

# **FUEL EFFICIENCY, IN-CYLINDRICAL PRESSURE AND HRR PREDICTION**

## **A PROJECT REPORT**

*Submitted by*

**DIKCHA SINGH [RA2011027010096]  
K SANTHANA LAKSHMI [RA2011027010129]**

*Under the Guidance of*

**DR. A. SHANTHINI**

(Associate Professor, Department of Data Science and Business Systems)

*In partial fulfillment of the Requirements for the Degree  
of*

**BACHELOR OF TECHNOLOGY  
COMPUTER SCIENCE ENGINEERING  
with specialization in Big Data Analytics**



**DEPARTMENT OF DATA SCIENCE AND BUSINESS  
SYSTEMS  
FACULTY OF ENGINEERING AND TECHNOLOGY  
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY  
KATTANKULATHUR – 603203**

**NOVEMBER 2023**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  
**KATTANKULATHUR-603203**

**BONAFIDE CERTIFICATE**

Certified that this project report titled "**FUEL EFFICIENCY, IN-CYLINDRICAL PRESSURE AND HRR PREDICTION**" is the bonafide work of **DIKCHA SINGH [RA2011027010096]** AND **K SANTHANA LAKSHMI [RA2011027010129]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

SIGNATURE

**DR. A. SHANTHINI  
GUIDE**  
Associate Professor  
Dept. of DSBS

SIGNATURE

**DR. M.LAKSHMI  
HEAD OF THE DEPARTMENT**  
Dept. of DSBS

Signature of Internal Examiner

Signature of External Examiner

## **ABSTRACT**

The escalating global vehicle count, particularly in emerging economies, poses significant challenges to energy reserves and environmental sustainability. This research endeavors to tackle this pressing issue by focusing on the identification of crucial intrinsic factors that influence fuel efficiency and the prediction of in-cylinder pressure within a mixture of fuels. Our comprehensive investigation delves into a range of machine learning models and formula-based calculations aimed at discerning the optimal fuel blends that not only maximize efficiency but also minimize harmful emissions. The study takes into account pivotal factors such as fuel density, calorific value, and brake power to construct models for predicting fuel efficiency and in-cylinder pressure. Our findings underscore the paramount importance of fuel mixtures and their profound impact on the efficiency of various brake power levels. Furthermore, this research involves a comparative analysis of multiple machine learning models, including Multiple Linear Regression, Random Forest Regressor, and Decision Tree Regressor, Gradient Boosting Regressor to offer valuable insights into their respective performance characteristics.

## ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr T.V.Gopal**, for his invaluable support.

We wish to thank **Dr Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr M. Lakshmi**, Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our program coordinators **Dr. A Shanthini**, Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for her inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Dr. S. Sharanya**, Assistant Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Dr. A Shanthini**, Associate Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under her mentorship. She provided me with the freedom and support to explore the research topics of my interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Data Science and Business Systems staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

Dikcha Singh (RA2011027010096)

K. Santhana Lakshmi (RA2011027010129)

# TABLE OF CONTENTS

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iii
	LIST OF TABLES	vii
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1.	<b>INTRODUCTION</b>	1
1.1	GENERAL	2
1.2	PURPOSE	2
1.2.1	SCOPE	3
1.2.2	NEED FOR FUEL EFFICIENCY PREDICTION	3
1.3	SUSTAINABLE TECHNOLOGY	4
1.4	MOTIVATION AND PROBLEM STATEMENT	5
2	<b>LITERATURE REVIEW</b>	6
2.1	FUEL EFFICIENCY PREDICTION	6
2.2	FUEL CONSUMPTION PREDICTION MODEL USING ML	8
2.3	VEHICLE FUEL CONSUMPTION PREDICTION USING ML	9
2.4	VEHICLE FUEL CONSUMPTION PREDICTION	10
2.5	FUEL EFFICIENCY MODELING AND PREDICTION FOR AV	12
3	<b>SYSTEM ARCHITECTURE AND DESIGN</b>	14
3.1	ARCHITECTURE DIAGRAM	14
3.2	USE CASE DIAGRAM	15
3.3	DATA FLOW DIAGRAM	16
4	<b>PROPOSED METHODOLOGY</b>	17
4.1	DATA COLLECTION	17
4.2	DATA CLEANING	17
4.3	DATA PROCESSING	17
4.4	TRAINING AND TESTING MODELS	18
4.5	IDENTIFICATION AND PREDICTION	19
4.6	DATA VISUALIZATION	19
4.7	RESULTS AND ACCURACY	20

<b>5</b>	<b>MODULE STUDY</b>	<b>21</b>
5.1	DATA PREPARATION AND CLEANING	21
5.2	EFFICIENCY CALCULATION	21
5.3	MACHINE LEARNING FOR PRESSURE PREDICTION	21
5.4	HEAT RELEASE RATE (HRR) ANALYSIS	22
5.5	MODEL PERFORMANCE AND RESULTS	22
5.6	CONCLUSION AND RECOMMENDATIONS	22
5.7	DOCUMENTATION AND FUTURE WORK	22
<b>6</b>	<b>CODING AND TESTING</b>	<b>23</b>
6.1	DATA PREPROCESSING	23
6.2	EFFICIENCY CALCULATION	30
6.3	PRESSURE PREDICTION	31
6.4	HRR ANALYSIS	37
<b>7</b>	<b>RESULTS AND DISCUSSIONS</b>	<b>39</b>
7.1	PERFORMANCE MATRIX	39
<b>8</b>	<b>CONCLUSION</b>	<b>41</b>
<b>9</b>	<b>FUTURE ENHANCEMENTS</b>	<b>42</b>
<b>10</b>	<b>REFERENCES</b>	<b>43</b>
	<b>APPENDIX</b>	<b>44</b>
	<b>PAPER PUBLICATION STATUS</b>	<b>45</b>

## LIST OF TABLES

6.2.1	Mass of fuels after calculation.....	30
6.3.1	Dataset for pressure prediction.....	31
6.3.2	Tabulating errors after prediction.....	37
7.2.1	Final pressure and HRR.....	40

## **LIST OF FIGURES**

3.1	Architecture diagram.....	14
3.2	Use Case diagram.....	15
3.3	Data Flow diagram.....	16

## ABBREVIATIONS

<b>AI</b>	Artificial Intelligence
<b>ML</b>	Machine Learning
<b>AV</b>	Automotive Vehicle
<b>SVM</b>	Support Vector Machine
<b>KNN</b>	K Nearest Neighbour
<b>OBD</b>	On Board Diagnostics
<b>HRR</b>	Heat Release Rate
<b>MPG</b>	Miles per Gallon
<b>CNSL</b>	Cashew Nut Shell Oil
<b>BP</b>	Brake Power
<b>MAF</b>	Mass Air Flow
<b>VS</b>	Vehicle Speed
<b>RPM</b>	Revolutions Per Minute
<b>TPS</b>	Throttle Position Sensor
<b>IoT</b>	Internet of Things
<b>GPS</b>	Global Positioning System
<b>MII</b>	Mutual Information Index
<b>KPI</b>	Key Performance Indicators
<b>RMSE</b>	Root Mean Square Error
<b>MSE</b>	Mean Square Error

# **CHAPTER 1**

## **INTRODUCTION**

As the number of vehicles on the planet experiences an unprecedented upsurge, especially in developing nations such as China and India, the resultant increase in the production of harmful greenhouse gasses and pollutants presents an immediate threat to the ecological balance of our planet. Simultaneously, the constant depletion of non-renewable resources, particularly petroleum, increases the need for creative and long-lasting solutions. In the midst of this environmental hell, smart route planning shows great promise as a potential solution to the ecological cost imposed by traditional transportation methods. However, a more thorough approach is necessary due to the complex terrain of vehicular dynamics.

To meet these demands, our research sets out on an ambitious and multi-pronged mission to decipher the complexities of combustion dynamics, forecast the illusive heat release rate, identify the ideal fuel efficiency, and precisely compute cylinder pressure at 720 angles. This quest for automotive enlightenment is supported by a strong framework that combines the formidable powers of machine learning algorithms with the clarity provided by advanced visualization techniques and the methodical approach of formulaic thinking, rather than being left to the whims of chance.

Our goal goes beyond simple scholarly curiosity as we explore the intersection of scientific research and technology innovation. We see a time where efficiency and sustainability coexist harmoniously, completely changing what it means to use cars. Our study aspires to be a catalyst for revolutionary change, leading the way for a peaceful cohabitation between humanity's automotive goals and the imperative of environmental stewardship through the synthesis of state-of-the-art technology and thorough scientific methods.

## **1.1 GENERAL**

With developing nations such as China and India leading the way in the global automobile industry, the number of vehicles on the road is increasing at an unsustainable rate, leading to a serious depletion of non-renewable resources like petroleum. This unrelenting path is further exacerbated by the simultaneous production of undesired greenhouse gasses and emissions, posing a threat to the fragile ecological balance of our world.

Strategic route design becomes a powerful tool in the armory of limiting environmental effect in response to this vehicular juggernaut. A sophisticated strategy with the potential to reduce fuel use, this methodology struggles with its broad applicability. There isn't a single, universally applicable answer due to the complexity of various driving situations and infrastructure quirks.

In this context, by eschewing traditional fossil fuels, our research efforts pioneer a new direction in the field of sustainable mobility. We steer clear of the negative effects linked to conventional petrol and diesel in favor of the use of biofuels. Our two main goals in this revolutionary voyage are to determine the biofuels efficiency quotient for use in automobiles and to forecast the complex dynamics of cylinder pressure.

## **1.2 PURPOSE**

Our scientific endeavors are driven by a tremendous mission to transform the field of engine performance prediction while pursuing efficiency and innovation at all costs. The idea to move beyond the conventional limitations of laboratory testing and usher in a time where engine performance can be predicted without physically operating the engines in controlled circumstances is at the heart of this audacious ambition. This paradigm change promises significant time and cost savings, but it also profoundly conserves the priceless energy and knowledge of our technical team.

We set out on a path that not only addresses the financial costs related to engine testing but also frees up the valuable time of our knowledgeable personnel by avoiding the need for lengthy laboratory experiments. Once tied to the demands of physical testing, this newly acquired financial and temporal capital can now be carefully diverted to more meaningful and creative endeavors. The benefits of this efficiency reach beyond our research labs and into domains where our technical workforce's combined knowledge and creativity can be used to tackle urgent problems, drive ground-breaking ideas, and make significant contributions to advancements outside of engine testing.

Essentially, our effort to forecast engine performance without requiring actual experiments constitutes both a responsible use of resources and a significant technological advancement. It is evidence of our dedication to effectiveness, sustainability, and the best possible use of human resources. As we navigate this unexplored area, we are focused not just on the revolutionary potential in our labs but also on the wider range of opportunities where the benefits of our vision might materialize in a more efficient, effective, and significant future.

### **1.2.1 SCOPE**

Before moving on to the domain of real experiments, we can build a solid foundation by utilizing analogous outcomes from simulations. This method not only saves money and gives us the luxury of time, but it also acts as a tactical prelude to the practical stage. We may simplify our experimental design and concentrate resources on the most promising directions by using simulations to determine the points that perform the best. The precision rates achieved in the simulated environment instill confidence in the validity of our methodology, providing a robust basis for the upcoming real-world experiments. Essentially, this two-pronged strategy balances theoretical accuracy with real-world relevance, maximizing the effectiveness and efficiency of the research process.

### **1.2.2 NEED OF FUEL EFFICIENCY PREDICTION**

This initiative is critical for a number of reasons. First of all, it directly tackles the critical issue of diesel engine fuel efficiency optimisation in cylindrical pressure scenarios, which is a cornerstone in many businesses and the transportation sector. Determining the optimal fuel mixtures for various loads is essential for enhancing overall engine performance and, in turn, operating economy.

Practical insights into real-world applications are offered by the inclusion of a wide variety of over 20 fuels and their combinations. This specificity is essential since it makes it possible to customize solutions for various sectors and circumstances, guaranteeing that the project's conclusions have quick and direct application.

Because diesel engines are widely used and biofuels have potential, there is a substantial focus on CNSL as the parent fuel along with other fuels like methanol and diesel. This project acts as a useful manual for putting biofuels into practice successfully, overcoming the challenges that come with using them directly, and proving that they are a competitive fuel substitute.

Not only is the combination of machine learning and data analytics a technological requirement, but it is also a luxury. A common shortcoming of mechanical engineers is their inability to precisely handle huge datasets. The project's use of these cutting-edge tools enables effective analysis, saving time and money while guaranteeing a degree of precision that is essential for making decisions.

Moreover, by reducing the hazardous pollutants that cars spew into the environment, the project directly supports environmental sustainability. Optimizing fuel combinations is a concrete step towards developing more environmentally friendly energy conversion processes that are in line with international initiatives to mitigate climate change and lower carbon footprints.

### **1.3 SUSTAINABLE TECHNOLOGY**

We are firmly in the field of sustainability technology because of our participation in this project. As follows:

- 1. Energy Efficiency Optimisation:** It helps optimize the energy efficiency of diesel engines by concentrating on determining the optimal fuel combinations and forecasting Heat Release Rate. This is a crucial component of sustainability since it lowers operating expenses and total energy usage.
- 2. Integrating Biofuel:** The project focus on conventional fuels in addition to biofuels like CNSL is in line with sustainability objectives. Because they are frequently made from renewable resources, which lessens the need for non-renewable fossil fuels and lessens their negative effects on the environment, biofuels are thought to be more environmentally friendly.
- 3. Decrease in Impact on the Environment:** The project's main objective, reducing the amount of hazardous pollutants that cars emit, directly tackles environmental issues. This emission decrease helps to develop more environmentally friendly and sustainable industrial and transportation methods.
- 4. Accurate Machine Learning:** Using Data analytics and machine learning in your project is a technological advance that has consequences for sustainability. You help to reduce the total environmental impact of the research, minimize needless testing, and increase resource efficiency by optimizing the analysis process and increasing precision in identifying the best fuel combinations.
- 5. Encouraging Eco-Aware Decision-Making:** Your initiative provides data-driven insights that can steer the transportation and industry sectors towards more environmentally friendly practices. This involves selecting gasoline blends that support wider environmental and sustainability goals while simultaneously improving performance. Our effort in figuring out the best fuel mixtures for diesel engines is essentially an application of sustainable technology in real life. It tackles the requirement for energy efficiency, investigates fuel substitutes that are less harmful to the environment, and uses cutting edge technology to make operations more sustainable. Research of this nature is essential to bringing in a day when sustainability and technology live side by side.

## **1.4 MOTIVATION**

Meet both the ends of the most practical energy conversion device. One side is diesel engines are needed in various sectors like industry and transport since it is most reliable.

Other side we have biofuels that are on par with diesel but they have some drawbacks if directly used.

Through this project we can find which fuel combination at which load gives best performance and to predict the Cylindrical Pressure. Since Mechanical engineers don't know ML and Data Analytics it is difficult for them to work with huge datasets and precision.

In this project we have 20 + Fuels and its combination of Fuels. We are going to find which combination of amount of fuel gives best Efficiency with Several Loads such as 5.2 Brake Power, 4.16, 3.12, 2.08, 1.04 . Our parent Fuel is CNSL(Cashew Nut Shell Oil) combined with other Fuels like Cashew nut Shell oil, Orange Peel oil etc.

### **1.4.1 PROBLEM STATEMENT**

We have 20 + Fuels and its combination of Fuels. We are going to find which combination of amount of fuel gives best Efficiency with Several Loads such as 5.2 Brake Power, 4.16, 3.12, 2.08, 1.04 and to predict the cylindrical Pressure and HRR. Our parent Fuel is CNSL(Cashew Nut Shell Oil) combined with other Fuels like Coconut oil, Cotton seed oil etc.

# CHAPTER 2

## LITERATURE STUDY

### 2.1 FUEL EFFICIENCY PREDICTION

Authors:M. Aditya Vamsi, B. Raja Rishita, M. Amin Qurishi, N. Siva Sandeep

#### OBJECTIVE:

The primary objective of the paper is to develop prediction models using machine learning techniques to estimate the fuel efficiency (Miles per Gallon - MPG) of various vehicles. The focus is on creating accurate models that can assess fuel consumption based on known specifications.

1. **Model Development:** The paper aims to employ machine learning algorithms such as Linear Regression, Random Forest, Decision Tree, and KNN to construct prediction models. Additionally, deep learning concepts will be implemented to explore their effectiveness in fuel efficiency prediction.
2. **Comparison of Models:** The research intends to compare different machine learning models to evaluate their performance in terms of error reduction and predictive accuracy. This involves assessing models such as Random Forest, Decision Tree, KNN, and Linear Regression to identify the one with the least error and better efficiency.
3. **Dataset Utilization:** The study aims to utilize datasets relevant to machine learning practitioners to train and test the prediction models. These datasets include information on various cars, incorporating features such as cylinders, displacement, horsepower, and weight.
4. **Industry Impact:** The paper aims to contribute to the automotive industry by providing a tool that can assist car makers in optimizing their processes for increased fuel efficiency. The accurate prediction models can potentially give manufacturers a competitive edge in the market.

## **MOTIVATIONS:**

1. **Rising Fuel Costs:** The continuous increase in fuel costs serves as a significant motivation for car manufacturers to optimize fuel efficiency in their vehicles. The paper seeks to address this concern by providing a tool that can accurately predict and improve fuel efficiency.
2. **Consumer Preferences:** As customers become more discerning about vehicle features, the motivation is to create more fuel-efficient cars to meet consumer demands. Accurate prediction models can guide manufacturers in adapting their processes to enhance fuel efficiency.
3. **Market Competition:** In a highly competitive market, having a more fuel-efficient vehicle can set a company apart from its competitors. The paper aims to contribute to this aspect by developing models that can lead to the creation of more fuel-efficient and attractive vehicles.
4. **Machine Learning Advancements:** The motivation behind using machine learning techniques is the advancements in the field. By leveraging these techniques, the paper aims to improve the accuracy of fuel efficiency predictions, allowing for better decision-making in the automotive industry.
5. **Environmental Impact:** Increasing fuel efficiency not only benefits manufacturers economically but also contributes to environmental sustainability. The motivation includes a broader goal of reducing the overall carbon footprint by promoting more fuel-efficient vehicles.

## **2.2 FUEL CONSUMPTION PREDICTION MODEL USING MACHINE LEARNING AND MATHEMATICAL METHODS**

Authors :Xianwei Xie, Baozhi Sun, Xiaohe Li, Tobias Olsson, Neda Maleki and Fredrik

### **OBJECTIVE:**

The primary objective is to develop precise fuel consumption prediction models for ships. The paper explores two types of models - a white-box model based on mathematical methods and a black-box model based on machine learning.

- 1. Apply Data Cleaning Methods:** The study introduces a data preprocessing cleaning method based on the Kwon formula. This method aims to eliminate noise in the data generated during the acceleration and deceleration processes, ensuring the accuracy of the prediction models.
- 2. Evaluate Model Performance:** The paper employs ship model test data and regression methods to evaluate the accuracy of the developed models. It also discusses the application of the data-cleaning method in the preprocessing of the black-box model and assesses its impact on model performance.
- 3. Validate Models Under Simulated Conditions:** The study validates the performance of the models by correlating predicted fuel consumption rates with ship speed under simulated conditions. This validation step ensures that the models can accurately represent real-world scenarios.
- 4. Enhance Decision Support for Shipping Companies:** The ultimate goal is to provide decision support for shipping companies. Accurate fuel consumption prediction models can aid in navigation status analysis, energy conservation, and emission reduction, contributing to the overall sustainability of maritime operations.

By addressing these objectives, the paper aims to contribute valuable insights and practical tools for the maritime industry to improve fuel efficiency, reduce operating costs, and make informed decisions aligned with environmental goals.

## **MOTIVATION:**

The motivation for this paper arises from the increasing emphasis on environmental sustainability in the international shipping community. With a focus on reducing greenhouse gas emissions, there is a growing need for accurate fuel consumption prediction models in the maritime industry. Shipping companies are also keen on improving the energy efficiency of their vessels to cut down on fuel costs, which constitute a significant portion of their operating expenses. As ships become equipped with a multitude of sensors, there is a vast amount of data available that can be leveraged for optimizing energy usage.

## **2.3 VEHICLE FUEL CONSUMPTION PREDICTION USING MACHINE LEARNING**

Authors: Mohamed A. HAMED, Mohammed H. Khafagy, Rasha M. Badry

### **OBJECTIVE:**

The objective of this paper is to enhance the accuracy of fuel consumption prediction models using machine learning, specifically employing the Support Vector Machine (SVM) algorithm. The focus is on minimizing fuel consumption, which is crucial for economic improvement in businesses and addressing the broader global concern regarding fuel usage. The paper aims to contribute a machine learning model that predicts vehicle fuel consumption based on key parameters: Mass Air Flow (MAF), Vehicle Speed (VS), Revolutions Per Minute (RPM), and Throttle Position Sensor (TPS) features.

The proposed model is tested on a vehicle's On-Board Diagnostics (OBD) dataset, involving 18 features. The goal is to achieve high accuracy in fuel consumption prediction, measured by the R-Squared metric. The paper contends that the SVM algorithm, when applied to fuel consumption prediction, outperforms other related works using the same algorithm.

## **MOTIVATION:**

Fuel consumption is of significant interest to individuals, businesses, and the global economy, given its impact on economic factors. Fluctuations in fuel prices can influence the economic landscape, making accurate fuel consumption prediction crucial for effective planning and decision-making.

The motivation behind this research is to leverage machine learning, particularly the SVM algorithm, to develop a robust model for fuel consumption prediction. The utilization of OBD data, which is part of the Internet of Things (IoT) technique, allows for real-time and remote monitoring of vehicles. This data-driven approach, combined with machine learning techniques, can lead to improved accuracy in predicting fuel consumption.

The paper addresses a gap in existing literature by proposing a model that considers both RPM\_TPS-based and VS\_MAF-based equations for fuel consumption prediction. It emphasizes the importance of these equations and their parameters, asserting that they are key to accurately measuring fuel consumption rates.

Furthermore, the paper discusses the significance of feature weighting and selection, emphasizing the importance of these processes before applying the SVM algorithm. The motivation is not only to predict fuel consumption but also to identify and prioritize the most influential features contributing to accurate predictions.

In summary, the objective is to advance fuel consumption prediction models using SVM, and the motivation lies in the broader implications for economic improvement, business optimization, and addressing global concerns related to fuel consumption through the application of machine learning techniques.

## **2.4 VEHICLE FUEL CONSUMPTION PREDICTION METHOD BASED ON DRIVING BEHAVIOR DATA COLLECTED FROM SMARTPHONES**

Authors: Ying Yao, Xiaohua Zhao ,Chang Liu ,Jian Rong, Yunlong Zhang

### **OBJECTIVE:**

The objective of this study is to develop a method for predicting vehicle fuel consumption based on driving behavior, with a specific focus on utilizing data collected from mobile phone sensors. As the energy consumption of private cars in China continues to rise, reaching significant levels, the need to address and reduce energy consumption in the transportation sector becomes paramount. This paper aims to contribute to the solution by investigating the correlation between driving behavior and energy consumption and leveraging mobile phone data to predict fuel consumption accurately.

The central goal is to construct a fuel consumption prediction model based on Global Positioning System (GPS) data obtained from smartphones. Through this model, the study seeks to establish a connection between driving behavior data collected from mobile phones and fuel consumption data from On-Board Diagnostics (OBD) terminals. The intention is not only to improve real-time monitoring databases with enhanced error tolerance but also to provide a theoretical foundation for urban traffic fuel consumption management and policy effectiveness evaluation.

## **MOTIVATION:**

The motivation behind this research stems from the escalating energy consumption of private cars in China, emphasizing the urgency to address this issue for sustainable urban transportation development. Driving behavior is identified as a crucial factor influencing energy consumption, and studies have shown that improvements in driving behavior can lead to significant fuel economy gains.

While existing research has explored prediction models for energy consumption based on driving behavior, this study introduces a novel approach by incorporating data collected from mobile phone sensors. Traditional data sources, such as OBD devices and questionnaires, have limitations in terms of scale, cost, and data quality. The motivation is to overcome these limitations by leveraging mobile phone technology for larger-scale data collection, offering a more detailed and easily accessible dataset for analyzing driving behavior.

The study acknowledges the challenges associated with using mobile phone data, including potential inaccuracies due to factors like phone placement and vibrations caused by vehicle movement. However, it proposes a method to calibrate and utilize this data effectively for predicting fuel consumption. The adoption of mobile phone terminals for data collection is seen as a cost-effective alternative to OBD equipment installation, providing a practical solution for traffic management departments to monitor urban traffic fuel consumption more accurately.

In conclusion, the motivation is to explore the untapped potential of mobile phone data in predicting vehicle fuel consumption, considering its widespread use and potential for large-scale data collection. The study aims to bridge the gap between driving behavior and fuel consumption prediction, offering valuable insights for energy-efficient urban transportation management.

## **2.5 FUEL EFFICIENCY MODELING AND PREDICTION FOR AUTOMOTIVE VEHICLES: A DATA-DRIVEN APPROACH**

Authors: Xunyuan Yin, Zhaojian Li, Sirish L. Shah, Lisong Zhang, Changhong Wang

### **OBJECTIVE:**

The primary objective of this study is to develop fuel efficiency prediction models for common automobiles based on an extensive and informative vehicle database. The focus is on identifying the key characteristics that significantly influence fuel efficiency and utilizing machine learning techniques to build accurate prediction models. The study explores five different machine learning techniques, with a specific emphasis on quantile regression, which has shown superior performance compared to other methods. The overarching goal is to contribute valuable insights for car designers and manufacturers to enhance fuel economy by considering specific characteristics that impact fuel efficiency.

### **MOTIVATION:**

The global increase in the number of vehicles, particularly in developing countries like China and India, has led to significant concerns regarding energy resource consumption and environmental impact. The negative consequences, such as the consumption of non-renewable resources and the generation of greenhouse gasses, necessitate effective strategies to reduce fuel consumption in vehicles. One promising approach is the development of route planning strategies; however, their applicability is limited in many areas. Therefore, this study aims to fill the gap by identifying major characteristics affecting fuel economy and developing predictive models for fuel efficiency.

The motivation stems from the need to provide car designers and manufacturers with an objective model that can guide improvements in fuel economy. By leveraging real-world vehicle data, the study aims to analyze and model fuel efficiency, considering features such as engine displacement, vehicle size, weight, and aerodynamic performance. The adoption of quantile regression as a prediction method is motivated by its demonstrated effectiveness in capturing the nuances of fuel efficiency prediction compared to other regression methods.

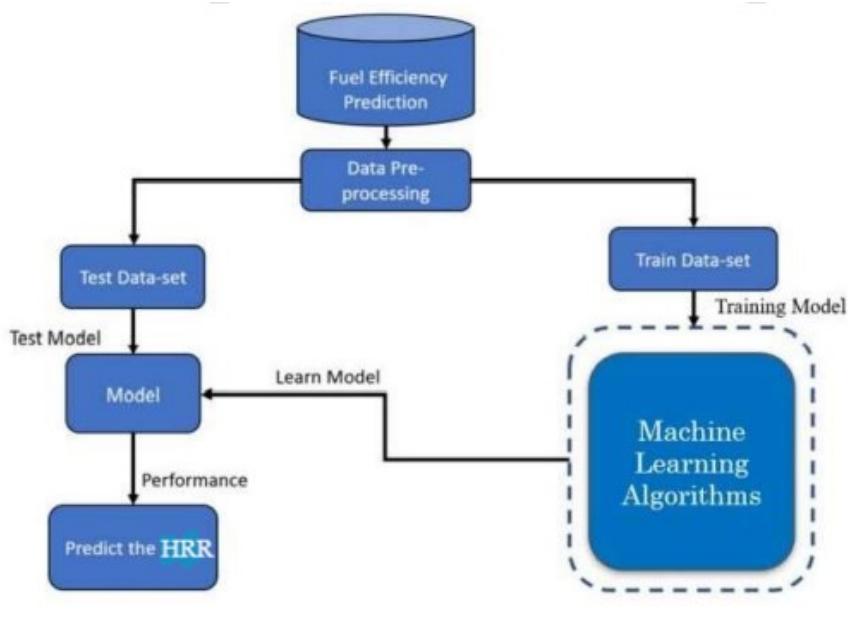
Furthermore, the study recognizes the challenges associated with fuel consumption models based on vehicle dynamics, which require detailed knowledge of the physical structure of vehicles. The proposed approach focuses on using a mutual information index (MII) to identify significant characteristics affecting fuel efficiency, providing a more adaptable and practical solution.

In summary, the motivation is to contribute to the understanding of fuel efficiency in common automobiles, provide practical guidance for car designers, and offer predictive models that can enhance fuel economy by considering specific characteristics. The study aims to demonstrate the effectiveness of quantile regression and the relevance of identified features in predicting fuel efficiency accurately.

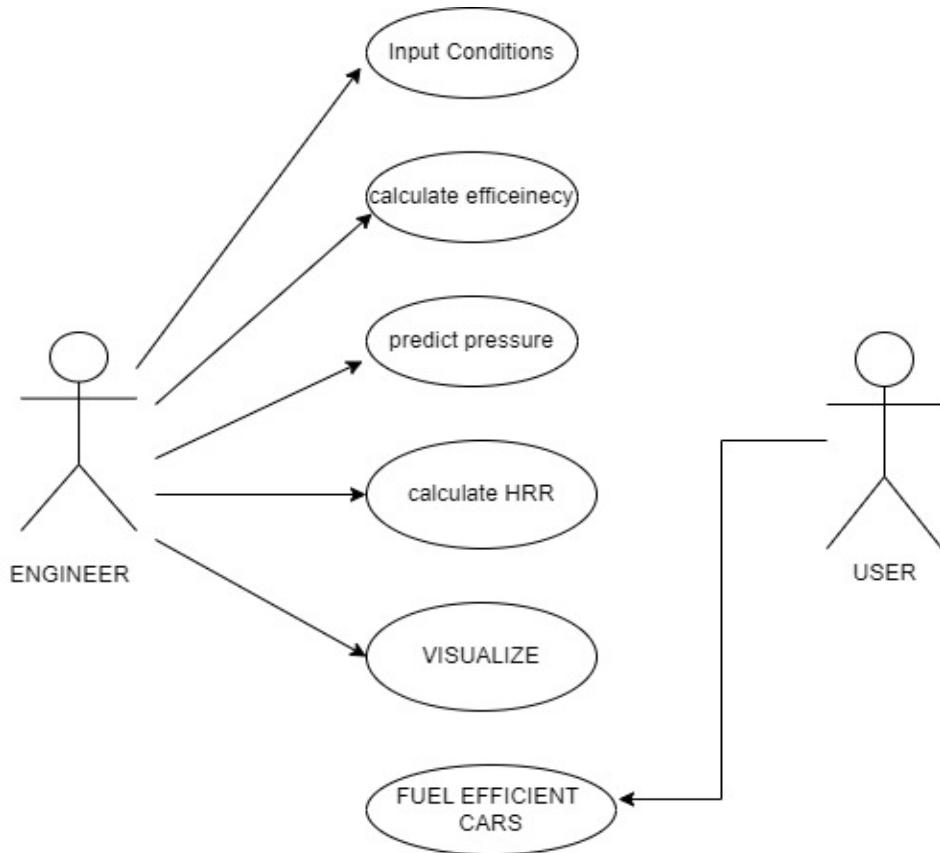
# CHAPTER 3

## SYSTEM ARCHITECTURE AND DESIGN

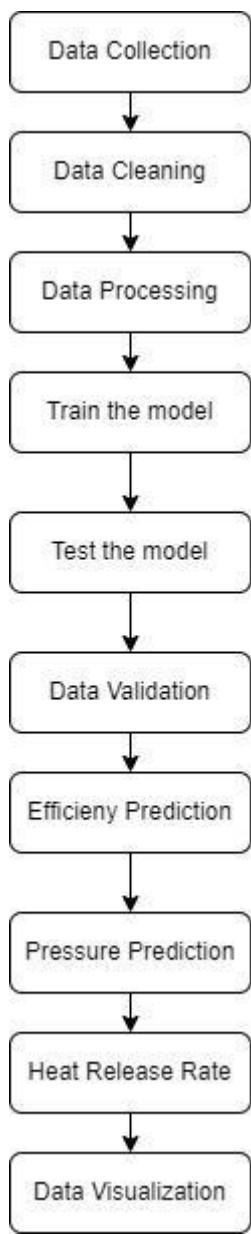
### 3.1 ARCHITECTURE DIAGRAM



### 3.2 USE CASE DIAGRAM



### 3.3 DATA FLOW DIAGRAM



## CHAPTER 4

### PROPOSED METHODOLOGY

#### 4.1 Data Collection:

##### 1. Data Variables:

The parameters like fuel types, fuel combinations, loads (Brake Power), pressure, calorific value, viscosity, volume, crank angle, density, mass of fuel, cetane are collected.

##### 2. Data Sources:

The Data is collected from the Experimental setup and stored in the database.

#### 4.2 Data Cleaning:

##### 1. Handling Missing Data:

Identify and address missing values in the dataset. Imputing missing values based on statistical measures, removing rows or columns with substantial missing data.

##### 2. Standardization:

Standardize the units of measurement to maintain consistency across different data points. This is crucial for accurate analysis and interpretation.

#### 4.3 Data Processing:

##### 1. Data Integration:

Dataset is sourced from multiple places, we integrate the data to create a unified dataset. Ensured consistency in variable names, units, and formats. First Efficiency are found and the values are integrated for pressure dataset.

##### 2. Data Transformation:

Transformations to variables using mathematical operations (formula method) to find the efficiency by using the efficiency formula and to find the mass of fuel and HRR value identification.

### **3. Normalization and Scaling:**

Normalize or scale numerical variables on different scales. This ensures that all variables contribute equally to the analysis, preventing those with larger magnitudes from dominating the results.

### **4. Data Splitting:**

Dividing the dataset into training and testing sets. This ensures that the model is trained on one subset of the data and tested on another, providing a more robust evaluation of its performance.

## **4.4 TRAINING AND TESTING MODELS**

**The models considered in our analysis include:**

### **1. Multiple Linear Regression:**

Multiple linear regression is a straightforward approach to modeling the relationship between multiple independent variables and a dependent variable, making it a suitable choice for our multi-factor analysis of In-cylinder pressure.

### **2. Random Forest Regressor:**

Random forests are an ensemble learning method known for their ability to handle complex relationships in data. They are capable of capturing non-linear patterns and interactions between factors, making them a valuable tool for predicting pressure.

### **3. Decision Tree Regressor:**

Decision trees are a simple yet effective model for regression tasks. We found that decision tree regressors performed well in capturing the impact of various factors on in-cylinder pressure.

### **4. Gradient Boosting Regressor:**

Gradient boosting is an ensemble technique that combines the predictions of multiple weak learners to create a strong predictive model. Gradient boosting regressors are known for their high accuracy in regression tasks.

#### **4.5 IDENTIFICATION AND PREDICTION:**

Efficiency is calculated using formula methods of efficiency and pressure is predicted using machine learning algorithms like Decision Tree , Random Forest , Linear Regression and Gradient Boosting and HRR calculated using the formula method of HRR.

##### **1. Conversion Brake Power to Equivalent Load Formula:**

$$\text{Brake Power} = (2 * \pi * n * t) / 60000$$

$$n=1500, r=0.185, t=w^*r^*9.81$$

##### **2. Efficiency Formula:**

$$\text{Mass of fuel 1} = ((\text{amount of fuel 1} * 0.000001) / 60) * \text{density of fuel 1}$$

$$\text{Mass of fuel 2} = ((\text{amount of fuel 2} * 0.000001) / 60) * \text{density of fuel 2}$$

$$\text{Heat input} = \text{mass of fuel 1} * \text{calorific value of fuel 1} + \text{mass of fuel 2} * \text{calorific value of fuel 2}$$

$$\text{Efficiency} = \text{Brake Power} / \text{Heat input}$$

##### **3. Heat Release Rate Formula:**

$$\text{HRR} = (((1.35 / (1.35 - 1)) * (\text{Pressure} * ((\text{Volume} - dV) / 1))) + ((1 / (1.35 - 1)) * (\text{Volume} * ((\text{Pressure} - dP) / 1))))$$

#### **4.6 Data Visualization:**

Data visualization is a key component of the data analysis process, allowing to communicate complex patterns and insights in a visually accessible format.

##### **1. Normalization Visualizations:**

- The paper discusses data normalization to facilitate analysis. Visualizing the effects of normalization, such as before-and-after plots, can help readers understand how scaling impacts the data and why it is essential for certain analyses.

##### **2. Regression Performance Plots:**

- For the regression models, visualizations like scatter plots comparing predicted fuel efficiency against actual values are likely included. This allows readers to assess the accuracy of the predictive models and understand how well they align with the observed data.

### **3. Feature Importance Plots:**

- In the context of machine learning model training, the paper may include visualizations highlighting the importance of different features in predicting fuel efficiency. This could be done through bar charts or other graphical representations.

### **4. Validation Results Visualization:**

- When presenting the results of model validation, the paper might include visualizations to compare predicted and actual fuel efficiency values. This helps in assessing the robustness and generalization capabilities of the predictive models.

## **4.7 Results and Accuracy :**

### **Performance Matrix:**

#### **1. Random Forest Regressor:**

Mean Square Error: 0.234809795

Root Mean Square Error: 0.484571765

R Square: 0.998782242

#### **2. Decision Tree Regressor:**

Mean Square Error: 0.33849539

Root Mean Square Error: 0.58180357

R Square: 0.998244513

#### **3. Gradient Boosting Regressor:**

Mean Square Error: 0.840088763

Root Mean Square Error: 0.916563562

R Square: 0.995643176

#### **4. Linear Regression:**

Mean Square Error: 151.1745303

Root Mean Square Error: 12.29530522

R Square: 0.215986548

# **CHAPTER 5**

## **MODULE STUDY**

### **5.1 Module 1: Data Preparation and Cleaning**

#### **1. Data Collection and Cleaning:**

- Handling missing values.
- Standardizing the dataset.
- Removing outliers using the z-score test.

### **5.2 Module 2: Efficiency Calculation**

#### **1. Mass Calculation:**

- Applying the provided formulas to calculate the mass of each fuel.
- Converting brake power to equivalent load.

#### **2. Efficiency Iteration:**

- Implementing a Python loop for efficiency calculations.
- Setting and applying an efficiency threshold.

#### **3. Best Efficiency Identification:**

- Storing the best efficiency fuel combination in an Excel sheet.

### **5.3 Module 3: Machine Learning for Pressure Prediction**

#### **1. Data Preparation for Pressure Prediction:**

- Selecting relevant features for pressure prediction.
- Utilizing the best efficiency fuel combination data.

#### **2. Model Training and Validation:**

- Splitting the dataset into training and testing sets.
- Evaluating and selecting the best-performing machine learning model.

#### **3. Pressure Prediction:**

- Using the selected model to predict in-cylinder pressure for all 720 angles.

## **5.4 Module 4: Heat Release Rate (HRR) Analysis**

### **1. HRR Calculation:**

- Applying the provided formula for HRR.
- Incorporating pressure, volume, and relevant changes.

### **2. Visualization of HRR:**

- Creating visualizations to represent the calculated Heat Release Rate.

## **5.5 Module 5: Model Performance and Results**

### **1. Performance Metrics:**

- Storing performance metrics for machine learning models.
- Documenting accuracy and error rates.

### **2. Results Storage:**

- Compiling key results, predictions, and efficiency data in Excel sheets.

## **5.6 Module 6: Conclusion and Recommendations**

### **1. Summary of Findings:**

- Summarizing the identified best efficiency fuel combination.
- Highlighting pressure predictions and HRR analysis.

### **2. Recommendations for Further Study:**

- Suggestions for refining models or exploring additional factors.

## **5.7 Module 7: Documentation and Future Work**

### **1. Methodology Documentation:**

- Detailed documentation of the methods and assumptions used.
- Transparency in data processing and analysis.

### **2. Future Work:**

- Suggestions for potential future improvements or expansions on the study.

# CHAPTER 6

## CODING AND TESTING

### 6.1 DATA PREPROCESSING AND ANALYSIS

```
# Supress Warnings
import warnings
warnings.filterwarnings('ignore')

# Import the numpy and pandas package
import numpy as np
import pandas as pd

# Data Visualisation
import matplotlib.pyplot as plt
import seaborn as sns

d = pd.DataFrame(pd.read_csv("pro1.csv"))
d.head(15)
```

	Unnamed: 0	FID	cf	Load	Fuel1	Fuel2	Total_Vol	Mass_Fuel	Heat_Input	BP	BTE
0	13	4	34500	4.19	17.658	1.962	19.62	0.000311	10.739879	1.22	11.359532
1	14	4	34500	6.63	22.041	2.449	24.49	0.000389	13.405690	1.92	14.322276
2	17	5	34200	3.48	15.264	3.816	19.08	0.000298	10.199002	1.03	10.099027
3	18	5	34200	7.36	18.224	4.556	22.78	0.000356	12.176796	2.14	17.574410
4	19	5	34200	11.05	22.400	5.600	28.00	0.000438	14.967089	3.16	21.112990
5	20	6	33800	5.34	16.912	7.248	24.16	0.000372	12.585085	1.56	12.395626
6	21	6	33800	7.41	18.648	7.992	26.64	0.000411	13.876931	2.14	15.421277

```
d.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48 entries, 0 to 47
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Unnamed: 0    48 non-null    int64  
 1   FID          48 non-null    int64  
 2   cf            48 non-null    int64  
 3   Load          48 non-null    float64 
 4   Fuel1         48 non-null    float64 
 5   Fuel2         48 non-null    float64 
 6   Total_Vol    48 non-null    float64 
 7   Mass_Fuel    48 non-null    float64 
 8   Heat_Input   48 non-null    float64 
 9   BP            48 non-null    float64 
 10  BTE           48 non-null    float64 
dtypes: float64(8), int64(3)
memory usage: 4.2 KB

[ ] d.shape
(48, 11)

[ ] list(d.columns)
['Unnamed: 0',
 'FID',
 'cf',
 'Load',
 'Fuel1',
 'Fuel2',
 'Total_Vol',
```

First we need to load the datasets from the given csv files and analyze and normalize it to make data trained to make it easy for modeling over the algorithms.

```
d.describe()

   Unnamed: 0      FID      cf     Load    Fuel1    Fuel2  Total_Vol  Mass_Fuel  Heat_Input      BP      BTE
count  48.000000  48.000000  48.000000  48.000000  48.000000  48.000000  48.000000  48.000000  48.000000  48.000000
mean   54.604167  14.416667  35985.416667  7.766667  16.168188  4.897021  21.065208  0.000326  11.700825  2.213750  18.600367
std    23.820909  6.235906  1596.270865  2.888487  3.414496  1.770067  3.828418  0.000060  1.938370  0.800565  5.060900
min    13.000000  4.000000  32000.000000  3.090000  11.277000  1.962000  14.270000  0.000223  8.399613  1.000000  9.431531
25%    34.750000  9.000000  34725.000000  5.700000  12.967500  3.496500  17.960000  0.000276  10.142955  1.650000  14.092301
50%    56.500000  15.000000  36600.000000  7.400000  15.850000  4.866500  20.710000  0.000322  11.531144  2.140000  19.638165
75%    75.250000  20.000000  36975.000000  10.870000  18.708750  5.745750  24.237500  0.000373  13.398800  3.060000  22.624036
max    92.000000  24.000000  38400.000000  14.620000  23.886000  8.940000  29.800000  0.000453  16.084606  4.000000  27.850374

[ ] # Checking Null values
d.isnull().sum()

   Unnamed: 0      0
FID            0
cf             0
Load           0
Fuel1          0
Fuel2          0
Total_Vol     0
Mass_Fuel      0
Heat_Input     0
BP             0
BTE            0
dtype: int64
```

```
d.isnull().sum()

   Unnamed: 0      0
FID            0
cf             0
Load           0
Fuel1          0
Fuel2          0
Total_Vol     0
Mass_Fuel      0
Heat_Input     0
BP             0
BTE            0
dtype: int64
```

```
# Outlier Analysis
sns.boxplot(d['BTE'])
plt.show()

[ ] np.mean(d['BTE']),np.std(d['BTE'])
#max 34 & min 7

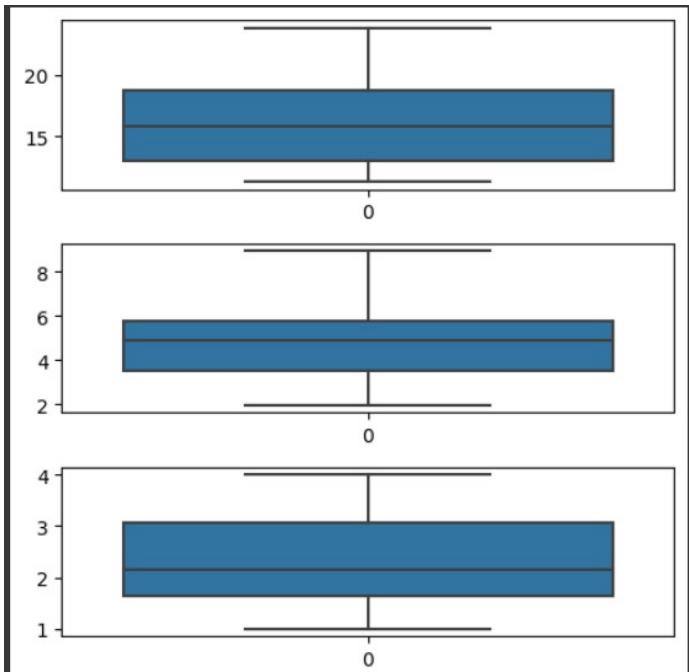
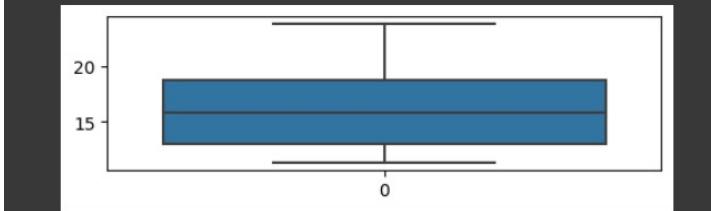
(18.600366825583333, 5.007904859637905)
```

```
[ ] np.mean(d['Fuel1']),np.std(d['Fuel1'])
#max=25 and min=1
(16.16818750000002, 3.37874064098007)

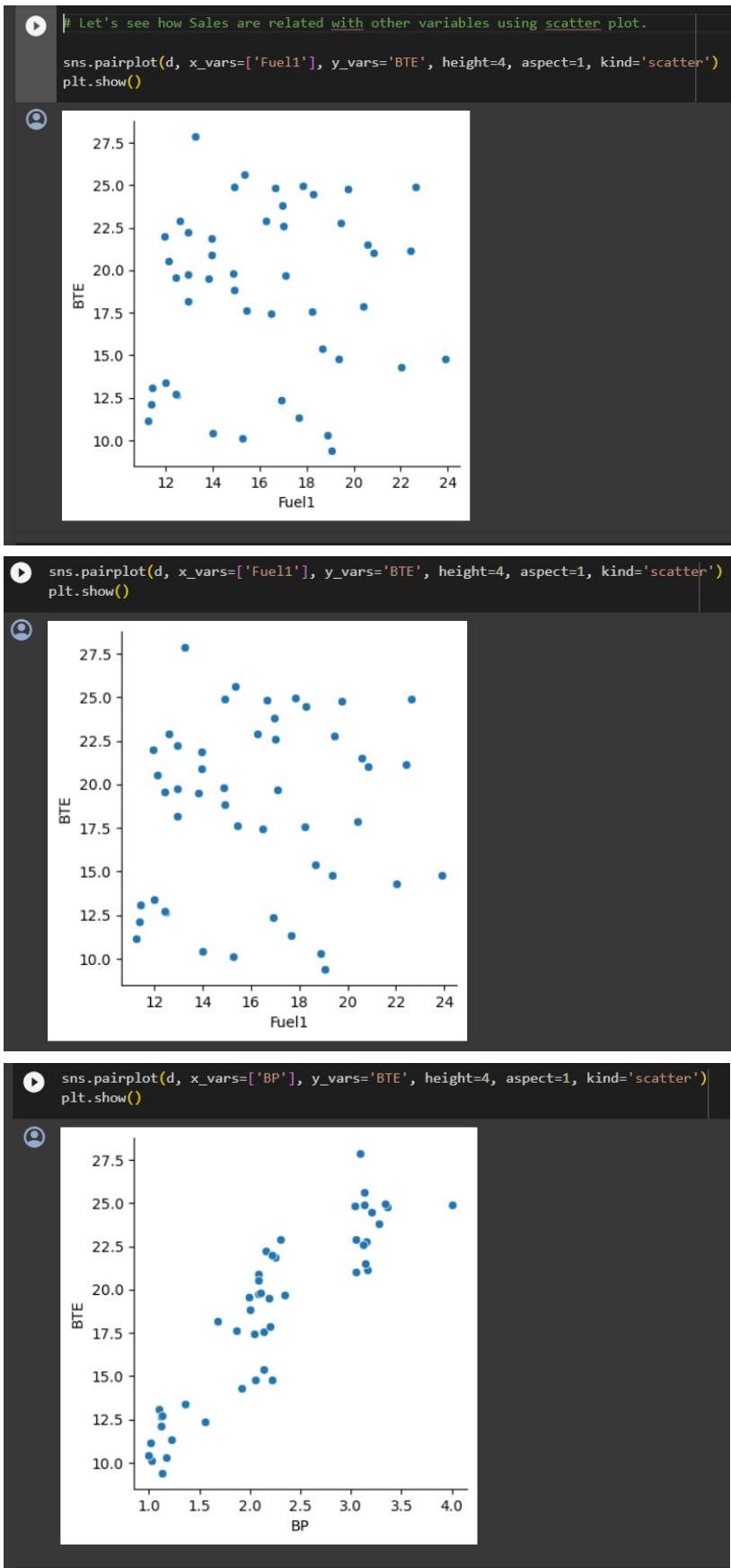
[ ] np.mean(d['Fuel2']),np.std(d['Fuel2'])
#max=9 and min=1
(4.89702083333333, 1.751531894865931)

[ ] np.mean(d['BP']),np.std(d['BP'])
#max=4 and min=1
(2.21375, 0.7921821786474792)

[ ] # Outlier Analysis
fig, axs = plt.subplots(3, figsize = (5,5))
plt1 = sns.boxplot(d['Fuel1'], ax = axs[0])
plt2 = sns.boxplot(d['Fuel2'], ax = axs[1])
plt3 = sns.boxplot(d['BP'], ax = axs[2])
plt.tight_layout()
```



Since the dataset contains outliers we try to use box plots because box plots provide a quick visual summary of the variability of values in a dataset. They show the median, upper and lower quartiles, minimum and maximum values, and any outliers in the dataset. Outliers can reveal mistakes or unusual occurrences in data.



Here we are normalizing data and scaling it so that data is not scattered and tends to fall under a normal range for easy modeling.

```

[ ] dfa.to_csv('pro1.csv')

▶ from sklearn.preprocessing import StandardScaler
import numpy as np
object = StandardScaler()
object.fit_transform(dfa)

@ array([[ -1.68811057, -0.94040168, -1.2513529 ,  0.44093722, -1.67568792,
          -0.3814897 , -0.2563561 , -0.50099542, -1.2544463 , -1.4458811 ],
         [-1.68811057, -0.94040168, -0.39768065,  1.73816612, -1.39764559,
          0.90403765,  1.05548497,  0.88884318, -0.37081117, -0.85426753],
         [-1.52605196, -1.13032852, -1.49975753, -0.2676108 , -0.61718593,
          -0.52403278, -0.47850494, -0.78298545, -1.49429012, -1.69758406],
         [-1.52605196, -1.13032852, -0.14227871,  0.60845526, -0.19469861,
          0.45265125,  0.50331311,  0.24815076, -0.09309727, -0.20486752],
         [-1.52605196, -1.13032852,  1.14872564,  1.84441872,  0.40135105,
          1.83056764,  1.88844548,  1.70288888,  1.19448534,  0.50173146],
         [-1.36399334, -1.38356432, -0.84900738,  0.22014489,  1.34224171,
          0.816928 ,  0.77992534,  0.46101497, -0.82525209, -1.23898942],
         [-1.36399334, -1.38356432, -0.12478542,  0.7339458 ,  1.76701274,
          1.47157027,  1.42880027,  1.13452785, -0.09309727, -0.63481435],
         [-1.20193473, -1.57349117, -1.09041469, -1.23542703,  1.76929645,
          -0.2838213 , -0.39883016, -0.80986042, -1.07771927, -1.03795702],
         [-1.03987611, -2.5231254 , -0.15977199,  0.27401112,  1.38677416,
          0.88555985,  0.7622519 ,  0.09423082,  0.15936991,  0.21845591],
         [-1.03987611, -2.5231254 ,  0.66241235,  1.3886276 ,  2.30825324,
          2.30571122,  2.15101742,  1.45895637,  1.05562839,  0.48628401],
         [-0.8778175 , -0.43393009, -1.63620515, -0.64408244, -0.79874128,
          -0.94374294, -0.91772637, -1.08810287, -1.53216019, -1.63713461],
         [-0.8778175 , -0.43393009, -0.68107185, -0.21788814, -0.59320692,
          -0.46859936, -0.44188476, -0.57226741, -0.43392796, -0.19252342],
         [-0.8778175 , -0.43393009,  1.01227802,  1.05832702,  0.02225433,
          0.9541917 ,  0.98302557,  0.97237322,  1.44695252,  1.23157391],
```

Here we are standardizing data because in machine learning preprocessing is essential to ensure that features are on a consistent scale. This practice facilitates fair comparisons between features, enhances algorithm stability, and prevents biases in models sensitive to feature scales. By scaling features to a common mean and standard deviation, standardization improves the convergence speed of optimization algorithms, aids regularization techniques, and contributes to overall model interpretability. In essence, standardization is a fundamental step that promotes better model performance, robustness, and generalization across diverse datasets.

new\_df = new\_df[(new\_df['Fuel2'] >= 1) & (new\_df['Fuel2'] <= 9)]  
new\_df

	Unnamed: 0	FID	cf	Load	Fuel1	Fuel2	Total_Vol	Mass_Fuel	Heat_Input	BP	BTE
0	13	4	34500	4.19	17.658	1.962	19.62	0.000311	10.739879	1.22	11.359532
1	14	4	34500	6.63	22.041	2.449	24.49	0.000389	13.405690	1.92	14.322276
2	17	5	34200	3.48	15.264	3.816	19.08	0.000298	10.199002	1.03	10.099027
3	18	5	34200	7.36	18.224	4.556	22.78	0.000356	12.176796	2.14	17.574410
4	19	5	34200	11.05	22.400	5.600	28.00	0.000438	14.967089	3.16	21.112990
5	20	6	33800	5.34	16.912	7.248	24.16	0.000372	12.585085	1.56	12.395626
6	21	6	33800	7.41	18.648	7.992	26.64	0.000411	13.876931	2.14	15.421277
7	24	7	33500	4.65	11.994	7.996	19.99	0.000303	10.147454	1.36	13.402377
8	29	8	32000	7.31	17.094	7.326	24.42	0.000371	11.881566	2.34	19.694373
9	30	8	32000	9.66	20.860	8.940	29.80	0.000453	14.499209	3.05	21.035631
10	33	9	35300	3.09	13.992	3.498	17.49	0.000272	9.613765	1.00	10.401752
11	34	9	35300	5.82	15.432	3.858	19.29	0.000300	10.603174	1.87	17.636228
12	35	9	35300	10.66	19.744	4.936	24.68	0.000384	13.565907	3.36	24.767972
13	39	10	34800	7.58	12.936	5.544	18.48	0.000279	9.708524	2.16	22.248489
14	40	10	34800	12.00	17.850	7.650	25.50	0.000385	13.396503	3.34	24.931880
15	43	11	34500	4.19	18.891	2.099	20.99	0.000330	11.369093	1.17	10.291058
16	44	11	34500	8.07	20.421	2.269	22.69	0.000356	12.289887	2.20	17.900897

new\_df = d[(d['Fuel1'] >= 11) & (d['Fuel1'] <= 25)]  
new\_df

1	14	4	34300	0.00	22.041	2.449	24.49	0.000308	10.900090	1.52	14.322270
2	17	5	34200	3.48	15.264	3.816	19.08	0.000298	10.199002	1.03	10.099027
3	18	5	34200	7.36	18.224	4.556	22.78	0.000356	12.176796	2.14	17.574410
4	19	5	34200	11.05	22.400	5.600	28.00	0.000438	14.967089	3.16	21.112990
5	20	6	33800	5.34	16.912	7.248	24.16	0.000372	12.585085	1.56	12.395626
6	21	6	33800	7.41	18.648	7.992	26.64	0.000411	13.876931	2.14	15.421277
7	24	7	33500	4.65	11.994	7.996	19.99	0.000303	10.147454	1.36	13.402377
8	29	8	32000	7.31	17.094	7.326	24.42	0.000371	11.881566	2.34	19.694373
9	30	8	32000	9.66	20.860	8.940	29.80	0.000453	14.499209	3.05	21.035631
10	33	9	35300	3.09	13.992	3.498	17.49	0.000272	9.613765	1.00	10.401752
11	34	9	35300	5.82	15.432	3.858	19.29	0.000300	10.603174	1.87	17.636228
12	35	9	35300	10.66	19.744	4.936	24.68	0.000384	13.565907	3.36	24.767972
13	39	10	34800	7.58	12.936	5.544	18.48	0.000279	9.708524	2.16	22.248489
14	40	10	34800	12.00	17.850	7.650	25.50	0.000385	13.396503	3.34	24.931880
15	43	11	34500	4.19	18.891	2.099	20.99	0.000330	11.369093	1.17	10.291058
16	44	11	34500	8.07	20.421	2.269	22.69	0.000356	12.289887	2.20	17.900897
17	46	12	36700	7.19	14.920	3.730	18.65	0.000289	10.611174	2.00	18.848055
18	47	12	36700	11.61	19.432	4.858	24.29	0.000377	13.820130	3.15	22.792839

```

▶ new_df.info()

👤 <class 'pandas.core.frame.DataFrame'>
Int64Index: 48 entries, 0 to 47
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Unnamed: 0    48 non-null    int64  
 1   FID          48 non-null    int64  
 2   cf           48 non-null    int64  
 3   Load         48 non-null    float64 
 4   Fuel1        48 non-null    float64 
 5   Fuel2        48 non-null    float64 
 6   Total_Vol   48 non-null    float64 
 7   Mass_Fuel   48 non-null    float64 
 8   Heat_Input  48 non-null    float64 
 9   BP           48 non-null    float64 
 10  BTE          48 non-null    float64 
dtypes: float64(8), int64(3)
memory usage: 4.5 KB

```

```

▶ c=['FID','cf',
     'Load',
     'Fuel1',
     'Fuel2',
     'Total_Vol',
     'Mass_Fuel',
     'Heat_Input',
     'BP','BTE']
dfa=new_df[c]
dfa

```

	FID	cf	Load	Fuel1	Fuel2	Total_Vol	Mass_Fuel	Heat_Input	BP	BTE
0	4	34500	4.19	17.658	1.962	19.62	0.000311	10.739879	1.22	11.359532
1	4	34500	6.63	22.041	2.449	24.49	0.000389	13.405690	1.92	14.322276
2	5	34200	3.48	15.264	3.816	19.08	0.000298	10.199002	1.03	10.099027
3	5	34200	7.36	18.224	4.556	22.78	0.000356	12.176796	2.14	17.574410
4	5	34200	11.05	22.400	5.600	28.00	0.000438	14.967089	3.16	21.112990
5	6	33800	5.34	16.912	7.248	24.16	0.000372	12.585085	1.56	12.395626
6	6	33800	7.41	18.648	7.992	26.64	0.000411	13.876931	2.14	15.421277
7	7	33500	4.65	11.994	7.996	19.99	0.000303	10.147454	1.36	13.402377
8	8	32000	7.31	17.094	7.326	24.42	0.000371	11.881566	2.34	19.694373
9	8	32000	9.66	20.860	8.940	29.80	0.000453	14.499209	3.05	21.035631
10	9	35300	3.09	13.992	3.498	17.49	0.000272	9.613765	1.00	10.401752
11	9	35300	5.82	15.432	3.858	19.29	0.000300	10.603174	1.87	17.636228

After the removal of outliers from the dataset, the refined set comprising Brake Power variables 1 to 4 and Fuel variables 1 to 9 is ready for storage in Excel sheets, marking a critical phase in the data preprocessing pipeline. This meticulous curation ensures that the data used for subsequent prediction and analysis is devoid of anomalous values that might skew the results.

## 6.2 EFFICIENCY CALCULATION

```
#Butanol 10
fuel_1_range = [16.46,20.11,28.18,18.05,20.49,27.43,41.24,12.81,19.13,8.91,8.37,26.28,33.83,17.658,22.041,27.756,26.955,15.264,18.224,22.4,16.912,18.648,23.429,9.324,11.994,14.682,17.604,21.522,6.909,17.09
fuel_2_range=[]
for i in fuel_1_range:
    fuel_2_range.append(i*0.1)
    # Values from 1 to 48 (inclusive)
calorific_value1 = 35800
brake_power = 5.2
density1 = 958
density2=811.6
calorific_value2=33100
# Assuming you have the formulas defined as functions, for example:
def predict_efficiency(fuel_amount_1, fuel_amount_2, calorific_value1, calorific_value2, brake_power, density1,density2):
    # Your formula to calculate efficiency based on the given inputs
    mass_of_fuel1 = ((fuel_amount_1 * 0.000001) / 60) * density1
    mass_of_fuel2=((fuel_amount_2 * 0.000001) / 60) * density2
    heat_input = mass_of_fuel1 * calorific_value1 + mass_of_fuel2 * calorific_value2
    efficiency = brake_power / heat_input
    return efficiency
    # Limit efficiency to a maximum value of 32
    efficiency = min(efficiency, 32)

return efficiency

# List to store combinations that meet the efficiency criteria
efficient_combinations = []

# Loop through all combinations of fuel_1 and fuel_2
for fuel_amount_1 in fuel_1_range:
    # Use the formula-based prediction function to get efficiency
    predicted_efficiency = predict_efficiency(fuel_amount_1, fuel_amount_1*0.1, calorific_value1,calorific_value2 ,brake_power,density1,density2)
    """if 23 <= predicted_efficiency <= 32:"""
    efficient_combinations.append((fuel_amount_1, fuel_amount_1*0.1, predicted_efficiency))
```

```
"""if 23 <= predicted_efficiency <= 32:"""
efficient_combinations.append((fuel_amount_1, fuel_amount_1*0.1, predicted_efficiency))

# Print the combinations that meet the efficiency criteria
if efficient_combinations:
    for fuel_amount_1, fuel_amount_2, predicted_efficiency in efficient_combinations:
        eff=(brake_power/predicted_efficiency)*100
        if eff<33 and eff>31.99999999999:
            print(f"For Fuel 1 amount = {fuel_amount_1}, Fuel 2 amount = {fuel_amount_2}, "
                  f"Brake power = {brake_power}, "
                  f"Heat Input = {predicted_efficiency}, ,f"Efficiency={eff}")

    else:
        print("No combinations meet the efficiency criteria.")
```

```
For Fuel 1 amount = 26.28, Fuel 2 amount = 2.628, Brake power = 5.2, Heat Input = 16.19846464799998, Efficiency=32.101807874995345
For Fuel 1 amount = 25.935, Fuel 2 amount = 2.5935, Brake power = 5.2, Heat Input = 15.985813571, Efficiency=32.52884175650193
For Fuel 1 amount = 25.935, Fuel 2 amount = 2.5935, Brake power = 5.2, Heat Input = 15.985813571, Efficiency=32.52884175650193
```

First we are calculating mass by applying the provided formulas to calculate the mass of each fuel. Then converting brake power to equivalent load.

After mass calculation we are iterating efficiency by implementing a Python loop for efficiency calculations, setting and applying an efficiency threshold.

At last we are storing the best efficiency fuel combination in an Excel sheet.

fuel	fuel1	fuel2	brake power	heat power	efficiency	mass_of_fuel1	mass_of_fuel2	total
butanol10	26.28	2.628	5.2	16.19846465	32.10180787	0.000419604	0.00003554808	0.00045515208
	25.935	2.5935	5.2	15.98581357	32.52884176	0.0004140955	0.00003508141	0.00044917691
diesel	26.955	0	5.2	16.038225	32.42254052	0.00037737	0	0.00037737
	27.12	0	5.2	16.1364	32.22527949	0.000433016	0	0.000433016

Fig.6.2.1 Mass of fuel after calculation

### 6.3 PRESSURE PREDICTION USING MACHINE LEARNING ALGORITHMS

Crank_angle	Volume	cf_value	density	viscosity	mass_of_fuel	pressure	cetane
1	40.16	42500	840	4.59	0.00028	10.44	1.45
2	40.34	42500	840	4.59	0.00028	10.44	1.45
3	40.65	42500	840	4.59	0.00028	10.44	1.42
4	41.09	42500	840	4.59	0.00028	10.44	1.38
5	41.65	42500	840	4.59	0.00028	10.44	1.38
6	42.33	42500	840	4.59	0.00028	10.44	1.31
7	43.14	42500	840	4.59	0.00028	10.44	1.25
8	44.07	42500	840	4.59	0.00028	10.44	1.25
9	45.12	42500	840	4.59	0.00028	10.44	1.18
10	46.29	42500	840	4.59	0.00028	10.44	1.15

fig.6.3.1 Dataset for pressure prediction

```

1. FUEL: DIESEL
2. MODEL: GradientBoostingRegressor
3. MASS OF FUEL:0.00043

import pandas as pd
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

met
# Load the dataset
data = pd.read_csv("DIESEL.csv") # Replace with the actual path

# Separate features and target
X = data[["Crank_angle", "mass_of_fuel", "Volume", "cf_value", "viscosity", "load", "cetane"]]
y = data["pressure"]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a Random Forest Regressor model
model1 = GradientBoostingRegressor(n_estimators=100, random_state=42)

# Fit the model to the training data
model1.fit(X_train, y_train)

# Make predictions on the test data

# Make predictions on the test data
Predictions = model1.predict(X_test)

# Calculate the Mean Squared Error
mse = mean_squared_error(y_test, Predictions)
print(f"Mean Squared Error: {mse}")
mass_values = [0.00043]
angles = list(range(1, 721))

# Create a DataFrame for new data
new_data_rows = []

for mass_value in mass_values:
    for angle in angles:
        new_data_rows.append({
            "Crank_angle": angle,
            "mass_of_fuel": mass_value,
            "Volume": data.at[angle - 1, "Volume"],
            "cf_value": data.at[angle - 1, "cf_value"],
            "viscosity": data.at[angle - 1, "viscosity"],
            "cetane": data.at[angle - 1, "cetane"],
            "load": 18.25
        })

new_data = pd.DataFrame(new_data_rows)

# Predict pressures for the new data
predicted_pressures = model1.predict(new_data[X.columns]) # Use the same columns as in X

# Reshape the predictions to matrices where each row corresponds to a mass of fuel
num_angles = len(angles)
num_mass_values = len(mass_values)

```

```

predicted_pressures_matrix = predicted_pressures.reshape(num_mass_values, num_angles, -1)

# Display predicted pressures for all angles and both mass values
for i, mass_value in enumerate(mass_values):
    print(f"Predicted Pressures for Mass of Fuel {mass_value}:")
    for j, angle in enumerate(angles):
        pressures = predicted_pressures_matrix[i, j, :]
        print(f"Angle {angle}: Pressures {pressures}")
    print("\n")
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

# Predict pressures for the new data
predicted_pressures = model1.predict(X)

# Compute metrics
mse = mean_squared_error(y, predicted_pressures)
rmse = np.sqrt(mse)
r2 = r2_score(y, predicted_pressures)

print(f"Root Mean Squared Error (RMSE): {rmse}")
print(mse)
print(f"R-squared (R2) Score: {r2}")

```

```

Mean Squared Error: 0.9635192635951386
Predicted Pressures for Mass of Fuel 0.00043
Angle 1: Pressures [0.99852942]
Angle 2: Pressures [0.99852942]
Angle 3: Pressures [0.99316126]
Angle 4: Pressures [1.26239005]
Angle 5: Pressures [1.41251009]
Angle 6: Pressures [1.60720309]
Angle 7: Pressures [1.54272751]
Angle 8: Pressures [1.54272751]
Angle 9: Pressures [1.47815268]
Angle 10: Pressures [0.76532476]
Angle 11: Pressures [0.62760149]
Angle 12: Pressures [0.62760149]
Angle 13: Pressures [0.62760149]
Angle 14: Pressures [0.79027304]
Angle 15: Pressures [0.78452616]
Angle 16: Pressures [0.78452616]
Angle 17: Pressures [0.77261799]
Angle 18: Pressures [0.75052811]
Angle 19: Pressures [0.60936016]
Angle 20: Pressures [0.75600074]

```

By using gradient boosting regressor we are predicting the pressures like this using 3 more algorithms mainly random forest regressor, linear regression and decision tree regressor. After which we will tabulate rmse, r square, mse in which we will select the model with lowest mse value.

1. FUEL: DIESEL
2. MODEL: RandomForestRegressor
3. MASS OF FUEL:0.00043

```

import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv("DIESEL.csv") # Replace with the actual path

# Separate features and target
X = data[["Crank_angle", "mass_of_fuel", "Volume", "cf_value", "viscosity", "load", "cetane"]]
y = data["pressure"]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a Random Forest Regressor model
model = RandomForestRegressor(n_estimators=100, random_state=42)

# Fit the model to the training data
model.fit(X_train, y_train)

```

```

# Make predictions on the test data
predictions = model.predict(X_test)

# Calculate the Mean Squared Error
mse = mean_squared_error(y_test, predictions)
print(f"Mean Squared Error: {mse}")
mass_values = [0.00043]
angles = list(range(1, 721))

# Create a DataFrame for new data
new_data_rows = []

for mass_value in mass_values:
    for angle in angles:
        new_data_rows.append({
            "Crank_angle": angle,
            "mass_of_fuel": mass_value,
            "Volume": data.at[angle - 1, "Volume"],
            "cf_value": data.at[angle - 1, "cf_value"],
            "viscosity": data.at[angle - 1, "viscosity"],
            "cetane": data.at[angle - 1, "cetane"],
            "load": 18.25
        })

new_data = pd.DataFrame(new_data_rows)

# Predict pressures for the new data
predicted_pressures = model.predict(new_data[X.columns]) # Use the same columns as in X

# Reshape the predictions to matrices where each row corresponds to a mass of fuel
num_angles = len(angles)
num_mass_values = len(mass_values)

```

```

predicted_pressures_matrix = predicted_pressures.reshape(num_mass_values, num_angles, -1)

# Display predicted pressures for all angles and both mass values
for i, mass_value in enumerate(mass_values):
    print(f"Predicted Pressures for Mass of Fuel {mass_value}")
    for j, angle in enumerate(angles):
        pressures = predicted_pressures_matrix[i, j, :]
        print(f"Angle {angle}: Pressures {pressures}")
    print("\n")
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

# Predict pressures for the new data
predicted_pressures = model.predict(X)

# Compute metrics
mse = mean_squared_error(y, predicted_pressures)
rmse = np.sqrt(mse)
r2 = r2_score(y, predicted_pressures)

print(f"Root Mean Squared Error (RMSE): {rmse}")
print(mse)
print(f"R-squared (R2) Score: {r2}")

```

1. FUEL: DIESEL
2. MODEL: LinearRegression
3. MASS OF FUEL:0.00028

```

▶ import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv("DIESEL.csv") # Replace with the actual path

# Separate features and target
X = data[["Crank_angle", "mass_of_fuel", "Volume", "cf_value", "viscosity", "load", "cetane"]]
y = data["pressure"]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a Random Forest Regressor model
model2 = LinearRegression()

# Fit the model to the training data
model2.fit(X_train, y_train)

# Make predictions on the test data
predictions = model2.predict(X_test)

```

```

# Calculate the Mean Squared Error
mse = mean_squared_error(y_test, predictions)
print(f"Mean Squared Error: {mse}")
mass_values = [0.00028]
angles = list(range(1, 721))

# Create a DataFrame for new data
new_data_rows = []

for mass_value in mass_values:
    for angle in angles:
        new_data_rows.append({
            "Crank_angle": angle,
            "mass_of_fuel": mass_value,
            "Volume": data.at[angle - 1, "Volume"],
            "cf_value": data.at[angle - 1, "cf_value"],
            "viscosity": data.at[angle - 1, "viscosity"],
            "cetane": data.at[angle - 1, "cetane"],
            "load": 10.44
        })

new_data = pd.DataFrame(new_data_rows)

# Predict pressures for the new data
predicted_pressures = model2.predict(new_data[X.columns]) # Use the same columns as in X

# Reshape the predictions to matrices where each row corresponds to a mass of fuel
num_angles = len(angles)
num_mass_values = len(mass_values)

predicted_pressures_matrix = predicted_pressures.reshape(num_mass_values, num_angles, -1)

```

```

# Display predicted pressures for all angles and both mass values
for i, mass_value in enumerate(mass_values):
    print(f"Predicted Pressures for Mass of Fuel {mass_value}:")
    for j, angle in enumerate(angles):
        pressures = predicted_pressures_matrix[i, j, :]
        print(f"Angle {angle}: Pressures {pressures}")
    print("\n")

from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

# Predict pressures for the new data
predicted_pressures = model2.predict(X)

# Compute metrics
mse = mean_squared_error(y, predicted_pressures)
rmse = np.sqrt(mse)
r2 = r2_score(y, predicted_pressures)

print(f"Root Mean Squared Error (RMSE): {rmse}")
print(mse)
print(f"R-squared (R2) Score: {r2}")

```

```
1. FUEL: DIESEL  
2. MODEL: DecisionTreeRegressor  
3. MASS OF FUEL:0.00037
```

```
▶ import pandas as pd  
from sklearn.tree import DecisionTreeRegressor  
from sklearn.model_selection import train_test_split  
from sklearn.metrics import mean_absolute_error, mean_squared_error  
from sklearn.metrics import accuracy_score  
import matplotlib.pyplot as plt  
  
# Load the dataset  
data = pd.read_csv("DIESEL.csv") # Replace with the actual path  
  
# Separate features and target  
X = data[["Crank_angle", "mass_of_fuel", "Volume", "cf_value", "viscosity", "load", "cetane"]]  
y = data["pressure"]  
  
# Split the data into training and testing sets  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)  
  
# Create a Random Forest Regressor model  
model3 = DecisionTreeRegressor()  
  
# Fit the model to the training data  
model3.fit(X_train, y_train)  
from sklearn.metrics import mean_squared_error, r2_score  
import numpy as np
```

```
# Predict pressures for the new data  
predicted_pressures = model3.predict(X)  
  
# Compute metrics  
mse = mean_squared_error(y, predicted_pressures)  
rmse = np.sqrt(mse)  
r2 = r2_score(y, predicted_pressures)  
  
print(f"Root Mean Squared Error (RMSE): {rmse}")  
print(mse)  
print(f"R-squared (R2) Score: {r2}")  
mass_values = [0.00037]  
angles = list(range(1, 721))  
  
# Create a DataFrame for new data  
new_data_rows = []  
  
for mass_value in mass_values:  
    for angle in angles:  
        new_data_rows.append({  
            "Crank_angle": angle,  
            "mass_of_fuel": mass_value,  
            "Volume": data.at[angle - 1, "Volume"],  
            "cf_value": data.at[angle - 1, "cf_value"],  
            "viscosity": data.at[angle - 1, "viscosity"],  
            "cetane": data.at[angle - 1, "cetane"],  
            "load": 18.25  
        })  
  
new_data = pd.DataFrame(new_data_rows)
```

```

# Predict pressures for the new data
predicted_pressures = model.predict(new_data[X.columns]) # Use the same columns as in X

# Reshape the predictions to matrices where each row corresponds to a mass of fuel
num_angles = len(angles)
num_mass_values = len(mass_values)

predicted_pressures_matrix = predicted_pressures.reshape(num_mass_values, num_angles, -1)

# Display predicted pressures for all angles and both mass values
for i, mass_value in enumerate(mass_values):
    print(f"Predicted Pressures for Mass of Fuel {mass_value}")
    for j, angle in enumerate(angles):
        pressures = predicted_pressures_matrix[i, j, :]
        print(f"Angle {angle}: Pressures {pressures}")
    print("\n")

```

	mass_of_fuel	model	mse	rmse	rsquare
Diesel	0.00028	GradientBoostingRegressor	0.7163811845	0.8463930437	0.9962463063
Diesel	0.00037	GradientBoostingRegressor	0.7163811845	0.8463930437	0.9962463063
Diesel	0.00043	GradientBoostingRegressor	0.7163811845	0.8463930437	0.9962463063
Diesel	0.00028	RandomForestRegressor	0.102447078	0.3200735509	0.9994631979
Diesel	0.00037	RandomForestRegressor	0.102447078	0.3200735509	0.9994631979
Diesel	0.00043	RandomForestRegressor	0.102447078	0.3200735509	0.9994631979
Diesel	0.00028	linear regression	149.4408675	12.22460091	0.216959838
Diesel	0.00037	linear regression	149.4408675	12.22460091	0.216959838
Diesel	0.00043	linear regression	149.4408675	12.22460091	0.216959838
Diesel	0.00028	DecisionTreeRegressor	0.1313038194	0.3623586889	0.9993119943
Diesel	0.00037	DecisionTreeRegressor	0.3536286764	0.1250532407	0.9993447461

fig.6.3.2 Tabulating errors after prediction

## 6.4 HEAT RELEASE RATE PREDICTION OR ANALYSIS

```

import pandas as pd

# Load the dataset from the CSV file
data = pd.read_csv("Diesel - trail - Sheet1.csv")

# Initialize an empty list to store HRR values
hrr_values = []

# Iterate through all angles in the data
for angle_index in range(len(data)):
    # Calculate HRR using the provided formula
    if angle_index == 0:
        # If it's the first angle, use a default value of 0
        hrr = 0
    else:
        # Calculate HRR using the formula
        pressure_current = data.iloc[angle_index]["pressure"]
        pressure_previous = data.iloc[angle_index - 1]["pressure"]
        volume_current = data.iloc[angle_index]["Volume"]
        volume_previous = data.iloc[angle_index - 1]["Volume"]

        hrr = ((3.85714285714 * (pressure_current * ((volume_current - volume_previous) / 1))) + (2.85714285714 * (volume_current * ((pressure_current - pressure_previous) / 1)))

    # Append the calculated HRR to the list
    hrr_values.append(hrr)

# Now you have a list of HRR values for all angles in the provided data

# Iterate through the angles and pressures to print them alongside HRR values
for angle_index in range(len(data)):
    angle = data.iloc[angle_index]["Crank_angle"]
    pressure = data.iloc[angle_index]["pressure"]
    hrr = data.iloc[angle_index]["HRR"]
    print(f"Angle: {angle}, Pressure: {pressure}, HRR: {hrr}")

```

```

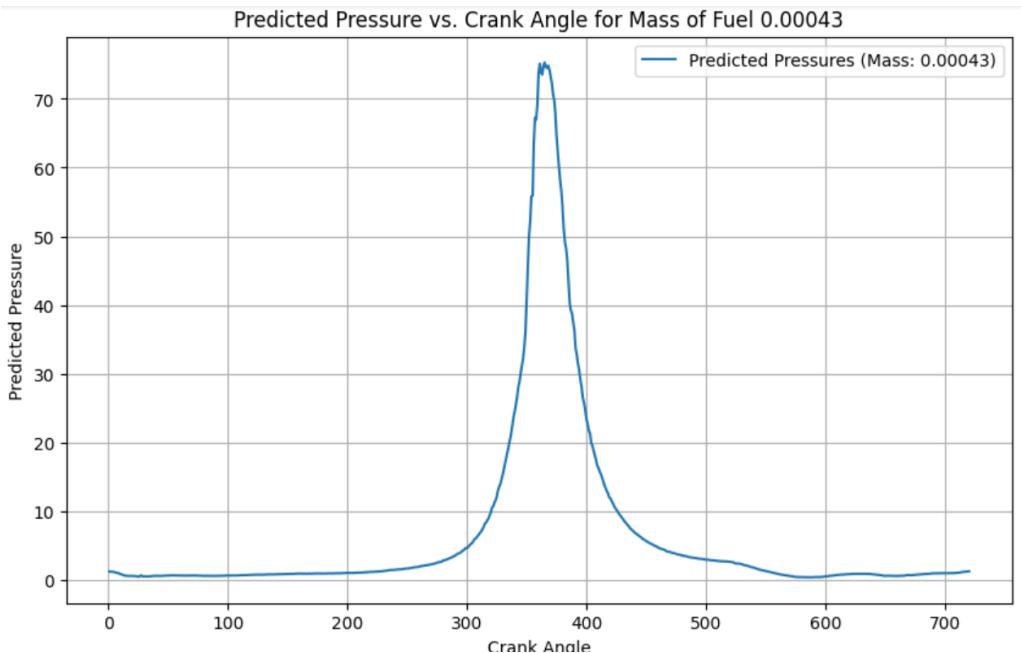
# Iterate through the angles and pressures to print them alongside HRR values
for angle_index in range(len(data)):
    angle = data.iloc[angle_index]["Crank_angle"]
    pressure = data.iloc[angle_index]["pressure"]
    hrr = hrr_values[angle_index]
    print(f"Angle: {angle}, Pressure: {pressure}, HRR: {hrr}")

```

```

Angle: 1.0, Pressure: 1.45, HRR: 0
Angle: 2.0, Pressure: 1.45, HRR: 1.0067142857135782
Angle: 3.0, Pressure: 1.42, HRR: -1.7863714285692318
Angle: 4.0, Pressure: 1.38, HRR: -2.353942857139875
Angle: 5.0, Pressure: 1.38, HRR: 2.980799999997766
Angle: 6.0, Pressure: 1.31, HRR: -5.030057142851202
Angle: 7.0, Pressure: 1.25, HRR: -3.4900714285669214
Angle: 8.0, Pressure: 1.25, HRR: 4.483928571425248
Angle: 9.0, Pressure: 1.18, HRR: -4.24499999994536
Angle: 10.0, Pressure: 1.15, HRR: 1.2220714285715566
Angle: 11.0, Pressure: 1.04, HRR: -9.74199999988872
Angle: 12.0, Pressure: 1.04, HRR: 5.656114285710083
Angle: 13.0, Pressure: 1.01, HRR: 1.667399999998808
Angle: 14.0, Pressure: 0.91, HRR: -9.119928571417951
Angle: 15.0, Pressure: 0.94, HRR: 11.079771428562012
Angle: 16.0, Pressure: 0.94, HRR: 6.85259999994926
Angle: 17.0, Pressure: 0.91, HRR: 2.094814285714027
Angle: 18.0, Pressure: 0.94, HRR: 12.865628571417702
Angle: 19.0, Pressure: 0.94, HRR: 8.1215999999399
Angle: 20.0, Pressure: 0.94, HRR: 8.520428571422265

```

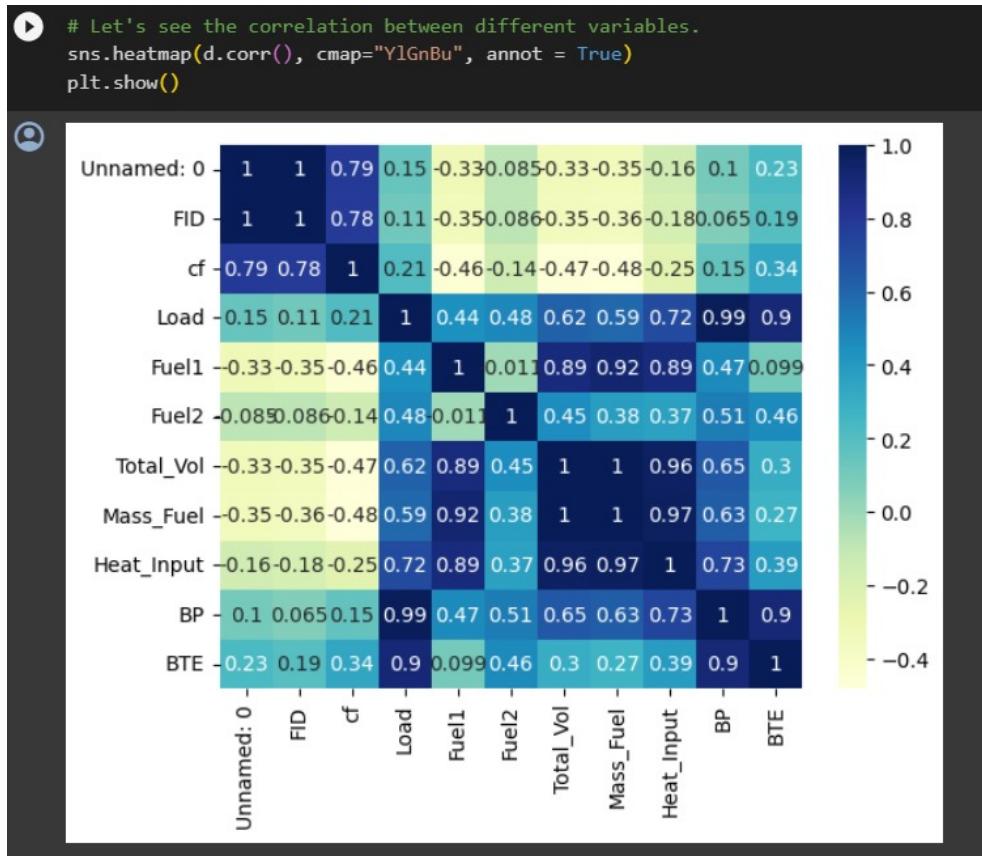


The process of Heat Release Rate (HRR) calculation involves applying a provided formula that considers pressure, volume, and relevant changes in a combustion system. Finally we visualize the HRR we found through the whole prediction process.

# CHAPTER 7

## RESULTS AND DISCUSSIONS

### 7.1 PERFORMANCE MATRIX



During analysis we create performance metrics because they serve as indispensable tools across diverse domains, offering a visual means to analyze and interpret complex data sets. By leveraging color gradients and matrix formats, these visualizations facilitate the identification of patterns, trends, and anomalies, providing valuable insights for decision-making and strategic planning. Whether applied in business analytics to compare the performance of different products or in sports to assess team dynamics, heatmaps condense intricate information into an accessible and user-friendly format. Beyond their aesthetic appeal, heatmaps are instrumental in highlighting correlations between variables, tracking Key Performance Indicators (KPIs), and optimizing processes by pinpointing bottlenecks. The visual clarity of heatmaps enhances communication, making them essential tools for conveying complex information to diverse audiences, ultimately aiding organizations in making informed decisions and driving continuous improvement.

## 7.2 RESULT

Angle	Cylinder Volume (cc)	Diesel				BUTANOL 10			
		0.00043		0.00037		0.00045		0.00044	
		Pressure	HRR	Pressure	HRR	Pressure	HRR	Pressure	HRR
Angle 1	40.16	1.2179	32.84015154	1.2179	32.84015154	0.7368	19.86	0.7368	19.86
Angle 2	40.34	1.2139	0.1303822	1.2139	0.1303822	0.7392	0.078	0.7392	0.078
Angle 3	40.65	1.206	0.235956	1.206	0.235956	0.7517	0.235	0.7517	0.235
Angle 4	41.09	1.189	0.4013702857	1.189	0.4013702857	0.7197	0.497	0.7197	0.497
Angle 5	41.65	1.155	0.65408	1.155	0.65408	0.6262	1.247	0.6262	1.247
Angle 6	42.33	1.0975	0.98328	1.0975	0.98328	0.5941	0.544	0.5941	0.544
Angle 7	43.14	1.0568	0.8318310857	1.0568	0.8318310857	0.5381	0.858	0.5381	0.858
Angle 8	44.07	1.0278	0.7338379714	1.0278	0.7338379714	0.4658	1.077	0.4658	1.077
Angle 9	45.12	0.9781	1.0368345	0.9781	1.0368345	0.3999	1.011	0.3999	1.011
Angle 10	46.29	0.8681	1.8465897	0.8681	1.8465897	0.3096	1.334	0.3096	1.334

fig.7.2.1 Final pressure and HRR

Upon the completion of machine learning predictions for the final Heat Release Rate (HRR) and pressure values, a crucial step is the systematic storage of this valuable information in Excel sheets. Organizing the predicted data in Excel facilitates a structured and accessible format for further analysis and comparison. This methodical approach ensures that the results of the machine learning model, capturing the dynamic relationships between variables, are readily available for comprehensive scrutiny and interpretation. By leveraging Excel's tabular structure, the predicted HRR and pressure values can be efficiently arranged, enabling easy reference and analysis. This organized storage not only streamlines subsequent data manipulation but also enhances the overall workflow for any future analyses, contributing to a more efficient and informed decision-making process based on the insights gleaned from the machine learning predictions.

## **CHAPTER 8**

### **CONCLUSION**

In this paper, we have explored the prediction of fuel efficiency and In-cylinder pressure in a mixture of fuels. We considered various factors that influence these parameters, such as volume, mass of fuel, cetane number, crank angle, density, viscosity, and calorific value. We employed both formula-based calculations and machine learning models to predict fuel efficiency and In-cylinder pressure.

Our findings indicate that optimizing fuel mixtures can significantly enhance vehicle fuel efficiency across different brake powers. Additionally, the application of machine learning models, such as the Random Forest Regressor and Decision Tree Regressor, can lead to highly accurate predictions of In-cylinder pressure, which is crucial for improving engine performance and reducing emissions.

# **CHAPTER 9**

## **FUTURE ENHANCEMENT**

As we look ahead, there are several avenues for future research and development. The study can be extended to consider more complex fuel mixtures and real-world data from various engines and vehicle types. Furthermore, the application of the models and methodologies developed in this study within the automotive industry has the potential to drive the development of more environmentally friendly and fuel-efficient vehicles.

Future work may also explore the integration of sensor data and real-time feedback to optimize engine performance on the fly, taking into account changing operating conditions and fuel mixtures. This could lead to significant improvements in fuel efficiency and emission reduction, contributing to a more sustainable and eco-friendly transportation system.

## CHAPTER 10

### REFERENCES

- [1] M. Aditya Vamsi, B. Raja Rishita, M. Amin Qurishi, N. Siva Sandeep, K. Raghu Ram. (2022). fuel efficiency prediction .*JETIR Journal of Emerging Technologies and Innovative Research*, 2022, 1-10.
- [2] Xie, X., Sun, B., Li, X., Olsson, T., Maleki, N., & Ahlgren, F. (2023). Fuel Consumption Prediction Models Based on Machine Learning and Mathematical Methods. *Journal of Marine Science and Engineering*, 11(4), 738.
- [3] Mohamed A. HAMED, Mohammed H.Khafagy and Rasha M.Badry, “Fuel Consumption Prediction Model using Machine Learning” *International Journal of Advanced Computer Science and Applications(IJACSA)*, 12(11), 2021.
- [4] Yao, Y., Zhao, X., Liu, C., Rong, J., Zhang, Y., Dong, Z., & Su, Y. (2020). Vehicle fuel consumption prediction method based on driving behavior data collected from smartphones. *Journal of Advanced Transportation*, 2020, 1-11.
- [5] Yin, X., Li, Z., Shah, S. L., Zhang, L., & Wang, C. (2015, October). Fuel efficiency modeling and prediction for automotive vehicles: A data-driven approach. In *2015 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 2527-2532). IEEE.

## **APPENDIX**

Appendices are provided to give supplementary information, which if included in the main text may serve as a distraction and cloud the central theme under discussion. Common examples of information included in appendices are listing of computer programs used to obtain your results, documentation of experimental setups, standards required for your work, tables of raw data, and part drawings. The Program/Code has to be included only in Appendix.

## **PAPER PUBLICATION STATUS**

Publication process not yet started