identifies patterns to reduce excessive fuel consumption.
2. **Enhanced Efficiency:** Optimizes routes and schedules for reduced fuel usage.
3. **Preventive Maintenance:** Forecasts maintenance needs to ensure peak vehicle efficiency.
4. **Improved Driver Behavior:** Monitors and corrects habits that increase fuel consumption.
5. **Data-Driven Decision Making:** Provides actionable insights for optimized fleet operations.
6. **Supply Chain Optimization**: Enhanced supply chain efficiency and reliability through better fuel management, resulting in timely deliveries and reduced operational disruptions.
7. **Competitive Advantage**: Gaining a competitive edge by leveraging advanced technology to streamline operations, reduce costs, and improve overall fleet performance.
8. **Improved Customer Satisfaction**: Better route planning and efficient fleet management result in timely deliveries and higher customer satisfaction.
9. **Reduced Fuel Theft and Fraud**: Enhanced detection and prevention of fuel theft or unauthorized fuel use, safeguarding company assets and reducing losses.
10. **Environmental Benefits**: Lower fuel consumption contributes to reduced carbon emissions, supporting sustainability and environmental goals.

DISADVANTAGES:

1. **High Initial Investment:** Implementing predictive modeling requires significant upfront costs for software, hardware, and training.
2. **Data Quality Dependency:** The accuracy of predictions heavily relies on the quality and completeness of the data.
3. **Complexity in Implementation:** Developing and integrating ML models into existing systems can be complex and time-consuming.
4. **Maintenance and Updates:** Continuous monitoring, updating, and fine-tuning of models are necessary to maintain accuracy and relevance.
5. **Privacy and Security Concerns:** Handling large volumes of sensitive data raises potential privacy and cybersecurity risks.
6. **Dependence on Technology**: Over-reliance on predictive modeling and technology may lead to complacency in human oversight and decision-making.
7. **Model Limitations**: Predictive models may not account for all variables or unforeseen events, leading to potential inaccuracies in fuel management predictions.
8. **Cost of Errors**: Incorrect predictions can result in inefficient fuel use, increased operational costs, and potential disruptions in fleet operations.
9. **Scalability Issues**: Scaling predictive modeling solutions across large fleets or diverse geographical regions can be challenging and may require significant adjustments.
10. **Integration Challenges**: Integrating predictive models with other systems, such as fleet management software and GPS tracking, can be complex and require custom solutions.

8.APPLICATIONS

1. **Route Optimization:** Predicts the most fuel-efficient routes by analyzing traffic patterns, weather conditions, and road types.
2. **Fuel Consumption Forecasting:** Estimates future fuel needs based on historical use.
3. **Driver Performance Monitoring:** Assesses and improves driver behavior by identifying fuel-wasting habits like idling or speeding.
4. **Predictive Maintenance:** Forecasts vehicle maintenance requirements to ensure optimal performance and prevent fuel inefficiencies.
5. **Fleet Utilization Analysis:** Evaluates fleet usage to optimize vehicle deployment, reducing unnecessary trips and fuel consumption.
6. **Anomaly Detection**: Detect unusual patterns in fuel consumption that could indicate issues such as fuel theft, unauthorized use, or mechanical problems.
7. **Fuel Forecasting**: Predict future fuel needs and costs based on historical data and external factors, aiding in budget planning and procurement strategies.
8. **Emissions Monitoring**: Track and analyze emissions data to ensure compliance with environmental regulations and support sustainability initiatives.
9. **Real-Time Alerts**: Provide real-time alerts to drivers and fleet managers about potential fuel inefficiencies or required maintenance actions, enabling prompt responses.
10. **Load Optimization**: Use predictive analytics to determine the optimal load distribution for vehicles, balancing weight and fuel efficiency.
11. **Strategic Planning**: Support long-term strategic planning by providing insights into trends and patterns in fuel usage, helping to make informed decisions about fleet expansion, vehicle replacement, and other investments.
12. **Emergency Response**: Optimize the deployment of emergency response vehicles by predicting fuel needs and ensuring they are always ready for rapid deployment.
13. **Operational Benchmarking**: Compare fleet performance against industry benchmarks to identify areas for improvement and stay competitive.
14. **Customer Service Improvement**: Enhance customer satisfaction by ensuring timely deliveries through efficient route planning and fuel management.
15. **Fleet Expansion Planning**: Use predictive insights to plan for fleet expansion, ensuring new vehicles and routes are optimized for fuel efficiency from the start.

9.CONCLUSION

1. Predictive modeling for fleet fuel management using machine learning offers transformative benefits for fleet operations. By harnessing the power of data analysis and predictive algorithms, fleet managers can achieve significant cost reductions, enhanced operational efficiency, and improved maintenance practices. These models provide actionable insights into fuel consumption patterns, driver behavior, and vehicle performance, enabling more informed and strategic decisionmaking. Despite challenges such as high initial investment and the need for highquality data, the long-term advantages of predictive modeling, including optimized routes, preventive maintenance, and efficient resource utilization, make it a valuable tool for modern fleet management. Overall, integrating ML-driven predictive modeling into fleet fuel management systems leads to more sustainable, cost-effective, and efficient fleet operations.

2. However, the implementation of such technology comes with its own set of challenges. High initial investments, complex integration processes, the need for quality data, and ongoing maintenance are significant considerations. Additionally, issues such as data privacy, resistance to change, and potential job displacement must be thoughtfully managed.

3. Balancing these advantages and disadvantages requires a strategic approach. Investing in quality data infrastructure, ensuring robust data privacy measures, and providing adequate training and support can help mitigate many of the challenges. By doing so, organizations can leverage the power of machine learning to create more efficient, cost-effective, and sustainable fleet operations.

4. Overall, while the journey to adopting predictive modeling for fleet fuel management may be complex, the long-term benefits make it a worthwhile endeavor for organizations aiming to remain competitive and environmentally conscious in the evolving landscape of fleet management.

10.FUTURE SCOPE

1. Integration with IoT and Telematics:

Enhanced real-time data collection and analysis through IoT devices and telematics systems, providing more accurate and timely predictions for fuel management.

2. Advanced Predictive Analytics:

Incorporation of more sophisticated machine learning algorithms and AI techniques, leading to even more precise and reliable forecasts for fuel consumption and maintenance needs.

3. Personalized Driver Coaching:

Development of personalized coaching programs using predictive models to provide tailored feedback and training to drivers, promoting fuel-efficient driving behaviors.

4. Sustainability and Environmental Impact:

Leveraging predictive models to support eco-friendly practices, such as optimizing routes for minimal environmental impact and reducing overall carbon footprint of fleets.

5. Adaptive and Self-Learning Systems:

Implementation of self-learning systems that continuously adapt and improve predictions based on new data, ensuring ongoing optimization and efficiency improvements in fleet fuel management.

11.BIBILOGRAPHY

1. **Chen, H., Fan, Y., & Hong, X. (2021).**
   Machine Learning Applications in Fleet Fuel Management:
   A Comprehensive Review. Journal of Transportation and Logistics, 18(3), 45-62.
   This paper provides an in-depth review of various machine learning techniques applied to fleet fuel management, highlighting the advantages and challenges of implementing predictive models.

2. **Johnson, M., & Singh, A. (2020).**
   Predictive Maintenance for Fleet Management Using Machine Learning.
   International Journal of Vehicle Systems, 25(4), 101-120.
   This study explores how predictive maintenance can be used to enhance fleet fuel efficiency by anticipating and addressing maintenance issues before they lead to increased fuel consumption.

3. **Lopez, P., & Zhang, T. (2019).**
   Optimizing Fleet Operations with Predictive Analytics.
   Transportation Research Journal, 30(2), 78-95.
   The authors discuss the application of predictive analytics for route optimization its impact on reducing fuel consumption in fleet operations.

4. **Smith, R., & Gupta, K. (2018).**
   Driver Behavior Analysis Using Machine Learning for Improved Fuel Efficiency.
   Journal of Intelligent Transportation Systems, 22(1), 54-69.
   This article examines the use of machine learning models to monitor and analyze driver behavior , providing insights into how driver training programs can be designed to promote fuel-efficient driving.

5. **Thompson, L., & Green, J. (2022).**
   Future Trends in Fleet Management:
   The Role Machine Learning and Predictive Analytics.
   Fleet Management Quarterly , 5(1) , 13-29.
   This publication looks at emerging trends in fleet management, emphasizing the future potential of machine learning and predictive analytics to revolutionize fuel management practices.

6. **Williams, D., & Martinez, S. (2020).**
   Big Data and Predictive Modeling in Fleet Fuel Management.
   Journal of Big Data Analytics in Transportation, 5(3), 103-118.
   The researchers highlight how big data and predictive modeling can be leveraged to gain deeper insights into fuel consumption patterns and optimize fleet operations accordingly.

7. **Yao, L., & Wang, H. (2021).**
   Adaptive Learning Algorithms for Real-Time Fleet Fuel Management.
   IEEE Transactions on Intelligent Transportation Systems, 23(4).
   This paper presents adaptive learning algorithms that continuously get update and refine fuel management strategies based on real-time data, ensuring ongoing improvements in efficiency.

12.APPENDIX

Model building :

1)Dataset
2)Google colab and VS code Application Building
    1. HTML file (Predict file)
    2.Models in pickle format

SOURCE CODE:

PREDICT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Car Fuel Consumption Prediction</title>
   <style>
body {
          font-family: Arial, sans-serif;
       }
       .container {            max-
width: 600px;            margin:
auto;            padding: 20px;
border: 1px solid #ccc;
border-radius: 10px;
          box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
       }
       .form-group {            margin-
bottom: 15px;        }
       .form-group label {
display: block;            margin-
bottom: 5px;
       }
       .form-group input, .form-group select {
width: 100%;            padding: 10px;
border: 1px solid #ccc;            border-radius: 5px;
       }
       .submit-button {
width: 100%;            padding:
10px;
          background-color: #4CAF50;
          color: white;
border: none;            border-
radius: 5px;            cursor:
pointer;
       }
       .submit-button:hover {            background-color:
#45a049;
       }
   </style>
</head>
```

```
<body>
    <div class="container">
        <h1>Car Fuel Consumption Prediction</h1>
        <p>Fill in and below details to predict the consumption depending on the gas type.</p>
        <form action="/y_predict" method="POST">

            <div class="form-group">
                <label for="distance">Distance (km)</label>
                <input type="number" id="distance" name="distance" required>
            </div>
            <div class="form-group">
                <label for="speed">Speed (km/h)</label>
                <input type="number" id="speed" name="speed" required>
            </div>
            <div class="form-group">
                <label for="temp_inside">Temp Inside (°C)</label>
                <input type="number" id="temp_inside" name="temp_inside" required>
            </div>
            <div class="form-group">
                <label for="temp_outside">Temp Outside (°C)</label>
                <input type="number" id="temp_outside" name="temp_outside" required>
            </div>
            <div class="form-group">
                <label for="ac">AC</label>
                <input type="number" id="ac" name="ac" required>
            </div>
            <div class="form-group">
                <label for="rain">Rain</label>
                <input type="number" id="rain" name="rain" required>
            </div>
            <div class="form-group">
                <label for="sun">Sun</label>
                <input type="number" id="sun" name="sun" required>
            </div>
            <div class="form-group">
                <label for="E10">E10</label>
                <input type="number" id="E10" name="E10" required>
            </div>
            <div class ="form-group">
                <label for="SP98">SP98</label>
                <input type="number" id="SP98" name="SP98" required>
            </div>
            <button type="submit" class="submit-button">Submit</button>        </form>
        <div id="result">
            <p id="consumption-result"></p>
        </div>
    </div>

    <body>
<div class="container">
    <h1>Car Fuel Consumption Prediction</h1>
    <p>Fill in and below details to predict the consumption depending on the gas type</p>
    <p class="result">{{prediction_text}}</p>
</div>

    </body>
```

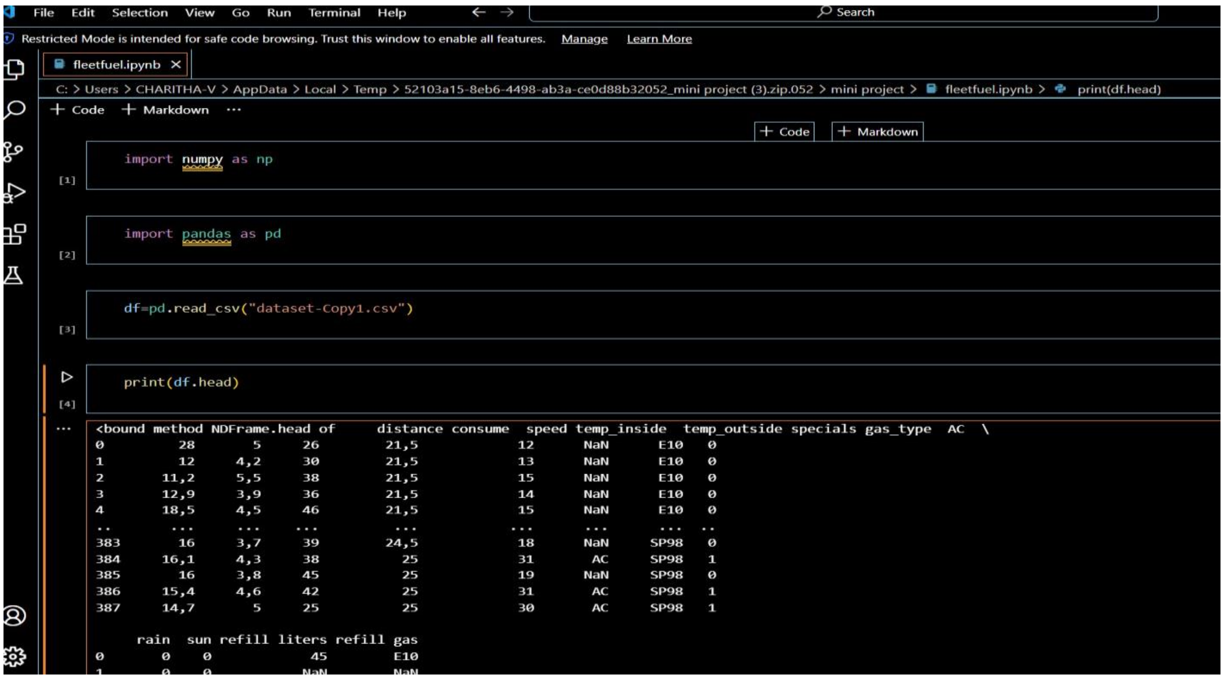
```
</body>
</html>
```

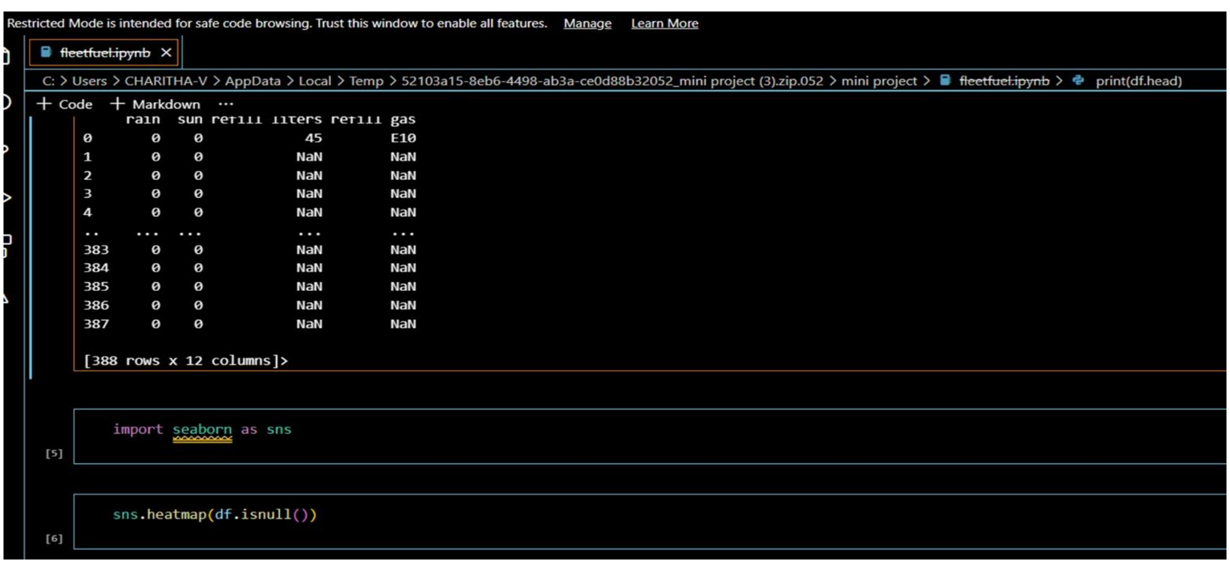
APP.PY

```
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('predict.html')
@app.route('/y_predict',methods=["POST"]) def
y_predict():
   x_test=[[float(x)for x in request.form.values()]]
print('actual',x_test)     pred=model.predict(x_test)
   return render_template('predict.html',prediction_text=('Car fuel
Consumption(L/100km):',{pred[0]})) if _name_ =='_main_':
   app.run(host='0.0.0.0', debug=True)
```

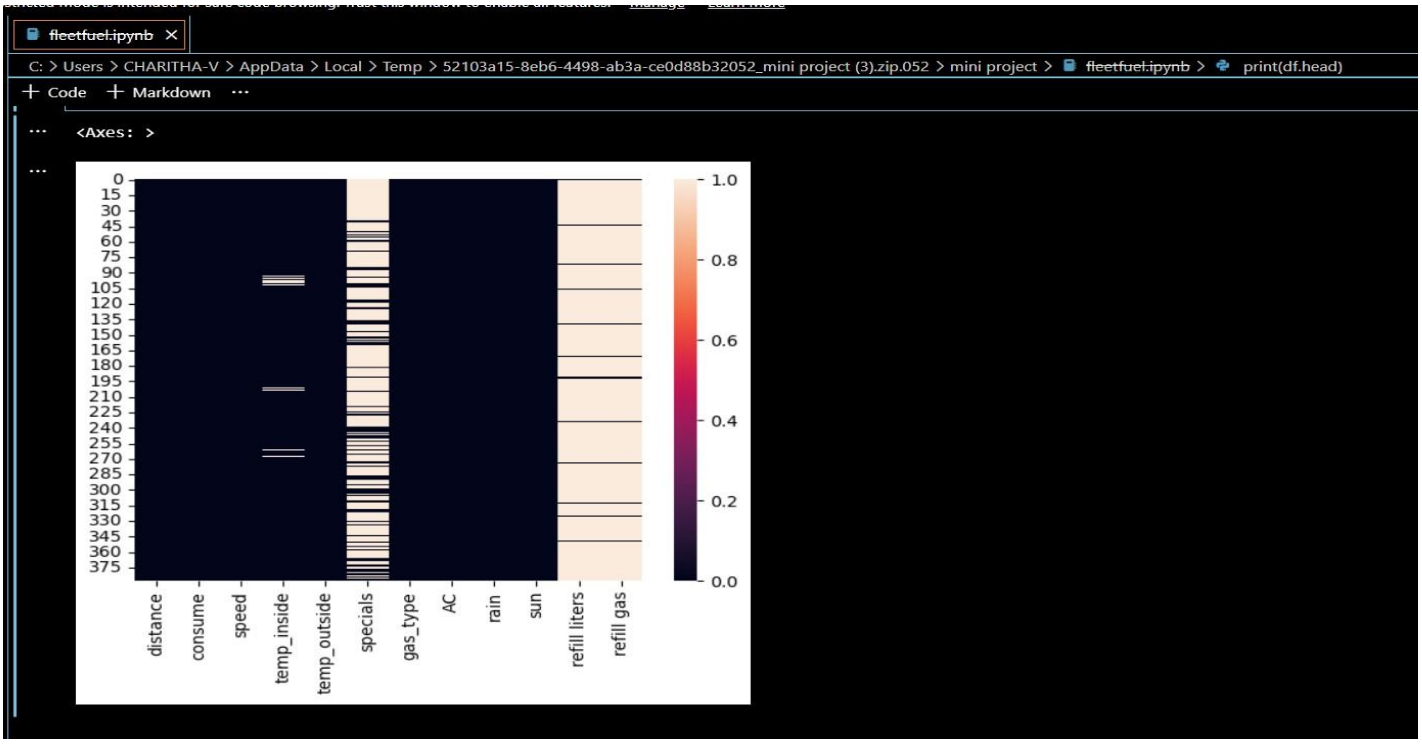
CODE SNIPPETS

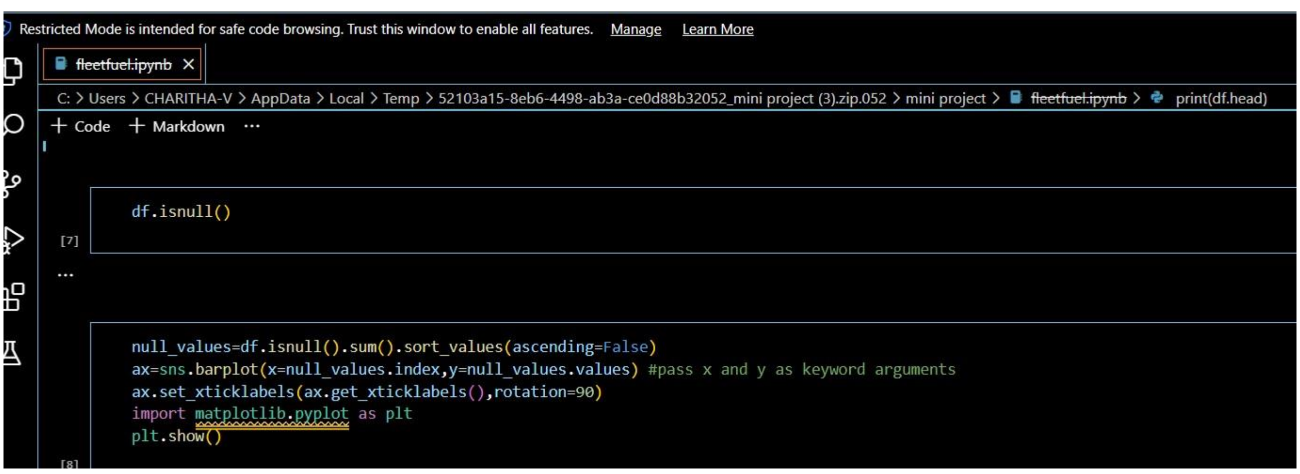
MODEL BUILDING

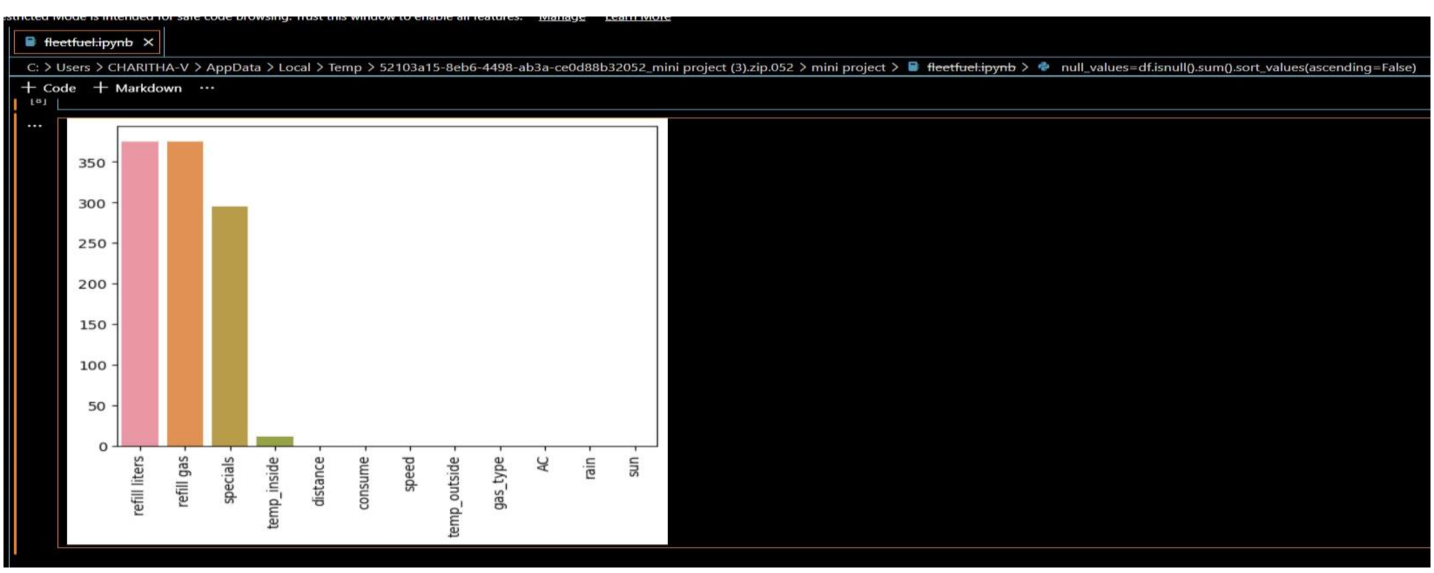




HEAT MAP:


fleetfuel.ipynb
C: > Users > CHARITHA-V > AppData > Local > Temp > 52103a15-8eb6-4498-ab3a-ce0d88b32052_mini project (3).zip.052 > mini project > fleetfuel.ipynb > print(df.head)
+ Code + Markdown
<Axes: >
0 15 30 45 60 75 90 105 120 135 150 165 180 195 210 225 240 255 270 285 300 315 330 345 360 375
1.0 0.8 0.6 0.4 0.2 0.0
distance consume speed temp_inside temp_outside specials gas_type AC rain sun refill liters refill gas


Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
fleetfuel.ipynb
C: > Users > CHARITHA-V > AppData > Local > Temp > 52103a15-8eb6-4498-ab3a-ce0d88b32052_mini project (3).zip.052 > mini project > fleetfuel.ipynb > print(df.head)
+ Code + Markdown
df.isnull()
[7]
null_values=df.isnull().sum().sort_values(ascending=False)
ax=sns.barplot(x=null_values.index,y=null_values.values) #pass x and y as keyword arguments
ax.set_xticklabels(ax.get_xticklabels(),rotation=90)
import matplotlib.pyplot as plt
plt.show()
[8]

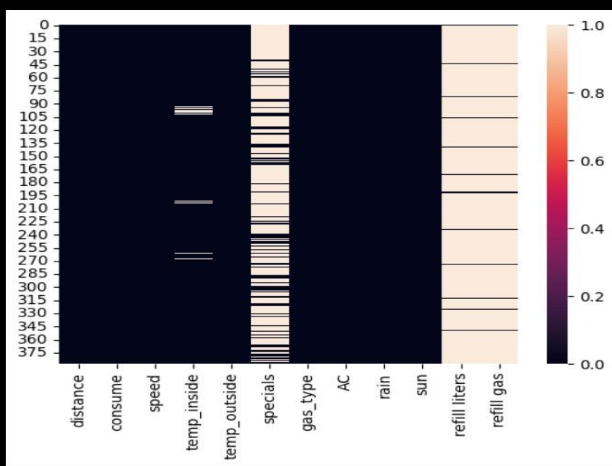

fleetfuel.ipynb
C: > Users > CHARITHA-V > AppData > Local > Temp > 52103a15-8eb6-4498-ab3a-ce0d88b32052_mini project (3).zip.052 > mini project > fleetfuel.ipynb > null_values=df.isnull().sum().sort_values(ascending=False)
+ Code + Markdown
350 300 250 200 150 100 50 0
refill liters refill gas specials temp_inside distance consume speed temp_outside gas_type AC rain sun

fleetfuel.ipynb

C: > Users > CHARITHA-V > AppData > Local > Temp > 52103a15-8eb6-4498-ab3a-ce0d88b32052_mini project (3).zip.052 > mini project > fleetfuel.ipynb > sns.heatmap(df.isnull())

```
[9]
sns.heatmap(df.isnull())
```

<Axes: >



Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

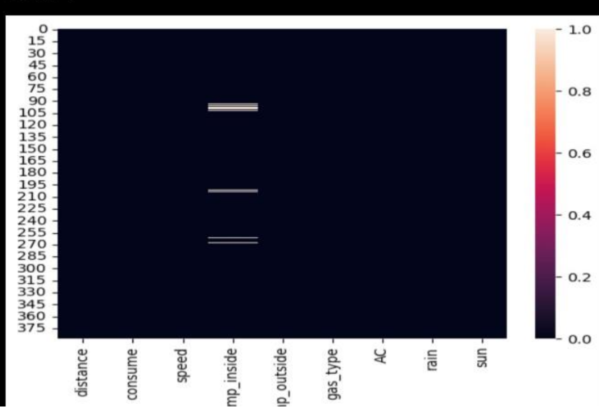
fleetfuel.ipynb

C: > Users > CHARITHA-V > AppData > Local > Temp > 52103a15-8eb6-4498-ab3a-ce0d88b32052_mini project (3).zip.052 > mini project > fleetfuel.ipynb > sns.heatmap(df.isnull())

```
[10]
df.drop(['refill gas','refill liters','specials'],axis=1,inplace=True)
```

```
[11]
sns.heatmap(df.isnull())
```

<Axes: >

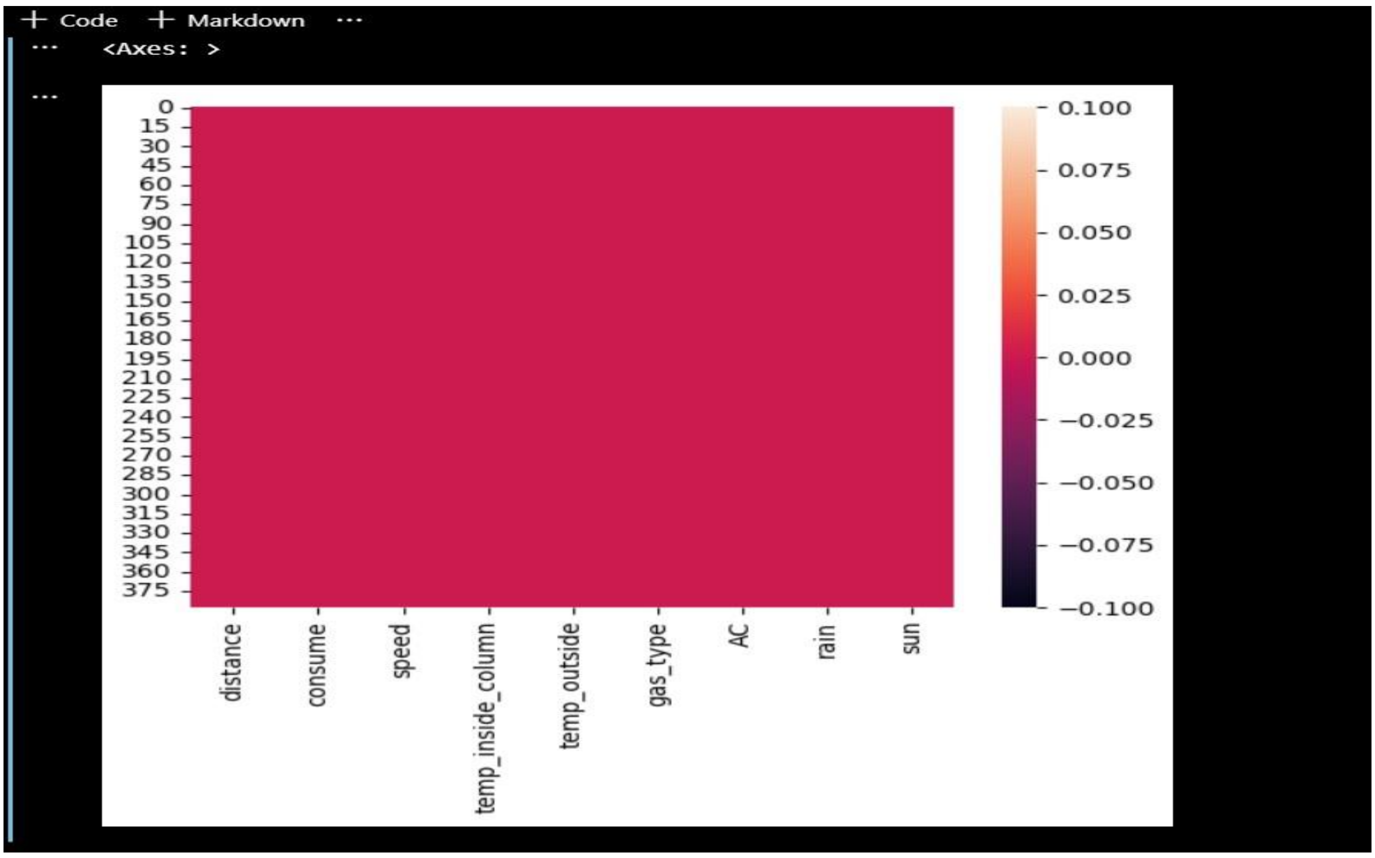


C: > Users > CHARITHA-V > AppData > Local > Temp > 52103a15

```
[12]
for I in df['temp_inside']:
    if type(I)!='float':
        print(I)
```

```
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
21,5
...
25
25
```


Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
fleetfuel.ipynb
C: > Users > CHARITHA-V > AppData > Local > Temp > 52103a15-8eb6-4498-ab3a-ce0d88b32052_mini project (3).zip.052 > mini project > fleetfuel.ipynb > sns.heatmap(df.isnull())
+ Code + Markdown
df.rename(columns={'temp_inside':'temp_inside_column'},inplace=True)
df['temp_inside_column']=df['temp_inside_column'].str.replace(',','.').astype(float)
temp_inside_mean=df['temp_inside_column'].mean()
df['temp_inside_column'].fillna(temp_inside_mean,inplace=True)
[13]
temp_inside_mean=np.mean(df['temp_inside_column'])
[14]
df['temp_inside_column'].fillna(temp_inside_mean,inplace=True)
[15]
sns.heatmap(df.isnull())
[16]


+ Code + Markdown
<Axes: >
0
15
30
45
60
75
90
105
120
135
150
165
180
195
210
225
240
255
270
285
300
315
330
345
360
375
distance
consume
speed
temp_inside_column
temp_outside
gas_type
AC
rain
sun
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100

fleetfuel.ipynb ×

C: > Users > CHARITHA-V > AppData > Local > Temp > 52103a15-8eb6-4498-ab3a-ce0d88b32052_mini project (3).zip.052 > mini project > fleetfuel.ipynb > sns.heatmap(df.isnull())

+ Code + Markdown ···

```
[17]
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
l=LinearRegression()
```

```
[18]
# Assuming 'df' is your original pandas DataFrame
x = df.drop(['consume','gas_type'], axis=1)
x = x.replace(',', '.', regex=True)  # Apply replace on the DataFrame

# Now you can proceed to convert 'x' to a NumPy array if needed
x_array = x.to_numpy()
```

```
[19]
y=df['consume']
```

```
[20]
x=x.values
y=y.values
```

```
[21]
x_array = x
```

+ Code + Markdown ···

```
[21]
x_array = x
```

```
[22]
x_train,x_test,y_train,y_test=train_test_split(x_array,y,test_size=0.3,random_state=42)
```

```
[23]
y_train = np.array([float(val.replace(',', '.')) for val in y_train])  # Fix: Convert y_train elements to floats
```

```
[24]
l = LinearRegression()
l.fit(x_train, y_train)
```

···

```
[25]
x_train.shape
```

··· (271, 7)

+ Code + Markdown ···

```
[26]  y_pred=l.predict(x_test)
```

```
[27]  print(l.coef_,l.intercept_)
```

```
··· [ 0.00523674 -0.02371772 -0.14711979 -0.03724498  0.41456804  0.61676684
     -0.06407861] 9.38930814225712
```

```
[28]  import numpy as np
```

```
[29]  y_test = np.array([float(str(val).replace(',', '.')) for val in y_test])
      y_pred = np.array([float(str(val).replace(',', '.')) for val in y_pred])
```

```
[30]  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)))
```

```
··· 0.7424532609047078
    0.6635761182069617
    0.8616572757800561
```

C: > Users > CHARITHA-V > AppData > Local > Temp > 52103a15-8eb6-4498-ab3a-ce0d88b32052_mini project (3).zip.052 > mini project > fleetf

+ Code + Markdown ···

```
[31]  dum1=pd.get_dummies(df['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]
```

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

```
[33]  df.drop('gas_type',axis=1,inplace=True)
```

+ Code + Markdown

```
[34]  x1=df.drop('consume',axis=1)
```

```
[34]  x1=df.drop('consume',axis=1)
```

```
[35]  y1=df['consume']
```

```
[36]  x1.columns
```

```
··· Index(['distance', 'speed', 'temp_inside_column', 'temp_outside', 'AC', 'rain',
           'sun', 'E10', 'SP98'],
          dtype='object')
```

```
# Convert numerical features to numeric type after replacing ','
for col in df.drop('consume',axis=1).columns:
    if df[col].dtype == 'object':  # Check if the column is of object type (likely string)
        df[col] = df[col].str.replace(',', '', regex=True).astype(float)  # Replace commas and convert to float
x1=df.drop('consume',axis=1).values
y1=df['consume'].str.replace(',','',regex=True).astype(float).values
[37]
```

```
[38]
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x1,y1,test_size=0.3,random_state=42)
from sklearn.linear_model import LinearRegression
l=LinearRegression()
x_train,x_test,y_train,y_test=train_test_split(x1,y1,test_size=0.3,random_state=42)
```

```
[39]
l.fit(x_train,y_train)
...
```

```
[40]
y_pred_1=l.predict(x_test)
print(y_pred_1)
```

```
[42.45427384 47.68318004 46.60145886 47.41816003 39.59832298 51.13256594
 48.62779179 48.54002754 52.44039345 45.51182096 34.11186147 43.01162566
 58.66395262 38.52055261 46.04766373 48.56250368 52.28230535 46.64841556
 50.77304586 47.32897415 32.93562282 46.17741991 43.83041128 48.01756819
 42.79292169 42.81240863 28.33997879 36.99328366 45.72847169 31.22210207
 44.57130935 48.3487666  42.34761831 39.86922831 55.36867075 45.06481804
 40.43733289 31.67491553 45.99293684 56.31877664 38.58640683 47.68016133
 49.16182765 37.48495332 40.90291361 38.2138211  45.54422488 44.99677097
 45.15931042 44.07121017 43.82972677 49.76943788 50.05857493 48.24636735
 40.48294271 41.95514614 60.37739081 48.92032016 41.42453273 42.52986518
 50.40946468 46.93361327 41.00708736 44.27801755 37.13974892 41.64769417
 50.48472818 38.2653709  39.34648245 42.138677   35.93710478 40.39377212
 46.94906379 44.76203895 47.68182974 42.25818633 48.33374493 45.94260108
 48.99267426 46.19315088 47.99170617 45.76026644 37.29983913 44.28569861
```

\+ Code \+ Markdown ···

```
 42.57145922 39.18800714 45.91235978 34.46031928 36.6704684  51.35652418
 45.13819827 42.56652507 46.60145886 47.31637032 45.87394677 28.95640109
 38.70584224 39.2430194  50.7676322  39.99808222 35.67346344 39.14390839
 57.40053549 55.22407044 40.16025848 45.73339057 56.67815221 44.39494785
 41.82242902 50.06354846 43.4986596 ]
```

```
[41]
from sklearn import metrics
print(np.sqrt(metrics.mean_squared_error(y_test,y_pred_1)))
```

```
16.89840612732394
```

```
[42]
x_train.shape
```

```
(271, 9)
```

```
[43]
x_train[0]
```

```
array([123.0, 62, 21.5, 6, 0, 0, 0, True, False], dtype=object)
```

```
[44]
import joblib
joblib.dump(l,'model.save')
```

```
['model.save']
```

DESCRIPTIVE ANALYSIS